Comparative Study of Learning Models for DDoS Traffic Detection Based on Network Flow Features

Zuming Liang
Communication Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract: The growing complexity of cyberattack behaviors has made network traffic classification a critical task in intelligent intrusion detection. This study investigates the adaptability of different learning architectures for Distributed Denial-of-Service (DDoS) traffic detection based on structured flow-level statistical features from the CIC-IDS2017 dataset. Three representative models, namely Multi-Layer Perceptron (MLP), Convolutional Neural Network-Long Short-Term Memory-Attention (CNN-LSTM-Attention), and Extreme Gradient Boosting (XGBoost), were implemented and evaluated under a consistent preprocessing and experimental protocol. Accuracy, Precision, Recall, F1-score, Receiver Operating Characteristic (ROC) curve, and Area Under the Curve (AUC) were used to compare overall detection performance and class-level behavior. Experimental results show that XGBoost achieved the best performance, with an accuracy of 0.9999 and an AUC of 1.0000. The MLP model also obtained highly competitive results, reaching an accuracy of 0.9997 and an AUC of 1.0000. In contrast, the CNN-LSTM-Attention model showed limited effectiveness, with an accuracy of 0.5752 and an AUC of 0.4956, because the structured statistical features did not provide meaningful temporal dependencies for sequence learning. These results indicate that model performance is strongly related to the compatibility between feature representation and learning mechanism. Under the current CIC-IDS2017 binary DDoS setting, tabular learning models such as XGBoost and MLP showed better suitability for structured flow-level features than the tested sequence-based architecture.

Keywords: Intrusion Detection System (IDS); Network Traffic Classification; Deep Learning; XGBoost; CNN-LSTM-Attention; Multi-Layer Perceptron (MLP); Structured Traffic Features; DDoS Traffic Detection

1. Introduction

The continuous growth of online services, cloud infrastructures, and Internet of Things (IoT) applications has made network environments increasingly complex. This complexity has also provided more opportunities for large-scale cyberattacks. Among them, Distributed Denial-of-Service (DDoS) attacks are highly disruptive because they can flood target systems with massive malicious traffic and seriously affect service availability [1]. As a result, intrusion detection systems (IDS) are widely used to identify abnormal traffic patterns and support network security protection [2]. Traditional IDS methods usually depend on rule-based or signature-based detection strategies [3]. These approaches are useful for recognizing known attack behaviors, but their reliance on predefined patterns makes them less effective against unseen attacks and newly emerging variants [4].

Data-driven detection methods have therefore attracted increasing attention in intrusion detection and traffic classification research [5]. Unlike manually defined rules, machine learning and deep learning models can learn representative patterns directly from traffic data. Classical machine learning models, including Support Vector Machine (SVM), Random Forest (RF), Multi-Layer Perceptron (MLP), and XGBoost, have been applied to structured traffic feature classification and have achieved competitive results [6]. In particular, MLP is able to model nonlinear relationships through fully connected layers, whereas XGBoost is effective for tabular feature learning because of its gradient boosting mechanism and ability to capture complex feature interactions [7].

Deep learning architectures, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and attention mechanisms, have also been introduced into security detection studies [8]–[10]. Hybrid

 Copyright @ STEMM Institute Press

models such as CNN-LSTM-Attention have been reported to achieve competitive detection performance in related security classification tasks [11]. Nevertheless, many existing studies emphasize accuracy improvement while insufficiently considering whether the selected model architecture is consistent with the intrinsic structure of the input features.

In many intrusion detection datasets, traffic records are represented as flow-level statistical attributes rather than continuous packet sequences. These features are naturally structured as tabular data and do not necessarily contain explicit temporal dependencies. Although sequence-based models are designed for temporal representation learning, artificially constructed pseudo-sequences may fail to provide meaningful sequential information. As a result, increasing architectural complexity does not always lead to better classification performance, especially when the feature representation is not compatible with the learning mechanism.

Based on this consideration, this study compares different learning architectures for structured network traffic statistical feature classification. Three representative models, namely MLP, CNN-LSTM-Attention, and XGBoost are evaluated under a consistent preprocessing and experimental protocol. Model performance is assessed using Accuracy, Precision, Recall, F1-score, Receiver Operating Characteristic (ROC), Area Under the Curve (AUC), and confusion matrix analysis.

The main contributions of this study are as follows:

(a) A unified preprocessing procedure is established for structured network traffic statistical features, including data cleaning, duplicate removal, label encoding, numerical feature selection, and constant-feature removal.

(b) MLP, XGBoost, and CNN-LSTM-Attention are compared under the same preprocessing protocol and train-test split ratio to analyze their adaptability to structured flow-level features, with CNN-LSTM-Attention serving as a representative sequence-based comparison model.

(c) The experimental findings show that tabular learning models such as MLP and XGBoost are more suitable for the adopted structured traffic features, whereas sequence-based architectures may perform poorly when genuine temporal dependencies are absent.

2. Methodology

Figure 1 outlines the workflow used in this study for network traffic classification. The procedure consisted of four main stages: preprocessing, feature construction, model learning, and performance comparison. This study focused on the adaptability between feature structures and learning architectures under a unified experimental framework.

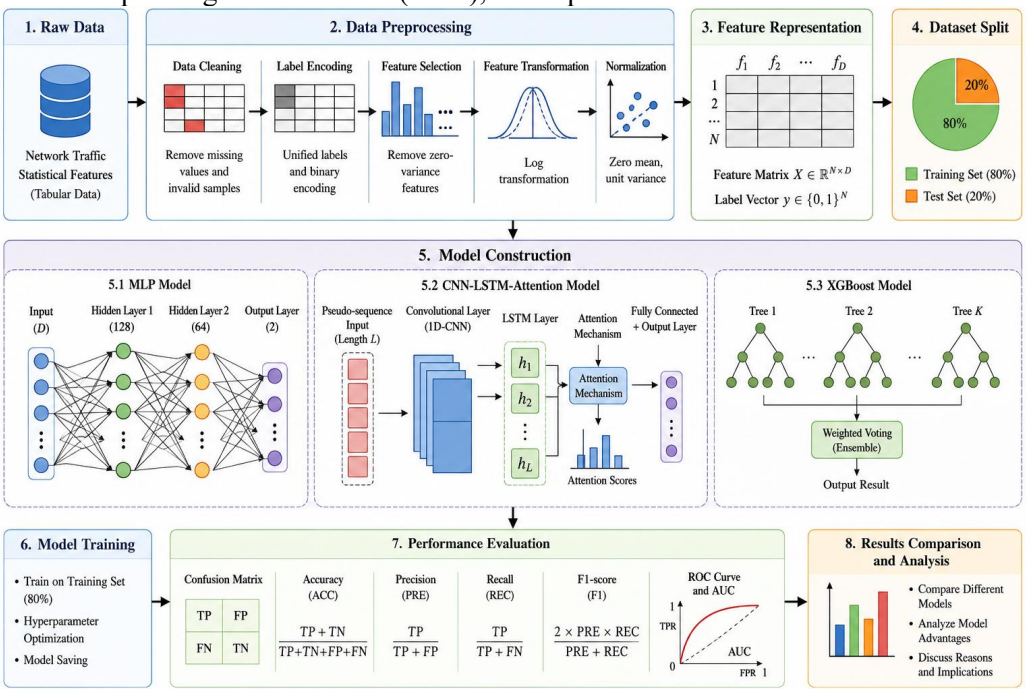

Figure 1. Overall Workflow of the Proposed Comparative Learning Framework for Structured Network Traffic Classification

2.1 Dataset Description

The dataset used in this study was derived from the CIC-IDS2017 dataset, which was released by the Canadian Institute for Cybersecurity for intrusion detection evaluation [12]. The dataset contains benign traffic and multiple common attack categories, and provides both raw PCAP files and flow-level CSV files generated using CICFlowMeter.

In this study, benign traffic and DDoS traffic samples were selected to construct a binary classification task. Each sample represented an independent network flow and consisted of structured flow-level statistical features. After duplicate removal, the final dataset contained 223,112 traffic samples with 68 valid flow-level statistical features. Unlike raw packet streams, these features belonged to structured tabular data rather than naturally ordered temporal sequences.

2.2 Data Preprocessing

Before training, the raw flow records were cleaned and converted into a form suitable for model input. Repeated flow records were discarded to reduce the possibility of duplicated samples affecting the train-test evaluation. Invalid samples containing missing or infinite values were also discarded, and traffic labels were converted into binary form. Features with zero variance were removed to reduce redundancy.

Only numerical features were retained, and constant-valued features were removed to reduce redundant information. Numerical features were standardized when required to improve optimization stability, especially for neural network-based models.

After preprocessing, the feature matrix was denoted as:

$X \in \mathbb{R}^{N \times D}$

where N denotes the number of traffic samples and D denotes the feature dimension. The corresponding label vector is represented as:

$y \in \{0,1\}^N$

2.3 Feature Representation

The input attributes described each network flow through statistical measurements rather than ordered packet-level sequences. Therefore, the adopted features were treated as structured tabular representations in this study.

For MLP and XGBoost, the processed feature vectors were directly used as model inputs. For the CNN-LSTM-Attention model, pseudo-sequential samples were constructed to satisfy the sequential input requirement. Specifically, pseudo-sequences were generated after the train-test split using a sliding-window strategy with a window length of 12 and a step size of 1. Each pseudo-sequence consisted of 12 consecutive flow-level feature vectors according to their order in the processed dataset. The label of each pseudo-sequence was assigned according to the label of the last flow sample within the window.

Since the selected CIC-IDS2017 records were flow-level statistical attributes rather than real packet-level temporal traces, the generated pseudo-sequences did not represent genuine temporal traffic evolution. Therefore, the CNN-LSTM-Attention model was included as a representative sequence-based comparison model to examine whether a commonly used sequence-learning architecture could still be effective when applied to pseudo-sequential representations of structured flow features.

2.4 Model Construction

Three representative learning architectures were investigated: MLP, CNN-LSTM-Attention, and XGBoost. These models were selected to represent fully connected neural learning, sequence-based deep learning, and tree-based ensemble learning, respectively.

The MLP model was employed as a fully connected neural network for structured feature learning. The CNN-LSTM-Attention model combined convolutional feature extraction, sequential dependency learning, and attention-based feature weighting, also it was included to examine whether a commonly used sequence-based architecture remains effective when applied to pseudo-sequential representations of structured flow-level features. XGBoost was adopted as a tree-based ensemble learning model for structured tabular feature classification

2.5 Experimental Settings

After preprocessing, 80% of the samples were used for training and the remaining 20% were used for testing. A consistent preprocessing procedure and split ratio were adopted for all models. For the CNN-LSTM-Attention model, the processed feature vectors were further transformed into pseudo-sequential inputs to meet the requirement of sequence-based learning. For the MLP model, two hidden layers with 128 and 64 neurons were used, and the model was

 Copyright @ STEMM Institute Press

trained for 50 epochs with a learning rate of 0.001. The CNN-LSTM-Attention model was trained for 20 epochs. For XGBoost, the number of estimators, maximum tree depth, and learning rate were set to 200, 6, and 0.1, respectively.

The neural network models were optimized using the Adam optimizer with cross-entropy loss. The XGBoost parameters were selected empirically according to the classification task. The experiments were carried out in Python using commonly used machine learning and deep learning toolkits.

2.6 Performance Evaluation

Model effectiveness was assessed from both threshold-based and class-level perspectives. Accuracy, Precision, Recall, F1-score, ROC curve, and AUC were used as quantitative indicators, while confusion matrices were examined to analyze the prediction distribution between benign and DDoS traffic samples.

3. Experimental Results and Analysis

3.1 Experimental Results Overview

This section compares the performance of MLP, XGBoost, and CNN-LSTM-Attention on structured network traffic statistical features. All models followed the same preprocessing protocol and 80:20 train-test split ratio, while model-specific input construction was applied when necessary.

Accuracy, Precision, Recall, F1-score, ROC curve, AUC, and confusion matrix were used to analyze model performance. These metrics reflect overall prediction accuracy, class-level classification balance, and discriminative ability.

Figures 2–5 report the MLP results, including accuracy, loss, ROC curve, and confusion matrix. Figures 6–7 show the ROC curve and confusion matrix of XGBoost. Figures 8–11 present the training behavior, ROC curve, and confusion matrix of CNN-LSTM-Attention.

The comparison reveals clear differences among the three models. XGBoost and MLP produced highly accurate results, while CNN-LSTM-Attention showed a strong tendency to predict the malicious class and failed to form effective decision boundaries. The following subsections analyze these results in terms of convergence behavior, discriminative capability, and classification balance.

3.2 MLP Results Analysis

The MLP model was first evaluated to investigate the effectiveness of fully connected neural networks for structured network traffic statistical feature classification.

Figures 2 and 3 show the accuracy and loss curves of the MLP model after duplicate removal and feature standardization. As shown in Figure 2, both training and testing accuracy stayed at a high level during the entire training process. The training accuracy increased from approximately 0.9984 to around 0.9997, while the testing accuracy followed a similar trend and remained close to the training curve. The small gap between the two curves indicates that the model achieved stable generalization without obvious overfitting.

Figure 3 further presents the training loss variation. The loss decreased sharply during the first few epochs and then gradually converged to a very low level. This trend indicates that the optimization process was stable and that the model parameters were effectively updated during training.

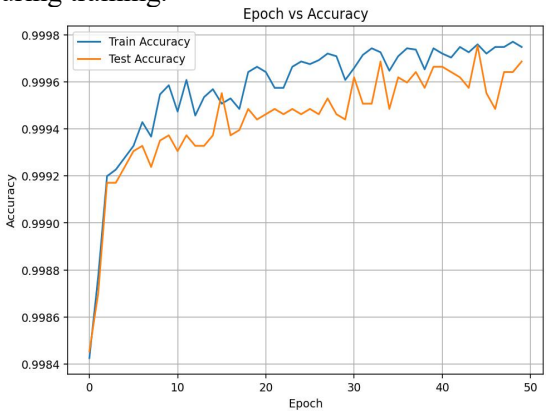

Figure 2. Accuracy Curve of MLP Model

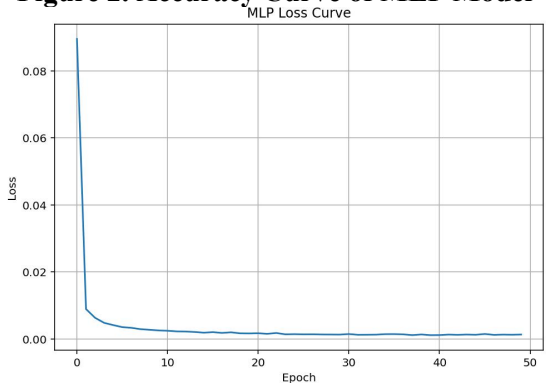

Figure 3. Loss Curve of MLP Model

The accuracy and loss curves jointly suggest that the MLP model can effectively learn discriminative relationships from standardized flow-level statistical features. The stable convergence also indicates that fully connected neural networks are suitable for structured

network traffic classification under the adopted preprocessing setting.

Beyond convergence behavior, the discriminative capability of the MLP model was further evaluated using ROC analysis. Figure 4 presents the ROC curve of the MLP model on the testing dataset. The ROC curve approached the ideal classification region, indicating that the model maintained a high true positive rate while keeping a very low false positive rate under different decision thresholds. The AUC value reached 1.0000, showing near-perfect separability between benign and malicious traffic samples.

To further investigate the detailed classification behavior, the confusion matrix of the MLP model is presented in Figure 5.

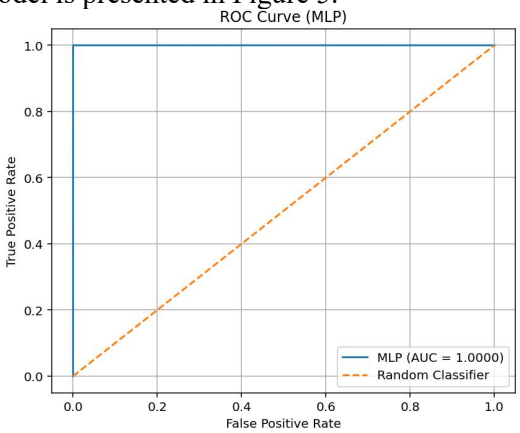

Figure 4. ROC Curve of MLP Model

The confusion matrix in Figure 5 reports the MLP model correctly classified 19,007 benign samples and 25,596 malicious samples. Only seven benign samples were misclassified as malicious traffic, and seven malicious samples were incorrectly identified as benign traffic.

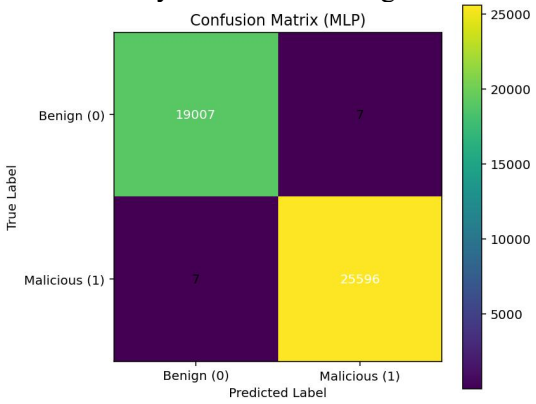

Figure 5. Confusion Matrix of MLP Model

The extremely small number of false positives and false negatives indicates that the MLP model achieved balanced classification performance for both traffic categories. This behavior also reflects that the model did not rely on a single dominant class, but learned effective decision boundaries from the structured flow-level features.

Overall, the MLP results show stable convergence and balanced classification on structured flow-level traffic features, indicating that fully connected networks can model tabular traffic data effectively.

3.3 XGBoost Results Analysis

The XGBoost model was further evaluated to examine the effectiveness of ensemble learning for structured network traffic statistical feature classification. The analysis focused on discriminative capability and detailed classification behavior.

The ROC result in Figure 6 shows that the curve approached the ideal classification region. This indicates that the model could distinguish benign and DDoS traffic with a high detection rate and few false alarms across different thresholds. The AUC reached 1.0000, indicating highly separable prediction scores for the two traffic classes.

This observation indicates that the adopted flow-level statistical features contained highly distinguishable structural patterns, which could be effectively captured by the gradient boosting mechanism of XGBoost.

To further examine its detailed classification behavior, the corresponding confusion matrix is analyzed in Figure 7.

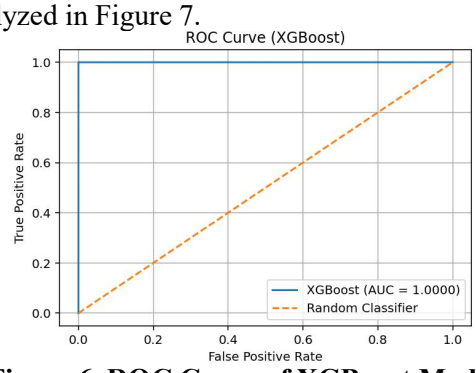

Figure 6. ROC Curve of XGBoost Model

Figure 7 shows the confusion matrix of the XGBoost model on the testing dataset. The model correctly identified 19,018 benign samples and 25,600 malicious samples. Only one benign sample was predicted as malicious, while four malicious samples were incorrectly classified as benign.

The very small numbers of false positives and false negatives indicate that XGBoost maintained balanced classification performance across both classes. In particular, the low false

 Copyright @ STEMM Institute Press

positive count suggests that the model introduced few unnecessary alarms for benign traffic, while the low false negative count indicates that most malicious samples were successfully detected.

Together with the ROC result in Figure 6, the confusion matrix suggests that XGBoost effectively exploited the separable patterns and nonlinear relationships within the structured flow-level statistical features.

Based on the above experimental results, the overall effectiveness of XGBoost on structured traffic statistical features is further discussed below.

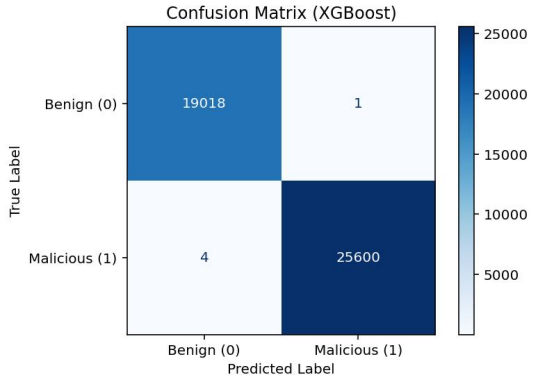

Figure 7. Confusion Matrix of XGBoost Model

The experimental results indicate that XGBoost achieved the best overall classification performance among the compared models. The model produced only five misclassified samples in the testing set, demonstrating highly accurate and balanced classification for both benign and malicious traffic categories.

The strong performance of XGBoost can be attributed to its compatibility with structured tabular traffic features. As a tree-based ensemble learning model, XGBoost can effectively capture nonlinear relationships among flow-level statistical attributes without relying on temporal dependency assumptions.

These results suggest that XGBoost was well suited to the feature representation adopted in this study. More importantly, the findings indicate that model selection should be guided by the characteristics of the input features, rather than by model complexity alone.

3.4 CNN-LSTM-Attention Results Analysis

The CNN-LSTM-Attention model was evaluated as a sequence-based comparison model to examine whether pseudo-sequential representations of structured flow-level features could support effective sequence learning.

Figures 8 and 9 present the accuracy and loss curves of the CNN-LSTM-Attention model during training. As shown in Figure 8, both training and testing accuracy remained almost unchanged throughout the training process. The training accuracy stabilized around 0.5735, while the testing accuracy stayed around 0.5752, which was close to the proportion of malicious samples in the testing set. This indicates that the model did not learn an effective decision boundary, but tended to follow a majority-class prediction pattern.

The curve in Figure 9 reflects a similar limitation from the optimization perspective. The loss decreased only slightly from approximately 0.684 to 0.6826 and then remained nearly stable. Although no severe numerical instability was observed, the limited loss reduction suggests that the model failed to extract useful discriminative information from the pseudo-sequential input.

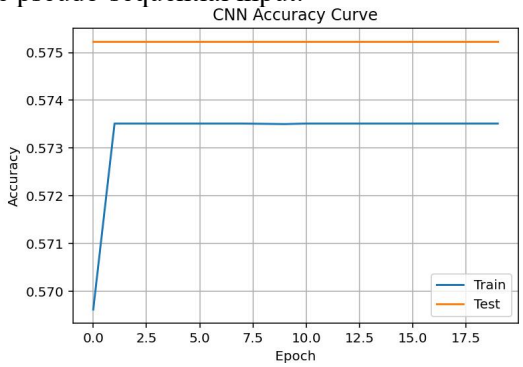

Figure 8. Accuracy Curve of CNN-LSTM-Attention Model

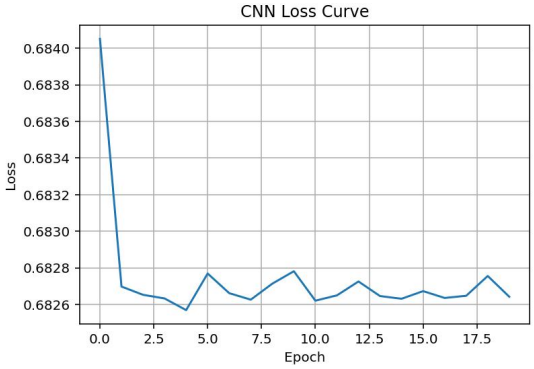

Figure 9. Loss Curve of CNN-LSTM-Attention Model

These results indicate that the artificially constructed pseudo-sequences did not provide meaningful temporal dependencies for the CNN-LSTM-Attention architecture. Therefore, the model's poor training behavior should be interpreted as evidence of architectural mismatch with the structured statistical feature representation.

Copyright @ STEMM Institute Press

Beyond convergence behavior, the discriminative capability of the CNN-LSTM-Attention model was further evaluated using ROC analysis.

Figure 10 presents the ROC curve of the CNN-LSTM-Attention model on the testing dataset. The curve almost overlaps with the diagonal reference line, indicating that the model had little discriminative ability between benign and malicious traffic samples.

The AUC value was 0.4956, which was close to random classification performance. This result shows that the prediction scores produced by the CNN-LSTM-Attention model could not provide effective class separation. Combined with the accuracy and loss curves shown in Figures 8 and 9, the ROC result further supports that the model failed to learn meaningful representations from the pseudo-sequential traffic features.

This weak ROC performance suggests that the constructed pseudo-sequences did not contain useful temporal dependencies for the sequence-based architecture.

To further examine the detailed prediction behavior of the model, the corresponding confusion matrix is analyzed in Figure 11.

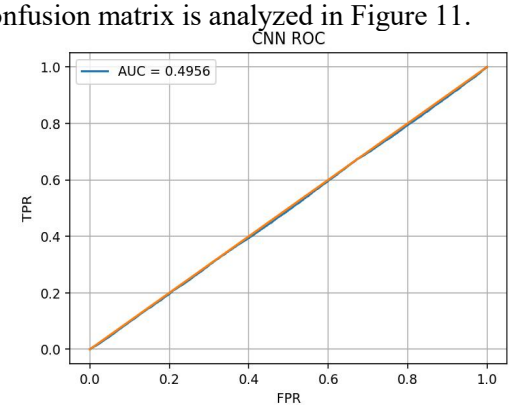

Figure 10. ROC Curve of CNN-LSTM-Attention Model

Figure 11 illustrates the confusion matrix of the CNN-LSTM-Attention model on the testing dataset. The model classified all testing samples as malicious traffic. Specifically, 25,662 malicious samples were correctly identified, whereas all 18,950 benign samples were misclassified as malicious traffic.

This result reveals a complete prediction bias toward the malicious class. Although the model achieved full recall for malicious traffic, it failed to recognize any benign traffic samples, leading to an extremely high false positive rate and severe classification imbalance.

Such behavior indicates that the CNN-LSTM-Attention model did not learn an effective decision boundary from the pseudo-sequential traffic representation. Instead, the model degenerated into a majority-class prediction pattern. This further suggests that the constructed pseudo-sequences were not suitable for representing the structured flow-level statistical features used in this study.

Combined with the accuracy, loss, and ROC analyses, the confusion matrix confirms that the CNN-LSTM-Attention architecture showed limited applicability to the current feature representation. Therefore, its performance should be interpreted as evidence of architectural mismatch rather than effective sequence-based classification.

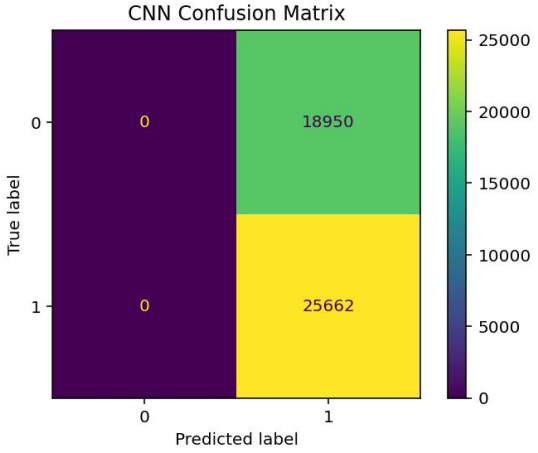

Figure 11. Confusion Matrix of CNN-LSTM-Attention Model

These results show that the CNN-LSTM-Attention model was not well matched with the current structured flow-level representation. The pseudo-sequence construction failed to provide useful temporal information, causing the model to collapse into a majority-class prediction pattern. Therefore, it suggests a clear mismatch between the pseudo-sequential representation and the sequence-based architecture under the current experimental setting.

3.5 Comparative Discussion

Table 1 summarizes the classification performance of the three models on the structured network traffic statistical feature dataset.

XGBoost obtained the best overall result, with an accuracy of 0.9999 and an AUC of 1.0000. Its macro-averaged Precision, Recall, and F1-score all reached 0.9999, showing highly balanced performance for both benign and malicious traffic samples.

The MLP model also achieved competitive performance, with an accuracy of 0.9997 and an

 Copyright @ STEMM Institute Press

AUC of 1.0000. The confusion matrix results further indicate that only a small number of samples were misclassified, suggesting that fully connected neural networks can effectively model discriminative relationships in standardized flow-level statistical features.

In contrast, the CNN-LSTM-Attention model showed limited effectiveness on the same preprocessing protocol and train-test split ratio. Its accuracy was 0.5752 and the AUC was only 0.4956, which was close to random classification. The confusion matrix showed that the model classified all testing samples as malicious traffic, indicating severe majority-class prediction bias and poor class balance.

Table 1. Performance Comparison of Different Models

Model	Accuracy	Precision	Recall	F1-score	AUC
MLP	0.9997	0.9997	0.9997	0.9997	1.0000
XGBoost	0.9999	0.9999	0.9999	0.9999	1.0000
CNN-LSTM-Attention	0.5752	0.2876	0.5000	0.3652	0.4956

Note: Precision, Recall, and F1-score are reported as macro averages. For the CNN-LSTM-Attention model, pseudo-sequential samples were generated using a sliding window after the train-test split; therefore, the effective number of evaluated instances may slightly differ from models using original two-dimensional feature vectors.

The near-perfect performance of XGBoost and MLP may also be related to the high separability of the selected DDoS and benign flow-level statistical features in CIC-IDS2017. The performance gap was mainly associated with the compatibility between feature representation and learning architecture. Since the adopted traffic features were structured flow-level statistical attributes rather than naturally ordered temporal sequences, under the current CIC-IDS2017 binary DDoS setting, tabular learning models such as XGBoost and MLP were more suitable for the adopted structured flow-level feature representation. The results also suggest that increasing model complexity does not necessarily improve intrusion detection performance when the input features do not provide meaningful temporal dependencies.

4. Conclusion

This study compared three learning architectures for DDoS traffic detection based on structured flow-level statistical features from the CIC-IDS2017 dataset. MLP, CNN-LSTM-Attention, and XGBoost were evaluated under a consistent preprocessing and experimental protocol.

The results showed clear performance differences among the three models. XGBoost achieved the best overall performance on the selected CIC-IDS2017 DDoS subset, with an accuracy of 0.9999 and an AUC of 1.0000. The MLP model also obtained highly competitive results, reaching an accuracy of 0.9997 and an AUC of 1.0000. In contrast, the CNN-LSTM-Attention model achieved only 0.5752 accuracy and 0.4956 AUC, and its confusion matrix showed that all testing samples were classified as malicious traffic.

These findings indicate that the effectiveness of a detection model is closely related to the compatibility between feature representation and learning mechanism. Since the adopted traffic records were structured statistical attributes rather than naturally ordered temporal sequences, tabular learning models such as XGBoost and MLP were better suited to this task. The poor performance of CNN-LSTM-Attention suggests that pseudo-sequence construction could not provide useful temporal information for sequence-based learning.

These findings indicate that architectural complexity alone is not sufficient to guarantee improved intrusion detection performance. For structured traffic feature classification, selecting a model that matches the feature structure is more important than simply adopting a more complex deep learning architecture.

These findings are limited to the selected CIC-IDS2017 benign/DDoS binary classification setting and should not be directly generalized to all intrusion detection datasets or attack categories without further validation.

Several limitations remain. The experiments were conducted on a single benchmark dataset, and the constructed pseudo-sequences could not fully represent real packet-level temporal behavior. Future work will evaluate the framework on more recent and diverse intrusion detection datasets, explore real sequential traffic representations, and investigate lightweight models for practical network security applications.

References

[1] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp.

39–53, 2004.
[2] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019.
[3] N. Masri, Y. A. Sultan, A. N. Akkila, A. Almasri, A. Ahmed, A. Y. Mahmoud, I. Zaqout, and S. S. Abu-Naser, “Survey of Rule-Based Systems,” International Journal of Academic Information Systems Research (IJAISR), vol. 3, no. 7, pp. 1–22, 2019.
[4] H. Arif, A. K. S. Ali, and H. A. Nabi, “IoT Security through ML/DL: Software Engineering Challenges and Directions,” ICCK Journal of Software Engineering, vol. 1, no. 2, pp. 90–108, 2025.
[5] S. Rezaei and X. Liu, “Deep learning for encrypted traffic classification: An overview,” IEEE Communications Magazine, vol. 57, no. 5, pp. 76–81, 2019.
[6] A. Verma and V. Ranga, “Machine Learning based Intrusion Detection Systems for IoT Applications,” Wireless Personal Communications, vol. 111, no. 4, pp. 2287–2310, 2020.
[7] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
[11] N. Wang and H. Kang, “An Enhanced Detection Method of Hardware Trojan Based on CNN-Attention-LSTM,” Computer Networks and Communications, vol. 2, no. 2, pp. 361–373, 2024.
[12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.

 Copyright @ STEMM Institute Press