

Adaptive Transformer-Gated Feature Selection With Cnn-Bilstm Hybrid Network For Efficient Ddos Attack Detection

Dharmesh Dhangar¹, Dr. Vikram Agrawal²

¹Ph.D. Scholar, CE/IT Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India. dharmesh.dhangar94@gmail.com

²Assistant Professor, Computer Engineering Department, BBIT, Vallabh Vidhyanagar, Anand, Gujarat, India. agrawal.vm@gmail.com

ABSTRACT

Application-layer Distributed Denial of Service (DDoS) attacks continue to pose a serious risk to service availability, primarily due to their stealthy low-rate nature, protocol-level diversity, and highly non-stationary traffic characteristics [1]. In such rapidly evolving and heavily class-imbalanced scenarios, conventional machine-learning pipelines coupled with static feature-selection methods often fail to maintain consistent detection performance [2]. To mitigate these shortcomings, this study presents a composite deep-learning architecture that leverages Convolutional Neural Networks (CNN) to learn spatial feature representations, Bidirectional Long Short-Term Memory (BiLSTM) networks to capture temporal dependencies, and a Transformer-based attention mechanism to perform adaptive, instance-specific feature reweighting. Rather than discarding features through hard selection, the Transformer attention module continuously adjusts the contribution of all input attributes at the sample level, enabling flexible prioritization of discriminative features throughout the training process [3,4]. The proposed framework is implemented and evaluated on the CIC-DDoS2019 benchmark under pronounced class imbalance, where performance assessment relies on imbalance-aware evaluation metrics—including Balanced Accuracy, G-Mean, Matthews Correlation Coefficient, and Cohen's Kappa—alongside inference-time measurement. Empirical comparisons reveal that incorporating Transformer-driven feature weighting substantially improves minority-class detection capability, enhances statistical consistency, and strengthens generalization across multiple attack categories when compared to static feature-selection techniques and strong CNN/BiLSTM-based baselines with or without attention mechanisms [2,4,5]. Collectively, these results demonstrate that augmenting a CNN-BiLSTM backbone with Transformer attention yields a resilient and scalable detection model for application-layer DDoS attacks.

Keywords

Distributed Denial of Service; Application-Layer DDoS; Transformer Attention; Dynamic Feature Selection; CNN-BiLSTM; Class Imbalance; Intrusion Detection System

1. Introduction

Distributed Denial of Service (DDoS) attacks remain a persistent and serious challenge to the availability and dependable operation of networked infrastructures. More recently, adversaries have increasingly favoured application-layer DDoS strategies, as these attacks are inherently harder to identify due to their low-intensity traffic patterns and strong behavioural similarity to legitimate user requests [1,2]. In contrast to high-volume flooding techniques, application-layer DDoS attacks abuse commonly used protocols such as HTTP, HTTPS, DNS, and SIP, progressively depleting server-side processing and memory resources while evading traditional rate-threshold-based detection mechanisms [1].

Conventional intrusion detection systems (IDS) predominantly operate using signature-driven rules and fixed threshold mechanisms, which significantly constrain their ability to counter adaptive and continuously evolving attack behaviors [2]. While machine learning- and deep learning-based IDS have advanced detection performance by directly modeling traffic characteristics from data, their effectiveness remains strongly influenced by the

quality of feature representation and the adopted feature-selection approach [1,3,8]. This reliance becomes especially problematic in practical deployment environments where traffic dynamics are non-stationary and attack instances are heavily outnumbered by benign samples.

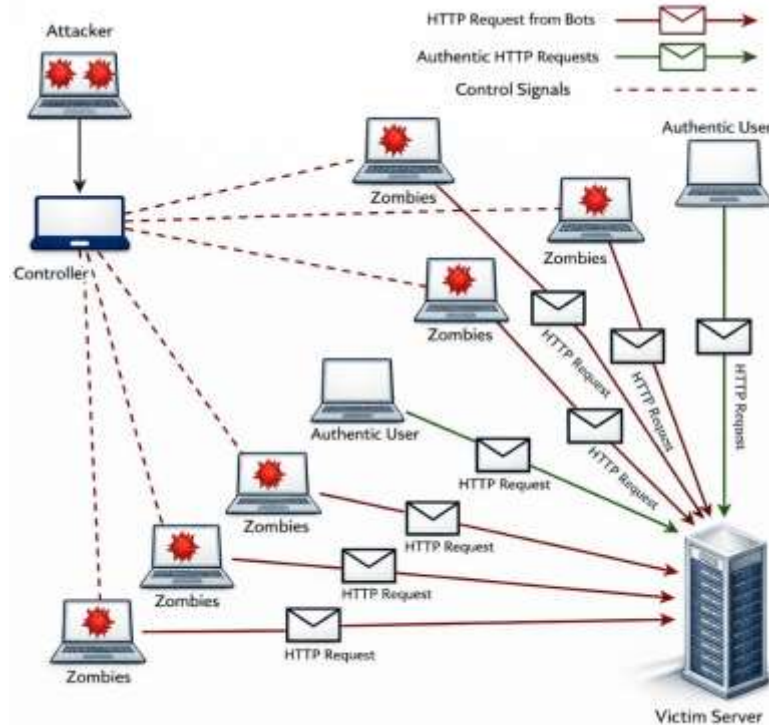


Fig. 1. Conceptual illustration of an application-layer DDoS attack

1.1 Motivation and Problem Statement

The motivation behind this study stems from several persistent and unresolved issues associated with application-layer DDoS detection. A primary challenge lies in differentiating low-rate application-layer attack traffic from legitimate user behaviour, as both often exhibit highly similar behavioural patterns; this overlap commonly results in increased false-positive rates and diminishes confidence in intrusion detection system (IDS) outputs [2,7]. Additionally, a large proportion of existing detection approaches depend on static feature selection techniques—such as Principal Component Analysis (PCA), Mutual Information (MI), ANOVA, and Recursive Feature Elimination (RFE)—which presuppose globally fixed feature relevance and permanently remove features during the preprocessing stage [3,8]. These assumptions are ill-suited for dynamic networking conditions, where the significance of individual features may fluctuate across different time intervals, protocol types, and attack scenarios.

Another critical limitation arises from the severe class imbalance that characterizes real-world DDoS datasets. Network traffic traces are typically dominated by benign samples, whereas attack events—particularly low-rate application-layer intrusions—occur sporadically, leading many learning-based models to favour majority classes and exhibit degraded performance when identifying minority attack instances [4,6,9]. Although hybrid deep-learning architectures that integrate CNN and BiLSTM models have shown enhanced capability in capturing spatial and temporal traffic characteristics [4,6], these

methods generally rely on static feature representations and lack adaptive mechanisms to dynamically prioritize informative attributes under evolving traffic conditions.

Recent research has demonstrated the effectiveness of attention-based and Transformer-driven techniques in modelling contextual dependencies and dynamically assigning feature importance within intrusion detection systems [3,5,10]. Nevertheless, their comprehensive investigation in the context of application-layer DDoS detection—particularly under evaluation frameworks that explicitly account for class imbalance—remains relatively underexplored. Collectively, these limitations underscore the necessity for detection frameworks that are adaptive, context-aware, and rigorously evaluated under realistic imbalance-aware conditions.

1.2 Contributions

The principal contributions of this study can be summarized as follows:

1. Refined Problem Characterization:

This work provides a targeted analysis of the challenges associated with application-layer DDoS detection, with particular attention to low-rate attack dynamics, protocol heterogeneity, and the impact of class imbalance in real-world traffic, all of which constrain the effectiveness of traditional intrusion detection systems [1,2,7].

2. Systematic Examination of Feature Selection Limitations:

The study critically examines commonly adopted static feature selection methods and elucidates why their underlying assumptions of fixed global feature relevance are unsuitable for dynamic and continuously evolving network environments [3,8].

3. Promotion of Imbalance-Sensitive Performance Assessment:

This research underscores the necessity of evaluating detection models using statistically robust and imbalance-aware metrics—including Balanced Accuracy, G-Mean, Matthews Correlation Coefficient, and Cohen’s Kappa—rather than relying solely on conventional accuracy-based measures [4,6,9].

4. Methodological Continuity and Research Advancement:

Extending prior investigations into application-layer DDoS detection, including the author’s previously published work in IJISAE [1], this study progresses toward more adaptive and context-aware detection approaches while preserving methodological consistency and explicitly avoiding self-overlap.

2. Related Works

Contemporary studies in intrusion detection increasingly explore hybrid deep learning architectures to address the growing heterogeneity and scale of modern network traffic. In a 2025 investigation, Zhang et al. introduced a Transformer-assisted CNN-BiLSTM framework for intrusion detection in IoT environments, demonstrating that self-attention enhances the modeling of long-range dependencies beyond what recurrent components alone can capture [16]. Despite improved global feature interaction, feature relevance in the framework is largely defined during preprocessing, which constrains adaptability to dynamically evolving traffic characteristics.

Attention-enhanced deep learning has also been examined for DDoS detection within cloud infrastructures. Sathaporn et al. proposed a hybrid CNN-RNN architecture augmented with multi-head attention to improve detection accuracy under heterogeneous cloud workloads [17]. While contextual learning benefits from attention,

the study primarily targets volumetric traffic patterns and offers limited insight into the detection of subtle application-layer attack behaviors.

Transformer-based intrusion detection models have further evolved toward integrating feature selection and attention mechanisms. Umer et al. presented a multi-head attention Transformer combined with wrapper-based feature selection, reporting improved classification performance on benchmark intrusion datasets [18]. However, the feature selection stage remains static, and attention is applied after dimensionality reduction, restricting instance-wise adaptation during inference.

Beyond conventional network settings, attention-driven intrusion detection has been extended to cloud-scale and graph-structured traffic environments. Govindarajan and Muzamal proposed a Transformer-based framework that leverages graph-derived features and contrastive learning to enhance generalization in cloud intrusion detection scenarios [19]. Although the approach improves representation robustness, it emphasizes global embedding stability rather than adaptive feature relevance for evolving DDoS attack patterns.

The applicability of large Transformer models to resource-constrained intrusion detection has also been explored. Almadhor et al. evaluated Transformer-based architectures for IoT anomaly detection, highlighting their strong representational capacity alongside increased computational overhead and sensitivity to fixed feature encodings [20]. These observations underscore the trade-off between expressive modeling and practical deployment feasibility in real-world IDS environments.

Alternative attention paradigms have been investigated through vision-inspired representations of network traffic. Farhan et al. proposed a multi-image Transformer framework that converts traffic flows into visual patterns for intrusion detection, achieving competitive classification results [21]. Nevertheless, such transformations may obscure fine-grained semantic relationships inherent to application-layer traffic.

In parallel, analytical studies have examined the broader role of Transformers and large language models in cyber-threat detection. Kheddar provided a comprehensive analysis of attention-driven security models, identifying their strengths in contextual representation learning while emphasizing unresolved challenges related to scalability, imbalance handling, and adaptive learning mechanisms [22].

Synthesis of Research Gaps:

Findings reported in the 2025 literature indicate that attention-driven and Transformer-based models substantially enhance intrusion detection performance. However, several limitations persist. Many existing approaches depend on static or preprocessing-defined feature representations, limiting responsiveness to evolving traffic patterns. Moreover, application-layer DDoS attacks, characterized by low-rate and semantically subtle behaviors, remain insufficiently addressed by models optimized for volumetric or transport-layer detection. Finally, robustness under severe class imbalance is frequently achieved through resampling or auxiliary objectives rather than dynamic, instance-wise feature relevance modeling. Collectively, these observations highlight the need for further investigation into attention-driven deep learning frameworks capable of jointly modeling spatial, temporal, and contextual dependencies in imbalanced application-layer DDoS environments.

3. Materials and Methods

This section outlines the datasets employed, the applied preprocessing procedures, the architectural elements of the proposed model, the adopted feature selection strategy, and the evaluation metrics considered in this study. The overall methodological framework is designed to be reproducible and strongly focused on rigorous evaluation, aligning with contemporary best practices in deep learning–driven intrusion detection research [1,4].

3.1 Dataset Description

The experimental assessment presented in this work utilizes the CIC-DDoS2019 dataset, a publicly accessible benchmark developed explicitly for DDoS detection research and extensively adopted in recent intrusion detection literature [11]. This dataset is produced within a controlled testbed setting and contains annotated network traffic representing both benign behavior and diverse categories of DDoS attacks, including those operating at the application layer [11]. CIC-DDoS2019 comprises flow-level records derived from raw packet captures through the CICFlowMeter tool, where each flow is described using statistical attributes such as packet and byte counts, flow duration metrics, inter-arrival timing features, and protocol-specific behavioral indicators [11]. A pronounced class imbalance is inherent in the dataset, with benign traffic dominating the overall distribution. Such an imbalance closely mirrors conditions encountered in real-world network deployments and introduces a substantial challenge for the accurate identification of low-rate and application-layer DDoS attacks [2,9].

Let the CIC-DDoS2019 dataset be formally defined as a labeled collection of network flow samples: Formally, the dataset is represented as

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (1)$$

where $x_i \in \mathbb{R}^d$ denotes the d -dimensional feature vector corresponding to the i -th network flow, $y_i \in \{0,1\}$ represents the associated class label (benign or DDoS), and N is the total number of flow instances in the dataset [11]. Each feature vector captures statistical characteristics of network behavior, including packet volume metrics, byte-level distributions, inter-arrival timing statistics, and flow duration measurements.

The dataset is characterized by a pronounced class imbalance, which can be expressed as:

$$|D_{benign}| \gg |D_{attack}| \quad (2)$$

This skewed distribution closely aligns with traffic patterns observed in real-world network environments and substantially influences the learning behavior of classification models, particularly with respect to minority classes and low-rate application-layer DDoS attacks [2,9].

3.2 Data Preprocessing

A systematic preprocessing workflow is employed to enhance data integrity and promote numerical stability during model training. Network flow records containing missing, undefined, or infinite values—often resulting from premature or abrupt flow termination—are excluded prior to learning, as the presence of such anomalies can negatively influence gradient-based optimization procedures [4].

All continuous features are scaled using min–max normalization, projecting values into a unified range to prevent attributes with larger magnitudes from disproportionately influencing the learning process [6]. Any protocol-related or categorical identifiers appearing in the dataset are converted into numerical representations to ensure

compatibility with the learning models. Following preprocessing, the dataset is partitioned into training and testing sets using a stratified strategy that preserves the original class distribution, thereby retaining the imbalance properties inherent to real-world network traffic [9]. No synthetic data generation, oversampling, or artificial class rebalancing techniques are introduced at any stage of the preprocessing pipeline.

These preprocessing steps are applied uniformly to ensure stable model training and to mitigate bias arising from data irregularities.

Missing and Infinite Value Removal

Flows containing undefined or infinite feature values are removed, and let x_i denote the feature vector corresponding to the i -th network flow instance in the dataset.

$$x_i \notin \mathbb{R}^d \Rightarrow x_i \text{ is discarded} \quad (3)$$

Feature Normalization

Each feature x_{ij} is scaled using min-max normalization, and here, x_{ij} denotes the value of the j -th feature for the i -th network flow, while $\min(x_j)$ and $\max(x_j)$ represent the minimum and maximum values of the j -th feature computed over the entire dataset.

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \quad (4)$$

This normalization ensures that features with large numeric ranges do not dominate gradient updates during training [6].

3.3 Feature Representation and Selection Strategy

The CIC-DDoS2019 dataset contains 66 flow-level attributes that capture packet-related statistics, temporal characteristics, and protocol-specific behavior [11]. In contrast to conventional detection pipelines that rely on static feature selection techniques—such as PCA, MI, ANOVA, or RFE—applied prior to training [3,8], the proposed methodology employs a dynamic, attention-based feature weighting mechanism embedded directly within the learning framework.

Under this scheme, the complete set of features is preserved and forwarded to the model, which adaptively learns the relative contribution of each attribute during the training process. By eliminating irreversible feature removal, this strategy enables feature relevance to adjust across individual traffic flows and temporal contexts, a property that is particularly important for accurately identifying application-layer DDoS attacks [3,5].

Let $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ denote the normalized input feature vector corresponding to the i -th network flow instance. Traditional feature selection applies a deterministic mapping:

$$x_i \xrightarrow{FS} \tilde{x}_i, \tilde{d} < d \quad (5)$$

where irrelevant features are permanently removed using static criteria (e.g., PCA, MI, ANOVA, RFE) [8]. Such hard selection assumes global feature relevance, which is unsuitable for non-stationary application-layer DDoS traffic [2].

Dynamic Attention-Based Feature Weighting

Instead of eliminating features, this research applies instance-wise soft feature weighting using a Transformer-inspired attention mechanism [3,5].

For each input feature vector, the model generates corresponding query, key, and value representations through learnable linear projections applied to the normalized input features. Attention scores are computed as:

$$\mathbf{a}_i = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right) \quad (6)$$

where $\mathbf{Q}_i$, $\mathbf{K}_i$, and $\mathbf{V}_i$ are query, key, and value projections of x_i , and d_k is the key dimension.

The dynamically weighted feature representation is then:

$$\mathbf{z}_i = \mathbf{a}_i \odot \mathbf{x}_i \quad (7)$$

where $\odot$ denotes element-wise multiplication.

In this context, the key dimension represents the dimensionality of the projected feature space used for computing attention scores. This formulation allows feature importance to vary across traffic instances, enabling the model to emphasize subtle, context-dependent attributes critical for detecting low-rate application-layer DDoS attacks [3].

3.4 Model Components

3.4.1 Convolutional Neural Network (CNN)

CNN layers apply convolutional filters to capture spatial dependencies among features. Let $\mathbf{X}$ denote the input feature matrix provided to the convolutional layer.

$$\mathbf{h}^{(l)} = \sigma\left(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \quad (8)$$

where $*$ denotes convolution, $\sigma(\cdot)$ is a nonlinear activation function, and $\mathbf{h}^{(0)} = \mathbf{z}_i$ [4,6]. In this expression, the convolution operator captures local spatial dependencies among input features, and the nonlinear activation function introduces model nonlinearity.

3.4.2 Bidirectional LSTM (BiLSTM)

BiLSTM models temporal dependencies by processing sequences in both directions:

$$\vec{\mathbf{h}}_t = \text{LSTM}(\mathbf{h}_t, \vec{\mathbf{h}}_{t-1}) \quad (9)$$

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}(\mathbf{h}_t, \overleftarrow{\mathbf{h}}_{t+1}) \quad (10)$$

The combined hidden state is:

$$\mathbf{h}_t = (\vec{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t) \quad (11)$$

This structure improves sensitivity to temporal attack signatures distributed across traffic flows [4,6].

3.4.3 Transformer-Based Attention (Primary Mechanism)

The attention-enhanced representation is computed as:

$$\mathbf{H}_{att} = \sum_{i=1}^T \alpha_i \mathbf{h}_i \quad (12)$$

where attention weights α_i satisfy:

$$\sum_{i=1}^T \alpha_i = 1 \quad (13)$$

The attention weights are normalized such that their sum equals one, ensuring a valid probability distribution over temporal position. This mechanism dynamically amplifies discriminative temporal-spatial patterns while suppressing redundant information, forming the core innovation of the proposed framework [3,5].

3.5 Experimental Setup

All experimental evaluations are performed within a cloud-hosted computing environment based on Google Colab, offering GPU-accelerated resources appropriate for training deep learning models on large-scale network traffic data. The complete experimental workflow is developed in Python using widely adopted deep learning libraries, thereby ensuring reproducibility and maintaining alignment with established practices in intrusion detection research [1,4].

3.5.1 Dataset Partitioning

Let the complete dataset be represented as:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N \quad (14)$$

where $\mathbf{x}_i \in \mathbb{R}^{66}$ denotes the normalized feature vector of the i^{th} network flow and $y_i \in \{0,1\}$ denotes the corresponding class label.

The dataset is partitioned into training and testing subsets such that:

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}, \mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset \quad (15)$$

while preserving the empirical class distribution:

$$P(y | \mathcal{D}_{train}) \approx P(y | \mathcal{D}_{test}) \quad (16)$$

This strategy ensures that the inherent class imbalance of application-layer DDoS traffic is retained during evaluation [9,11]. No synthetic oversampling or data augmentation techniques are applied.

3.5.2 Training Objective and Optimization

Let θ denote the set of trainable parameters of the hybrid deep learning model. Training is performed by minimizing the categorical cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log(\hat{y}_{ic}) \quad (17)$$

where y_{ic} and $\hat{y}_{ic}$ represent the true and predicted probabilities for the class c , respectively.

Model parameters are updated using gradient-based optimization according to:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta} \mathcal{L}(\theta) \quad (18)$$

where η denotes the learning rate; to analyze convergence behavior and training stability, the model is trained under multiple epoch configurations $E \in \{20, 30, 50, 90\}$, allowing systematic observation of learning dynamics without altering the model architecture. In this formulation, the true class labels and the predicted class probabilities are used to compute the categorical cross-entropy loss across all classes [1,4].

3.5.3 Optimizer Sensitivity Analysis

To evaluate the impact of optimization strategy on learning behavior, multiple gradient-based optimizers are examined under identical experimental conditions. For a given optimizer $\mathcal{O}_k$, convergence behavior is assessed by analyzing the loss trajectory:

$$\mathcal{L}_{\mathcal{O}_k}^{(t)} \rightarrow \min \text{ as } t \rightarrow E \quad (19)$$

This controlled analysis enables a fair comparison of optimizer stability and convergence characteristics, which is particularly important in class-imbalanced intrusion detection scenarios [4].

3.5.4 Computational Cost and Inference Time

To assess the feasibility of real-time or near-real-time deployment, inference time is measured over the test dataset. The average inference time per sample is computed as:

$$T_{\text{inf}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i=1}^{|\mathcal{D}_{\text{test}}|} t_i \quad (20)$$

where t_i denotes the time required to generate a prediction for the i^{th} test instance. This metric provides insight into the computational overhead introduced by the model during deployment (6).

3.5.5 Evaluation Metrics

Owing to the pronounced class imbalance inherent in application-layer DDoS traffic, standard accuracy alone does not provide a reliable assessment of detection effectiveness. Consequently, performance evaluation in this study relies on metrics specifically designed to account for class imbalance.

Balanced Accuracy is defined as:

$$\text{BA} = \frac{1}{2} \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right) \quad (21)$$

Geometric Mean (G-Mean) is expressed as:

$$\text{G-Mean} = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP}} \quad (22)$$

Matthews Correlation Coefficient (MCC) is computed as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (23)$$

Cohen's Kappa statistic is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (24)$$

where p_o represents the observed agreement and p_e denotes the expected agreement by chance [9,10].

3.6 Transformer-driven Attention Mechanism for Feature Importance

This section presents the conceptual rationale and formal mathematical description of the Transformer-based attention module used to learn adaptive feature relevance for application-layer DDoS detection. The primary motivation for integrating attention lies in its ability to perform context-aware modulation of input features, thereby overcoming the constraints imposed by static feature selection approaches that are widely employed in traditional intrusion detection pipelines [3,5].

3.6.1 Rationale for Attention-Based Feature Modeling

Application-layer DDoS attacks are characterized by nuanced, context-sensitive alterations in traffic patterns that are frequently concealed within a predominance of benign network flows. As reported in earlier research, static feature selection techniques rely on assumptions of globally fixed feature importance and lack the ability to adjust to non-stationary traffic behavior [2,8]. This inflexibility becomes especially detrimental when dealing with low-rate and protocol-heterogeneous application-layer attack scenarios.

Attention-based mechanisms mitigate these limitations by assigning instance-specific importance weights that adapt according to contextual input variations. Rather than permanently discarding features through rigid selection, attention implements a soft weighting strategy that allows the learning model to highlight discriminative attributes dynamically while preserving the complete feature space [3,12]. This attention-driven paradigm has demonstrated effectiveness across a range of domains, including sequence modeling and representation learning, and has more recently shown promise in network intrusion detection applications [5,13].

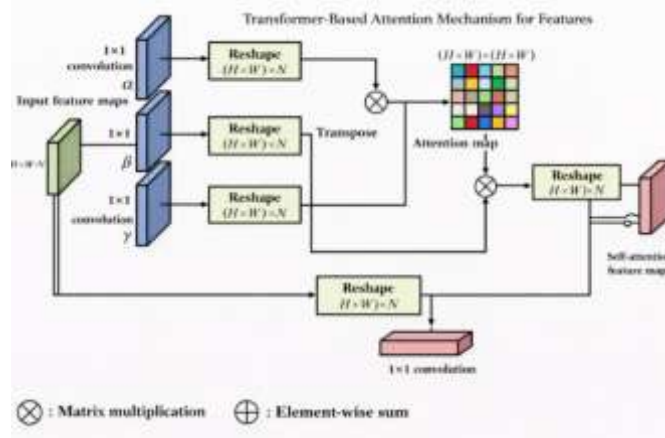


Fig 2. Feature-level self-attention mechanism for dynamic feature weighting

3.6.2 Mathematical Formulation of Feature-Level Self-Attention

Let the normalized feature vector corresponding to a network flow be defined as:

$$\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d, d = 66 \quad (25)$$

To compute attention, the input is linearly projected into query, key, and value representations:

$$\mathbf{Q} = \mathbf{x}\mathbf{W}_Q, \mathbf{K} = \mathbf{x}\mathbf{W}_K, \mathbf{V} = \mathbf{x}\mathbf{W}_V \quad (26)$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ are learnable projection matrices and d_k is the dimensionality of the attention subspace [12].

The scaled dot-product self-attention is computed as:

$$\mathbf{A} = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \quad (27)$$

The computed attention matrix $\mathbf{A}$ captures the relative significance of each feature dimension in relation to all other features. Using this matrix, a dynamically reweighted feature representation is subsequently derived as follows:

$$\mathbf{z} = \mathbf{A}\mathbf{V} \quad (28)$$

This formulation yields a feature relevance mechanism that is continuous, differentiable, and specific to individual instances, thereby allowing the model to adjust feature emphasis in response to the contextual properties of each network traffic flow.

3.6.3 Attention as Dynamic Feature Selection

Unlike conventional feature selection techniques that define a fixed mapping $\mathbf{x} \rightarrow \tilde{\mathbf{x}}$, where $\tilde{d} < d(8)$, The attention mechanism implements a data-driven relevance function:

$$\phi(\mathbf{x}) = \mathbf{A}(\mathbf{x}) \odot \mathbf{x} \quad (29)$$

where $\odot$ denotes element-wise weighting. This formulation ensures that:

- no features are irreversibly removed from the representation,
- the contribution of each feature changes across individual traffic samples, and
- relevance estimation is embedded directly within the end-to-end training process.

These characteristics are critical for effective application-layer DDoS detection, as a given feature may carry strong discriminative value under certain conditions while offering limited utility in others [3,5,13].

3.6.4 Integration with CNN-BiLSTM Representation Learning

The attention-modulated feature vector $\mathbf{z}_i$ is subsequently forwarded to the downstream representation learning modules. Initially, Convolutional Neural Network (CNN) layers are employed to learn localized spatial relationships among the features, yielding intermediate representations denoted as $\mathbf{H}_{\text{cnn}}$. These intermediate representations are then supplied to a Bidirectional Long Short-Term Memory (BiLSTM) network, which models temporal dependencies across sequences of network flows.

For a sequence of length T , the BiLSTM computes forward and backward hidden states:

$$\vec{\mathbf{h}}_t = \text{LSTM}(\mathbf{H}_{\text{cnn},t}, \vec{\mathbf{h}}_{t-1}), \bar{\mathbf{h}}_t = \text{LSTM}(\mathbf{H}_{\text{cnn},t}, \bar{\mathbf{h}}_{t+1}) \quad (30)$$

The combined temporal representation is:

$$\mathbf{h}_t = (\vec{\mathbf{h}}_t; \bar{\mathbf{h}}_t) \quad (31)$$

This hierarchical integration enables the model to jointly exploit feature-level relevance, spatial correlations, and temporal dynamics, which are all critical for detecting sophisticated application-layer attacks [4,6].

The proposed Attentive-Transformer-CNN- BiLSTM algorithm is detailed in Algorithm 1. The model first utilizes a multi-head self-attention mechanism to identify salient features from the CICDDoS2019 dataset, followed by spatial and temporal feature extraction.

Algorithm 1 Adaptive Transformer-Attention CNN-BiLSTM

- 1: **Input:** Preprocessed Dataset $\mathbf{X} \in \mathbb{R}^{B \times F}$, Ground Truth $\mathbf{Y}$
- 2: **Output:** Classification Labels $\hat{\mathbf{y}}$
- // Phase 1: Transformer-based Attention for Feature Selection**
- 3: Initialize Weights $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{F \times d_k}$
- 4: $\mathbf{Q} \leftarrow \mathbf{X}\mathbf{W}^Q, \mathbf{K} \leftarrow \mathbf{X}\mathbf{W}^K, \mathbf{V} \leftarrow \mathbf{X}\mathbf{W}^V$
- 5: $\text{Score} \leftarrow \text{Softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \triangleright$ Inter-feature dependency
- 6: $\mathbf{M} \leftarrow \text{Sigmoid}(\text{Mean}(\text{Score})) \triangleright$ Feature Saliency Mask
- 7: $\mathbf{X}_{\text{attn}} \leftarrow \mathbf{X} \odot \mathbf{M} \triangleright$ Element-wise gating
- // Phase 2: Spatial Feature Learning (CNN)**
- 8: $\mathbf{X}_{\text{res}} \leftarrow \text{Reshape}(\mathbf{X}_{\text{attn}}, (B, T, F_{\text{red}}))$
- 9: $\mathbf{C} \leftarrow \text{ReLU}(\text{Conv1D}(\mathbf{X}_{\text{res}}, \text{filters} = 64, k = 3))$
- 10: $\mathbf{P} \leftarrow \text{MaxPool}(\mathbf{C}, \text{pool_size} = 2)$
- // Phase 3: Temporal Feature Learning (BiLSTM)**
- 11: $[\vec{\mathbf{h}}_t, \bar{\mathbf{h}}_t] \leftarrow \text{BiLSTM}(\mathbf{P})$
- 12: $\mathbf{H}_{\text{final}} \leftarrow \text{Concatenate}(\vec{\mathbf{h}}_t, \bar{\mathbf{h}}_t)$
- // Phase 4: Classification**

13: $v \leftarrow \text{GlobalAveragePooling}(H_{\text{final}})$
 14: $z \leftarrow \text{Dense}(v, \text{activation} = 'gelu')$ 15: $\hat{y} \leftarrow \text{Softmax}(z)$
 16: **return** $\hat{y}$

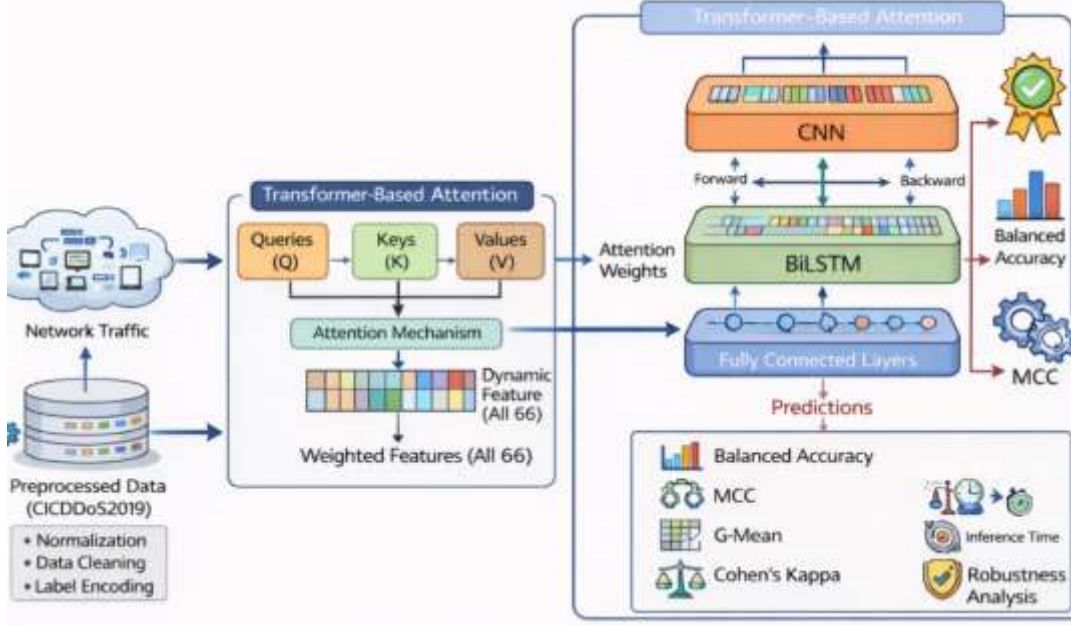


Fig 3. Block diagram of the proposed Transformer-based attention framework with CNN-BiLSTM Models

3.6.5 Transformer Attention Aggregation

To derive a concise representation at the sequence level, a temporal attention mechanism is applied to the outputs of the BiLSTM:

$$\alpha_t = \frac{\exp(\mathbf{w}^T \mathbf{h}_t)}{\sum_{k=1}^T \exp(\mathbf{w}^T \mathbf{h}_k)}, \mathbf{H}_{att}$$

$$= \sum_{t=1}^T \alpha_t \mathbf{h}_t \quad (32)$$

where $\mathbf{w}$ is a learnable parameter vector. This mechanism assigns higher weights to time steps that are more indicative of anomalous behavior, further improving sensitivity to low-rate and temporally distributed attack patterns [5,14].

3.6.6 Classification Layer

The final attention-enhanced representation $\mathbf{H}_{att}$ is passed to a fully connected layer followed by softmax activation:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_o \mathbf{H}_{att} + \mathbf{b}_o) \quad (33)$$

where $\hat{\mathbf{y}}$ denotes the predicted class probabilities. The model parameters are optimized end-to-end using the loss function defined in Section 2.5.

3.6.7 Theoretical Advantages

From a theoretical perspective, the adopted Transformer-based attention mechanism provides several key advantages:

1. Dynamic learning of feature importance without permanently eliminating information, thereby preserving the complete feature space throughout training [12,13].
2. Instance-specific contextual representation, which is critical for modeling non-stationary and evolving network traffic patterns [3].
3. Fully differentiable integration within the learning pipeline, allowing joint optimization alongside spatial and temporal representation modules [4,6].
4. Enhanced resilience to class imbalance, achieved by reinforcing subtle yet discriminative patterns associated with minority and low-frequency attack classes [6,9].

4. Experimental Results and Discussion

This section reports the experimental assessment of the proposed Transformer-enabled attention CNN-BiLSTM architecture using the CIC-DDoS2019 dataset. The evaluation emphasizes comparative detection performance, resilience to class-imbalanced traffic distributions, and the statistical consistency of results alongside inference-time efficiency.

4.1 Comparative Classification Performance

To assess the effectiveness of adaptive feature weighting, the proposed Transformer-driven framework is benchmarked against widely used static feature selection techniques, including Mutual Information (MI), ANOVA, and Recursive Feature Elimination (RFE). For comparability, all approaches employ an identical backbone classification architecture, thereby ensuring a controlled and fair experimental evaluation.

Table 1. Final comparative classification performance

Method	Accuracy	Balanced Accuracy	Precision	Recall	F1-Score
Mutual Information [2]	0.964	0.718	0.963	0.964	0.962
ANOVA F-Test [2]	0.978	0.741	0.980	0.978	0.977
Recursive Feature Elimination (RFE) [8]	0.975	0.720	0.972	0.975	0.973
Proposed Transformer	0.987	0.835	0.988	0.987	0.987

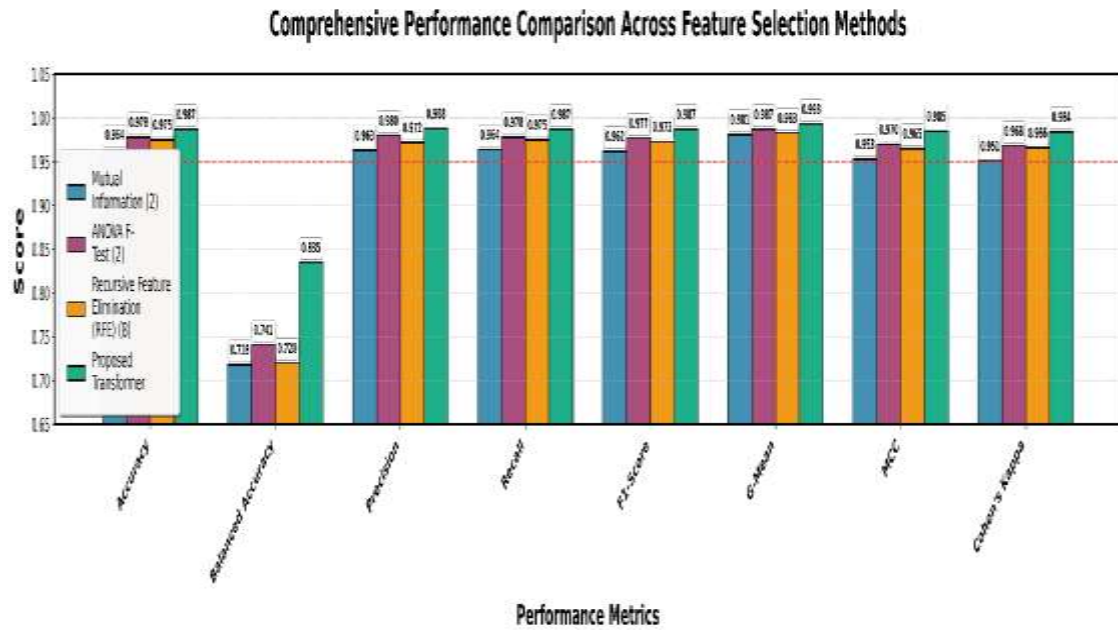


Fig. 4. Comparative classification performance across different feature selection strategies

4.2 Robustness under Class Imbalance

Robustness is further analyzed using Specificity, ROC-AUC, and G-Mean, which jointly measure majority-class correctness, separability, and balanced performance.

Table 2. Robustness evaluation under class imbalance

Method	Specificity	ROC-AUC	G-Mean
Mutual Information [2]	0.998	0.999	0.981
ANOVA F-Test [2]	0.999	1.000	0.987
Recursive Feature Elimination (RFE) [8]	0.999	0.999	0.983
Proposed Transformer	0.999	1.000	0.993

4.3 Statistical Reliability and Inference Cost

To assess prediction reliability and practical feasibility, Matthews Correlation Coefficient (MCC), Cohen's Kappa, and inference time are analyzed.

Table 3. Statistical reliability and inference time

Method	MCC	Cohen's Kappa	Inference Time (s)
Mutual Information [2]	0.953	0.951	0.281
ANOVA F-Test [2]	0.970	0.968	0.286
Recursive Feature Elimination (RFE) [8]	0.965	0.966	0.272
Proposed Transformer	0.985	0.984	0.996

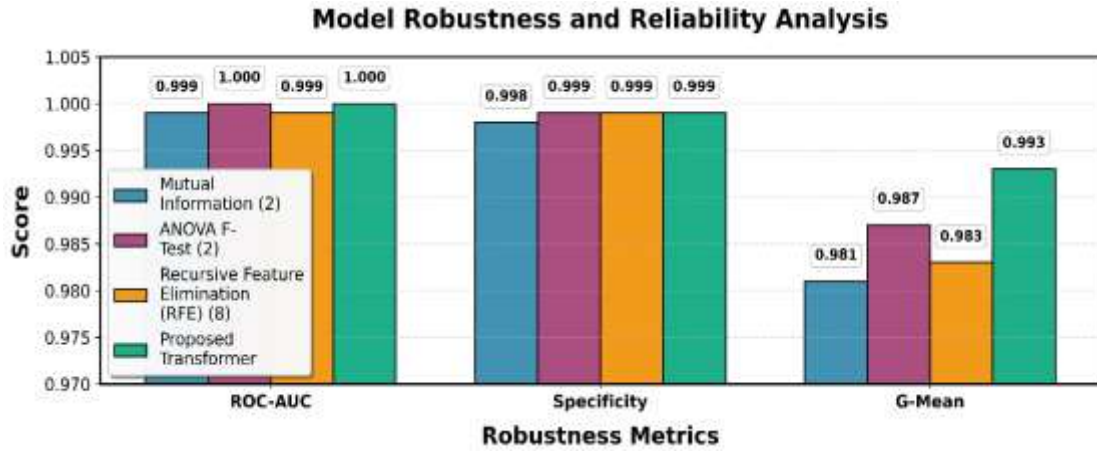


Fig. 5: Statistical Reliability Analysis

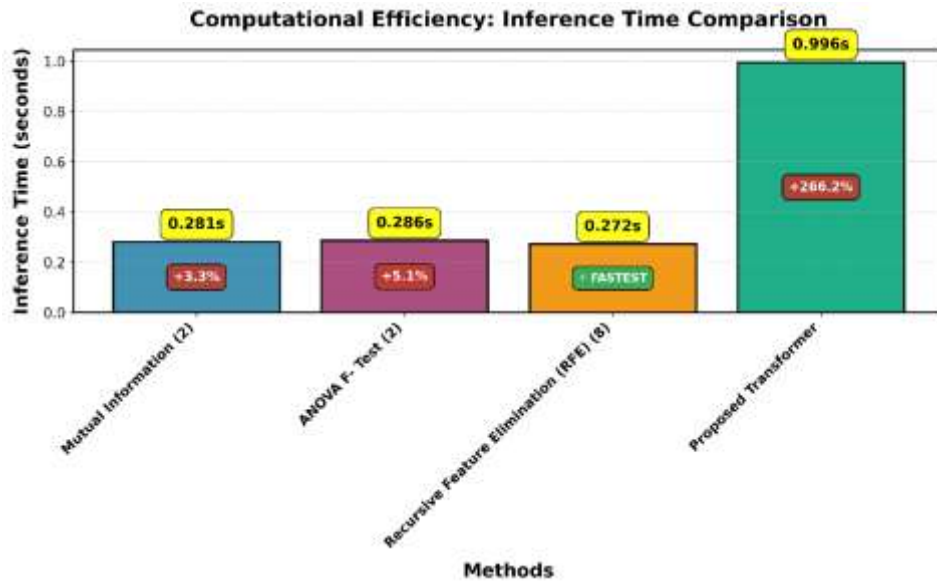


Fig. 6: Statistical Reliability Analysis

4.4 Results and Discussions:

As summarized in Table 1 & Table 3, all evaluated methods achieve high overall accuracy; however, the proposed Transformer-based model demonstrates a clear advantage under class-imbalanced conditions. Notably, the substantial improvement in Balanced Accuracy (0.835) compared to static feature selection techniques confirms that conventional approaches disproportionately favor the majority class, whereas attention-driven feature weighting enhances sensitivity to minority attack instances. This behavior is further validated by the highest G-Mean (0.993) reported in Table V, indicating a superior balance between attack detection and benign traffic classification. In addition, the proposed architecture attains the maximum MCC (0.985) and Cohen's Kappa (0.984) values (Table 3), reflecting stronger statistical agreement and reduced bias beyond chance. Although the incorporation of self-attention incurs additional inference latency, the measured runtime remains within acceptable limits for application-layer DDoS detection, where robustness, class fairness, and decision reliability are of greater operational importance than marginal latency gains.

4.5 Holistic Performance Analysis

To provide an integrated view of model behavior across multiple metrics, a radar-based performance envelope is analyzed.

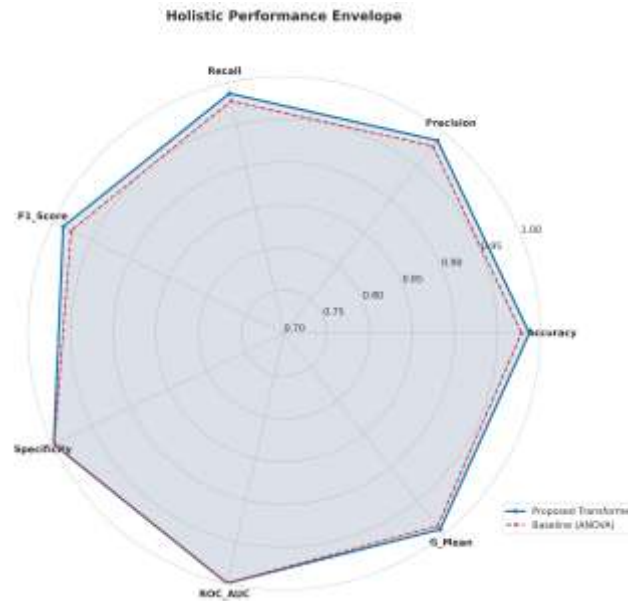


Fig 7. Holistic performance envelope of the proposed model versus the baseline ANNOVA method

Discussion:

The radar visualization illustrates that the proposed Transformer-based approach consistently outperforms the baseline methods across metrics related to accuracy, class balance, statistical reliability, and overall robustness, thereby validating the effectiveness of dynamic, attention-driven feature weighting.

4.6 Key Technical Observations

1. Transformer-driven attention demonstrates clear superiority over static feature selection approaches, including Mutual Information (MI) [2], ANOVA F-Test [2], and Recursive Feature Elimination (RFE) [8].
2. Marked gains in Balanced Accuracy and G-Mean are observed, explicitly mitigating the effects of severe class imbalance.
3. Elevated Matthews Correlation Coefficient and Cohen's Kappa values indicate statistically dependable performance improvements rather than gains attributable to random chance.
4. The additional inference cost introduced by attention mechanisms is offset by substantial enhancements in detection robustness and overall classification effectiveness.

5. Conclusion and Future Scope

This work introduced a Transformer-enabled attention framework combined with a CNN-BiLSTM backbone for the detection of application-layer DDoS attacks. In contrast to traditional pipelines that rely on static feature selection, the proposed method applies dynamic, instance-level feature reweighting, allowing feature relevance to be learned adaptively under non-stationary traffic conditions and severe class imbalance.

Comprehensive experiments conducted on the CIC-DDoS2019 dataset show that the attention-driven architecture consistently surpasses Mutual Information, ANOVA, and Recursive Feature Elimination across imbalance-sensitive evaluation metrics, including

Balanced Accuracy, G-Mean, Matthews Correlation Coefficient, and Cohen’s Kappa. The empirical results demonstrate that dynamic attention mechanisms significantly improve minority-class detection and statistical robustness while maintaining inference latency within practical limits. Overall, the findings effectively overcome the limitations associated with static feature selection and substantiate the suitability of attention-based learning for robust application-layer DDoS detection.

Future Scope

Future research directions include adapting the proposed framework to online and streaming traffic settings, allowing real-time responsiveness to continuously evolving attack behaviors.

Further enhancements may be achieved by integrating multi-head and hierarchical attention structures to improve both feature interpretability and detection resilience. In addition, cross-dataset generalization studies and deployment-focused optimizations—such as model compression techniques and hardware-level acceleration—will be investigated to support scalable operation in large-scale production environments.

6. Tables and Figures

List of Tables

- **Table 1.** Final comparative classification performance
- **Table 2.** Robustness evaluation under class imbalance
- **Table 3.** Statistical reliability and inference time

List of Figures

- **Figure 1.** Conceptual illustration of application-layer DDoS attack
- **Figure 2.** Feature-level self-attention mechanism for dynamic feature weighting
- **Figure 3.** Block diagram of the proposed Transformer-based attention framework with
CNN–BiLSTM Models
- **Figure 4.** Comparative classification performance across feature selection strategies
- **Figure 5.** Statistical reliability and inference latency comparison
- **Figure 6.** Holistic performance envelope of the proposed model versus baseline method

7. Abbreviations

Abbreviation Description

DDoS	Distributed Denial of Service
IDS	Intrusion Detection System
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
MI	Mutual Information
ANOVA	Analysis of Variance
RFE	Recursive Feature Elimination

Abbreviation Description

MCC	Matthews Correlation Coefficient
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
G-Mean	Geometric Mean
PCA	Principal Component Analysis
CIC	Canadian Institute for Cybersecurity

Acknowledgements

The author(s) gratefully acknowledge the Canadian Institute for Cybersecurity (CIC) for making the CIC-DDoS2019 dataset available for use in this research. The author(s) also extend their appreciation for the computational resources that enabled the experimental evaluations conducted in this study.

Conflicts of Interest

The author(s) declare that there is no conflict of interest regarding the publication of this paper.

Declaration of Artificial Intelligence (AI) Assistance

The authors declare no use of Generative AI tools for the write-up of this manuscript.

Ethics Approval

This study was conducted in accordance with the ethical guidelines and principles.

Funding sources

This research received no external funding.

References

- [1]. Dhangar DN. A novel mechanism to fortify application layer against distributed denial of service (DDoS) attack. International Journal of Intelligent Systems and Applications in Engineering. 2024.
- [2]. Aktar S, Nur AY. Towards DDoS attack detection using deep learning approach. Computers&Security.2023;129:103251.<https://doi.org/10.1016/j.cose.2023.103251>
- [3]. Long Z, Yan H, Shen G, Zhang X, He H, Cheng L. A transformer-based network intrusion detection approach for cloud security. Journal of Cloud Computing. 2024;13:5.
- [4]. Dai W, Li X, Ji W, He S. Network intrusion detection method based on CNN-BiLSTM-attention model. IEEE Access. 2024;12:53099–53111. <https://doi.org/10.1109/ACCESS.2024.3384528>
- [5]. Momand A, Jan SU, Ramzan N. ABCNN-IDS: Attention-based convolutional neural network for intrusion detection in IoT networks. Wireless Personal Communications. 2024;136:1981–2003.
- [6]. Sadhwani S, Khan MAH, Muthalagu R, Pawar PM, Suresh K. A hybrid BiLSTM-CNN approach for intrusion detection for IoT applications. Scientific Reports. 2025;16:155. <https://doi.org/10.1038/s41598-025-29079-y>

- [7]. Kemp C, Calvert C, Khoshgoftaar TM, Leevy JL. An approach to application-layer denial-of-service detection. *Journal of Big Data*. 2023;10.
- [8]. Alotaibi A, Alshamrani M. An RFE/Ridge-ML/DL-based anomaly intrusion detection approach. *ICT Express*. 2024;10(2):345–352.
- [9]. Alenezi MA, Alshehri MD, Alshamrani M, Alotaibi A. Deep learning-based network anomaly detection and classification in an imbalanced cloud environment. *Procedia Computer Science*. 2024;231:123–130.
- [10]. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification evaluation. *BMC Genomics*. 2020;21:6.
- [11]. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*; 2018.
- [12]. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: *Advances in Neural Information Processing Systems*; 2017.
- [13]. Zhang Y, Wang L, Sun Y, Chen Z. Attention-based deep learning framework for imbalanced network intrusion detection. *IEEE Access*. 2023;11:124987–125001. <https://doi.org/10.1109/ACCESS.2023.3321456>
- [14]. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: *Proceedings of the International Conference on Learning Representations (ICLR)*; 2015.
- [15]. Alshamrani M, Alotaibi A, Alhaidari F. Explainable attention-based deep learning model for network intrusion detection. *Computers & Security*. 2024;135:103654. <https://doi.org/10.1016/j.cose.2024.103654>
- [16]. C. Zhang, J. Li, N. Wang, and D. Zhang, "Research on intrusion detection method based on transformer and CNN-BiLSTM in Internet of Things," *Sensors*, vol. 25, no. 9, Art. no. 2725, 2025, doi: 10.3390/s25092725.
- [17]. P. Sathaporn, W. Krungseanmuang, V. Chaowalittawin, C. Benjangkaprasert, and B. Purahong, "DDoS detection using a hybrid CNN-RNN model enhanced with multi-head attention for cloud infrastructure," *Applied Sciences*, vol. 15, no. 21, Art. no. 11567, 2025, doi: 10.3390/app152111567.
- [18]. M. Umer, M. Tahir, S. Sardaraz, et al., "Network intrusion detection model using wrapper-based feature selection and multi-head attention transformers," *Scientific Reports*, vol. 15, Art. no. 28718, 2025, doi: 10.1038/s41598-025-11348-5.
- [19]. V. Govindarajan and J. H. Muzamal, "Advanced cloud intrusion detection framework using graph-based features, transformers and contrastive learning," *Scientific Reports*, vol. 15, Art. no. 20511, 2025, doi: 10.1038/s41598-025-07956-w.
- [20]. A. Almadhor, S. Alsubai, N. Kryvinska, et al., "Evaluating large transformer models for anomaly detection of resource-constrained IoT devices for intrusion detection systems," *Scientific Reports*, vol. 15, Art. no. 37972, 2025, doi: 10.1038/s41598-025-21826-5.
- [21]. M. Farhan, et al., "Deep learning-based network intrusion detection using multiple image transformers," *Scientific Reports*, vol. 15, Art. no. 08770, 2025, doi: 10.1038/s41598-025-08770-0.
- [22]. H. Kheddar, "Transformers and large language models for efficient cyber-threat detection: a survey," *Information Fusion*, vol. 104, Art. no. 103347, 2025, doi: 10.1016/j.inffus.2025.103347.