ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

A Hybrid Transformer-Based Deep Neural Network for DDoS Detection: A Comparative Evaluation Across Modern Architectures

Nitipon Pongphaw[1], Mune Sukumaradat[2] and Prommin Buaphan[3]

ABSTRACT

DDoS attacks remain a major threat to network infrastructures. While deep learning is applied for detection, prior studies often lack standardized comparison and stability evaluation under consistent settings. This study systematically evaluates nine deep learning models including MLP, CNNs, ResNet1D, and attention-augmented architectures under consistent experimental settings and introduces two novel models: TDNN (Transformer-based) and ATDNN (attention-enhanced) for capturing complex traffic patterns. Using a balanced real-world dataset, all models were trained over five independent runs, with performance assessed via accuracy, precision, recall, and F1-score. TDNN achieved the highest performance (Accuracy: 0.9653 ± 0.0018; Precision: 0.9659 ± 0.0016; Recall: 0.9653 ± 0.0018; F1-score: 0.9653 ± 0.0018), while simpler models such as DNN, MLP, and LSTMClassifier also performed competitively with lower variance. The study further analyzes learning behaviors and evaluates deployment potential, highlighting that well-tuned deep learning models, particularly TDNN, can support real-time DDoS detection in enterprise and edge computing environments.

Article information:

Keywords: DDoS Detection, Deep Learning, Transformer, Attention Mechanism, Model Comparison, Cyber Threat Mitigation

Article history:
Received: August 6, 2024
Revised: September 25, 2025
Accepted: October 9, 2025
Published: October 18, 2025 (Online)

DOI: 10.37936/ecti-cit.2025194.263424

1. INTRODUCTION

Distributed Denial of Service (DDoS) attacks, which incapacitate servers and networks through overwhelming traffic, remain a critical cybersecurity challenge. Their increasing scale and sophistication, often coordinated via botnets, underscore the necessity for adaptive detection and mitigation strategies, as traditional approaches frequently fail to cope with the evolving attack landscape.

To address these challenges, researchers have increasingly explored deep learning techniques, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid architectures. These models can learn complex traffic patterns and distinguish malicious activity from normal traffic. Previous studies have shown that combining multiple deep learning architectures, incorporating attention mechanisms, or using distributed processing frameworks improves detection performance. However, many existing works have limitations, such as heterogeneous experimental settings, lack of standardized evaluation, insufficient analysis of model stability, and high computational cost, particularly for deployment in resource-constrained environments.

In this study, we present a rigorous and comprehensive evaluation of nine deep learning models for DDoS detection, including MLP, DNN, CNNs, CNNs1D_DNN, ResNet1D, ATRN, LSTMClassifier, TDNN, and ATDNN. We introduce two novel architectures: TDNN, which integrates a Transformer encoder, and ATDNN, which incorporates a feature-wise attention mechanism, designed to capture complex and sequential patterns in network traffic. All models are trained under identical conditions, with standardized preprocessing, feature selection, and normalization pipelines, and evaluated using multiple metrics, training dynamics, and confusion matrices. This work provides practical insights into developing

[1,2,3]The authors are with the Department of Electrical and Computer Engineering, Faculty of Science and Engineering, Kasetsart University Chalermphrakiat Sakon Nakhon Province Campus, Sakon Nakhon, Thailand, Email: nitipon.p@ku.th, mune.s@ku.th and prommin.b@ku.th
[1]Corresponding author: nitipon.p@ku.th

effective and deployable deep learning-based DDoS detection systems.

2. LITERATURE

2.1 Introduction to DDoS Attacks and the Importance of Deep Learning

DDoS attacks are one of the most disruptive forms of cyber threats, overwhelming target systems by injecting massive volumes of malicious traffic and denying service to legitimate users. The escalating frequency and sophistication of these attacks pose major challenges. Traditional intrusion detection systems that rely on static signatures or simple thresholds often fail to adapt to this dynamic environment, resulting in high false alarm rates and insufficient protection.

Recognizing these limitations, recent research emphasizes the importance of intelligent and adaptive detection mechanisms capable of continuously learning from traffic data. As highlighted in [1], anomaly-based detection that can capture patterns in large-scale flows is crucial for timely defense against evolving threats.

Deep learning, a branch of machine learning based on multilayer neural networks, is a promising approach as it can automatically extract both low- and high-level patterns from raw traffic, making it well-suited for handling the complexity and variability of real-world DDoS attacks. For example, Prabha and Srinivasan [2] demonstrated that enhanced Deep Convolutional Neural Networks (DCNNs) achieved rapid and accurate detection of DDoS attacks, outperforming traditional baselines. Similarly, Ramzan *et al.* [3] showed that Recurrent Neural Networks (RNNs) and CNNs could capture both sequential and spatial patterns, significantly improving discrimination between normal and malicious flows.

The adaptability of these models offers clear advantages: CNNs excel at capturing localized traffic anomalies, RNNs model sequential dependencies over time, and hybrid architectures have been shown to integrate these strengths. Moreover, attention-based mechanisms have been explored to further improve sensitivity to critical traffic features. Together, these developments demonstrate that deep learning is not just a tool for higher accuracy, but a foundation for adaptive, scalable, and robust DDoS detection systems.

2.2 Deep Learning Models for DDoS Attack Detection and Classification

Several studies highlight the benefits of combining different deep learning paradigms. CNNs effectively capture localized anomalies and have shown high accuracy [3,4], with distributed frameworks (e.g., SSK-DDoS [5]) enabling near real-time detection. However, they are less suited for modeling long-term dependencies. RNNs and LSTMs capture sequential dependencies in traffic flows [6], though they often face training instability and high computational cost on large-scale data.

Combining CNNs and RNNs [7], ensemble approaches [8], or integrating deep models with MLPs [9] enhances robustness across diverse conditions, yet most focus only on temporal/spatial patterns, neglecting feature-wise dependencies. Attention improves sensitivity to critical features [10], but has mainly been applied at temporal or convolutional levels, with limited exploration of feature-wise importance. Adaptive CNNs [11,12] and distributed stream-processing integrations [5] improve efficiency for constrained environments, though further work on compression and edge deployment is needed.

Existing deep learning–based DDoS detection approaches have shown significant progress. Attention mechanisms have further improved feature prioritization, while distributed frameworks enable real-time scalability. However, most prior studies suffer from key limitations. First, many rely on sequential packet data or flow-level time series, limiting their applicability to feature-based tabular data commonly used in real-world detection systems. Second, attention has often been applied at the temporal or convolutional level, with little focus on feature-wise importance, an aspect crucial for distinguishing correlated traffic characteristics. Finally, comparative studies are frequently limited to a few models evaluated under inconsistent conditions, hindering fair performance assessment and practical benchmarking.

Taken together, prior studies highlight that while CNNs and LSTMs are effective in modeling localized or sequential patterns, they remain constrained in handling tabular traffic features and feature-wise dependencies. To overcome these shortcomings, the following section presents our methodological framework, including two novel architectures that explicitly model long-range interactions and adaptively weight critical features.

2.3 Research gap

While the primary gaps summarize the limitations above, a closer examination reveals additional challenges. Many prior works report only average accuracy, without assessing variability or statistical significance across multiple runs, leaving questions about model stability and generalizability. Analyses of training dynamics, overfitting behavior, and confusion matrices are rarely provided, limiting interpretability. Moreover, complex models such as residual, attention-based, or hybrid architectures often lack empirical evidence of consistent superiority over simpler designs. Finally, high computational costs and deployment challenges in resource-constrained environments highlight the need for new solutions that balance accuracy, robustness, and efficiency un-

der standardized evaluation conditions. These observations motivate our proposed TDNN and ATDNN frameworks, which explicitly model long-range interactions and adaptively weight critical features.

3. METHODOLOGY

In this study, we propose a systematic methodology to evaluate nine deep learning architectures for DDoS detection including MLP, DNN, CNN variants, ResNet1D, ATRN, LSTMClassifier, TDNN, and ATDNN under identical experimental conditions. The pipeline comprises four key stages: data preparation, feature selection, normalization, and model development. Standardized training settings and hyperparameters were applied to ensure fair comparison, and model performance was assessed using accuracy, precision, recall, F1-score, learning curves, and confusion matrices. Statistical significance testing (t-tests) was conducted to validate performance differences across multiple runs, while the best-performing model (TDNN) was further tested on an independent dataset to examine its generalization. **Fig. 1** summarizes the overall methodology.

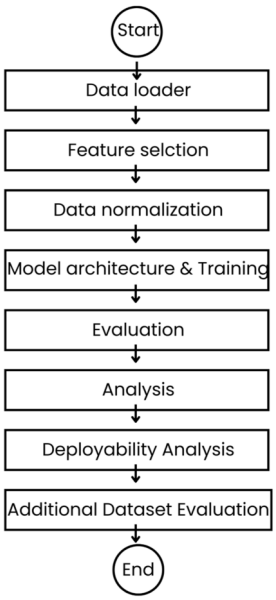


Fig.1: *Flowchart of methodology pipeline.*

3.1 Overview of proposed models and contributions

Unlike previous approaches that mainly applied CNNs or LSTMs to traffic features, our work introduces two key architectural innovations to overcome long-standing limitations in DDoS detection. **TDNN** integrates a lightweight Transformer encoder into a fully connected pipeline, enabling long-range feature dependency modeling in tabular data without relying on sequential inputs. **ATDNN** further enhances this by applying feature-wise attention before classification, dynamically reweighting feature importance, a capability often overlooked in prior models that focus on temporal or convolutional attention. Together, these designs transform the detection pipeline into a context-aware, feature-adaptive architecture that achieves high accuracy with practical efficiency.

In addition, this study fills a major research gap by conducting a comprehensive, standardized comparison of nine representative deep learning models under identical conditions. Through consistent preprocessing, unified hyperparameters, statistical significance testing, deployability analysis, and cross-dataset evaluation, our framework provides both novel architectures and one of the most rigorous performance evaluations in modern DDoS detection literature.

3.2 Data preprocessing

3.2.1 Data loader

The DDoS dataset was downloaded from [13] in Kaggle ("devendra416/ddos-datasets") and loaded into a Pandas DataFrame. The label column 'Label' with classes 'ddos' and 'benign' was converted to binary values (1 for ddos, 0 for benign). To ensure balanced training, 25,000 samples per class were randomly selected, while the remaining data was split into validation (5,000 per class) and test sets (8,000 per class). All subsets were shuffled and saved as CSV files under organized directories (data/train, data/val, data/test), providing a balanced and reproducible dataset for model development.

3.2.2 Feature selection

During feature selection, the training, validation, and test datasets were loaded and cleaned to remove irrelevant columns such as 'Unnamed: 0', 'Flow ID', 'Src IP', 'Dst IP', and 'Timestamp', keeping only relevant features. The 'Label' column was encoded numerically for consistency. The datasets were then merged for a comprehensive correlation analysis using the Pearson correlation matrix, visualized as a heatmap (**Fig.2**). This process identified highly correlated features that could be removed to reduce multicollinearity and improve overall model performance.

3.2.3 Data normalization

Based on the Pearson correlation heatmap analysis, we applied the following feature selection rules: (1) Features with very high correlation ($|r| \geq 0.9$) to others were considered redundant and removed to reduce multicollinearity, (2) Features with moderate to low correlation and strong relevance to traffic dynamics were retained, and (3) Statistical interpretability and domain knowledge such as packet rates, inter-arrival times, and byte counts were prioritized. Following these rules, we selected ten features: 'Fwd Pkts/s', 'Bwd Pkts/s', 'Pkt Len Mean', 'Flow IAT

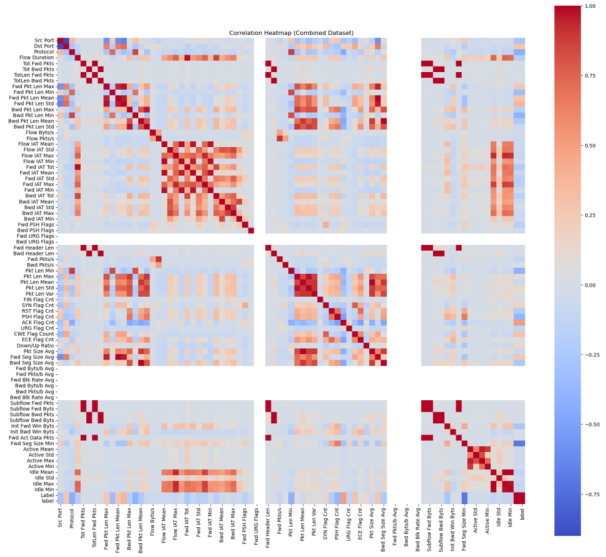


Fig.2: *Heatmap of feature correlation.*

Mean', 'Idle Mean', 'Tot Fwd Pkts', 'Flow Pkts/s', 'Subflow Fwd Byts', 'Active Mean', and 'Pkt Len Std'.

These features were extracted and preprocessed: infinite values were replaced with NaNs, and missing values were imputed using the maximum value of each column. Min–Max normalization was then applied to scale all features to the [0,1] range, with the scaler fitted only on the training set and applied to validation and test sets to prevent data leakage. The normalized feature matrices and corresponding labels were converted into PyTorch tensors, wrapped into TensorDataset objects, and loaded into DataLoaders with a batch size of 64. The training loader used shuffling for randomization, while validation and test loaders preserved the original order, ensuring consistent and stable model training and evaluation.

3.3 Model

In this study, we proposed and selected deep learning models as summarized in **Table 1**, with their schematic architectures illustrated in **Fig. 3**. To ensure a fair comparison, all models were trained under identical experimental settings, including a batch size of 64, 20 training epochs, and five independent runs with different random seeds to assess statistical reliability and performance variability. All models employed consistent training configurations, using the ReLU activation function, the Adam optimizer with a fixed learning rate of 0.001, and the CrossEntropy loss function. By intentionally fixing these hyperparameters, we eliminate any tuning advantage of a particular architecture and focus the evaluation strictly on methodological differences. The proposed TDNN and ATDNN architectures incorporate key modules designed to address limitations in previous deep learning approaches.

The **FeatureAttention** module has a key parameter feature_dim=10 and learnable attention weights initialized with torch.randn(10). It operates by multiplying input features by softmax-

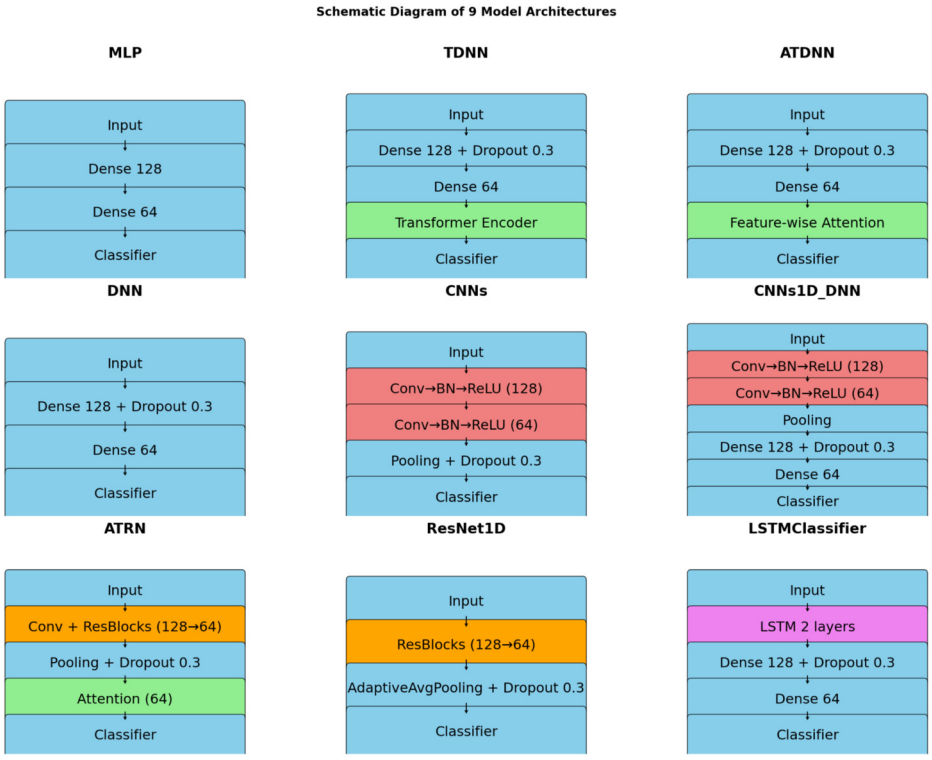


Fig.3: *Schematic diagram of 9 model architectures.*

normalized attention weights to emphasize the most important features, producing an output of shape [batch_size, 10]. Meanwhile, the **FeatureTransformer module** is configured with feature_dim=10, dim_feed forward=128, n_heads=5, num_layers=1, dropout=0.1, dropout=0.1, dim_feedforward=128, and batch_first= True. It reshapes the input to (B, 1, 10), passes it through a TransformerEncoder comprising multi-head attention and feedforward layers, and then squeezes the output back to shape [batch_size, 10].

Table 1: *Model Architecture and Training Configuration.*

Model	Architecture	Layer	Layer Size
MLP	MLPClassifier (sklearn)	2 Hidden Layers	128, 64
TDNN	FC + Transformer encoder layer	2 FC → Transformer Encoder → Classifier	128, 64
ATDNN	DNN + Feature-wise attention	2 FC → Attention → Classification	128, 64
DNN	2 FC layers	2 Hidden Layers	128, 64
CNNs	1D CNNs	(Conv → BN → ReLU) × 2 → Adaptive AvgPooling	128, 64
CNNs1D_DNN	1D CNNs + DNN classifier	2 Conv + BN + Pool → 2 FC → Classifier	128, 64
ATRN	1D ResNet + Attention	Conv − > Residual blocks − > Attention	128, 64
ResNet1D	1D ResNet (no attention)	Residual blocks	128, 64
LSTMClassifier	LSTM + Fully Connected Layers	LSTM (2 layers) → FC1 → FC2 (Classifier)	128, 64

3.4 Evaluate metrics

Model performance was evaluated on the test set using accuracy, precision, recall, and F1-score with macro averaging to account for class imbalance. Zero-division errors were handled with zero_division = 0. Confusion matrices were generated to visualize true/false positives and negatives, providing detailed insights into classification behavior. Evaluation results from multiple runs were collected for statistical analysis, including mean and standard deviation, to ensure robust and consistent performance.

4. RESULTS

Before presenting the experimental findings, we outline the objectives and key hypotheses of this study. The main goal was to evaluate and compare nine deep learning architectures MLP, DNN, CNNs, CNNs1D_DNN, ResNet1D, ATRN, LSTMClassifier, TDNN, and ATDNN for DDoS detection under standardized experimental conditions. The working hypothesis was that transformer-based models (TDNN) and models with feature-wise attention (ATDNN) would outperform conventional architectures due to their ability to capture long-range dependencies and emphasize critical traffic features.

4.1 Evaluation

Table 2: *Summarizes the evaluation results.*

Model	Accuracy	Precision	Recall	F1-score
MLP	0.9624± 0.0025	0.9629± 0.0024	0.9624± 0.0025	0.9624± 0.0025
TDNN	0.9653± 0.0018	0.9659± 0.0016	0.9653± 0.0018	0.9653± 0.0018
ATDNN	0.9361± 0.0043	0.9363± 0.0041	0.9361± 0.0043	0.9361± 0.0044
DNN	0.9632± 0.0009	0.9638± 0.0007	0.9632± 0.0009	0.9632± 0.0009
CNNs	0.8063± 0.1184	0.8533± 0.0781	0.8063± 0.1184	0.7929± 0.1323
ATRN	0.7010± 0.2200	0.8384± 0.1080	0.7010± 0.2200	0.6158± 0.2895
ResNet 1D	0.8447± 0.1579	0.8899± 0.0965	0.8447± 0.1579	0.8259± 0.1867
CNNs1D_ DNN	0.9035± 0.0302	0.9069± 0.0293	0.9035± 0.0302	0.9032± 0.0304
LSTM Classifier	0.9582± 0.0028	0.9587± 0.0028	0.9582± 0.0028	0.9582± 0.0028

According to **Table 2**, the experimental results support the hypothesis that transformer-based architectures, particularly TDNN, outperform conventional models in DDoS detection. TDNN achieved the highest accuracy (0.9653 ± 0.0018), precision (0.9659 ± 0.0016), recall (0.9653 ± 0.0018), and F1-score (0.9653 ± 0.0018), demonstrating both effectiveness and consistency. LSTMClassifier also performed strongly (accuracy 0.9582 ± 0.0028, precision 0.9587 ± 0.0028, recall 0.9582 ± 0.0028, and F1-score 0.9582 ± 0.0028), placing it close to TDNN and outperforming simpler feedforward models. DNN and MLP showed competitive results, with DNN exhibiting slightly better stability (lowest standard deviation in accuracy: ±0.0009) compared to MLP. While ATDNN maintained reasonable accuracy (0.9361 ± 0.0043), it fell behind TDNN, suggesting that the additional attention mechanism may not provide clear benefits for this dataset. CNNs1D_DNN demonstrated a good balance between performance (0.9035 ± 0.0302) and generalization. Conversely, CNNs,

ResNet1D, and ATRN had lower overall performance and higher variability, with ATRN showing the poorest and most unstable generalization (F1-score 0.6158 ± 0.2895).

4.2 Loss and accuracy curve

4.2.1 The best performing model

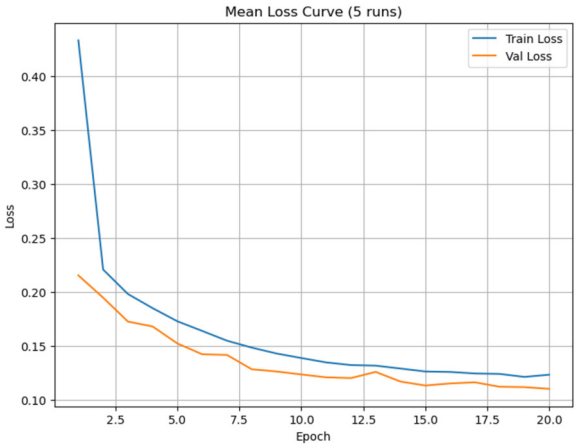


Fig.4: *Average training and validation loss of TDNN.*

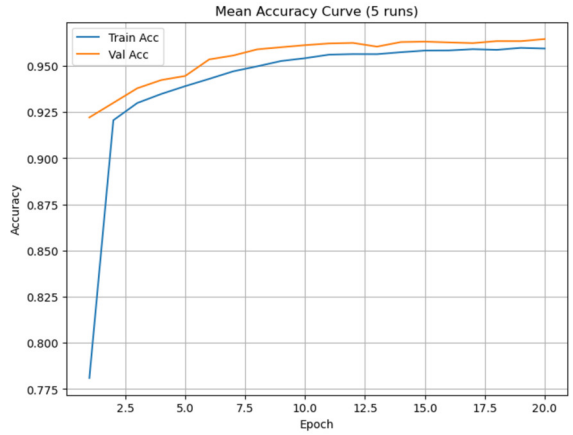


Fig.5: *Average training and validation accuracy of TDNN.*

According to **Fig.4-5** During training, the TDNN model exhibited near-ideal learning behavior, supporting the hypothesis that transformer-based architectures can effectively capture long-range dependencies. The training loss decreased smoothly and steadily across epochs, while the validation accuracy increased consistently with minimal divergence between training and validation metrics. This indicates that TDNN not only learns efficiently but also generalizes well to unseen samples, demonstrating robustness and stable performance across multiple runs. The steady rise in accuracy with little fluctuation further confirms its strong generalization capability.

4.2.2 High-performing models

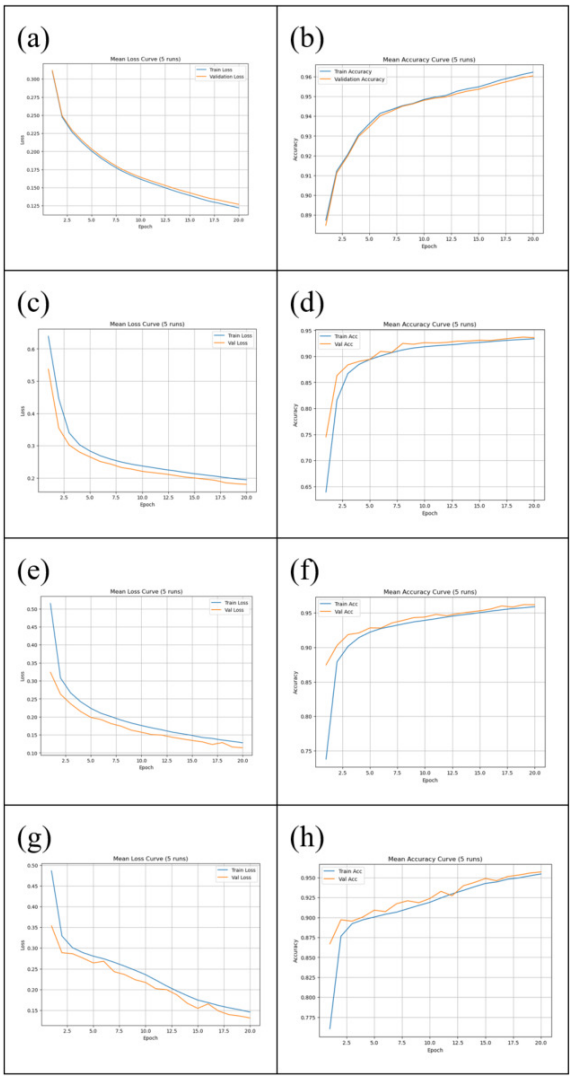


Fig.6: *Average training and validation loss and accuracy curves over multiple runs: (a) Loss curve of MLP, (b) Accuracy curve of MLP, (c) Loss curve of ATDNN, (d) Accuracy curve of ATDNN, (e) Loss curve of DNN, (f) Accuracy curve of DNN (g) Loss curve of LSTMClassifier, and (h) Accuracy curve of LSTMClassifier.*

In all plots, blue lines represent training metrics and orange lines represent validation metrics. The learning behavior of key models MLP, DNN, LSTMClassifier, and ATDNN was analyzed over five runs (**Fig. 6**). MLP (**Fig.6a-b**) and DNN (**Fig.6e-f**) showed smooth, stable learning with steady loss reduction and consistent validation accuracy, with training and validation curves remaining close, indicating good generalization. The LSTMClassifier (**Fig.6g-h**) was similarly stable with slight fluctuations, and the gap between the two curves was relatively small. ATDNN (**Fig.6c-d**), using feature-wise attention, also maintained relatively stable curves, though with minor validation loss fluctuations and a

slightly wider gap suggesting mild overfitting. Overall, ATDNN effectively captured critical traffic features, supporting the hypothesis that attention-based models like ATDNN better learn important feature relationships and outperform conventional architectures.

4.2.3 Underperforming models

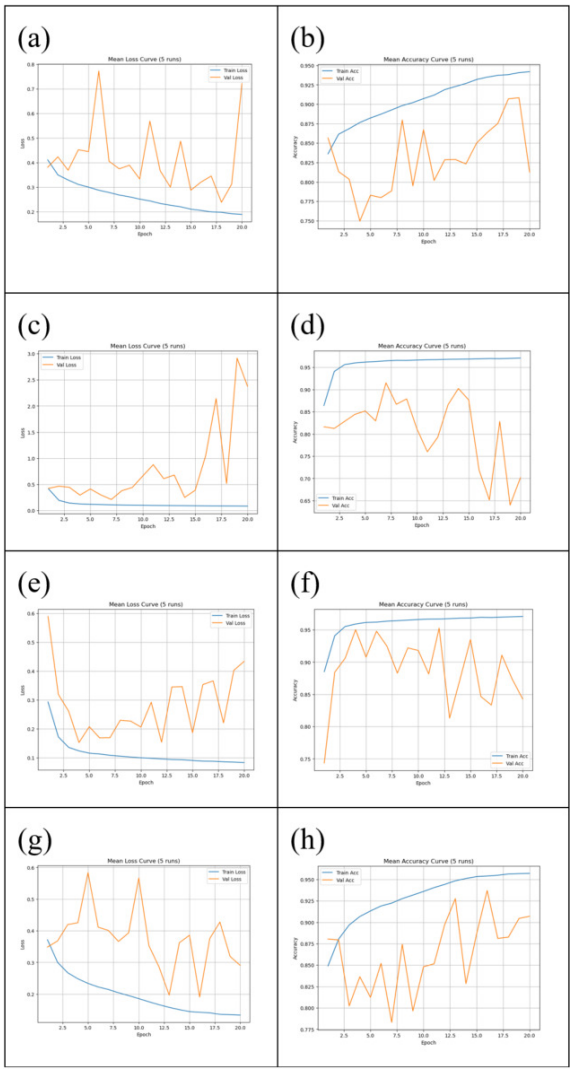


Fig.7: *Average training and validation loss and accuracy curves over multiple runs: (a) Loss curve of CNNs, (b) Accuracy curve of CNNs, (c) Loss curve of ATRN, (d) Accuracy curve of ATRN, (e) Loss curve of ResNet1D, (f) Accuracy curve of ResNet1D, (g) Loss curve of CNNs1D_DNN, and (h) Accuracy curve of CNNs1D_DNN.*

In all plots, blue lines represent training metrics and orange lines represent validation metrics. More complex architectures CNNs (**Fig.7a-b**), ResNet1D (**Fig.7e-f**), ATRN (**Fig.7c-d**), and CNNs1D_DNN (**Fig.7g-h**) showed less stable learning, with validation loss often plateauing or increasing after early epochs, resulting in larger gaps between training and validation curves, which indicates overfitting. ATRN exhibited the largest divergence between training and validation accuracy, suggesting poor generalization, while CNNs1D_DNN showed intermediate gap behavior. In contrast, the transformer-based TDNN and the feature-wise attention-based ATDNN demonstrated stable and smooth learning, with closely aligned training and validation curves, highlighting minimal divergence. This supports the hypothesis that these models effectively capture long-range dependencies and emphasize critical traffic features, leading to superior generalization.

4.3 Confusion matrices

4.3.1 The best performing model

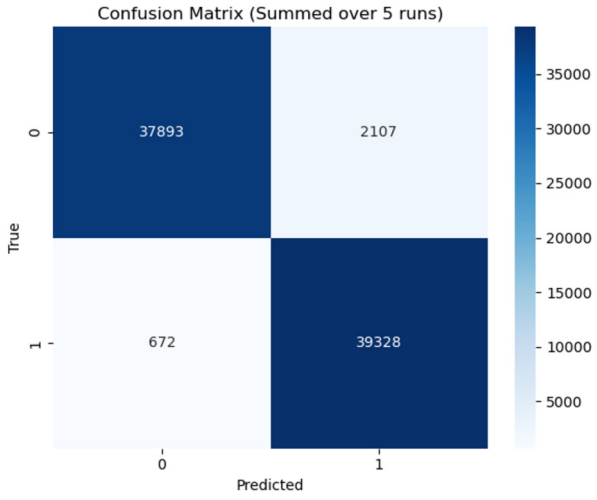


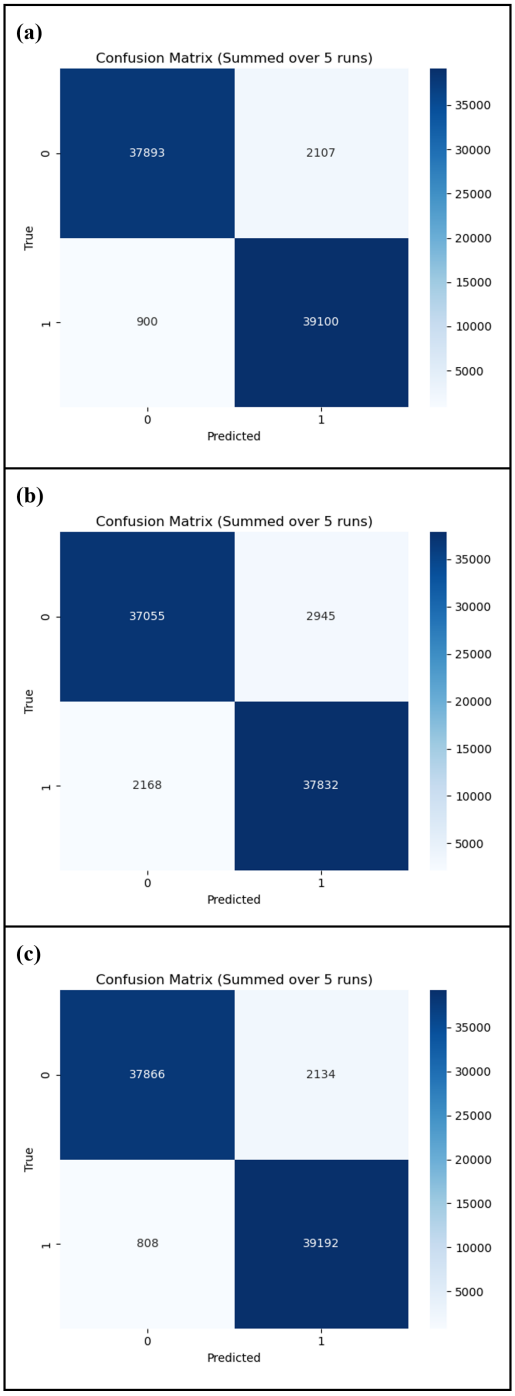
Fig.8: *Confusion matrices of TDNNl evaluated on the test set for five runs (each run consisting of 16,000 samples, 8,000 per class).*

According to **Fig. 8**, TDNN achieved the best classification performance, with confusion matrices showing high true positives and negatives and minimal false predictions. This supports the hypothesis that transformer-based architectures like TDNN capture long-range dependencies and key traffic features, yielding accurate discrimination between benign and DDoS traffic. Quantitative metrics (accuracy, precision, recall, F1-score) further confirm its robustness, generalization, and practical deployability.

4.3.2 High-performing models

According to **Fig.9**, the MLP (**Fig.9a**), DNN (**Fig.9c**), and LSTMClassifier (**Fig.9d**) models produced relatively clean confusion matrices across the five independent test runs, with only slightly more misclassifications compared to TDNN. While these conventional architectures demonstrated robust performance and low variance, their results partially support the hypothesis: although they generalize well under standardized conditions, they lack the

transformer-based capability to capture long-range dependencies and emphasize critical traffic features. In contrast, the ATDNN (**Fig.9b**), despite incorporating feature-wise attention, showed slightly degraded performance relative to TDNN. Its confusion matrices indicate a marginal increase in false negatives, suggesting that the added model complexity did not fully translate into improved generalization. These observations reinforce that TDNN's transformer-based design more effectively leverages long-range dependencies and critical feature information, as hypothesized.



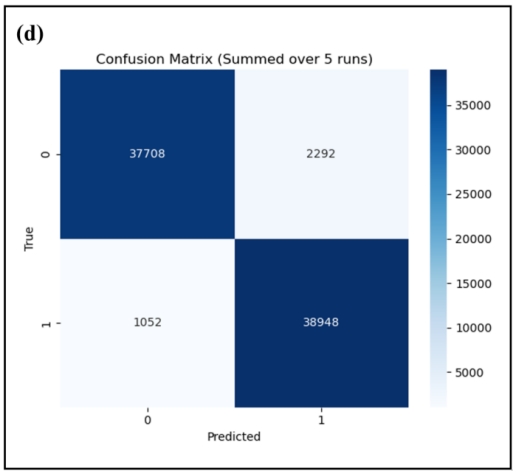
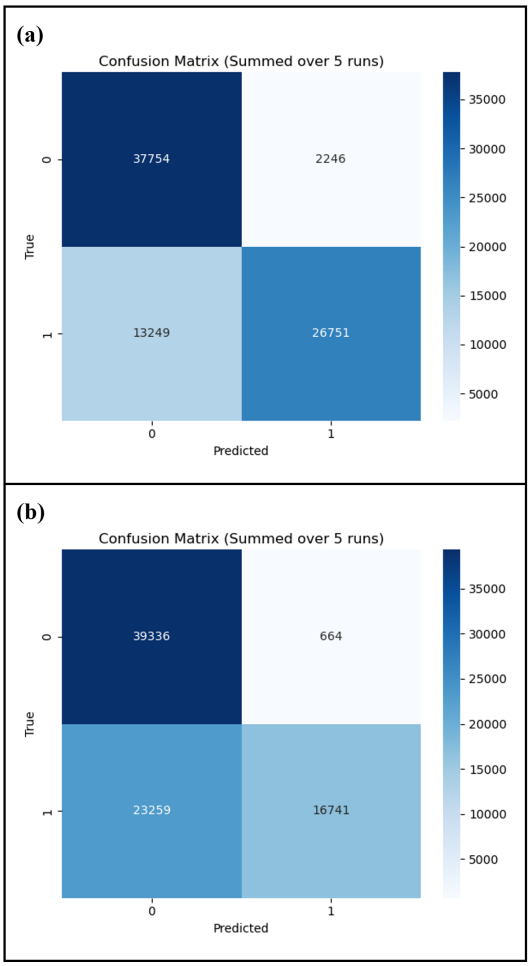


Fig.9: *Confusion matrices of each model evaluated on the test set for five runs (each run consisting of 16,000 samples, 8,000 per class): (a) MLP, (b) ATDNN, (c) DNN, and (d) LSTMClassifier.*

4.3.3 Underperforming models



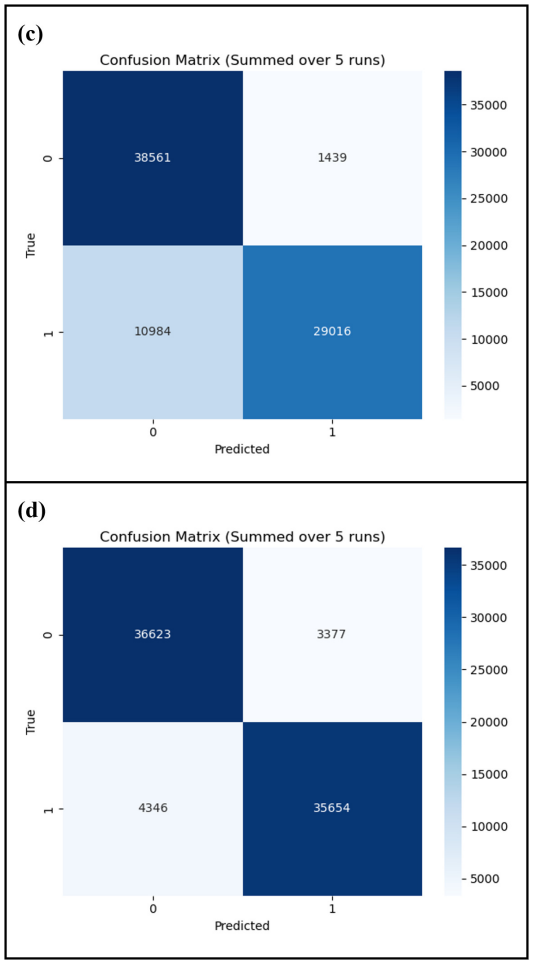


Fig.10: *Confusion matrices of each model evaluated on the test set for five runs (each run consisting of 16,000 samples, 8,000 per class):(a) CNNs, (b) ATRN, (c) ResNet1D, and (d) CNNs1D_DNN.*

According to **Fig. 10**, more complex convolutional and residual architectures, including CNNs1D_DNN (**Fig.10d**) and CNNs (**Fig.10a**), showed moderate performance, with confusion matrices reflecting occasional misclassifications of both benign and DDoS traffic. While these models provide a reasonable trade-off between computational cost and accuracy, their inconsistent generalization highlights a limitation of purely convolutional or residual designs. ResNet1D (**Fig.10c**) and ATRN (**Fig.10b**) exhibited the weakest results, with ATRN showing a disproportionately high number of false negatives. These findings support the hypothesis that transformer-based models, such as TDNN, and feature-wise attention models, such as ATDNN, can more effectively capture long-range dependencies and emphasize critical traffic features.

4.4 Statistical Analysis

1. H_0 (Null hypothesis): There is no difference in performance between TDNN and the comparison model (TDNN $\leq$ other model).
2. H_1 (Alternative hypothesis): TDNN outperforms the comparison model (TDNN $>$ other model).

Table 3: *Statistical Comparison of TDNN Performance Against Other Models Using t-Tests.*

Comparison	**t-statistic**	**p-value**	**Significance (α=0.05)**
TDNN vs MLP	1.1179	0.3262	Not Significant
TDNN vs DNN	3.7764	0.0195	Significant
TDNN vs LSTMClassifier	3.7206	0.0205	Significant
TDNN vs ATDNN	12.4064	0.0002	Significant

According to **Table 3**, we selected the top five performing models and conducted independent t-tests across multiple runs to determine whether the performance differences between TDNN and these models were statistically significant. The results show that TDNN significantly outperformed DNN ($p = 0.0195$) and LSTMClassifier ($p = 0.0205$), leading us to reject H_0 for these comparisons. In contrast, the difference between TDNN and MLP was not statistically significant ($p = 0.3262$), so we fail to reject H_0, indicating no strong evidence that TDNN is superior to MLP. Additionally, TDNN outperformed ATDNN with high significance ($p = 0.0003$), supporting H_1.

4.5 Deployability Analysis

The TDNN model was selected for deployability evaluation as it achieved the best performance. The deployability metrics are presented in **Table 4**.

Table 4: *Summarizes the evaluation results.*

Metric	**Value**
Inference time per sample	0.019 ms
CPU memory used (current)	1.35 KB
CPU memory peak	24.73 KB

Table 4 shows that TDNN is fast and lightweight, with 0.019 ms per sample and peak memory of 24.73 KB, confirming its suitability for real-time deployment in resource-limited settings.

4.6 Additional Dataset Evaluation

To assess the generalization capability of the proposed TDNN model, we evaluated it on the CICDDoS2019 dataset from [14], which includes nine classes: 'DrDoS_DNS', 'DrDoS_LDAP', 'DrDoS_MSSQL', 'DrDoS_NetBIOS', 'DrDoS_NTP', 'DrDoS_SNMP', 'DrDoS_SSDP', 'DrDoS_UDP', and 'BENIGN' (encoded as 1–8 for attacks and 0 for benign). The dataset was split into training (9,900 samples), validation (3,600 samples), and testing (4,500 samples). Feature selection was performed using the Pearson correlation matrix method described in Section 3.2.2, resulting in ten selected features: Fwd Packet Length Min, Average Packet Size, Inbound,

Packet Length Mean, Protocol, Down/Up Ratio, Destination Port, URG Flag Count, Flow Bytes/s, and Fwd Packet Length Max. As summarized in **Table 5**, the confusion matrices in **Fig.11** indicate minimal false positives and false negatives, while **Fig.12** and **Fig.13** present the corresponding loss and accuracy curves, respectively.

Table 5: *Summarizes the evaluation results of TDNN.*

Model	Accuracy	Precision	Recall	F1-score
TDNN	0.7044± 0.0067	0.6609± 0.0415	0.7044± 0.0067	0.6540± 0.0212

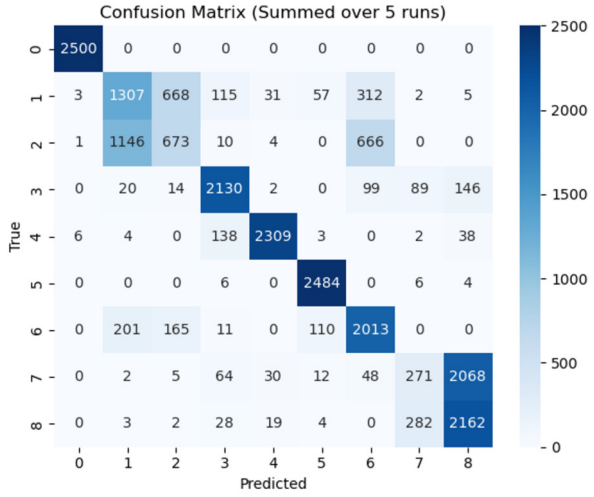


Fig.11: *Confusion matrices of TDNN evaluated on the test set for five runs (each run consisting of 4,500 samples, 500 per class).*

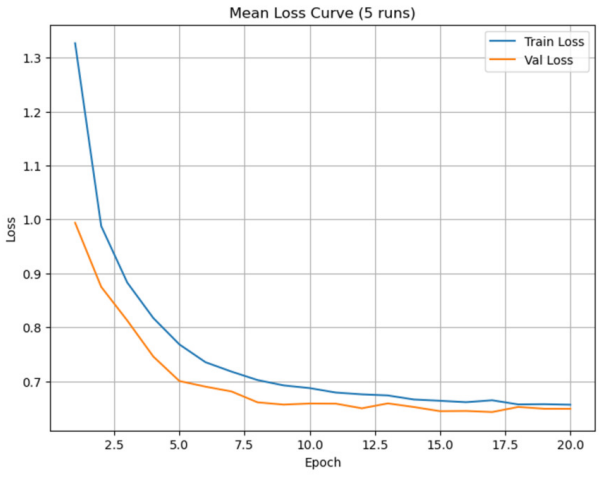


Fig.12: *Mean loss curve of TDNN in CICDDoS2019 dataset.*

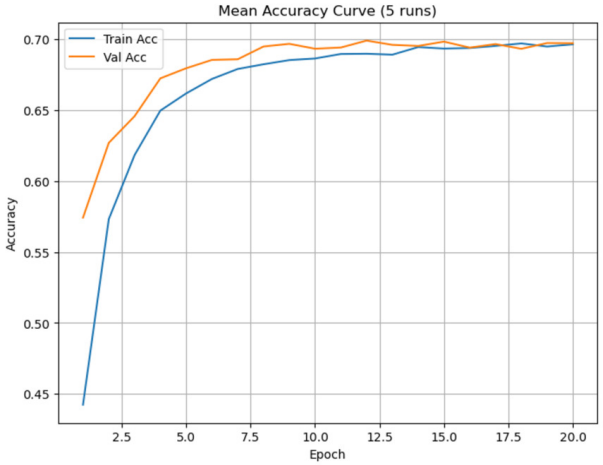


Fig.13: *Mean accuracy curve of TDNN in CICDDoS2019 dataset.*

According to **Table 5**, **Fig.11**, **Fig.12**, and **Fig.13** on the CICDDoS2019 dataset, the proposed TDNN model achieved an accuracy of 0.7044 ± 0.0067, precision of 0.6609 ± 0.0415, recall of 0.7044 ± 0.0067, and F1-score of 0.6540 ± 0.0212. Although these results are lower than those obtained on the original dataset, the model still demonstrates reasonable and consistent performance without any additional hyperparameter tuning. Notably, the loss and accuracy curves (**Fig.12** and **Fig.13**) progress in the same direction, with accuracy remaining relatively high while loss continues to decrease. This pattern suggests potential underfitting, likely caused by the increased complexity of the dataset, which now contains nine classes. Despite the added difficulty in discriminating between multiple attack types and benign traffic, TDNN maintains strong generalization capability, reflecting its robustness in handling more complex and multi-class scenarios.

5. DISCUSSION

5.1 Model analysis

The experimental results demonstrate that the proposed TDNN model outperforms all other evaluated architectures, consistent with the stated hypothesis. TDNN achieved the highest accuracy (96.53%) with the lowest variability across multiple runs, indicating strong generalization and robust performance. Notably, TDNN also maintained reasonable effectiveness when tested on a separate CICDDoS2019 dataset, despite the increased number of classes and without additional hyperparameter tuning. While ATDNN incorporates feature-wise attention, it slightly underperforms relative to TDNN (accuracy 93.61% ± 0.43%), and with only 9,858 parameters compared to 43,266 in TDNN, its ability to leverage attention and capture critical traffic patterns may be limited. The learning curves

(**Fig.4**–**Fig.6d-e**) show slightly higher fluctuations in ATDNN's validation loss, suggesting minor overfitting, while confusion matrices (**Fig.8-Fig.9b**) indicate a small increase in false negatives. These observations support the hypothesis that attention may inadvertently dilute important feature signals in this context, reducing generalization ability. Statistical analysis (**Table 3**) confirms the performance difference between TDNN and ATDNN is significant ($p = 0.0003$), reinforcing that feature-wise attention does not provide measurable benefits over the simpler TDNN architecture for this dataset.

Additionally, simpler feed-forward architectures such as MLP, DNN, and LSTMClassifier performed competitively and, in some cases, even outperformed more complex convolutional or residual networks. MLP, in particular, demonstrated stable learning curves with minor fluctuations, effectively generalizing to unseen data. This can be attributed to several factors: the DDoS detection task relies on tabular traffic features that do not require complex convolutional hierarchies; deeper architectures like ResNet1D and ATRN exhibited significant divergence between training and validation performance (**Fig.7**), indicating susceptibility to overfitting; and statistical analysis (**Table 3**) shows that MLP, DNN, and LSTMClassifier have low variance across runs, confirming stable and reliable performance. Finally, simpler models require less memory and computation, enabling faster convergence and reduced sensitivity to training noise, further supporting their strong generalization.

5.2 Limitations and Future Work

A limitation of this study is that all models were evaluated using fixed hyperparameters to ensure fair comparison. This may have constrained each model from reaching its optimal performance. In future work, we plan to incorporate automated hyperparameter tuning (e.g., using Optuna) to further enhance model accuracy and robustness. Additionally, we aim to explore model compression techniques to enable efficient deployment on resource-constrained edge devices and IoT infrastructures.

5.3 Practical Implications

The TDNN model achieved the highest detection accuracy (96.53%) on the primary dataset and maintained strong generalization (70.44%) on the independent CICDDoS2019 dataset. Its deployability metrics 0.019 ms inference latency per sample and 24.73 KB peak memory usage make it suitable for real-time, resource-constrained systems. TDNN is practical for integration into IDS or firewall pipelines, enabling rapid anomaly detection, automated alerting, and real-time mitigation of malicious traffic, offering a scalable and efficient solution for DDoS defense across enterprise, edge, and cloud environments.

6. CONCLUSIONS

This study presents a comprehensive evaluation of deep learning models for DDoS attack detection, addressing current research gaps in model comparability, performance consistency, and deployment viability. Among the nine models assessed, the proposed TDNN architecture delivered superior performance across all key metrics, while simpler models like DNN and MLP also demonstrated effectiveness with greater computational efficiency. Our findings underscore the importance of not only accuracy but also robustness and practicality in cybersecurity applications. These insights offer valuable guidance for researchers and practitioners seeking to implement deep learning-based DDoS defense mechanisms in real-world scenarios, particularly in resource-constrained and time-sensitive environments.

ABBREVIATION

Distributed denial of service	DDoS
Deep Neural Network	DNN
Multilayer Perceptron	MLP
Convolutional Neural Networks	CNNs
Transformer-based Deep Neural Network	TDNN
Attention-based Deep Neural Network	ATDNN
Long Short-Term Memory	LSTM
Deep Convolutional Neural Networks	DCNNs
Recurrent Neural Networks	RNNs
Attention-based ResNet1D	ATRN

AUTHOR CONTRIBUTIONS

Conceptualization, N.P., M.S. and P.B.; methodology, N.P.; software, N.P.; validation, N.P., M.S. and P.B.; formal analysis, N.P.; investigation, N.P.; data curation, M.S. and P.B.; writing—original draft preparation, N.P.; writing—review and editing, N.P., M.S. and P.B.; visualization, P.B.; supervision, N.P. All authors have read and agreed to the published version of the manuscript.

References

[1] A. Otiko, E. Edim, G. Iyang and E. Oyo-Ita, "A survey of AI methods for detection of DDoS attacks on networks," *Advances in Research*, vol. 25, no. 5, pp. 256–271, 2024.

[2] M. Prabha and B. Srinivasan, "Enhanced DDoS attack detection using improved deep convolutional neural networks," *International Journal of Progressive Research in Engineering Management and Science*, 2024.

[3] M. Ramzan *et al.*, "Distributed denial of service attack detection in network traffic using deep learning algorithm," *Sensors*, vol. 23, no. 20, p. 8642, 2023.

[4] H. Xu and H. Xian, "SCD: A detection system for DDoS attacks based on SAE-CNN networks,"

Frontiers in Computing and Intelligent Systems, vol. 5, no. 3, pp. 94–99, 2023.

[5] N. Patil, C. Krishna and K. Kumar, "SSK-DDoS: Distributed stream processing framework based classification system for DDoS attacks," *Cluster Computing*, vol. 25, no. 2, pp. 1355–1372, 2022.

[6] N. Katuk, M. Sinal, M. Alsamman and I. Ahmad, "An observational mechanism for detection of distributed denial-of-service attacks," *International Journal of Advances in Applied Sciences*, vol. 12, no. 2, pp. 121–132, 2023.

[7] M. Owaid and A. Hammoodi, "Evaluating machine learning and deep learning models for enhanced DDoS attack detection," *Mathematical Modelling and Engineering Problems*, vol. 11, no. 2, pp. 493–499, 2024.

[8] F. Alanazi, K. Jambi, F. Eassa, M. Khemakhem, A. Basuhail and K. Alsubhi, "Ensemble deep learning models for mitigating DDoS attack in software-defined network," *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 923–938, 2022.

[9] E. Effah, E. Osei, M. Dorgbefu and A. Tetteh, "Hybrid approach to classification of DDoS attacks on a computer network infrastructure," *Asian Journal of Research in Computer Science*, vol. 17, no. 4, pp. 19–43, 2024.

[10] S. Muthukumar and A. Ahamed, "A novel framework of DDoS attack detection in network using hybrid heuristic deep learning approaches with attention mechanism," *Journal of High Speed Networks*, vol. 30, no. 2, pp. 251–277, 2024.

[11] A. Akinwumi, A. Akingbesote, O. Ajayi and F. Aranuwa, "Detection of distributed denial of service (DDoS) attacks using convolutional neural networks," *Nigerian Journal of Technology*, vol. 41, no. 6, pp. 1017–1024, 2023.

[12] S. Rameshkumar, R. Ganesan and A. Merline, "Progressive transfer learning-based deep Q network for DDoS defence in WSN," *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2379–2394, 2023.

[13] M. D. Prasad, P. B. V. Prasanta and C. Amarnath, "Machine Learning DDoS Detection Using Stochastic Gradient Boosting," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 4, pp. 157–166, 2019.

[14] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2019.

Nitipon Pongphaw is currently pursuing a Bachelor's degree in the Department of Electrical and Computer Engineering at Kasetsart University, Chalermphrakiat Sakon Nakhon Province Campus, Sakon Nakhon, Thailand. His research interests include machine learning and deep learning, particularly their applications in energy systems and computer networks.

Mune Sukumaradat is a first-year undergraduate student in the Department of Computer Engineering at Kasetsart University, Chalermphrakiat Sakon Nakhon Province Campus. He has an academic focus on cybersecurity, with a particular emphasis on ethical hacking. Mune has developed practical expertise in the field through his active participation in Capture The Flag (CTF) competitions, which have equipped him with valuable skills in reverse engineering and forensic analysis.

Prommin Buaphan is an undergraduate student pursuing a Bachelor of Engineering in Computer Engineering at Kasetsart University, Chalermphrakiat Sakon Nakhon Province Campus. He has a strong interest in Artificial Intelligence and Cybersecurity, with a particular focus on applying Deep Learning techniques for network threat detection. In this research, he contributed to the design of the hybrid model architecture, dataset analysis and preprocessing, as well as the comparative performance evaluation of the models.