I. J. Engineering and Manufacturing, **2026, 1, 39-49**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2026.01.04

Development of a Deep Learning Model for Detecting DOS Attacks in Computer Networks

Obiageli M. Attoh*
Dennis Osadebay University, Delta State, Nigeria
E-mail: attoh.obiageli@dou.edu.ng
ORCID iD: https://orcid.org/0009-0008-8454-6604
*Corresponding Author

Oduware Okosun
University of Benin, Benin City, Nigeria
E-mail: oduware.aghama@uniben.edu
ORCID iD: https://orcid.org/0009-0003-9958-6303

Received: 11 September, 2025; Revised: 18 October, 2025; Accepted: 28 November, 2025; Published: 08 February, 2026

Abstract: This study investigates the application of deep learning techniques for the detection of Denial of Service (DoS) attacks in network traffic using the NSL-KDD dataset. A Deep Neural Network (DNN) model is proposed and optimized for intrusion detection. The model consists of a 41-feature input layer, two fully connected hidden layers containing 128 and 64 neurons respectively and a SoftMax activated output layer for multiclass classification. The hidden layer used ReLU activation function and the model was optimized using Adam optimizer. The dataset was preprocessed using feature encoding, normalization and label transformation. The dataset was used with its standard predefined split: KDDTrain+ for training/validation and KDDTest+ for testing. The training data was further divided into 80% for training and 20% for validation. The effectiveness of the DNN was compared against traditional machine learning models, including Logistic Regression, LightGBM (LGBM), and CatBoost. Key evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the effectiveness of each model in detecting network intrusions. The results demonstrate that the DNN model achieves an accuracy of 86% on the test dataset, consistently outperforming the traditional models across all key metrics. These findings highlight the advantages of deep learning for anomaly-based intrusion detection, particularly in handling complex network traffic patterns. This study contributes to advancing network security by leveraging the capabilities of DNNs for real-time DoS detection, scalability, and practical implementation in modern cybersecurity frameworks.

Index Terms: Denial of Service, Deep Learning, NSL-KDD Dataset, Intrusion Detection, Computer Network, Attacks

1. Introduction

The advancement of digital infrastructure and the growing dependence on internet-connected systems have underscored the very vital necessity of cyber security.[1]. Among the myriads of cyber threats, Denial of Service (DoS) attacks constitute a major and serious challenge. A DoS attack is an attempt to overwhelm a website or network, aiming to degrade its performance or render it entirely inaccessible.[2]. DoS attacks have become more frequent and increasingly sophisticated in recent years. In addition to mere disruption of services, such attacks might be translated into great financial losses, congestion of communication channels, loss of data, long-term reputation damage, emphasising the need for further research into DoS attack detection [3].

Addressing these attacks is therefore critically important, as they can cause significant economic damage and disruption. Conventional methods for detecting DoS attacks most often depends on signature-based techniques, anomaly-based detection systems or simple statistical analyses, which may not effectively identify novel or sophisticated attack patterns [4]. These methods can struggle with the high volume of traffic and complex patterns associated with modern DoS attacks. As noted by [5], traditional approaches frequently suffer from high false positive rates and limited scalability, underscoring the need for more recent ways to detect the attack. These limitations highlight the need for advanced techniques that can dynamically adapt and identify a wide range of attack patterns. Deep learning is a subset of machine learning that uses neural networks with multiple layers (deep neural networks) to model and learn

complex patterns and representations from data, and it has proven to be one of the most promising solutions for detecting DoS attacks [6].

Deep learning models have the capability of learning intricate patterns and anomalies in network traffic, making them well-suited for identifying sophisticated attacks that may elude traditional methods. [7].

This study uses the NSL-KDD dataset, a benchmark commonly adopted in intrusion detection research due to its improvements over the KDD Cup'99 dataset, particularly in addressing issues of redundancy and class imbalance, making it more suitable for effective model development and evaluation [8]. Beyond dataset limitations, prior research shows that there are other gaps in the model applicability. Most intrusion detection approaches have relied on traditional machine learning techniques with static feature engineering, limiting their generalization to new or complex attack patterns [9,10]. These gaps imply a need for advanced detection techniques that can adapt to evolving threats with high accuracy and robustness of detection, even when benchmark datasets are used to evaluate them.

To overcome these limitations, this study designs a DNN-based model for detecting DoS using the NSL-KDD dataset, laying emphasis on achieving better detection performance by optimizing the model architecture and improving the feature selection strategy. By addressing these knowledge gaps, the study aims to equip researchers with a more robust and effective way of detecting and mitigating DoS attacks, thus promoting improved security in network communications. The present study also constitutes the foundation for possible future research on applying advanced machine learning techniques to cybersecurity-related challenges.

2. Literature Review

Recent research works have shown that deep learning models significantly performs better than the old existing methods in identifying network intrusions.[11]. This is reflected in the present study, where nine machine learning models were analysed. Based on the test data, their performances were observed to be weaker than the proposed deep learning model. Specifically, LGBM, CatBoost, and Logistic Regression, was comparatively weaker with respect to accuracy and other key metrics like precision, recall, and F1 score. LGBM got an accuracy score of 74% on the test data, with a precision, F1 score and Recall as 12%, 12% and 14% respectively. Cat Boost performed slightly better, with a test accuracy of 80%, precision of 37%, recall of 52%, and an F1 score of 38%. Logistic Regression achieved a better accuracy of 83.12% on the test data, with a precision of 44.9%, recall of 43.3%, and F1 score of 48.9%. These models did not perform well and may lead to false classification of traffic, suboptimal decisions and outcomes.

Recent research has aimed to address the shortcomings of traditional IDS by leveraging on the capabilities of deep learning techniques. [12]. used a combination of Multilayer Perceptron (MLP) and K-means clustering on NSL-KDD. Although this approach improved feature categorization, the model achieved only 73.09% accuracy, highlighting limitations in feature selection and model optimization. Similarly, convolutional neural network (CNN) approaches achieved moderate improvements; a CNN model attained 79.48% accuracy on KDDTest+, demonstrating that deep learning enhances detection capabilities, yet substantial performance gains are still needed.

Further research explored Deep Neural Network (DNN)-based approaches. The model developed by [13] provides a significant contribution to DoS attack detection by implementing two deep learning approaches. Using the NSL-KDD dataset, the authors attained an accuracy of 79% with a Deep Neural Network (DNN) and 84% with a tabular deep learning model developed using the fastai library. Another study applied a DNN for binary and multiclass classification, achieving 80.7% and 76.5% accuracy, respectively [14]. The study showed that DNNs are effective in classifying network intrusions, outperforming traditional machine learning models. However, the authors noted the challenge of multiclass classification, as the performance slightly decreased, and suggested that further optimization and more advanced techniques could improve the model's robustness.

Overall, the literature indicates that deep learning improves detection over traditional methods, but limitations persist in feature optimization, multiclass classification, and generalization to evolving attack patterns. The proposed model addresses these gaps by employing a DNN with optimized feature selection and architecture, achieving enhanced detection performance on NSL-KDD. This study thus contributes to advancing intrusion detection systems by providing a robust framework for mitigating DoS attacks while informing future research on adaptive cybersecurity solutions.

The model being proposed, achieved an accuracy of 86% on the NSL-KDD dataset, this exemplifies the progress in this domain, focusing on real-time detection and scalability. By addressing previous limitations and employing robust methodologies, this model sets a new standard in the intrusion detection systems.

3. Materials and Methods

The experimental environment for this project was the Kaggle platform. The experimental analysis was conducted within a single Jupyter notebook using Python 3 programming language. Python's extensive ecosystem of libraries, including TensorFlow, PyTorch, Pandas, NumPy, and scikit-learn, facilitated preprocessing, feature engineering, model training, and evaluation. The proposed Intrusion Detection System architecture is as shown in Figure 1.

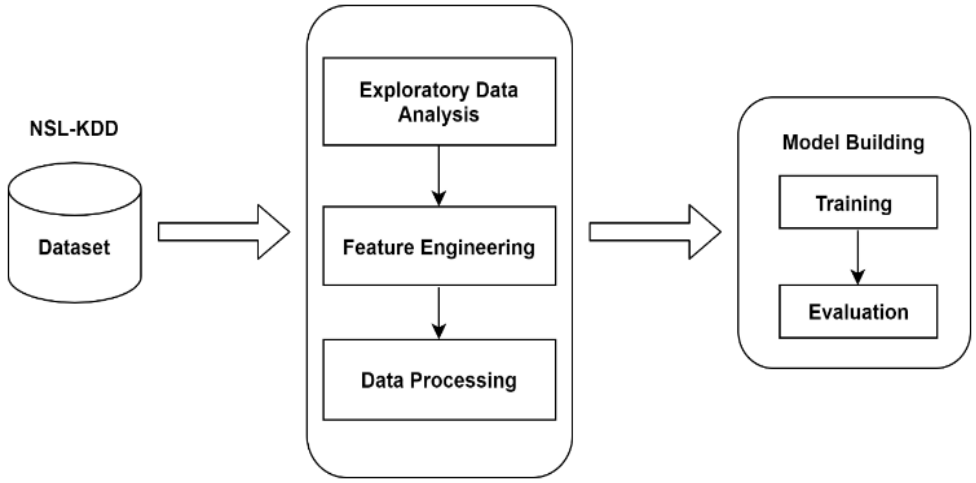

Fig. 1. The proposed Intrusion Detection System architecture

3.1 Importation of Dataset

The NSL-KDD dataset used was obtained from Kaggle verified source NSL-KDD – Version 1 (uploaded by Hassan06, 55.87 MB). This version corresponds to the official NSL-KDD release provided by the University of New Brunswick, ensuring source reliability. The KDDTrain+.TXT and KDDTest+.TXT files were used for the study as selected as they contain all 41 network traffic features and the associated class labels. These files include four major attack categories relevant to intrusion detection research: Denial of Service (DoS), Probe, User-to-Root (U2R), and Root-to-Local (R2L). NSL-KDD was used as it fixes major limitations found in the KDD'99 dataset.

3.2 Data Loading and Preprocessing.

The NSL-KDD dataset was used to develop and evaluate the model. The training and testing sets were first combined to ensure consistent preprocessing. Column names were standardized for clarity, and an exploratory analysis showed there were no missing values. However, the dataset had class imbalance, which could skew model predictions toward the more common classes. The categorical variables in the dataset were encoded using Label Encoding. This process converts each category into a corresponding integer. Label encoding was necessary to transform the categorical variables into numerical form that machine learning algorithms could process. After preprocessing, the dataset was split back into training, validation, and testing subsets while keeping the original distribution for a fair evaluation. Initial experiments with a baseline neural network showed limited performance. This highlighted the need for better preprocessing and feature representation. A second preprocessing pass ensured consistency between the training and test classes. It also updated feature encodings and reinforced scaling, preparing the dataset for effective training of the proposed Deep Neural Network (DNN).

3.3 Feature Engineering

For dimensionality reduction, Principal Component Analysis (PCA) was applied to further reduce the noise, and the simplify dataset enhancing computational efficiency. Components retaining 75% and 90% of the variance achieved accuracies of 84% and 84.5%, respectively. Using only the first 10 principal components, on the other hand, achieved an accuracy of 86%, which accounted for approximately 29% of the total variance. This proves that a few components can retain critical information for classification and thus reduce this dataset without compromising performance. While retaining more components preserves additional variance, it also increased computational complexity.

3.4 Modelling

The modeling process in this study focused on employing a deep learning algorithm as the primary tool for classifying network traffic within the NSL-KDD dataset. To evaluate the deep learning model's performance, nine other machine learning algorithms were also applied. These algorithms were used primarily to establish baseline performance, which helped in comparing the deep learning model's efficiency. Through the use of these various algorithms, the study aimed to validate the deep learning approach and assess its effectiveness in classifying network traffic into multiple categories. The machine learning models analyzed included Logistic Regression, Support Vector Classifier (SVC), Decision Tree Classifier, K-Nearest Neighbors Classifier (KNeighborsClassifier), Random Forest Classifier, Multi-Layer Perceptron Classifier (MLPClassifier), Extreme Gradient Boosting Classifier (XGBClassifier), Light Gradient Boosting Machine Classifier (LGBMClassifier), and CatBoost Classifier.

The Deep Neural Network model developed in this project consists of three layers: an input layer, two dense hidden layers, and an output layer. The model consists of 47,959(187.34KB) total parameters, out of which

47,959(187.34KB) are trainable and 0 (0.00B) is non-trainable. These parameters highlight the model's design efficiency, with all parameters being trainable, ensuring the network's ability to fully learn from the dataset.

3.5 Deep Learning Model Architecture

This model is built with 47,959 trainable parameters all of which are trainable, allowing the model to fully adapt and learn from the complex patterns within the dataset. The architecture consists of three Dense hidden layers with 64, 128, and 256 neurons, respectively, allowing the network to learn progressively more abstract representations of network traffic. Each hidden layer uses the ReLU activation function for efficient non-linear modeling. ReLU is computationally efficient and helps mitigate the vanishing gradient problem commonly observed in deep networks. It is defined mathematically as:

$$ReLU(x) = max(0, x) \tag{1}$$

This means that the function outputs 0 for negative inputs and returns the input value x for positive inputs.To reduce overfitting, a dropout rate of 30% was applied after each hidden layer

Table 1. Model architecture

Layer	Units	Activation	Dropout
Input		–	–
Hidden 1	64	ReLU	0.3
Hidden 2	128	ReLU	0.3
Hidden 3	256	ReLU	0.3
Output		Softmax	–

The deep learning model in this study, was designed utilizing Keras Functional API which provides flexibility in designing complex network architectures. The deep neural network (DNN) is optimized for multiclass classification tasks, with layers designed to effectively identify the non-linear relationships present in the data.

The output layer applies a Softmax activation function, producing a probability distribution across all attack categories for multiclass classification. The model was compiled with the Adam optimizer and trained using sparse categorical cross-entropy, both of which are well suited for multiclass intrusion-detection tasks.

3.6 Training the Deep Learning Model

The model was compiled and trained using parameters such as optimizer, loss function, and other training techniques. This model was developed using the Adam (Adaptive Moment Estimation) optimizer, which, according to the authors, is considered the best optimizer. It is a commonly used optimization algorithm in deep learning that leverages the benefits of AdaGrad and RMSProp methods. Adam ensures that the model's weight updates are more efficient, particularly in situations where data is sparse or noisy. These characteristics are common in datasets like NSL-KDD, where network traffic patterns can vary significantly. The model employs sparse categorical cross-entropy as its loss function, which is widely used for multiclass classification problem analysis, especially when target labels are integer-encoded. By using this loss function, the model can better differentiate between the various classes of network traffic, whether normal or a specific type of attack. The model was trained for 20 epochs with a batch size of 32, which strikes a balance between efficient GPU matrix operations and frequent weight updates during training. Training accuracy steadily increased from 92.50% to 98.62%, while training loss decreased from 0.3031 to 0.0487, indicating effective learning and optimization. Similarly, validation accuracy rose from 97.01% to 99.14%, and validation loss dropped from 0.0759 to 0.0315, suggesting the model's strong capability to generalize unseen data. By the 20th epoch, the model reached peak performance, indicating convergence.

To further reduce the risk of over fitting, early stopping was implemented. This approach is essential to prevent models from memorizing training data and instead learning to generalize. In addition to early stopping, learning rate reduction was implemented to further refine the model's learning process. When the validation performance plateaued, the learning rate was reduced, allowing the model to make smaller, more precise updates to the weights. This strategy prevents the model from overshooting the optimal set of weights and helps it refine its learning as training progresses.

These combined techniques enabled the model to efficiently learn and generalize well to unseen data, making it effective for network intrusion detection.

4 Result and Discussion

The results of this prediction experiment, after applying the test dataset, were expressed through a comparison between traditional machine learning models and the proposed deep learning model, along with the confusion matrix and associated performance metrics.

4.1 Comparison between traditional machine learning models and the proposed deep learning model

Among the traditional models analyzed in this study, the K Neighbors Classifier performed the best, achieving an accuracy of 84.99%, while Logistic Regression had the lowest performance across all metrics. Table 4 compares the performance of traditional machine learning models with the proposed deep learning model on the test dataset. All models—including Logistic Regression, SVC, KNN, and the proposed deep learning model—were evaluated under identical experimental conditions The proposed model outperformed all traditional models in key metrics, achieving the highest accuracy (86%). This demonstrates its effectiveness in accurately detecting network intrusions.

Table 2. Performance Comparison between some of the traditional machine learning models and the proposed deep learning model

Model	Accuracy	Precision	F1 Score	Recall
Logistic Regression	0.824678	0.427007	0.355098	0.384390
SVC	0.831063	0.455292	0.435936	0.487023
K Neighbors Classifier	0.849952	0.547379	0.466305	0.499209
Proposed Model	0.86	0.77	0.81	0.86

4.2 Confusion Matrix for Developed Intrusion Detection System

A confusion matrix is used to visualize the classification performance of the model. The confusion matrix in Fig. 2 assesses the performance of the multiclass intrusion detection system in classifying network traffic as normal or various attack types. Each row represents actual labels, while each column represents predicted labels, with off-diagonal values indicating misclassifications. The model effectively detected frequent attacks like neptune and guess_passwd but struggled with rarer attacks such as buffer_overflow and rootkit. The confusion matrix illustrates the distribution of False Positives, False Negatives, True Positives, and True Negatives, providing insight into the model's classification performance. Table 5. shows the extracted values of the confusion Matrix Arrangement for Classification using the test Dataset.

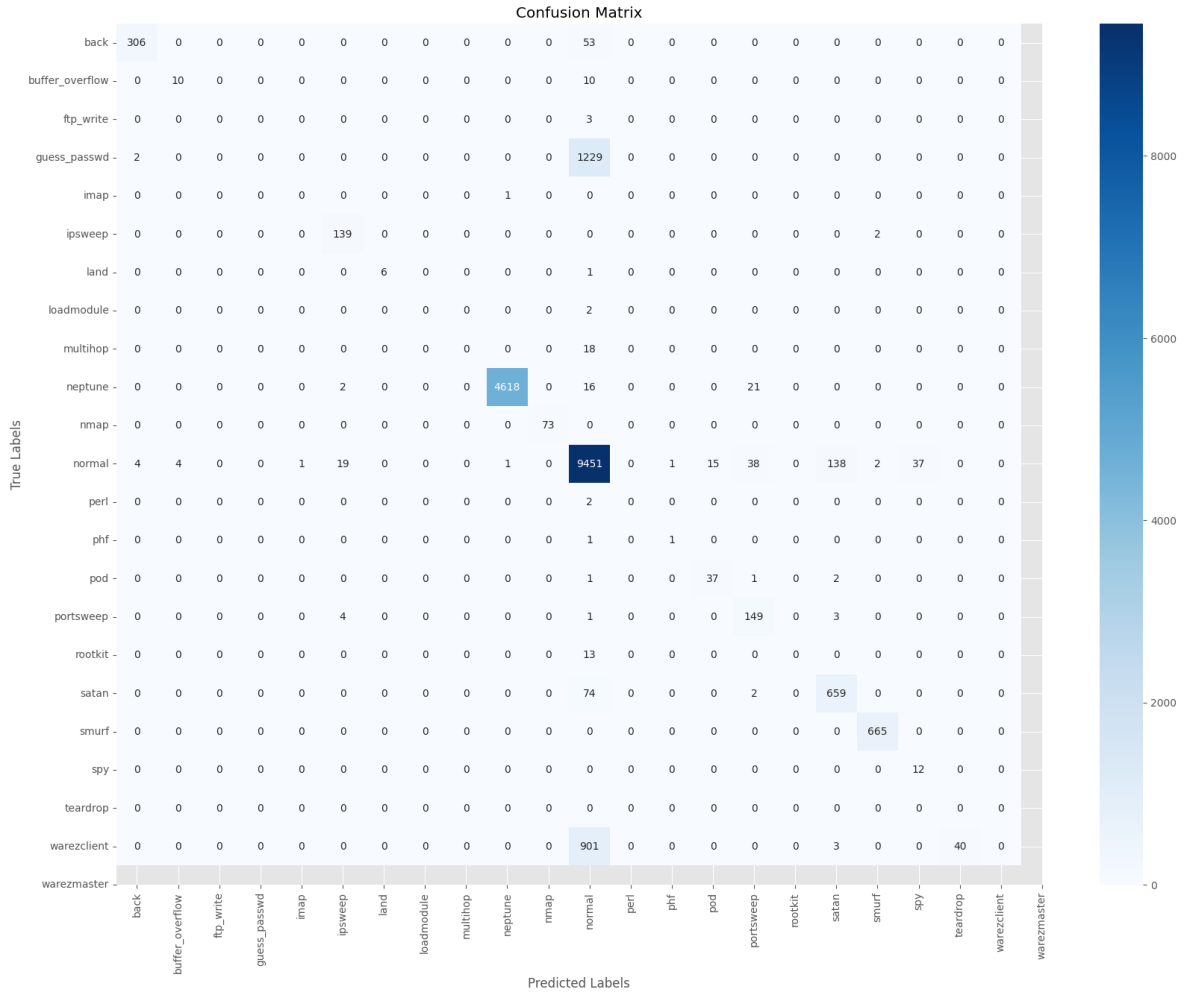

Fig. 2. Confusion Matrix for Classification

Table 3. Confusion Matrix Arrangement for Classification with Test Dataset

Class Type	Class	Attack Type	Total Class Samples Present	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)
Attack	back	DOS	359	306	15834	53	53
Attack	buffer_overflow	U2R	20	10	16179	0	10
Attack	ftp_write	R2L	8	3	16186	0	5
Attack	guess_passwd	R2L	1229	1229	14960	0	0
Attack	imap	R2L	1	0	16189	0	1
Attack	ipsweep	Probe	139	139	15965	0	0
Attack	land	DOS	6	6	16194	0	0
Attack	loadmodule	U2R	2	0	16193	0	2
Attack	multihop	R2L	18	0	16175	0	18
Attack	neptune	DOS	4637	4618	11536	0	19
Attack	nmap	Probe	73	73	15928	0	0
Normal	normal	Normal	9647	9451	6536	160	196
Attack	perl	U2R	2	0	16193	0	2
Attack	phf	U2R	4	0	16195	0	4
Attack	pod	DOS	38	37	16158	1	1
Attack	portsweep	Probe	153	149	15945	0	4
Attack	rootkit	U2R	13	13	16135	0	0
Attack	satan	Probe	733	659	15351	0	74
Attack	smurf	DOS	665	665	15325	0	0
Attack	spy	R2L	12	0	16178	0	12
Attack	teardrop	DOS	58	58	16132	0	0
Attack	warezclient	R2L	941	901	15186	0	40
Attack	warezmaster	R2L	20	20	16166	0	0

4.2.1 True Negatives

This implies that the classifier correctly identifies normal traffic as normal. The **True Negative Rate (TNR)** measures the proportion of normal instances that are correctly classified. It is defined mathematically as:

$$TNR = TN / (TN + FP) \quad (2)$$

Where:

- TN = True Negatives (correctly predicted normal samples)
- FP = False Positives (normal samples incorrectly classified as attack)

From Fig. 2, for the *back* attack class, the values are given as: TP =306, FP =6, FN= 53 and the total test samples N=16200

The True Negatives are calculated as:

$$TN = N - (TP + FP + FN) \quad (3)$$
$$TN = 16{,}200 - (306 + 6 + 53)$$
$$TN = 15{,}835$$

Substituting into Equation (2) results in:

$$TNR = 15{,}835 / (15{,}835 + 6)$$
$$TNR \approx 0.9996\ (99.96\%)$$

4.2.2 False Negatives

False negative implies that the classifier failed to identify an anomaly but classified it as a normal traffic. The False Negative Rate (FNR) is given mathematically defined as

$$FNR = FN / (FN + TP) \quad (4)$$

From Fig. 2, for the *back* attack class, the values are given as: TP =306, FP =6, FN= 53 and the total test samples N=16200

$$FNR = 53 / (53 + 306)$$
$$FNR \approx 0.1476\ (14.76\%)$$

4.2.3 False Positives

This implies that the classifier failed to identify a normal traffic but classified it as an attack. False Positive Rate (FPR) is mathematically represented as the ratio of incorrectly classified normal traffic instances (i.e., as attack) to the actual total number of normal traffic instances. It is defined mathematically as:

$$FPR = FP / (FP + TN) \quad (5)$$

From Fig. 2, for the *back* attack class, the values are given as: TP =306, FP =6, FN= 53, TN= 15834 and the total test samples N=16200

$$FPR = 6 / (6 + 15{,}834)$$
$$FPR \approx 0.00038\ (0.038\%)$$

4.2.4 True Positives

True positive implies that the classifier correctly identified attacks. The True Positive Rate (TPR) in mathematical form is given as the ratio between the number of the correctly predicted attacks and total number of samples that represent attacks. The TPR is defined mathematically as:

$$TPR = TP / (TP + FN) \quad (6)$$

From Fig. 2, for the *back* attack class, the values are given as: TP =306, FP =6, FN= 53, TN= 15834 and the total test samples N=16200

$$TPR = 306 / (306 + 53)$$
$$TPR \approx 0.8524\ (85.24\%)$$

4.3 Performance Analysis of DoS Attack Detection

The model demonstrates high effectiveness in detecting Denial-of-Service (DoS) attacks, achieving an overall accuracy of 98.73% across all DoS attack instances. It successfully identifies land, smurf, and teardrop attacks with 100% accuracy, while neptune and pod attacks also show high detection rates (99.6% and 97.4%, respectively). Table 4 provides a breakdown of the model's detection performance across multiple DoS attack classes, measuring True Positives (TP), False Positives (FP), False Negatives (FN), True Negatives (TN), and individual accuracy rates for each attack type.

Table 4. Analysis of DOS Attacks

Attack Class	Total Samples (Present)	True Positive (TP)	False Positive (FP)	False Negative FN)	True Negative TN)	Accuracy: True Positive/Total Samples
back	359	306	53	53	15834	85.2%
land	6	6	0	0	16194	100%
neptune	4637	4618	0	19	11536	99.6%
pod	38	37	1	1	16158	97.4%
smurf	665	665	0	0	15325	100%
teardrop	58	58	0	0	16132	100%
DOS - TOTAL	**5763**	**5690**	**54**	**73**		**98.73%**

4.4 Performance Metrics

After completing the model training, its effectiveness in detecting network intrusions was evaluated using key performance metrics. Fig. 3 presents the classification report, which provides a detailed breakdown of the model's performance. The following subsections analyse the core evaluation metrics—Precision, Recall, F1-Score, and Accuracy—highlighting their significance in assessing the reliability of the intrusion detection system.

4.4.1 Precision

The model achieved a precision of 77%, meaning that 77% of all predicted attack instances were indeed actual attacks. This implies that the model has a moderate level of false positives (FP) and suggests that some network traffic can mistakenly be classified as an attack.

4.4.2 Recall

The model detects 86% of all actual attack instances, meaning it has a strong ability to capture malicious activities. For intrusion detection systems, high recall is critical, as failing to detect an attack (False Negatives) can lead to security breaches.

4.4.3 F1-score

The F1-Score of 81% shows a good trade-off between precision and recall. It indicates that the model is reasonably effective in both detecting attacks (recall) and minimizing false positives (precision). This also suggests that the model is well-calibrated for intrusion detection, ensuring that it not only detects a high percentage of attacks but also reduces unnecessary false alarms. This balance is essential for maintaining an efficient and effective cyber security monitoring system.

4.4.4 Accuracy

The model correctly classifies 86% of the total test dataset, meaning that 86% of all network affic instances were correctly classified. An accuracy of 86% suggests that the model performs well overall, though there is still room for improvement in fine-tuning its ability to differentiate between attack and normal traffic.

Table 5. Performance Metrics of the deep learning model

PERFORMANCE METRICS	VALUE in %
Accuracy	86
Precision	77
F1-Score	81
Recall	86

```
In [70]: tools.evaluate_classification_model(test_new_Y, y_pred, class_indices)

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.85      0.91       359
           1       0.71      0.50      0.59        20
           2       0.00      0.00      0.00         3
           3       0.00      0.00      0.00      1231
           4       0.00      0.00      0.00         1
           5       0.85      0.99      0.91       141
           6       1.00      0.86      0.92         7
           7       0.00      0.00      0.00         2
           8       0.00      0.00      0.00        18
           9       1.00      0.99      1.00      4657
          10       1.00      1.00      1.00        73
          11       0.80      0.97      0.88      9711
          12       0.00      0.00      0.00         2
          13       0.50      0.50      0.50         2
          14       0.71      0.90      0.80        41
          15       0.71      0.95      0.81       157
          16       0.00      0.00      0.00        13
          17       0.82      0.90      0.86       735
          18       0.99      1.00      1.00       665
          20       0.24      1.00      0.39        12
          21       0.00      0.00      0.00         0
          22       0.00      0.00      0.00       944

    accuracy                           0.86     18794
   macro avg       0.47      0.52      0.48     18794
weighted avg       0.77      0.86      0.81     18794
```

Fig. 3. Classification report of the proposed model

4.5 Discussion of Results

The results presented in Table 3 illustrate the performance of the model in classifying both normal and attack traffic from the 22,544 samples in the test portion of the NSL-KDD dataset. The model demonstrated great capability to distinguish between normal and malicious traffic, highlighting its potential effectiveness in intrusion detection systems.

The model showed a strong ability to identify both normal and attack traffic. From the test samples, out of 9,647 normal samples, 9,451 were correctly classified as normal samples representing the true positives for the normal class.

Also, a total of 8,886 samples were accurately identified as attacks, reflecting the true positives for the attack class. This performance underscores the model's robust detection capabilities, particularly in correctly identifying attack samples, resulting in a high detection rate. However, 160 normal samples were misclassified as attacks, resulting in false positives. Although this number is small relative to the total normal samples, it indicates a slight tendency to misclassify benign traffic. Nonetheless, the model's ability to maintain a low false positive rate is beneficial, as it helps minimize the occurrence of false alarms, a critical factor for effective intrusion detection. Conversely, 245 attack samples were misclassified as normal, indicating the presence of false negatives. While the proportion of false negatives is relatively small, addressing these misclassifications would further improve the model's reliability in detecting a wider range of attacks.

For the Denial-of-Service (DoS) attack classification as seen in Table 4, the model performed exceptionally well, achieving a 98.73% accuracy rate. Out of 5,763 DoS attack samples, 5,690 were correctly classified, indicating a high number of true positives. This result highlights the model's proficiency in recognizing DoS attacks, which are a prominent type of attack in cybersecurity and a key focus of this study. Only 73 DoS samples were misclassified as other attack types, resulting in false negatives. Despite these misclassifications, the model's proficiency in accurately detecting DoS attacks demonstrates a high level of effectiveness and reliability in handling this specific category of attacks.

The proposed model's performance was compared to other machine learning models, including Logistic Regression, Support Vector Classifier (SVC), and K-Nearest Neighbors (KNeighbors Classifier), using metrics such as accuracy, precision, recall, and F1-score. As shown in Table 2, the proposed model outperformed the machine learning models across all metrics, demonstrating its superior effectiveness.

The model showed consistent improvement over 20 epochs, with steady improvements in both training and validation accuracy. By the 20th epoch, the model's performance reached its peak, suggesting it had converged. Training accuracy improved from 92.50% to 98.62%, while training loss decreased from 0.3031 to 0.0487 indicating effective learning and optimization. Validation accuracy increased from 97.01% to 99.14% while validation loss dropped from 0.0759 to 0.0315, demonstrating the model's ability to generalize effectively without over fitting.

The performance metrics as shown in Table 5, provides deeper insights into the model's capabilities. An accuracy of 86% signifies that the model correctly classified 86% of the samples across all categories, reflecting its effectiveness in differentiating between normal and attack traffic with a high degree of reliability. The precision of 77% reflects the model's ability to correctly identify attack samples out of all the samples it classified as attacks. While this is relatively good, it also indicates room for improvement in reducing false positives. The recall value of 86% demonstrates the model's ability to detect 86% of all actual attack samples, ensuring a high detection rate, making it valuable for detecting and preventing potential intrusions. The F1-score of 81% is the harmonic mean of precision and recall, this indicates a balanced measure of the model's overall performance.

The model has demonstrated outstanding performance in classifying normal and malicious traffic, with particularly remarkable results in detecting DoS attacks. This demonstrates the strong capability of the model in detecting DoS attacks, accurately classifying 5,690 out of 5,763 DoS samples, with an accuracy of 98.73%. On the other hand, the overall test accuracy of 86% exposes weaker generalization across other attack types, evidenced by misclassifications in the confusion matrix.

Summarily, the model's capability to achieve a low false positive rate alongside high accuracy in DoS attack detection underscores its suitability for real-world deployment in intrusion detection systems. Key performance metrics, such as high accuracy, recall, and F1-score, further validates the model's effectiveness in reliably identifying both benign and malicious traffic.

5 Conclusion

This project developed a Deep Neural Network (DNN) model to detect Denial of Service (DoS) attacks in computer networks using the NSL-KDD dataset. It also considered other attack types (User-to-Root, Root-to-Local, and Probe) and benign data. The model performed excellently, achieving a high accuracy, precision, recall, and F1-score, especially for DoS attack detection. It reached optimal accuracy at the 20th epoch, with low false positive and negative rates. Despite these, there are some limitations. The model's performance on modern or real-world datasets remains untested, limiting external validity. The computational cost of training and inference also raises concerns regarding real-time feasibility in operational environments. Also, despite high DoS detection accuracy, the model showed reduced performance for less frequent attack types. Real-world deployment would require continuous model updates, integration with existing network infrastructure, and resilience against evolving attack patterns. These considerations outline key avenues for future research, while promising, the current model represents an early step toward a fully deployable intrusion detection system.

Acknowledgment

The authors appreciate the staff and students of the Department of Computer Engineering at the University of Benin for their technical assistance, as well as the anonymous reviewers for their valuable feedback. They also thank their families for their love and encouragement and give thanks to God for strength, wisdom, and perseverance in completing the journal.

References

[1] M. Mittal, K. Kumar, and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” Soft Computing, vol. 27, pp. 13039–13075, 2023, doi: 10.1007/s00500-021-06608-1.
[2] National Cyber Security Centre, Denial of Service (DoS) Guidance Collection, ver. 1.0, rev. Mar. 25, 2024. [Online]. Available: https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection.
[3] V. Venkatesh and S. Rao, “Impact analysis of DDoS attacks on enterprise networks,” SN Computer Science, vol. 3, article 145, 2022.
[4] H. Singh and H. Sharma, “Intrusion detection techniques: A review,” International Journal of Computer Applications, vol. 182, no. 25, 2020.
[5] Kaspersky, “Africa Cyberthreat Landscape Report 2025,” Kaspersky, Apr. 2025. [Online]. Available: https://content.kaspersky-labs.com/se/media/en/africa-cyberthreat-landscape-report-2025.pdf (accessed Oct. 29, 2025).
[6] GMI Cloud, “Deep Learning: The Power Behind Modern AI Systems,” GMI Cloud. [Online]. Available: https://www.gmicloud.ai/glossary/deep-learning. [Accessed: Nov. 22, 2025].
[7] A. K. Silivery, K. Ram, and L. K. Suresh Kumar, “An Effective Deep Learning Based Multi-Class Classification of DoS and DDoS Attack Detection,” International Journal of Electrical and Computer Engineering Systems (IJECES), vol. 14, no. 4, pp. 421–431, Apr. 2023.
[8] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528..
[9] A. A. Salih, S. Y. Ameen, S. R. M. Zeebaree, M. A. M. Sadeeq, S. F. Kak, N. Omar, I. M. Ibrahim, H. M. Yasin, Z. N. Rashid, and Z. S. Ageed, “Deep Learning Approaches for Intrusion Detection,” Asian Journal of Research in Computer Science, vol. 9, no. 4, pp. 50–64, 2021.
[10] D. Ajalkar, V. Chavan and P. Bhosle, “Machine Learning Based Intrusion Detection System,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 5, no. 10, Apr. 2025, doi: 10.48175/IJARSCT-25679.
[11] M. Ramzan, M. Shoaib, A. Altaf, S. Arshad, F. Iqbal, Á. K. Castilla, and I. Ashraf, "Distributed Denial of Service attack detection in network traffic using deep learning algorithm," *Sensors*, vol. 23, no. 20, p. 8642, 2023, doi: 10.3390/s23208642.
[12] S. K. Alladi, "Effectively improving the efficiency and performance of an intrusion detection system using hybrid machine learning models," M.S. thesis, School of Computing, National College of Ireland, 2020.
[13] O. Edosa, A. E. Ibhaze, E. C. Ekoko, and P. E. Orukpe, "Development of an intrusion detection system leveraging deep learning model classification," Advances in Knowledge Based Systems, Data Science, and Cybersecurity, vol. 1, no. 1, pp. 38–48, 2024.
[14] S. Rawat, A. Srinivasan, V. Ravi, and U. Ghosh, "Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network," *Internet Technology Letters*, vol. 5, no. 1, p. e232, 2022.

Authors’ Profiles

Obiageli M. Attoh earned a B.Eng. in Electrical/Electronics Engineering and an M.Eng. in Computer Engineering from the University of Benin, Benin City, Nigeria. She also holds a Postgraduate Diploma in Educational Planning and Administration from the London College of Teachers and is currently pursuing a Ph.D. in Computer Engineering. She is a Lecturer at Dennis Osadebay University, Asaba, Nigeria, and has participated in various professional and community-based initiatives aimed at advancing engineering education and technology development. Her research contributions and professional engagements focus on cybersecurity, digital innovation, and educational technology. Obiageli is a member of the Council for the Regulation of Engineers (COREN), the Nigerian Society of Engineers (NSE) and the Association of Professional Women Engineers of Nigeria (APWEN), and she remains dedicated to mentoring the next generation of engineers.

Oduware Okosun received the B.Eng., M.Eng., and Ph.D. degrees in Computer Engineering from the University of Benin (UNIBEN), Benin City, Nigeria. Her major field of study is Computer Engineering. She has served in various academic and administrative positions, including her current role as Senior Lecturer and Head of the Computer Engineering Department (2023–2025). She has also contributed to several research projects and published articles in peer-reviewed journals. Her current research interests include wireless sensor networks, cloud computing, telecommunications engineering, and optimization techniques. Dr. Okosun is a member of the Council for the Regulation of Engineers (COREN), the Nigeria Society of Engineers (NSE), and the Association of Professional Women Engineers of Nigeria (APWEN). She has received recognition for her contributions to

engineering education and research, and actively participates in professional committees and editorial boards within the field.

How to cite this paper: Obiageli M. Attoh, Oduware Okosun, "Development of a Deep Learning Model for Detecting DOS Attacks in Computer Networks", International Journal of Engineering and Manufacturing (IJEM), Vol.16, No.1, pp. 39-49, 2026. DOI:10.5815/ijem.2026.01.04