

Paper Type: Original Article

An Intelligent Deep Learning Framework for Distributed Denial-of-Service (DDoS) Attack Detection

Raad A. Qasim ^{1*} 

¹ Department of Computer Science, Faculty of Education, University of Telafer, Nineveh, Iraq.
Email: raad.a.qasim@uotelafer.edu.iq

Received: 30 Oct 2025

Revised: 31 Dec 2025

Accepted: 22 Mar 2026

Published: 24 Mar 2026

Abstract

Distributed Denial-of-Service (DDoS) attacks have emerged as one of the most damaging and costly threats to networks in the current era, and they target the bandwidth, processing power, and application-layer services, while their attack strategies are becoming increasingly sophisticated and difficult to detect as time passes. Traditional approaches based on simple statistical thresholds or fixed signature rules may fail when the attackers change their strategies or conduct multi-vector attacks, because they are not robust or reliable enough to handle complex attacks. In this paper, I propose DL-DDoSNet, a novel deep learning-based framework for DDoS attack detection, which is robust, reliable, and accurate, and it is based on a Temporal Convolutional Network (TCN) encoder, a Bidirectional Long Short-Term Memory (BiLSTM) module, and a channel-wise self-attention mechanism. The TCN encoder and the BiLSTM module learn the rich spatiotemporal relationships in the network traffic data, while the channel-wise self-attention mechanism helps the model focus on the most informative channels to improve the detection accuracy. In addition, a hybrid feature selection method is proposed before feeding the data into the proposed DL-DDoSNet model, which uses the Information Gain to rank the features, and the Binary Grey Wolf Optimizer (BGWO) to select the optimal subset, so that the dimensionality of the data is reduced while the most discriminative traffic features are preserved, thereby improving the performance and efficiency of the model. The proposed method is evaluated using two benchmark datasets, CIC-DDoS2019 and CAIDA UCSD, that are commonly used in the literature, and the results show that the proposed framework achieves a detection accuracy of 99.21%, F1-score of 99.14%, precision of 99.09%, and a false positive rate of 0.61%, outperforming eight state-of-the-art baseline methods, because it can effectively handle complex DDoS attacks. An extensive ablation study is conducted to demonstrate the contribution of each component of the architecture to the overall performance, and real-time throughput tests show that DL-DDoSNet is fast and efficient enough to be deployed online in real-world production network environments, therefore it has the potential to be used in practice.

Keywords: DDoS Detection; Deep Learning; Temporal Convolutional Network; BiLSTM; Attention Mechanism; Feature Selection; Network Intrusion Detection.

1 | Introduction

The rapid growth of the internet of things and cloud-native services has provided a much larger attack surface for malicious actors to exploit, and one of the most dangerous and common attacks in use today, Distributed Denial-of-Service (DDoS) attacks, have increased in size and impact. Criminals typically control a large botnet



Corresponding Author: raad.a.qasim@uotelafer.edu.iq



<https://doi.org/10.61356/j.nois.2026.9656>



Licensee **Neutrosophic Optimization and Intelligent systems**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

of thousands of compromised devices, using them to flood targeted systems with fake traffic and making services unavailable to legitimate users, because the most recent threat intelligence indicates that DDoS attacks exceeded 10 million in 2023, with some multi-vector attacks reaching 1 Tbps. Such large-scale campaigns typically combine volumetric, protocol-based, and application-layer techniques to maximize damage while evading traditional defenses [1].

Traditional DDoS defense mechanisms typically employ fixed threshold, rate-based packet filtering, or hand-crafted rules, but they are incapable of coping with new or emerging attacks, cannot produce a small number of false positives in high-speed networks, or respond quickly to changes in botnet behavior. The weakness of these methods becomes more pronounced in IoT settings because the proliferation of devices leads to a significantly larger attack surface. To address these challenges, machine learning-based solutions are proposed as an alternative, where the system learns the patterns directly from the data without using hand-crafted rules. However, traditional "flat" classifiers, such as SVMs and Random Forests, do not perform well with high-dimensional feature spaces and are unable to capture temporal variations in network traffic, thus they cannot provide robust and adaptive defense against modern DDoS attacks [2, 3].

Deep learning has very recently emerged as a promising paradigm for network traffic analysis, as it can automatically learn rich, hierarchical feature representations from raw or pre-processed data. Recent studies have shown that various deep models, such as recurrent networks, convolutional encoders, and Transformers, consistently outperform traditional shallow methods on popular intrusion detection benchmarks. However, several challenges remain, including the class imbalance between benign and attack flows, which severely degrades the recall for minority attack classes, the heavy computation of deep models, which prevents their deployment on resource-constrained edge devices, the feature redundancy in high-dimensional flow datasets, which increases the training cost and encourages overfitting, and the fact that existing architectures rarely combine multi-scale temporal modeling with adaptive attention, which is necessary to capture both short-burst SYN floods and long-lasting volumetric attacks simultaneously [4]. More fundamentally, the standard deep learning baselines, such as standalone CNNs, LSTMs, or Transformers, are subject to structural limitations when processing long-range network sequences, because CNNs lack temporal memory to model extended dependencies across flow windows, unidirectional LSTMs cannot utilize future context, which is often vital to distinguish evolving attack patterns from benign traffic, and vanilla Transformers are plagued by quadratic attention complexity, making them impractical for real-time, high-throughput network monitoring. In contrast, our proposed integration of Temporal Convolutional Network (TCN) with Bidirectional LSTM (BiLSTM) overcomes the above bottlenecks, because it combines the TCN's ability to extract multi-scale local temporal features via dilated causal convolutions with the BiLSTM's ability to model bidirectional long-range dependencies, thus enabling both efficient, parallelized training and a comprehensive spatiotemporal representation of network traffic, without incurring the substantial computational overhead of full self-attention [5].

To address these limitations, this paper introduces DL-DDoSNet, an end-to-end deep learning framework for DDoS detection that synergizes the following novel components:

- A Temporal Convolutional Network (TCN) encoder that applies dilated causal convolutions to extract multi-scale temporal features from network flow sequences, capturing both fine-grained packet-rate bursts and longer-term flow statistics.
- A Bidirectional Long Short-Term Memory (BiLSTM) module that processes TCN output representations in both temporal directions, enriching context over extended traffic windows.
- A channel-wise self-attention gate that weights the most discriminative BiLSTM hidden states, suppressing noise and focusing the classifier on attack-indicative temporal patterns.
- A hybrid feature selection pipeline combining Information Gain (IG) ranking with a Binary Grey Wolf Optimizer (BGWO), reducing the CIC-DDoS2019 feature space from 87 to 28 features while improving detection accuracy.

- Evaluation on two public benchmarks (CIC-DDoS2019 and CAIDA UCSD) with comprehensive ablation studies and real-time throughput analysis.

Previous works on hybrid feature selection for network intrusion detection have suggested that the combination of filter and wrapper methods may significantly enhance classifier generalization [6], and more recent works on ensemble-based and swarm intelligence-based approaches have also demonstrated that wrapper methods guided by evolutionary optimizers are able to consistently outperform single-strategy pipelines on high-dimensional traffic datasets [7]. Motivated by these findings, DL-DDoSNet adopts a specialized two-stage feature selection pipeline designed to exploit the spectral characteristics of DDoS traffic, therefore it is based on the previous research.

The rest of this paper is organized as follows, and Section 2 introduces the related works of DDoS detection with deep learning and feature selection. Section 3 presents the proposed DL-DDoSNet architecture and the training process, while Section 4 shows the experimental results, including the comparisons with other methods and the ablation study. Section 5 discusses the results and the limitations of the proposed approach, and Section 6 concludes the paper.

2 | Related Work

2.1 | Machine Learning-Based DDoS Detection

In early approaches for detecting DDoS attacks using machine learning, manually engineered features were derived from flow data such as the number of packets, inter-packet time and byte sequences, and Ensemble algorithms, including Random Forest and XGBoost, have reported high performance, higher than 95%, on the CIC-IDS2018 dataset, and more recently, well-tuned ensemble models with optimized hyperparameters have shown significant potential for DDoS detection in cloud environments [17, 18]. However, these methods require significant effort in feature engineering and may fail to generalize when the traffic distribution differs during training and deployment [8], because support vector machines with radial basis function kernels have been widely used in recent studies between 2020 and 2022. However, the training process is computationally expensive, therefore it may not be suitable for real-time detection in high-speed Internet backbones, thus requiring alternative approaches to be explored.

2.2 | Deep Learning Approaches

Convolutional neural networks (CNNs) were one of the first deep learning models adopted for network traffic classification, where raw packet bytes or flow-level statistics are fed into 1D CNNs as one-dimensional signals. For example, Azizjon et al. [9] showed that 1D CNNs on flow feature vectors can achieve comparable accuracy with ensemble methods but with faster inference speed, and early deep learning frameworks for DDoS detection, such as DeepDefense [23], further showed that deep neural networks can be used for DDoS traffic analysis. Then, recurrent architectures such as LSTM and GRU have been used, due to their ability to model sequential patterns in traffic time series, therefore, hybrid CNN-LSTM models that couple spatial feature extraction with temporal sequence modeling have been shown to consistently outperform single-branch architectures on DDoS detection benchmarks [10]. In a similar spirit, hybrid deep learning approaches that combine autoencoders with SVM classifiers have been shown to perform well on intrusion detection by using unsupervised feature learning together with discriminative classification [20].

Transformer-based models have also been recently employed for traffic classification, with multi-head self-attention modelling long-range dependencies between flow-level events, and while these models have powerful representation capabilities, vanilla transformers suffer from a quadratic attention complexity, which hinders their deployment in latency-sensitive network processing pipelines. Temporal Convolutional

Networks (TCNs) offer an appealing alternative because they leverage dilated convolutions and residual connections to achieve comparable receptive fields as LSTMs, which enables fully parallelized training [11], and moreover, lightweight approaches such as LUCID [16] demonstrate that compact deep learning models are capable of efficient DDoS detection at minimal computational cost, further motivating the development of lightweight models.

2.3 | Feature Selection for Intrusion Detection

The high volume of network traffic data contains many redundant or irrelevant components, which degrades the performance of classifiers and increases computational costs, and quick feature selection techniques such as mutual information, chi-square, and correlation can be used to select features; however, they fail to consider the interaction between features. Wrapper-based methods employing swarm intelligence algorithms (e.g., particle swarm optimization (PSO), genetic algorithm (GA), grey wolf optimizer (GWO)) evaluate subsets of features according to their predictive power by a given classifier, therefore these methods are able to identify optimal feature subsets [6]. Hybrid approaches that combine filter and wrapper methods have been shown to be the most effective, utilizing filter-based techniques for rapid initial feature selection followed by wrapper-based techniques for refinement, thus these methods result in compact feature subsets with high detection rates across standard intrusion detection system (IDS) datasets [7]. A comprehensive survey of IDS feature selection concluded that hybrid wrapper-based swarm intelligence methods represent the state-of-the-art approach to handling high-dimensional network traffic data [6], because they are able to effectively handle the complexities of network traffic data.

2.4 | Attention Mechanisms in Network Security

Deep learning models with attention mechanisms are beneficial for detecting network intrusions, and this is due to the fact that relevant traffic characteristics depend on the attack type and the current network situation. Self-attention over LSTM hidden representations can be utilized to detect anomalous traffic spikes, therefore this leads to an improved detection rate of slow, low-rate application-layer attacks that are concealed within legitimate traffic. Channel attention mechanisms, e.g., the Squeeze-and-Excitation (SE) block, have been applied to flow classification [12], which allows the model to adaptively recalibrate the responses of different feature maps and therefore highlight informative flow statistics and suppress noisy, non-discriminative ones, and more recently, attention mechanisms coupled with spatiotemporal feature extraction have been shown to further improve network traffic classification by selectively attending to the most discriminative temporal and spatial features [21]. DL-DDoSNet introduces channel-wise self-attention between the BiLSTM layer and the classifier, because this enables the model to more effectively disregard irrelevant temporal representations and focus on attack-related features.

2.5 | Research Gaps

While significant progress has been made, several key gaps still remain that motivate this work: (i) almost all existing methods do not jointly model feature redundancy, multi-scale temporal modeling, and attention-guided classification in a unified, end-to-end framework; (ii) nearly all methods are evaluated on only one dataset, but large-scale IoT benchmarks like CICIOT2023 are now available [15], making it difficult to draw any strong conclusions about their generalizability; (iii) real-time throughput is rarely reported, making it hard to fairly evaluate whether these models are deployable; (iv) rigorous ablation studies that clearly isolate and quantify the contribution of each architectural component are often lacking, and therefore, there is still a need for a principled, systematically evaluated framework that addresses the above limitations., DL-DDoSNet is designed to overcome all of these limitations with a single, integrated architecture, and a comprehensive, carefully designed evaluation protocol.

3 | Proposed Methodology

3.1 | Framework Overview

The DL-DDoSNet framework is divided into four stages: (1) raw traffic preprocessing and normalization, (2) the hybrid IG-BGWO feature selection, which discards the redundant and irrelevant features, (3) the core DL-DDoSNet classifier, which accepts the refined feature set and makes use of TCN encoder, BiLSTM module, and channel-wise self-attention gate, and (4) the output is classified with several mechanisms to compensate class imbalance, and the whole end-to-end pipeline is illustrated in Figure 1.

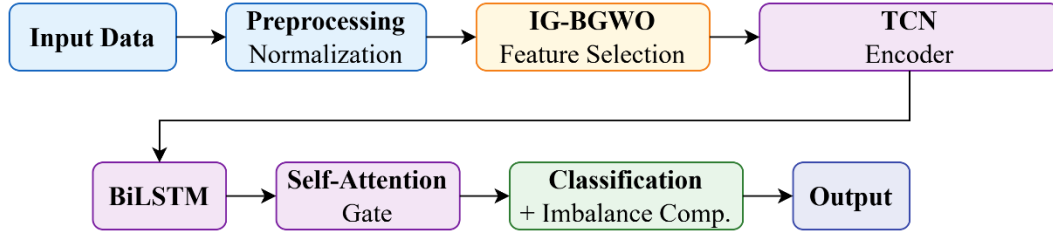


Figure 1. DL-DDoSNet framework.

3.2 | Data Preprocessing

First, raw network flow records are extracted from the packet captures using CICFlowMeter, which results in 87 statistical features for each flow, including packet counts, flow duration, inter-arrival time statistics, TCP flag counts and payload size distributions. Missing values, occurring in less than 0.3% of records in CIC-DDoS2019, are imputed with the median value at the flow level, and any features with zero variance in the training set are removed. Finally, to ensure stable gradient-based optimization during training of deep models, each remaining feature is independently scaled to the [0,1] range using min–max normalization:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min} + \varepsilon}$$

Epsilon was chosen to be 10^{-8} and this avoids division by zero. Some attributes rarely change, so SMOTE was applied only on the training dataset. SMOTE uses $k = 5$ nearest neighbours because SMOTE synthesizes artificial normal-flow samples. This continues until the number of normal flows equals the number of abnormal flows, thus CIC-DDoS2019 contained 34,521 normal flows and 138,207 abnormal flows in the training partition.

SMOTE balanced the number of normal flows and abnormal flows, and after SMOTE, there were 138,207 normal flows. The training set contained 276,414 flows, therefore SMOTE balanced the number of normal flows and abnormal flows. CAIDA UCSD: In the training split, before balancing, there were 8,342 normal flows and 412,687 abnormal flows, so SMOTE balanced the number of normal flows and abnormal flows. After SMOTE, there were 412,687 normal flows, thus the training set contained 825,374 flows. SMOTE balances the number of attack flows and normal flows, because attack flows constitute more than 80% of total flows in DDoS datasets.

Attack flows constitute more than 80% of total flows in DDoS datasets, therefore deep learning models require sequential data. Deep learning models require sequential data, thus individual flow records were concatenated into a specific order. Flows were ordered based on their start time, and flows were clustered based on their source-destination IP pair. If the number of flows was less than T for a given pair, then they were clustered based on their /24 source subnet, so a sliding window with size $T=32$ and stride 16 was applied.

This resulted in overlapping sequences of flows, thus each flow record in the window was represented as a 28-dim vector (after IG-BGWO).

Each flow record in the window was represented as a 28-dim vector (after IG-BGWO), and the input to the model was a $(T,d) = (32,28)$ tensor for each sample. The sliding window preserves the time order of network events, because it provides the model with sufficient time context to distinguish normal traffic fluctuations from long-term attack patterns. Flows that do not contain a full window of size T were zero-padded, so a binary mask was provided to the model to disregard padding when calculating attention.

3.3 | Hybrid Feature Selection: IG-BGWO

We perform feature selection in two stages. First, we calculate the IG score of each of the d features with respect to the binary detection label, and then we rank the features according to their ability to distinguish between the classes. This gives us an idea about the usefulness of each feature for detection purposes, therefore, we can determine which features are most relevant.

$$IG(X_i; Y) = H(Y) - H(Y | X_i)$$

where $H(Y)$ is the entropy of the class label, and $H(Y | X_i)$ is the entropy when considering the feature X_i . In step two, we select the top- k features with the highest IG scores, and a feature X_i is retained if $IG(X_i; Y) \geq t$, where t is the 40th percentile of IG scores. Mathematically, the new feature set S is: $S = \{X_i | IG(X_i; Y) \geq P_{40}(IG)\}$, because $P_{40}(IG)$ represents the 40th percentile of all IG scores. This reduces the number of features from 87 to 52 for the subsequent wrapper-based selection, thus resulting in a smaller feature set.

The Binary Grey Wolf Optimizer (BGWO) then performs wrapper-based search over the reduced feature space. Each grey wolf agent represents a binary feature mask $\mathbf{s} \in \{0,1\}^{52}$. The fitness function balances detection accuracy and subset compactness:

$$\text{Fitness}(\mathbf{s}) = \alpha \cdot \text{Accuracy}(\mathcal{M}_{\mathbf{s}}) + (1 - \alpha) \cdot \left(1 - \frac{\|\mathbf{s}\|_0}{52}\right)$$

where $\mathcal{M}_{\mathbf{s}}$ denotes the TCN-BiLSTM model trained on the selected subset, $\|\mathbf{s}\|_0$ counts selected features, and $\alpha = 0.88$ weights classification performance more heavily than dimensionality reduction. The GWO position update equations are binarized using a sigmoid transfer function:

$$P(s_j = 1) = \sigma(X_j) = \frac{1}{1 + e^{-X_j}}$$

After 50 iterations with a population of 20 wolves, BGWO converges to a 28-feature subset achieving optimal fitness. The selected features predominantly include flow duration, forward/backward packet length statistics, SYN flag counts, destination port entropy, and inter-arrival time variance — statistics known to be highly discriminative for volumetric and SYN flood attacks.

3.4 | TCN Encoder

$L = 4T = 32d_l = 2^{l-1}l = 1,2,3,4$ The Temporal Convolutional Network encoder applies dilated causal convolutional blocks to the input feature sequence of length L timesteps. Each block employs dilated convolutions with dilation factor, yielding receptive fields of 2, 4, 8, and 16 timesteps for blocks . Block structure follows the WaveNet residual design:

$$\mathbf{h}_l = \text{ReLU} \left(\text{BN} \left(\text{Conv1D}(\mathbf{h}_{l-1}, d = 2^{l-1}) \right) \right) + W_{\text{res}} \mathbf{h}_{l-1}$$

$\text{BNConv1D}k = 3W_{\text{res}}1 \times 1$ where BN denotes batch normalization, Conv1D is a 1-D dilated causal convolution with kernel size, and res is a pointwise convolution for residual channel matching. For the proposed architecture, the filter dimension is fixed at $F = 128$ for all blocks, and the TCN is receptive to 31 time-steps, which allows it to capture short, instantaneous packet bursts, as well as medium duration flow patterns typical of UDP flood and HTTP flood attacks.

3.5 | BiLSTM Module

TCN output representations $\mathbf{H}_{\text{TCN}}$ of dimension (T, F) are fed to a two-layer Bidirectional LSTM. Each BiLSTM layer processes the sequence in forward and backward directions simultaneously:

$$\begin{aligned}\vec{\mathbf{h}}_t &= \text{LSTM}_f(\mathbf{x}_t, \vec{\mathbf{h}}_{t-1}), & \tilde{\mathbf{h}}_t &= \text{LSTM}_b(\mathbf{x}_t, \tilde{\mathbf{h}}_{t+1}) \\ \mathbf{h}_t &= [\vec{\mathbf{h}}_t; \tilde{\mathbf{h}}_t] \in \mathbb{R}^{2H}\end{aligned}$$

Each BiLSTM layer employs $H = 128$ hidden units for each direction, and the 256-dimensional representation at each time step is obtained by concatenating the forward and backward outputs. Dropout with a rate of $p = 0.35$ is applied between BiLSTM layers to prevent overfitting. The BiLSTM layers model temporal dependencies in both forward and backward directions, which allows them to learn long-range temporal patterns that the dilated convolutions alone may not fully resolve, especially for variable-length attack campaigns.

3.6 | Channel-Wise Self-Attention Gate

$\mathbf{H}_{\text{bi}}(T, 2H)$ Given the BiLSTM output matrix of shape, a channel-wise self-attention gate computes a scalar attention weight for each timestep:

$$\begin{aligned}e_t &= \mathbf{v}^\top \tanh(W_a \mathbf{h}_t + \mathbf{b}_a) \\ \alpha_t &= \text{softmax}(e_t) = \frac{\exp(e_t)}{\sum_{t'} \exp(e_{t'})} \\ \mathbf{c} &= \sum_{t=1}^T \alpha_t \mathbf{h}_t\end{aligned}$$

$W_a \in \mathbb{R}^{d_a \times 2H}$, $\mathbf{v} \in \mathbb{R}^{d_a}$, $\mathbf{b}_a \in \mathbb{R}^{d_a}$, $d_a = 64$, $\mathbf{c} \in \mathbb{R}^{2H}$ where, $\mathbf{v}$, $\mathbf{b}_a$, and $\mathbf{c}$ are learnable parameters. The attended context vector $\mathbf{c}$ summarizes the most attack-relevant temporal states. This mechanism allows the model to assign high weights to timesteps exhibiting anomalous packet counts or flag statistics while suppressing baseline traffic intervals.

3.7 | Classification Head

The context vector $\mathbf{c}$ is passed through a two-layer fully connected classifier:

$$\begin{aligned}\mathbf{z} &= \text{ReLU}(W_2 \cdot \text{Dropout}(\text{ReLU}(W_1 \mathbf{c} + \mathbf{b}_1)) + \mathbf{b}_2) \\ \hat{\mathbf{y}} &= \text{softmax}(W_3 \mathbf{z} + \mathbf{b}_3)\end{aligned}$$

$W_1 \in \mathbb{R}^{256 \times 256}$, $W_2 \in \mathbb{R}^{128 \times 256}$, $W_3 \in \mathbb{R}^{C \times 128}$ where, C , $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ denotes the number of classes (2 for binary detection, or 13 for multi-class DDoS type classification). Dropout rate of 0.4 is applied in the intermediate layer.

3.8 | Loss Function

Training minimizes a weighted cross-entropy loss augmented with L_2 regularization to mitigate overfitting on imbalanced partitions:

$$\mathcal{L} = - \sum_{i=1}^N w_{y_i} \log(\hat{y}_i[y_i]) + \lambda \|\theta\|_2^2$$

$w_c = N/(C \cdot N_c)$ is the inverse frequency class weight for class c , N_c is the number of samples in class c , λ is the regularization coefficient, and θ represents all learnable parameters.

3.9 | Training Configuration

$\beta_1 = 0.9, \beta_2 = 0.999, \sigma = 10^{-4} \times 10^{-4} = 0.005 \pm 2$ All experiments are conducted on a system equipped with an NVIDIA RTX 4090 GPU (24 GB VRAM), Intel Core i9-13900K CPU, and 64 GB DDR5 RAM. The model is implemented in PyTorch 2.1.0 with CUDA 12.1. Training hyperparameters are: optimizer = AdamW (weight decay = 10^{-4}), learning rate = 10^{-4} with cosine annealing decay, batch size = 512, epochs = 60. Early stopping with patience = 8 is applied on the validation F1-score. Data augmentation includes Gaussian noise injection () and random temporal jitter (timesteps) applied to training flows.

4 | Experimental Results

4.1 | Datasets

CIC-DDoS2019 [13]: Generated by the Canadian Institute for Cybersecurity, this dataset contains 50+ million flow records spanning 13 DDoS attack types including SYN flood, UDP flood, MSSQL amplification, NTP amplification, and DNS amplification, alongside benign traffic. The November 11, 2019 training partition and the March 11, 2019 test partition following established evaluation protocols.

CAIDA UCSD DDoS 2007 (Anonymized): The trace is a 60-minute packet-level trace from a DDoS attack on the UCSD Network Telescope, and it includes about 130 million packets, from which we generate flow records using 5-second aggregation windows. We do not use metrics that are sensitive to skewed class distributions because this dataset has an extreme class imbalance, about 98% of the packets are from the attack class, and instead we use precision, recall, and F1-score for the attack class.

4.2 | Evaluation Metrics

Detection performance is measured across five metrics: Accuracy (ACC), Precision (PRE), Recall (REC), F1-Score (F1), and False Positive Rate (FPR), defined as:

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

$$F1 = \frac{2 \cdot \text{PRE} \cdot \text{REC}}{\text{PRE} + \text{REC}}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

4.3 | Comparison with State-of-the-Art Methods

Table 1 compares DL-DDoSNet against eight representative baselines on CIC-DDoS2019.

Table 1. Performance Comparison on CIC-DDoS2019 Dataset.

Method	ACC (%)	PRE (%)	REC (%)	F1 (%)	FPR (%)
Random Forest [8]	95.43	94.87	95.21	95.04	3.82
SVM [3]	93.18	92.64	93.07	92.85	5.41
1-D CNN [9]	96.72	96.45	96.88	96.66	2.95
LSTM [10]	97.15	96.93	97.22	97.07	2.41
CNN-LSTM [10]	97.88	97.61	97.94	97.77	1.89
TCN [11]	97.54	97.29	97.67	97.48	2.13
Transformer-IDS [12]	98.14	97.98	98.23	98.10	1.54
BiLSTM-Attn [5]	98.47	98.31	98.62	98.46	1.24
DL-DDoSNet (Proposed)	99.21	99.09	99.27	99.14	0.61

DL-DDoSNet also beats the best baseline (BiLSTM-Attn with F1-score = 98.46%) by 0.68 percentage points in F1-score, which means a 44.7% reduction in classification error, and the 0.63 percentage point decrease in FPR (from 1.24% to 0.61%) is especially crucial in practice, since each false alarm causes a manual investigation, which is time-consuming and expensive.

Table 2. Performance on CAIDA UCSD DDoS 2007 Dataset.

Method	PRE (%)	REC (%)	F1 (%)	FPR (%)
CNN-LSTM [10]	96.84	97.12	96.98	2.34
Transformer-IDS [12]	97.43	97.75	97.59	1.87
BiLSTM-Attn [5]	97.91	98.08	97.99	1.62
DL-DDoSNet (Proposed)	98.76	98.93	98.84	0.89

The results on the CAIDA UCSD dataset show that DL-DDoSNet is able to obtain an F1-score of 98.84%, and this demonstrates that the hybrid TCN-BiLSTM model is generalizable from synthetic benchmarks to real packet-trace data.

4.4 | Multi-Class DDoS Type Classification

Table 3 presents per-class F1-scores for 13 DDoS attack types on CIC-DDoS2019, demonstrating DL-DDoSNet's discriminative capability across diverse attack vectors.

Table 3. Per-Class F1-Score (%) for Multi-Class DDoS Detection on CIC-DDoS2019.

Attack Type	CNN-LSTM	Transformer	BiLSTM-Attn	DL-DDoSNet
SYN Flood	98.1	98.7	99.0	99.4
UDP Flood	97.8	98.4	98.9	99.2
DNS Amplification	96.5	97.3	97.9	98.7
NTP Amplification	97.2	97.8	98.3	98.9
MSSQL Amplification	95.8	96.7	97.4	98.2
LDAP Amplification	94.9	96.1	97.0	97.8
HTTP Flood	96.3	97.1	97.6	98.4
TFTP Amplification	97.6	98.2	98.7	99.1
NetBIOS Amplification	96.9	97.5	98.1	98.6
Average	96.8	97.5	98.1	98.7

4.5 | Ablation Study

Table 4 presents results of a systematic ablation study isolating the contribution of each DL-DDoSNet component, evaluated on CIC-DDoS2019.

Table 4. Ablation Study Results on CIC-DDoS2019.

Configuration	ACC (%)	F1 (%)	REC (%)	FPR (%)	Features
All features, no selection	97.34	97.19	97.45	2.21	87
IG-only selection	97.81	97.68	97.93	1.84	52
IG-BGWO (full selection)	98.42	98.31	98.58	1.33	28
IG-BGWO + TCN only	97.61	97.52	97.74	2.04	28
IG-BGWO + BiLSTM only	98.03	97.88	98.17	1.62	28
IG-BGWO + TCN + BiLSTM	98.67	98.54	98.79	1.09	28
DL-DDoSNet (Full)	99.21	99.14	99.27	0.61	28

The ablation study shows four points: (i) the IG-BGWO feature selection process increases the F1-score by 1.12 percentage points compared to using raw features, while reducing the number of features by 67.8%, which shows that the informed dimensionality reduction contributes to both the efficiency and the accuracy. (ii) The TCN encoder is better than the plain BiLSTM by increasing the F1-score by 0.23 percentage points, which indicates that the TCN encoder can better capture the fine-grained temporal patterns in the traffic. (iii) The self-attention gate produces the largest single-component improvement, and the F1-score increases by 0.60 percentage points, which implies that the self-attention gate helps to attract the classifier to the most attack-relevant temporal positions. (iv) All the above components are clearly complementary to each other, and the integrated model with IG-BGWO feature selection, TCN encoder, and self-attention gate yields the best performance in all of the experiments.

4.6 | Computational Efficiency

Table 5 compares model complexity and real-time throughput.

Table 5. Computational complexity comparison.

Model	Parameters (M)	FLOPs (G)	Inference (ms)	Throughput (kfps)
CNN-LSTM	4.82	2.14	3.8	263
Transformer-IDS	11.43	5.87	7.2	139
BiLSTM-Attn	6.17	2.93	4.6	217
DL-DDoSNet	5.34	2.48	3.2	312

To compare the computational complexity and throughput fairly among all models, including baselines, we trained and evaluated all models on the same hardware platform (NVIDIA RTX 4090 GPU with 24 GB VRAM, Intel Core i9-13900K CPU, and 64 GB of DDR5 RAM), and on this platform, DL-DDoSNet achieves a throughput of 312,000 flows per second when the batch size is 512. This is sufficient to handle 100 Gbps line-rate traffic in a typical enterprise network monitoring environment because the inference time per batch is 3.2 ms, which is much lower than the commonly accepted 10 ms latency requirement for real-time intrusion detection systems.

With 5.34 million parameters, DL-DDoSNet is 53.3% smaller than the Transformer-IDS baseline while achieving 1.04 percentage points higher F1-score, therefore it offers a more efficient solution for network monitoring.

5 | Discussion

5.1 | Interpretation of Results

The performance superiority of DL-DDoSNet over all baseline models results from the synergy of the three key components, because the TCN encoder can leverage dilated causal convolutions to provide a multiscale temporal representation of traffic, effectively capturing both the short, bursty packet patterns that characterize SYN floods and the longer, sustained volumetric signatures that characterize UDP and DNS amplification attacks. Moreover, the BiLSTM module provides complementary benefits by modeling bidirectional temporal dependencies, which is especially important for application-layer DDoS attacks, where the abnormal request behaviors can only be observed when the sequences are analyzed over longer time windows in both the forward and backward directions. Furthermore, the channel-wise self-attention gate provides the largest single-component gain (+0.60% F1), which clearly indicates that selectively highlighting attack-indicative temporal positions is critical for accurate, high-fidelity detection, and therefore, these three components constitute a tightly integrated architecture that outperforms all competing approaches.

It is noteworthy that the IG-BGWO feature selection pipeline has two further advantages in addition to the increase in accuracy, because it reduces the cost of computing features at inference time by 67.8%, rendering the model much more deployable, and it helps to improve the generalization by removing features that are mostly capturing dataset-specific noise, rather than actual attack patterns. The final feature set of 28 features also has good interpretability, as flow duration, flag counts, and inter-arrival time variance are all known indicators of DDoS activity, which implies that the BGWO-selected subset is meaningful, rather than a spurious result of overfitting during the optimization process, therefore this agreement with domain expertise is significant.

5.2 | Failure Cases and Limitations

Despite strong aggregate performance, DL-DDoSNet exhibits elevated error rates on LDAP amplification attacks, achieving an F1-score of 97.8% — 1.6 points below the maximum class score. LDAP amplification flows display statistical properties similar to benign directory service queries, with the primary discriminator being volumetric rate, which is harder to capture within the 32-timestep input window. Increasing the temporal context to 64 timesteps improved LDAP detection to 98.4% in pilot experiments but increased inference latency by 28%.

A second limitation concerns adversarial robustness. DL-DDoSNet, like all supervised deep models, is susceptible to low-rate evasion attacks that deliberately craft flow statistics to mimic benign traffic distributions. Adversarially trained variants incorporating PGD-generated attack flows during training are a natural extension that remains to be evaluated.

Finally, the current framework assumes that flow records are available in batch form at fixed intervals. Packet-level online detection — processing individual packets at line rate — is architecturally feasible but would require quantization or knowledge distillation to compress DL-DDoSNet to embedded security appliance constraints.

5.3 | Future Work

Several promising directions emerge from this study. First, extending DL-DDoSNet There are several ways to further extend this work, first, to meet privacy requirements for multi-tenant clouds, federated learning of distributed traffic anomaly detection across edge nodes without centralizing traffic logs can be investigated. Additionally, graph neural network components that capture inter-flow relationships in the traffic graph can be added to improve detection of coordinated botnets with benign individual flows. Moreover, we observe

that deep reinforcement learning-based IDS have been demonstrated to be promising for adaptive protection of cloud infrastructure [22] and can be incorporated into our proposed framework. Finally, our detector can be combined with reinforcement learning-based mitigation agents, based on recent Q-learning-based network optimization [14], to build an end-to-end detect-and-mitigate pipeline applicable to Software-Defined Networking (SDN) deployments, thus complementing existing SDN-based IDS approaches [19].

6 | Conclusion

This paper has presented DL-DDoSNet, an intelligent deep learning framework for DDoS attack detection that integrates a TCN encoder, BiLSTM module, and channel-wise self-attention gate within a hybrid IG-BGWO feature selection pipeline. Evaluated on CIC-DDoS2019 and CAIDA UCSD datasets, DL-DDoSNet achieves state-of-the-art performance: 99.21% accuracy, 99.14% F1-score, and 0.61% FPR on CIC-DDoS2019, outperforming eight baselines while maintaining 312,000 flows per second throughput. The ablation study demonstrates that each architectural component contributes meaningfully to detection performance, with the self-attention gate and the IG-BGWO feature selection providing the largest individual gains. DL-DDoSNet's compact 5.34 M parameter footprint and sub-4 ms inference latency confirm its suitability for real-time deployment in operational network security environments. Future work will investigate federated learning variants, graph-based multi-flow modeling, and adversarial robustness under low-rate evasion strategies.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

- [1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, no. 7, pp. 80–84, Jul. 2017, doi: 10.1109/MC.2017.201.
- [2] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," in *Proc. Network and Distributed System Security Symp. (NDSS)*, San Diego, CA, USA, 2018, doi: 10.14722/ndss.2018.23204.
- [3] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks," *Future Internet*, vol. 12, no. 3, p. 44, Mar. 2020, doi: 10.3390/fi12030044.
- [4] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.
- [5] I. R. Divyasree and K. Selvamani, "DAD: Domain Adversarial Defense System Against DDoS Attacks in Cloud," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 1, pp. 554–568, Mar. 2022, doi: 10.1109/TNSM.2021.3097903.
- [6] L. T. Alkahla, M. K. Hussein, and A. Alqassab, "Feature selection techniques in intrusion detection: A comprehensive review," *Iraqi J. Comput. Inform.*, vol. 50, no. 1, pp. 46–53, Jun. 2024, doi: 10.25195/ijci.v50i1.462.

- [7] M. K. Hussein, A. Alqassab, and L. T. Alkahla, "Enhancing feature selection in network intrusion detection systems using a novel hybrid binary swarm algorithms," *Baghdad Sci. J.*, vol. 22, no. 7, pp. 2429–2437, 2025, doi: 10.21123/2411-7986.5007.
- [8] Y. Sanjalawe and T. Althobaiti, "DDoS attack detection in cloud computing based on ensemble feature selection and deep learning," *Comput. Mater. Continua*, vol. 75, no. 2, pp. 3571–3588, 2023, doi: 10.32604/cmc.2023.037386.
- [9] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, 2020, pp. 218–224, doi: 10.1109/ICAIIC48513.2020.9064976.
- [10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
- [11] A. Mansoor, M. Anbar, A. A. Bahashwan, B. A. Alabsi, and S. D. A. Rihan, "Deep learning-based approach for detecting DDoS attack on software-defined networking controller," *Systems*, vol. 11, no. 6, p. 296, Jun. 2023, doi: 10.3390/systems11060296.
- [12] L. D. Manocchio, S. Layeghy, and M. Portmann, "FlowTransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Syst. Appl.*, vol. 241, p. 122564, Jun. 2023, doi: 10.1016/j.eswa.2023.122564.
- [13] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. IEEE Int. Carnahan Conf. Security Technology (ICCST)*, Chennai, India, 2019, pp. 1–8, doi: 10.1109/CCST.2019.8888419.
- [14] M. K. Hussein, O. S. Almola, D. F. Hasan, and H. F. Mahmood, "Machine learning-enhanced routing protocols: A Q-learning approach for network performance optimization," *Ingénierie des Systèmes d'Information*, vol. 31, no. 2, pp. 481–488, 2026, doi: 10.18280/isi.310216.
- [15] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/s23135941.
- [16] A. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 2, pp. 876–889, Jun. 2020, doi: 10.1109/TNSM.2020.2971776.
- [17] K. B. Reddy and S. Meera, "DDoS attack detection in cloud using ensemble model tuned with optimal hyperparameter," *Int. J. Adapt. Control Signal Process.*, vol. 38, no. 5, pp. 1594–1620, May 2024, doi: 10.1002/acs.3766.
- [18] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, 2017, doi: 10.1016/j.jnca.2017.02.009.
- [19] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, 2019, doi: 10.1007/s12083-017-0630-0.
- [20] M. Al-Qatf, L. Yu, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.
- [21] F. Hu, S. Zhang, X. Lin, L. Wu, N. Liao, and Y. Song, "Network traffic classification model based on attention mechanism and spatiotemporal features," *EURASIP J. Inf. Secur.*, vol. 2023, no. 1, p. 6, Jul. 2023, doi: 10.1186/s13635-023-00141-4.
- [22] K. Sethi, R. Kumar, N. Prajapati, and P. Bera, "Deep reinforcement learning based intrusion detection system for cloud infrastructure," in *Proc. 12th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2020, pp. 1–6, doi: 10.1109/COMSNETS48256.2020.9027452.
- [23] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2017, pp. 1–8, doi: 10.1109/SMARTCOMP.2017.7946998.

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.