Enhancing Web Server Security Using Deepfocusguard Based Skydive Optimization For AI-Based Ddos Prediction And Prevention

R. Raja[1], Dr. Venkadesh.R[2], Dr. C. Ezhilazhagan[3], A. M. Sarravanaprabhu[4]

[1]Ph.D., Department of Information and Communication, Anna University
rajavaibavi0609@gmail.com
[2]Professor and HOD, Department of Computer Science and Engineering, Velammal Institute of Technology, Chennai-601204, India
[3]Associate Professor, Department of Electronics and Communication Engineering, VelTech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India
[4]Assistant Professor, Department of Computer Science and Engineering, Velammal Institute of Technology, Panchetti, Thiruvallur, Tamil Nadu, India

Abstract

Modern web servers face an increasing threat from Distributed Denial-of-Service (DDoS) attacks, which can overwhelm resources, disrupt services, and compromise availability, yet traditional defense mechanisms often fail to cope with high traffic volume, evolving attack patterns, and adaptive adversaries. To overcome these challenges, this research proposes DeepFocusGuard Based SkyDive Optimization and is developed for accurate DDoS prediction, efficient prevention, and enhanced web server security. Initially the dataset transforms through data preprocessing, including feature transformation to integer format, handling of missing values to ensure data completeness, and label encoding to convert categorical attributes into machine-readable form. Feature selection is performed using correlation filtering to eliminate redundant attributes, followed by Adaptive Synthetic Sampling Approach (ADASYN) oversampling to balance traffic classes between normal and malicious flows. The processed data is then classified using the DeepFocusGuard framework, where Convolutional Neural Networks (CNN) are employed to extract spatial features from network traffic, Bidirectional Long Short-Term Memory (BiLSTM) captures temporal dependencies across sequential traffic patterns, and the Attention mechanism highlights critical traffic behaviors, collectively enhancing DDoS detection accuracy. To further strengthen performance, SkyDive Optimization developed by combining Pelican Optimization (PO) and Aquilo Optimizer (AO) is applied for adaptive hyperparameter tuning, improving convergence, reducing overfitting, and ensuring generalization across diverse traffic conditions. Experimental results demonstrate that the proposed model achieves superior accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, Attack Mitigation, False Blocking, and Server Availability compared to baseline traditional defenses and state-of-the-art methods, thereby ensuring stronger DDoS prevention and improved web server security. Future research will extend this work by incorporating explainable AI modules for interpretability, real-time adaptive learning against novel attacks, and large-scale deployment in heterogeneous web server environments.

Keywords: Web Server Security; DDoS attack; DDoS Prediction and Prevention; CNN and Attention

1. Introduction

The growing sophistication of DDoS attacks poses a critical challenge to the stability and resilience of modern network infrastructures [1]. As Internet of Things (IoT) devices, cloud platforms, and industrial control systems continue to expand, the attack surface for malicious actors has significantly increased, amplifying the risk of large-scale service disruptions [2]. Traditional rule-based intrusion detection systems often fall short in identifying complex and evolving attack patterns, particularly within Software-Defined Networking (SDN) and edge-enabled environments where traffic

dynamics are highly variable [3]. The sheer scale and velocity of modern DDoS attacks demand automated solutions capable of operating in real-time, far beyond the limitations of conventional signature-based methods [4]. To address these challenges, artificial intelligence (AI)-driven techniques, particularly deep learning frameworks, have emerged as powerful tools for DDoS detection due to their ability to autonomously learn discriminative traffic features, capture long-term temporal dependencies, and adapt to shifting threat landscapes [5]. Hybrid deep models such as those combining CNN, BiLSTM, and attention mechanisms have demonstrated strong potential in enhancing feature extraction, sequential learning, and contextual relevance, enabling higher detection accuracy and faster response in diverse environments [6]. Moreover, the adoption of explainable AI and attention-based layers is increasingly recognized as essential for improving transparency and trust in automated security systems [7].

Despite these advancements, several critical challenges remain unresolved in the current body of research. Many deep learning-based intrusion detection models are highly sensitive to noise and suffer from overfitting, which reduces robustness when applied to heterogeneous or large-scale real-world traffic [8]. Class imbalance within widely used DDoS datasets further constrains classification performance, limiting the ability to reliably identify stealthy or low-frequency attack variants [9]. Additionally, deep learning frameworks frequently incur high computational overhead, which makes them impractical for deployment in latency-sensitive, resource-constrained SDN or edge infrastructures [10]. Ensemble and hierarchical learning-based models have been explored to enhance adaptability [11]. Furthermore, adversaries continue to develop advanced evasion techniques, including adversarial traffic generation, to bypass detection mechanisms, exposing vulnerabilities in static and non-adaptive models [12]. Beyond technical challenges, real-time integration with heterogeneous telemetry sources introduces synchronization and data fusion difficulties, which can degrade detection accuracy and responsiveness [13]. Collectively, these limitations underscore the pressing need for a resilient, lightweight, and adaptive anomaly detection framework that can sustain performance under dynamic, large-scale, and adversarially evolving network conditions.

To overcome the above challenges, this research proposes DeepFocusGuard Based SkyDive Optimization, an AI-enabled DDoS prediction and prevention framework that integrates CNN, BiLSTM, and an Attention mechanism, optimized through the novel SkyDive Optimization strategy. The contributions of the research are as follows:

DeepFocusGuard Model: This model integrates CNN, BiLSTM, and Attention layers to capture spatial traffic patterns, sequential dependencies, and feature relevance from network flows. This layered fusion enhances accuracy, reduces false positives, and ensures adaptive detection of DDoS attacks in dynamic web server environments.

SkyDive Optimization: This optimization that combines the exploratory search strength of PO with the adaptive fine-tuning capability of the AO. This dual strategy enables efficient hyperparameter tuning, faster convergence, and robust generalization, ensuring stable performance across heterogeneous and large-scale network scenarios.

The remainder of this paper is structured as follows. Section 2 reviews the related work and highlights the limitations of existing AI-driven DDoS detection and prevention methods. Section 3 introduces the proposed DeepFocusGuard framework and the SkyDive Optimization strategy, detailing the data preprocessing steps, model architecture, and optimization workflow. Section 4 presents the experimental setup and performance evaluation, comparing the proposed framework against state-of-the-art benchmarks. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Literature Review

Recent advances in deep learning and AI-driven network security have significantly influenced DDoS detection frameworks. Elshewey et al. [14] proposed a hybrid CNN-GRU model for SDN environments, enhancing temporal learning and detection rates in dynamic traffic scenarios, while Golduzian [15] combined machine learning with statistical methods to design lightweight and explainable models for efficient DDoS prediction. Girubagari et al. [16] developed an attention-enabled deep learning approach for anomaly detection, highlighting the importance of prioritizing critical features and reducing false alarms, and Shou et al. [17] further optimized CNN-BiLSTM architectures with attention mechanisms to improve sequence modeling and resilience against complex intrusion patterns. Zhou and Ling [18] introduced cooperative CNN-BiLSTM detection in SDN, enabling distributed decision-making for better detection accuracy, while Zhang et al. [19] integrated Transformer blocks with CNN-BiLSTM for intrusion detection in IoT networks, improving contextual understanding. Yin et al. [20] proposed a multi-scale CNN-BiLSTM arbitration dense network specifically to detect low-rate DDoS attacks that conventional systems often overlook, emphasizing the need for multi-scale feature learning. Jebril et al. [21] applied transfer learning and regularization techniques to reduce overfitting and improve robustness in IoT scenarios, and Zhang et al. [22] implemented BiLSTM-based detection models in edge computing environments to achieve low-latency, accurate inference. Li et al. [23] introduced the ADFCNN-BiLSTM model incorporating attention and deformable convolution layers to enhance feature extraction and adaptability in intrusion detection tasks. Despite these notable advancements, several challenges persist that hinder robust deployment, scalability, and real-time reliability in web server environments. Many detection models struggle to maintain high accuracy against zero-day or adaptive DDoS attacks, as evolving attack patterns undermine their robustness in real-world traffic [14], [15]. Hybrid deep learning systems often misclassify legitimate traffic surges, such as flash crowds, as malicious flows, interrupting service availability and increasing false alarms [16], [17]. Although CNN–BiLSTM and attention-based architectures provide strong accuracy, their computational demands introduce inference delays that weaken real-time mitigation [18], [19]. Under large-scale DDoS floods, detection pipelines can be overwhelmed, and deep sequential models may fail to sustain performance beyond expected traffic thresholds [20], [21]. Furthermore, static hyperparameter tuning and limited regularization contribute to overfitting, reducing model generalization across heterogeneous and dynamic network conditions [22], [23]. To address these limitations, this research proposes DeepFocusGuard Based SkyDive Optimization, a hybrid AI-enabled model designed to ensure robust web server security. The framework preprocesses raw traffic data using feature transformation, missing value handling, and label encoding, complemented by correlation-based feature selection. It integrates CNN layers to extract spatial patterns, BiLSTM units to capture temporal dependencies, and an attention mechanism to highlight critical features distinguishing malicious traffic from normal flows. The model's hyperparameters are optimized using SkyDive Optimization, a novel hybrid algorithm combining PO and AO, enabling faster convergence, enhanced stability, and better generalization across diverse DDoS scenarios, thereby improving accuracy, scalability, and resilience in dynamic network environments.

3. Proposed Methodology

The main objective of this research is to ensure accurate detection and effective mitigation of DDoS attacks in web server environments, thereby maintaining high server availability and security. To achieve this, the research proposes an AI-enabled DDoS prediction and prevention scheme DeepFocusGuard that integrates a fusion of

CNN, Bi-LSTM, and Attention mechanisms, optimized by SkyDive Optimization, a fusion of PO, AO. First, the dataset is collected from the DDoS SDN dataset, which contains traffic records with features representing different network flow characteristics along with labels indicating normal or DDoS traffic. With the gathered dataset, preprocessing steps are applied to ensure data quality and suitability for AI-based learning. These steps include Feature Transformation, Missing Value Handling, Label Encoding. For feature selection, correlation filtering is employed to remove highly correlated features which provide redundant information. This step reduces dimensionality, improves training efficiency, and results in a refined dataset. To address class imbalance, the ADASYN technique is applied. Since traditional defenses such as firewall rules, IDS/IPS, and rate limiting, while partially effective, still allowed a significant portion of malicious traffic to remain undetected, we advance to an AI-driven classification approach to achieve higher detection accuracy and robust prevention. The balanced dataset is then fed into the proposed DeepFocusGuard, where CNN extracts spatial traffic patterns from network flows. BiLSTM captures temporal dependencies, learning how traffic evolves over time. Attention Mechanism highlights the most critical features in the data to improve classification accuracy. To further enhance the model's accuracy and generalization, SkyDive Optimization is employed to fine-tune hyperparameters. After that, the system enters the prevention phase to actively mitigate attacks. The prevention module applies multiple strategies that are Blocking, Rate Limiting, Redirection, and Adaptive Policies. This two-stage prediction and prevention pipeline ensures both proactive detection and real-time attack mitigation. Finally, the model's performance is evaluated using standard metrics such as accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, attack mitigation rate, false blocking rate and server availability. These results provide robust DDoS detection and prevention capabilities, outperforming traditional defense methods. The proposed Methodology diagram for AI-enabled DDoS prediction and prevention framework is shown in Figure 1

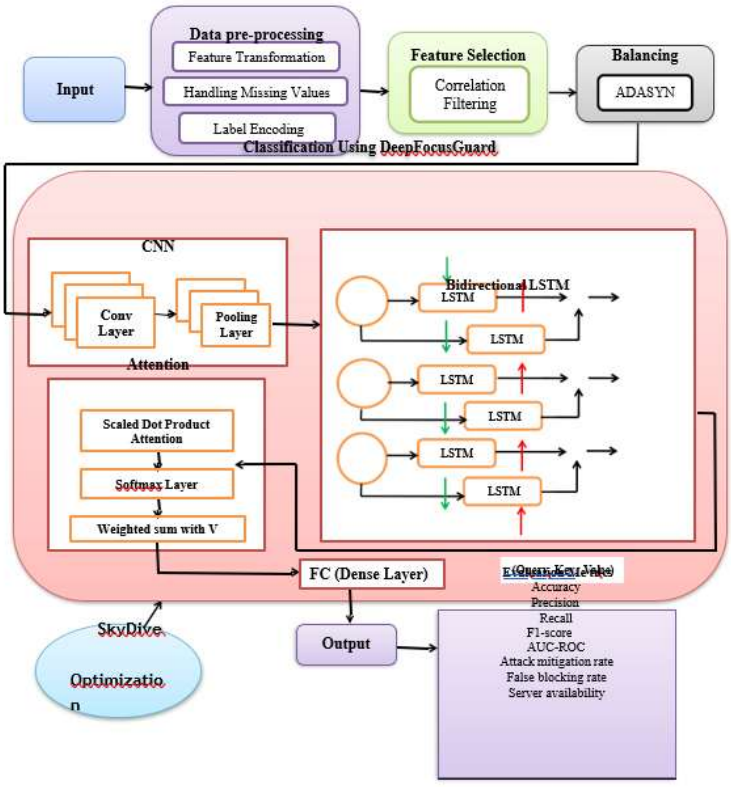


Figure 1: Proposed methodology Flow Diagram for DeepFocusGuard Based SkyDive Optimization

3.1 Dataset Collection

Initially data is collected from DDoS SDN dataset [26] was employed, consisting of approximately 104,345 traffic flows with 23 features representing flow characteristics such as duration, packet count, byte count, and packet rate. Each record is labeled as either normal traffic (0) or attack traffic (1), forming the basis for supervised classification. The dataset provides a comprehensive mix of benign and malicious flows, making it suitable for training and evaluating AI-based DDoS detection models. Formally, the dataset can be represented as

$$D = \{(x_i, y_i) x_i \in S^{23}, y_i \in \{0,1\}, i = 1,2,..,N\} \quad (1)$$

Where x_i is the input sample (network flow), $y_i \in \{0,1\}$ is the class label (0 = normal traffic, 1 = DDoS attack), N=104,345 is the total number of collected traffic samples, Here S denotes the feature space, n=23 is the total number of extracted features in the raw dataset.

3.2 Data Preprocessing

Data preprocessing is a critical step in transforming raw network flow data into a structured and high-quality format suitable for accurate DDoS prediction. The collected DDoS SDN dataset initially contained inconsistencies such as missing values, redundant features, and categorical labels, which could negatively influence the training and performance of the DeepFocusGuard model. To overcome these challenges, a systematic preprocessing pipeline was applied. First, feature transformation was performed to convert all input attributes into a uniform numerical format compatible with the learning model. Next, missing value handling was carried out to remove or impute incomplete records, ensuring the dataset's completeness and reliability. Label encoding was then applied to traffic classes, converting categorical labels into binary numerical values (0 for normal traffic and 1 for attack traffic). These steps collectively ensure that the dataset is clean, consistent, and fully compatible with the DeepFocusGuard framework. By addressing these irregularities at an early stage, the model is better positioned to capture meaningful traffic behaviors and distinguish between normal and malicious flows with higher precision.

3.2.1 Feature Transformation

Feature transformation is an essential preprocessing step to ensure that all raw network traffic attributes are converted into a suitable numerical format for deep learning. Since the collected dataset may contain mixed data types such as integers, floating-point values, or categorical strings, these must be standardized into a consistent representation. Formally, the transformation can be defined as a mapping function

$$T = S^n_{raw} \rightarrow S^n_{num} \quad (2)$$

Where, T is feature transformation function, S^n_{raw} denotes the raw feature space containing heterogeneous data types, S^n_{num} represents the transformed feature space with uniform numerical values, n is the total number of features in each network flow.

3.2.2 Missing Value Handling

Missing value handling is an essential step to ensure data completeness and consistency. The

raw dataset may contain incomplete traffic flows due to packet loss, logging errors, or corrupted entries. Such missing values, if left untreated, can distort feature distributions and degrade model performance. Therefore, records with incomplete information are either removed or imputed, resulting in a refined dataset with only valid samples.Formally, the operation can be defined as

$$H = S^{n}_{num} \rightarrow S^{n}_{clean} \quad (3)$$

Where, H is missing value handling function, S^{n}_{num} denotes numerical feature space (before cleaning), S^{n}_{clean} cleaned feature space (after removing/imputing missing values), n denotes 23 features in the original dataset.

3.2.3 Label Encoding

Label encoding is applied to transform categorical traffic labels into numerical values, ensuring compatibility with deep learning models. In this research, normal traffic is encoded as 0 and attack traffic as 1. This standardization enables the model to interpret traffic classes as binary outputs. Formally, the encoding can be defined as

$$E = S_{labels} \rightarrow \{0,1\} \quad (4)$$

Where, E denotes label encoding function, S_{labels} denotes original categorical label space (e.g., {Normal, Attack}), {0,1} represents numerical representation of the traffic labels (0 = normal, 1 = attack).

3.3 Feature Selection Using Correlation filtering

The collected dataset initially contained 23 traffic features describing different flow characteristics such as packet rate, flow duration, and bytes per flow. However, not all of these features provide unique information for the classification task. Some are highly correlated, meaning they carry similar information, which can lead to redundancy, higher computation costs, and risk of model overfitting. To address this, correlation filtering was applied to identify and remove redundant features. The Pearson correlation coefficient is used to measure the relationship between two features, expressed as

$$\rho(f_j, f_k) = \frac{Cov(f_j, f_k)}{\sigma(f_j)\sigma(f_k)} \quad (5)$$

Here, f_j and f_k represent any two distinct traffic features from the dataset, The correlation coefficient $\rho(f_j, f_k)$ represents correlation between features, $Cov(f_j, f_k)$ denotes covariance between the features, $\sigma(f_j)$, $\sigma(f_k)$ denotes standard deviations of the respective features, If the absolute correlation $\rho(f_j, f_k)$ is greater than a defined threshold (e.g., 0.9), one of the two features is removed, since it does not provide significant new information. As a result, the feature space was reduced from 23 to 20 dimensions, leading to a more compact and efficient dataset representation. Formally, the reduced dataset can be expressed as

$$D^{'} = \{(x_i^{'}, y_i) \mid x_i^{'} \in S^{20}, y_i \in \{0,1\}, i = 1,2,..., N^{'}\} \quad (6)$$

.

This ensures that the input to the DeepFocusGuard model contains only informative and non-redundant features, improving both efficiency and predictive performance.

3.4. Balancing ADASYN

To effectively address the issue of class imbalance in the collected network traffic dataset, ADASYN was applied. ADASYN generates synthetic samples for the minority class (DDoS attack) based on the density distribution of its neighbors in the feature space, thereby reducing classifier bias towards the majority class (normal traffic). For each minority class sample $x_i^{'}$, a synthetic sample x_{new} is generated as,

$$x_{new} = x_i^{'} + \delta.(x_i^{'} - x_j^{'}) \quad (7)$$

Where, $x_j^{'}$ represents a neighboring minority class sample that is similar to $x_i^{'}$, $\delta \in [0,1]$ is the random scaling factor controlling the position of the synthetic sample along the line segment between $x_i^{'}$ and $x_j^{'}$. This adaptive generation focuses more on minority samples that are harder to learn, improving class balance and ensuring the trained classifier can effectively discriminate normal traffic from DDoS attacks.

3.5 Baseline Traditional Defense

In order to provide a comparative reference for advanced AI-based detection techniques, traditional network defense mechanisms were applied sequentially to the collected network traffic. Initially, firewall rules, which block or restrict traffic from suspicious IP addresses, effectively filtering out potentially malicious flows at the network perimeter. Following this, Intrusion Detection and Prevention Systems (IDS/IPS), such as Snort or Suricata, were employed to detect known attack patterns using signature-based methods. Finally, rate limiting was applied to control the frequency of incoming traffic, preventing high-volume attacks from overwhelming the system.

Although traditional defense mechanisms can mitigate immediate threats and regulate network traffic, they are largely reactive, depending on static rules or previously known attack signatures. Moreover, when the volume of incoming requests is very high, these mechanisms struggle to maintain effectiveness and performance. Consequently, their ability to identify new or sophisticated attacks is limited, and they are generally ineffective at handling variations in network traffic. This underscores the importance of AI-enabled detection systems that can learn adaptively, identify malicious patterns, and generalize effectively across a wide range of dynamic attack scenarios.

3.6 Train Test Spliting

To evaluate the performance of the proposed detection framework, the preprocessed dataset was divided into training and testing subsets. The training set is used to train the model, allowing it to learn patterns and relationships within the data, while the testing set is reserved for evaluating the model's generalization performance on unseen samples. The dataset is split such that,

$$D_{train} \cup D_{test} = D, D_{train} \cap D_{test} = \phi \quad (8)$$

This split ensures that the model is trained on one subset and evaluated on another,

providing an unbiased assessment of its predictive capability on unseen network traffic.

3.7 Classification Using Proposed DeepFocusGuard Based SkyDive Optimization

The proposed DeepFocusGuard Model Optimized with SkyDive is an advanced hybrid deep learning framework designed for precise detection and prevention of DDoS attacks in web server environments. The framework first employs a CNN to extract spatial patterns from incoming traffic flows, followed by a Bi-LSTM network that captures temporal dependencies and sequential behaviors in the traffic data. An attention mechanism is then applied to highlight critical features, enabling the model to focus on the most relevant traffic patterns before the final classification layer predicts whether the traffic is normal or malicious. To maximize performance, the model is optimized using a hybrid SkyDive Optimization, which integrates PO and AO, two bio-inspired hyperparameter tuning methods. This combined approach effectively balances exploration and exploitation to identify the optimal network configuration. Leveraging both optimizers enhances the model's accuracy, adaptability, and robustness in detecting both known and evolving DDoS attacks. As a result, the proposed DeepFocusGuard framework provides real-time, intelligent threat detection and prevention of of DDoS attacks, making it well-suited for deployment in web server environments.

3.7.1 DeepFocusGuard architecture

The proposed DeepFocusGuard model is designed for detecting DDoS attacks in web server environments, where both local traffic patterns and their temporal behavior are critical. The model processes input network traffic features through a structured sequence of deep learning layers. An Attention layer then highlights the most relevant features for classification, and fully connected layers aggregate these features to produce the final binary output. The following section provides a detailed explanation of each layer.

i) Input Layer: The input layer of the proposed DeepFocusGuard model is responsible for receiving the preprocessed network traffic samples, which serve as the foundational data for detecting DDoS attacks. Each traffic sample is represented as a multidimensional feature vector containing F distinct features that describe various aspects of network activity, such as packet size, inter-arrival time, flow duration, protocol type, source and destination IP addresses, and other relevant statistics indicative of network behavior. Each input sample can be represented as a feature vector

$$X = [x_1, x_2, ..., x_F] \in S^F \quad (9)$$

Here, X denotes a single traffic sample, x_i represents the i-th feature, F is the total number of extracted features, and S^F defines the feature space for all samples. This structured representation allows the model to efficiently process and learn from the diverse characteristics of network traffic. The input layer serves as the foundation of the DeepFocusGuard architecture, feeding structured traffic information into subsequent CNN layers for spatial feature extraction.

ii) Convolutional Layer: The convolutional layer in the DeepFocusGuard model is responsible for extracting local spatial patterns from the input network traffic features. These patterns capture correlations and relationships between neighboring features that may indicate anomalies or attack behavior. Convolutional layers apply a set of learnable filters (kernels) across the input feature vectors to generate feature maps, which highlight

important local characteristics in the data. Mathematically, the convolution operation for a single filter can be expressed as,

$$h_j = f(\sum_{i=1}^{F} X_i * W_{i,j} + b_j) \quad (10)$$

Where h_j is the j-th feature map resulting from the convolution, X_i is the i-th input feature, $W_{i,j}$ represents the learnable weights of the convolutional filter, b_j is the bias term, and f(·) is the activation function (commonly ReLU) applied element-wise to introduce non-linearity.

iii) Max Pooling Layer: The max pooling layer is applied after the convolutional layer to reduce the dimensionality of the feature maps while preserving the most important information. This operation selects the maximum value within a defined window (pooling size), thereby retaining the strongest activation in that region. By doing so, the layer reduces computational complexity, prevents overfitting, and ensures translation invariance in the learned spatial patterns. Mathematically, max pooling can be expressed as,

$$p_j = \max\{h_j(m), h_j(m+1), ..., h_j(m+k-1)\} \quad (11)$$

Where h_j is the feature map from the convolutional layer, *k* is the pooling window size, *m* is the starting index of the window, and p_j is the pooled output.

iv) Bidirectional Layer: The Bidirectional LSTM layer processes the pooled feature representations obtained from the CNN block to capture temporal dependencies in both forward and backward directions. Unlike standard LSTMs, which only learn from past information, BiLSTMs incorporate context from both preceding and succeeding time steps, making them well-suited for modeling sequential network traffic behavior. This ability is essential in distinguishing between normal and malicious traffic patterns, where detection often depends on feature relationships spread across the entire sequence. Let the pooled feature sequence be represented as p_j. The forward LSTM processes the sequence in the standard order, while the backward LSTM processes it in reverse. The outputs are then concatenated to form the BiLSTM representation

$$h_t = \overrightarrow{LSTM}(p_t) \oplus \overleftarrow{LSTM}(p_t) \quad (12)$$

Where, p_t is the pooled feature at time step *t*, $\overrightarrow{LSTM}(p_t)$ is the forward hidden state, $\overleftarrow{LSTM}(p_t)$ is the backward hidden state, $\oplus$ denotes concatenation, h_t and is the combined BiLSTM output at time step *t*.

v) Attention Layer: The Attention mechanism is applied after the BiLSTM to selectively emphasize the most relevant temporal features. While BiLSTM generates rich contextual representations across all time steps, not every hidden state contributes equally to identifying malicious traffic. To address this, the Attention mechanism leverages the Query–Key–Value (QKV) framework to assign varying importance to each hidden state. The hidden states from the BiLSTM are first projected into three distinct spaces Queries (Q), Keys (K), and Values (V)

$$Q = HW_Q, K = HW_K, V = HW_V \quad (13)$$

Where, $H = [h_1, h_2, ..., h_T]$ is the sequence of BiLSTM hidden states, W_Q, W_K, W_V are learnable weight matrices, and T is the sequence length. The attention weights are then computed by measuring similarity between Queries and Keys, followed by a Softmax normalization

$$\alpha_t = soft\max(\frac{QK^T}{\sqrt{d_k}}) \quad (14)$$

Where d_k is the dimensionality of the key vectors, and α_t represents the normalized importance score for each time step *t*.The final context vector*c*, representing the weighted sum of the Values, is obtained as

$$c = \sum_{t=1}^{T} \alpha_t V_t \quad (15)$$

Here, T is the sequence length (number of BiLSTM outputs), c captures the most critical temporal patterns, ensuring that the downstream dense layers operate on the most informative features for accurate classification.

vi) Fully Connected Dense Layer*:* The fully connected (Dense) layer serves as the decision-making stage of the DeepFocusGuard model. The context vector *c*, obtained from the Attention mechanism, is passed into one or more dense layers where all neurons are fully connected to the previous layer's output. These layers aggregate the high-level spatial and temporal features, apply non-linear transformations, and project them into a more compact classification space. This enables the model to learn complex decision boundaries that distinguish normal traffic from malicious DDoS traffic. Mathematically, the operation of a dense layer can be expressed as,

$$z = f(W_d c + b_d) \quad (16)$$

Where *c* is the context vector from the Attention layer, W_d and b_d are the trainable weights and bias of the dense layer, f(·) is the activation function (e.g., ReLU), and *z* is the transformed feature representation output from the dense layer.

Where *c* is the context vector from the Attention layer, W_d and b_d are the trainable weights and bias of the dense layer, f(·) is the activation function (e.g., ReLU), and *z* is the transformed feature representation output from the dense layer.

vii) Output Layer: The output layer is the final stage of the DeepFocusGuard model, responsible for generating the classification result. Since the objective is to distinguish between normal and DDoS traffic, the output layer consists of a single neuron activated by a sigmoid function. This activation maps the aggregated features from the dense layer into a probability score ranging between 0 and 1, indicating the likelihood of the traffic sample belonging to the malicious class. Mathematically, the output is defined as

$$\hat{y} = \sigma(W_o z + b_o) \quad (17)$$

Where z is the feature representation from the dense layer, W_o and b_o are the trainable weight and bias parameters of the output neuron, σ(·) is the sigmoid activation function, $\hat{y}$ represents the predicted probability ($\hat{y} \in [0,1]$). During inference, a threshold is applied to $\hat{y}$ to assign the traffic sample to either the Normal or DDoS class. The architecture diagram of the proposed DeepFocusGuard model is shown in Figure 2.

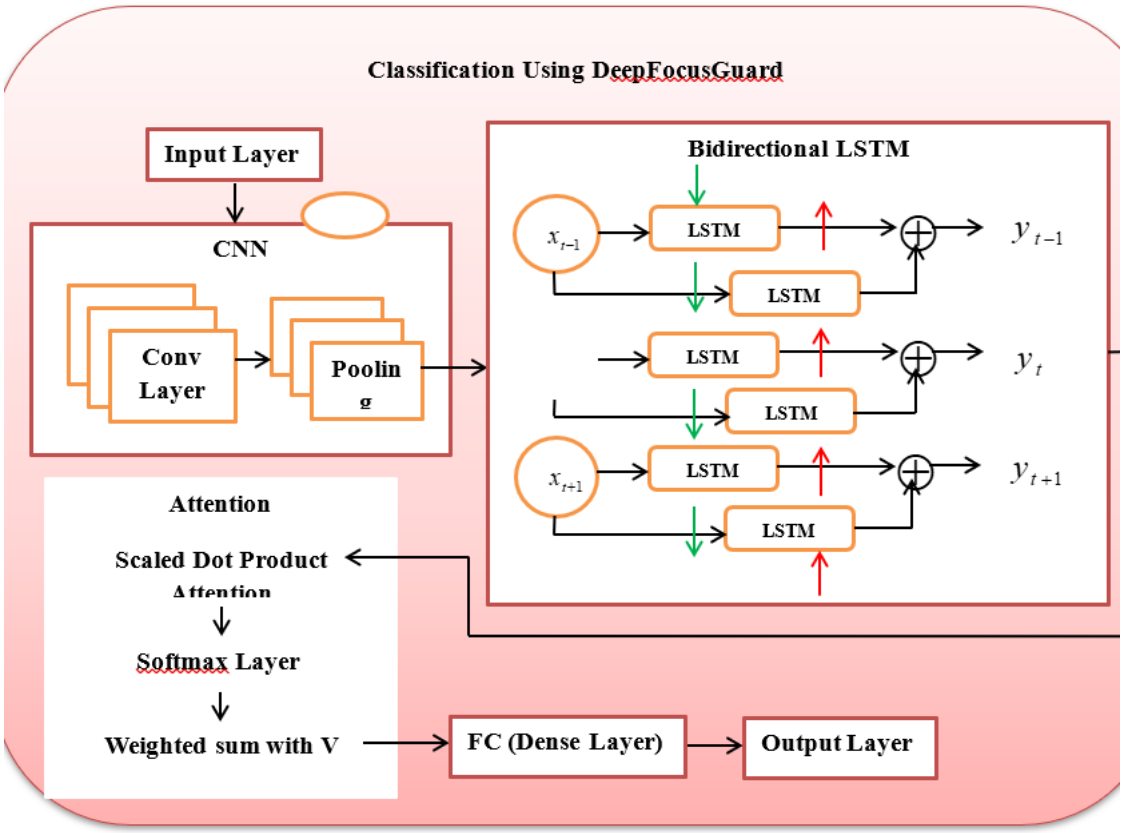


Figure 2: Architecture diagram of proposed DeepFocusGuard Model

3.7.2 SkyDive Optimization

To enhance the performance and robustness of the DeepFocusGuard model, this research employs a novel bio-inspired hyperparameter optimization algorithm called SkyDive Optimization, which combines PO and AO. Combining these two algorithms is beneficial because PO excels at global exploration by spreading candidate solutions widely across the search space, while AO focuses on local exploitation through adaptive fine-tuning in promising regions. This synergy helps avoid premature convergence and ensures a thorough search for optimal solutions. The exploration phase, driven by PO, mimics a skydiver's free-fall, allowing the algorithm to explore diverse hyperparameter settings and prevent the search from being trapped in local optima. The exploitation phase, inspired by AO's controlled glide, enables the algorithm to refine solutions using adaptive step sizes, focusing computational effort on regions with higher potential. By alternating between these phases, SkyDive Optimization balances exploration and exploitation, leading to faster convergence and more robust performance.

The workflow begins by initializing candidate solutions randomly within the hyperparameter space. In each iteration, the algorithm first applies the PO mechanism to explore new regions and diversify solutions. Next, the AO mechanism adaptively adjusts step sizes and fine-tunes the solutions toward promising areas. The best-performing solutions from both phases are then combined using a weighted approach to generate new candidate sets. This iterative process continues until convergence criteria are met, resulting in a set of optimized hyperparameters that enhance the detection capability of DeepFocusGuard, improving its ability to accurately distinguish between normal and malicious traffic in web server environments.

Motivation: Modern web servers face significant challenges in detecting and preventing DDoS attacks, which are increasingly characterized by high-volume traffic,

rapidly evolving attack strategies, and the limitations of traditional defense mechanisms in capturing both spatial and temporal behaviors of network flows. Existing detection systems often struggle with restricted generalization capability, increased false positive rates, and insufficient adaptability to highly dynamic traffic variations. Furthermore, issues such as imbalanced datasets, inefficient feature representation, and a lack of intelligent learning mechanisms hinder reliable DDoS detection. The motivation of this research is to develop an intelligent and automated deep learning framework, termed DeepFocusGuard, that effectively identifies and mitigates DDoS attacks by leveraging spatial features through CNN, sequential dependencies through BiLSTM, and critical pattern refinement through an Attention mechanism. To further enhance performance, the SkyDive Optimization algorithm is employed, which combines broad hyperparameter exploration with adaptive refinement, ensuring robust model scalability, improved detection precision, and resilient defense against real-world DDoS attacks in web server environments.

Mathematical representation of SkyDive Optimization

The SkyDive Optimization algorithm optimizes the hyperparameters of the DeepFocusGuard model for DDoS detection by finding the best combination of parameters. It balances global exploration and local exploitation by combining PO and AO. The goal is to maximize detection accuracy and minimize false positives while ensuring efficient convergence.

1. Initialization of Candidate Hyperparameter Population: The optimization process begins by generating a diverse set of candidate solutions, each representing a possible combination of hyperparameters. This initialization ensures that the search starts with broad coverage of the hyperparameter space and prevents early convergence to suboptimal solutions.

$$x_i^0 = Random(x_{\min}, x_{\max}), i = 1,2,\ldots,N \quad (18)$$

Where x_i^0 is initial hyperparameter vector for candidate i,, N is the total number of candidate solutions initialized, $x_{\min}$ and $x_{\max}$ define the allowed range for each parameter. This phase ensures that the search space is adequately populated with diverse configurations.

2. Fitness Evaluation of Candidates: After initialization, each candidate is evaluated by training the DeepFocusGuard model using its hyperparameters. The model's performance is then quantified through a fitness function, which measures how effectively it detects DDoS attacks.

$$f_i^t = \varepsilon(x_i^t) \quad (19)$$

Where f_i^t denotes the fitness score of candidate i at iteration t, ε is evaluation function mapping candidate hyperparameters to detection performance. This ensures that only performance-driven exploration is carried forward.

3. PO Update Rule in Exploration Phase: In the exploration phase, the algorithm encourages diversity by adjusting candidate solutions to explore a wider search space. The PO mechanism simulates a skydiver's free fall, where candidates are attracted toward the best solution while introducing randomness to escape local minima.

$$x_{PO}^{t+1} = x_i^t + \alpha.(x_{best}^t - x_i^t) + \beta.R \quad (20)$$

Where x_{best}^t is best candidate solution discovered up to iteration t, α controls how strongly candidates are pulled toward the best solution, β introduces randomness via R,

promoting exploration.

4. AO Update Rule in Exploitation Phase: Once potential solutions are identified, the algorithm refines them in the exploitation phase. The AO simulates a skydiver's glide toward stability, focusing on adjusting candidate solutions based on the population's mean and best-performing candidates.

$$x_{AO}^{t+1} = x_{best}^{t} + \gamma.(x_{mean}^{t} - x_{i}^{t}).F \quad (21)$$

Where γ determines how aggressively the solution is refined, x_{best}^{t} denotes best candidate at iteration *t*, x_{mean}^{t} mean position of all candidates at iteration *t*. F is the fine-tuning factor based on local search dynamics.

5. SkyDive Hybrid Update: The algorithm combines the outputs from both exploration and exploitation to determine the next generation of solutions. This hybrid update balances the strengths of PO and AO by weighting their contributions, allowing adaptive navigation between broad search and focused refinement.

$$x_{i}^{t-1} = \lambda.x_{PO}^{t+1} + (1-\lambda).x_{AO}^{t+1} \quad (22)$$

Where, λ is the balancing factor that controls the contribution from each phase.

6. Selection and Survival of Best Candidates: After updating the candidate solutions, their fitness is re-evaluated. Only the best-performing solutions are selected to survive into the next iteration, mimicking natural selection where stronger candidates persist while weaker ones are discarded.

$$x_{best}^{t-1} = \arg\max_{x} f(x) \quad (22)$$

Where, x_{best}^{t+1} is the candidate with the highest fitness score at iteration t+1. This phase employs elitism, diversity preservation, and competitive filtering to retain only the strongest solutions while maintaining adaptability.

7. Stopping Criterion and Convergence: The optimization loop continues until certain criteria are met, ensuring the process halts when either the best solution reaches a satisfactory level or computational resources are exhausted. This final step ensures efficiency and prevents unnecessary computations. The stopping condition is mathematically expressed as

$$\text{Stop if } (t \geq t_{\max}) \text{ or } f(x_{best}) \geq f_{target} \quad (23)$$

Where $t_{\max}$ maximum number of is iterations allowed, f_{target} is the desired threshold for performance. The optimization halts when maximum iterations, target fitness, or convergence stability is reached, ensuring efficiency and robustness.

Table 1: Pseudocode for SkyDive Optimization

Input: Hyperparameter search space with bounds $x_{\min}$, $x_{\max}$, Population size N, Maximum iterations $t_{\max}$, Target fitness f_{target}, Coefficients α (exploration), β (randomness), γ (exploitation), F (fine-tuning), Balancing factor λ (between exploration and exploitation)	
Output: Optimal hyperparameter set x_{best}	
1.	Initialization
	For i = 1 to N do

	Initialize candidate solution x_i randomly within bounds [x_{min}, x_{max}]
	End For
	Evaluate fitness f_i for each x_i
	Set x_{best} = best candidate from initial population
	Compute x_{mean} = mean of all candidates
2.	For t = 1 to t_{max} do
	For each candidate i = 1 to N do
	//--- Exploration Phase using PO ---
	Generate random vector R
	$x_{PO} = x_i + \alpha * (x_{best} - x_i) + \beta * R$
	// --- Exploitation Phase using AO ---
	$x_{AO} = x_{best} + \gamma * (x_{mean} - x_i) * F$
	// --- Hybrid Update ---
	$x_{new} = \lambda * x_{PO} + (1 - \lambda) * x_{AO}$
	// Update candidate
	$x_i = x_{new}$
	End For
	// Evaluate fitness of updated candidates
	For i = 1 to N do
	$f_i = EvaluateFitness(x_i)$
	End For
	// Update best candidate and mean
	x_{best} = candidate with highest f_i
	x_{mean} = average of all x_i
	// Check stopping criteria
	If $f(x_{best}) \geq f_{target}$ then
	Break
	End If
	End For
3.	Return the best solution

	Output x_{best}
	End

4. Results and Discussion

This section presents the evaluation outcomes of the Proposed DeepFocusGuard Based SkyDive Optimization for DDoS prediction and prevention in web server environments. The performance is assessed using the DDoS dataset, employing standard classification metrics such as accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, Attack Mitigation, False Blocking, and Server Availability. It demonstrates remarkable improvements in detection accuracy, adaptability, and robustness, attributed to the synergistic integration of CNN, Bi-LSTM, and Attention mechanisms within its hybrid deep learning architecture. The incorporation of SkyDive Optimization significantly enhances the model's ability to identify optimal parameter settings, thereby preventing overfitting and ensuring stable convergence. Furthermore, comparative evaluation against baseline and existing approaches highlights that the proposed model achieves higher classification accuracy, improved detection of minority attack classes, and greater consistency across multiple experimental runs and making it suitable for real-time DDoS prediction and prevention in dynamic web server environments.

4.1 Dataset Description

The experiment utilizes the DDoS SDN dataset, which simulates real-world web server traffic encompassing both benign requests and diverse DDoS attack patterns. The dataset contains over 104345 traffic records with 23 multiple flow-level and packet-level features, thereby supporting supervised classification. Each record represents a detailed snapshot of network activity, allowing the model to learn discriminative patterns that separate legitimate traffic from DDoS attack behavior. To ensure consistency and reliability, the dataset undergoes essential preprocessing steps, including feature transformation, handling of missing values, and label encoding which are used to improve data quality, ensure feature consistency, and enhance the learning efficiency.

4.2 Feature Selection Using Correlation filtering

Feature selection is applied using correlation filtering to eliminate redundant attributes that provide overlapping information. The correlation heatmap highlights strong dependencies among certain features, such as tot_dur, byteperflow, and pktrate, which are removed to reduce redundancy. This step reduces the feature space from 23 to 20 features, improving computational efficiency and enhancing the model's ability to learn distinct traffic patterns for accurate DDoS detection.

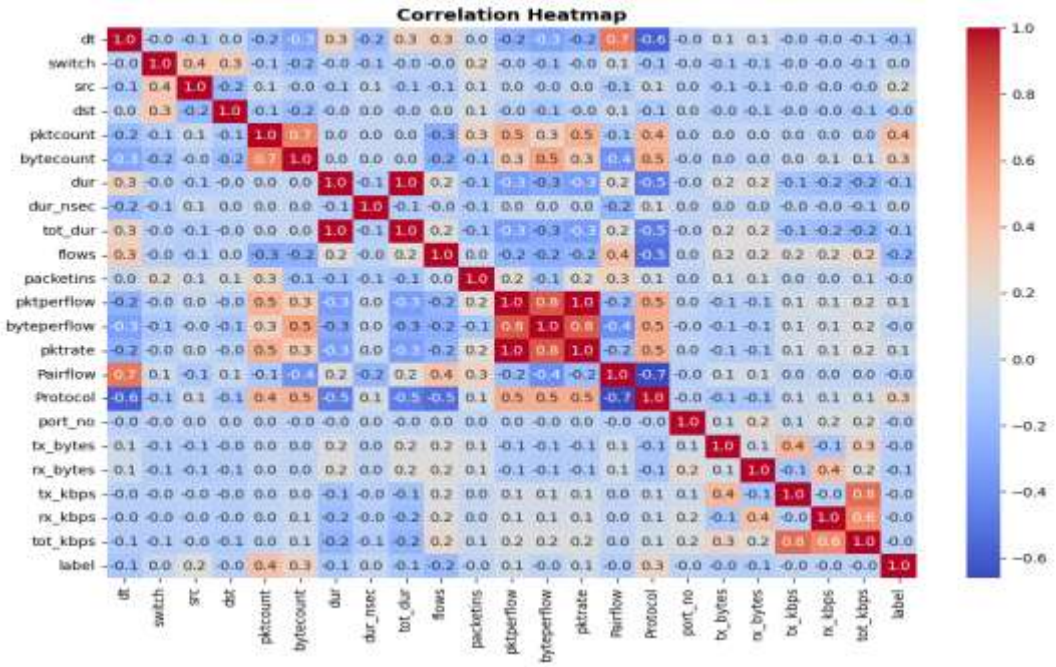


Figure 3: Correlation Heatmap of Network Traffic Features

Figure 3 illustrates the correlation heatmap of extracted traffic features from the dataset, where the color intensity represents the degree of linear relationship between feature pairs. Strong positive correlations are shown in red, while negative correlations are depicted in blue. To reduce multicollinearity and improve model efficiency, correlation filtering is applied, retaining only representative features with minimal redundancy. This ensures that the Proposed DeepFocusGuard model is trained on a compact yet informative feature set, enhancing learning performance and preventing overfitting.

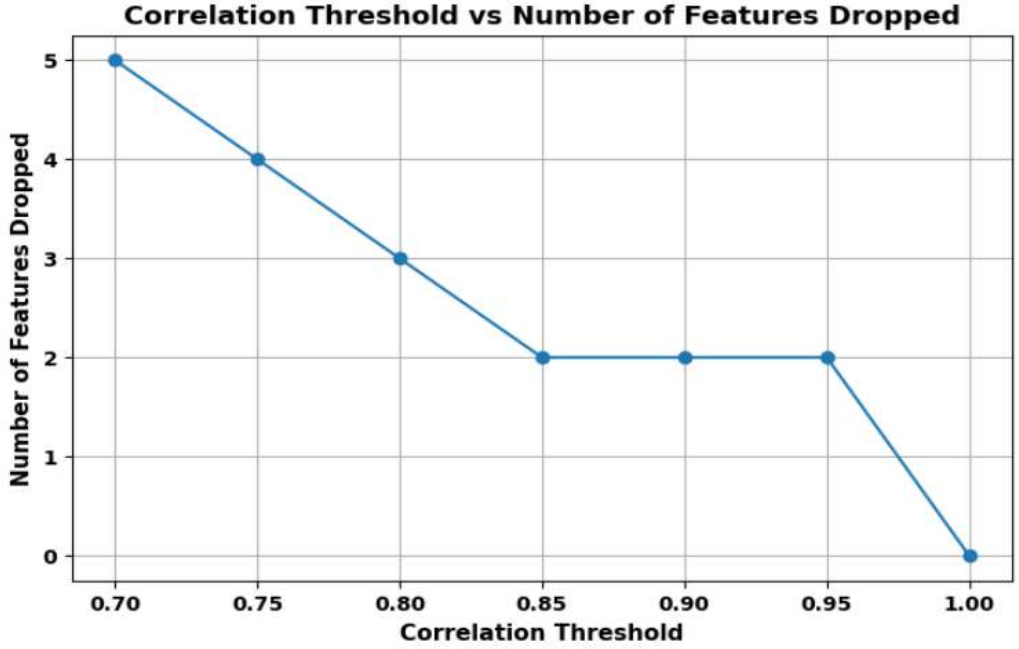


Figure 4: Relationship between Correlation Threshold and Number of Features Removed

Figure 4 presents the relationship between varying correlation thresholds and the number of features removed during correlation filtering. At lower thresholds (e.g., 0.70), more redundant features are discarded, reflecting stronger control over multicollinearity. As the threshold increases, fewer features are eliminated, retaining a larger set of correlated variables. At the extreme threshold of 1.0, no features are removed since only perfectly correlated features would be considered redundant. This analysis highlights the trade-off between feature dimensionality reduction and information retention, enabling the selection of an optimal threshold that balances redundancy removal with the preservation of discriminative traffic features for the Proposed DeepFocusGuard model Based SkyDive Optimization.

4.3 Balancing Using ADASYN

In AI-enabled DDoS detection for ISP networks, imbalanced datasets often lead to biased models where the majority class dominates the learning process. To overcome this challenge, resampling techniques such as ADASYN are used to generate additional synthetic samples for the minority class, thereby balancing the dataset and improving classification performance.

Before applying ADASYN, the dataset contains an uneven distribution of samples, with the normal web traffic class dominating the DDoS attack traffic class. Such imbalance leads to skewed learning, where the classifier tends to prioritize normal web traffic patterns while ignoring the DDoS attack. After applying ADASYN, the dataset becomes more balanced, as synthetic samples are intelligently generated for the minority class. This ensures that both classes contribute more equally to the training process, improving the model's sensitivity and reducing bias toward the majority class.

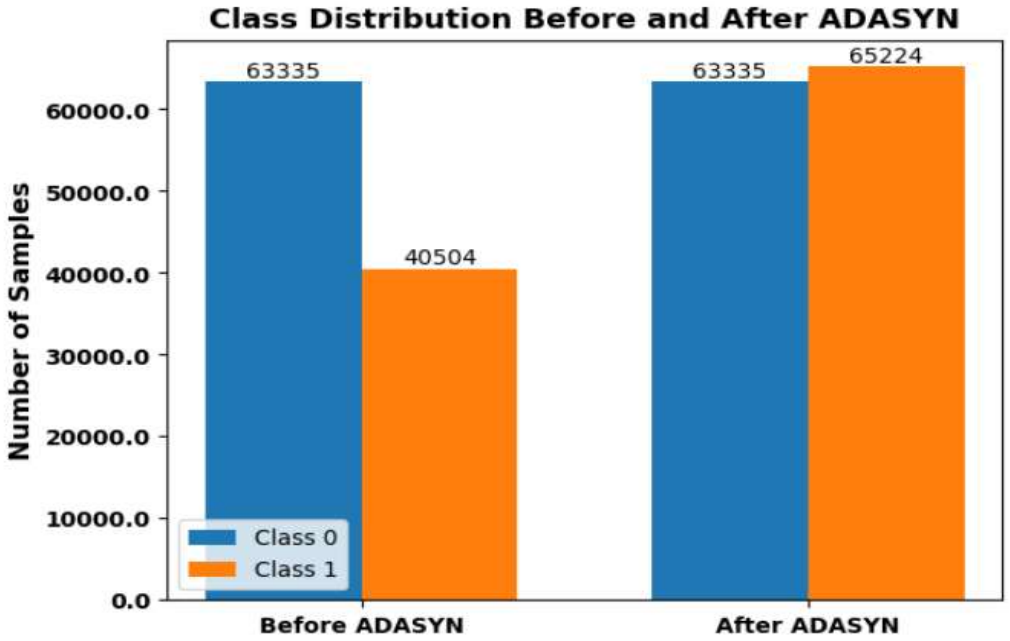


Figure 5: Class Distribution Before and After Applying ADASYN

Figure 5 clearly illustrates the impact of ADASYN on class distribution. The left bars represent the original imbalanced dataset where normal traffic (Class 0) dominates, while the right bars show the balanced dataset after ADASYN, where anomalous traffic (Class 1) is expanded to achieve near-equal representation. This balancing process ensures fairer training conditions and enhances the accurate DDoS attack capability.

4.4 Baseline Traditional Defense

In traditional web server security, multiple defense mechanisms such as firewalls,

intrusion detection/prevention systems (IDS/IPS), and rate-limiting are employed to mitigate malicious traffic. While these mechanisms reduce the malicious traffic to some extent, their effectiveness diminishes significantly under large-scale and evolving DDoS attack scenarios, highlighting the need for adaptive and AI-driven defense strategies.

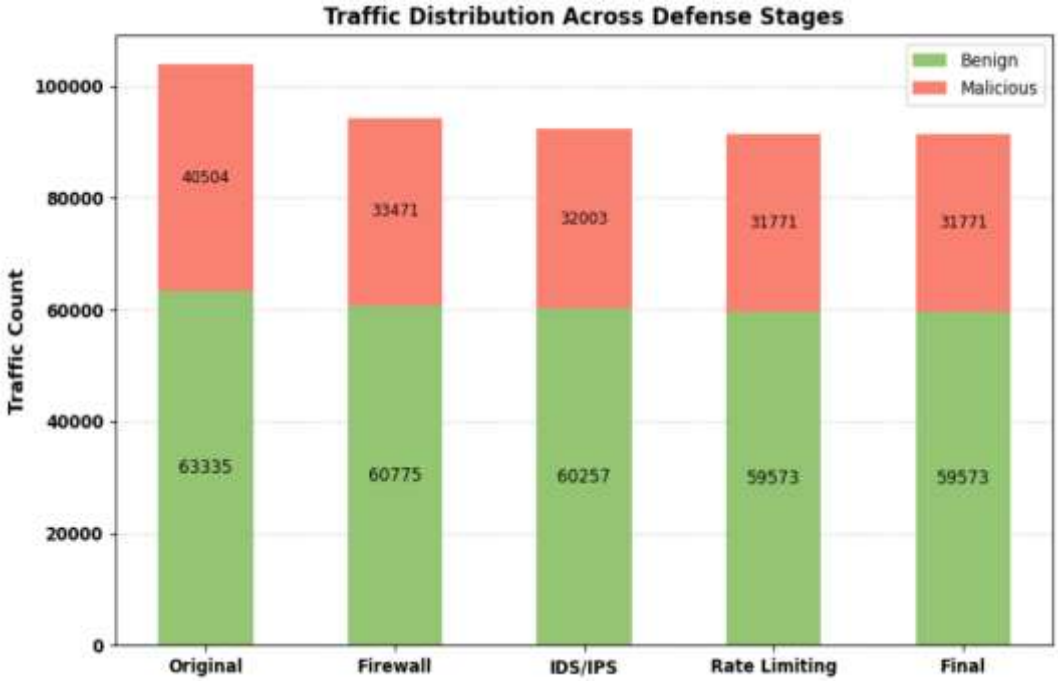


Figure 6: Visualization of Traffic Distribution Across Different Defense Stages, Including Firewall, IDS/IPS, and Rate Limiting

Figure 6 illustrates how traffic is distributed across various defense stages. Initially, a considerable portion of malicious traffic is present alongside benign flows. As the traffic progresses through defenses such as firewalls, IDS/IPS, and rate limiting, the malicious component is reduced; however, a substantial fraction still persists even in the final stage. This indicates that while traditional methods are able to filter attacks partially, they fail to provide complete protection in high-volume, dynamic ISP environments. This limitation motivates the need for the proposed DeepFocusGuard framework, which leverages deep learning and optimization strategies to deliver more accurate, scalable, and adaptive DDoS detection and prevention for web servers.

4.5 Model Accuracy and Loss Analysis

To assess the effectiveness of the Proposed DeepFocusGuard Framework Optimized with SkyDive Optimization, both training and test accuracy and loss were monitored across multiple epochs. These performance indicators provide key insights into how well the model learns from web server traffic data and its ability to generalize to unseen attack scenarios.

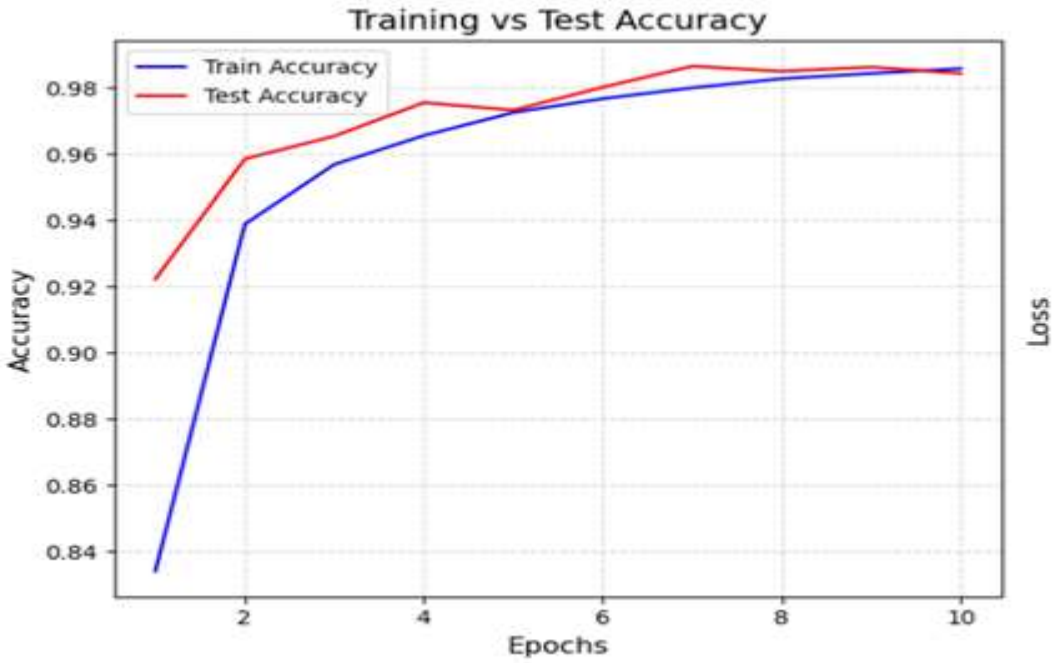


Figure 7: Training and Test accuracy progression over epochs.

Figure 7 illustrates the accuracy trends across training and test epochs. The blue curve represents the training accuracy, while the red curve indicates the test accuracy. Both curves exhibit a consistent upward trend, converging near 98%, which confirms the model's ability to learn effectively from training data while maintaining strong generalization on unseen test data. The rapid rise in accuracy during the initial epochs highlights efficient learning, while the stabilization of both curves in later epochs demonstrates robust convergence and minimal overfitting.

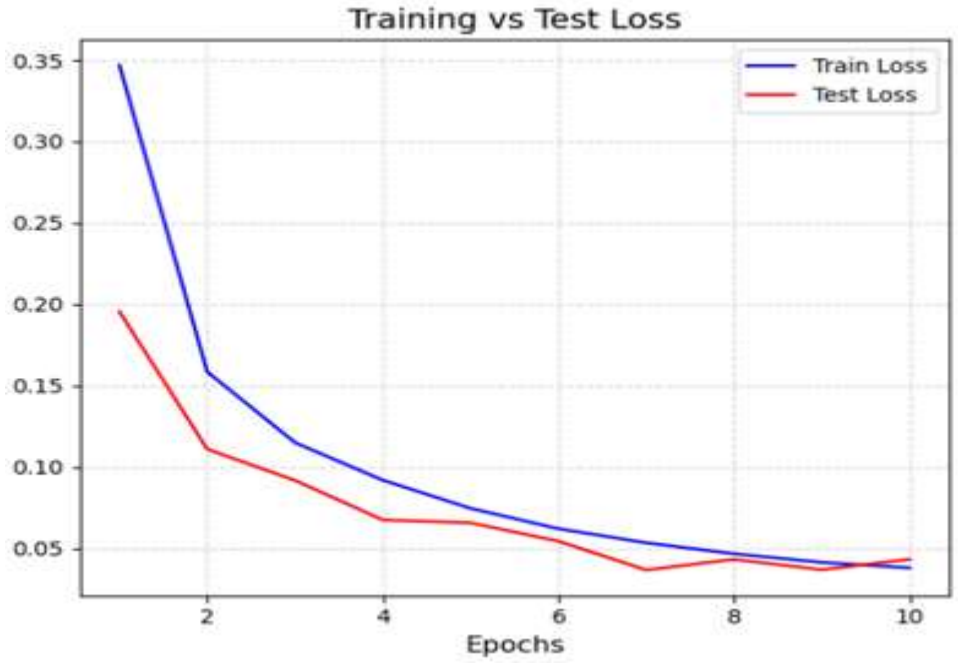


Figure 8: Training and Test loss progression over epochs.

Figure 8 depicts the loss trends across training epochs. The blue curve corresponds to the training loss, while the red curve represents the test loss. Both curves show a steep

decline during the initial epochs, followed by a gradual flattening below 0.05, which reflects efficient learning and strong convergence. The close alignment of training and test loss indicates that the model avoids overfitting and maintains stability across unseen data.

4.6 Optimization with SkyDive Optimization

In order to optimize the hyperparameters of the proposed DeepFocusGuard model, the SkyDive Optimization (STX) algorithm was employed. Specifically, the framework explores the PO Algorithm, inspired by cooperative hunting strategies, to enhance global exploration, and the AO, designed for adaptive fine-tuning and local exploitation. The synergistic combination of these optimization techniques ensures efficient parameter search, minimizes the risk of overfitting, and accelerates convergence toward highly accurate configurations. This makes the model both computationally efficient and highly reliable for real-time web server security tasks.

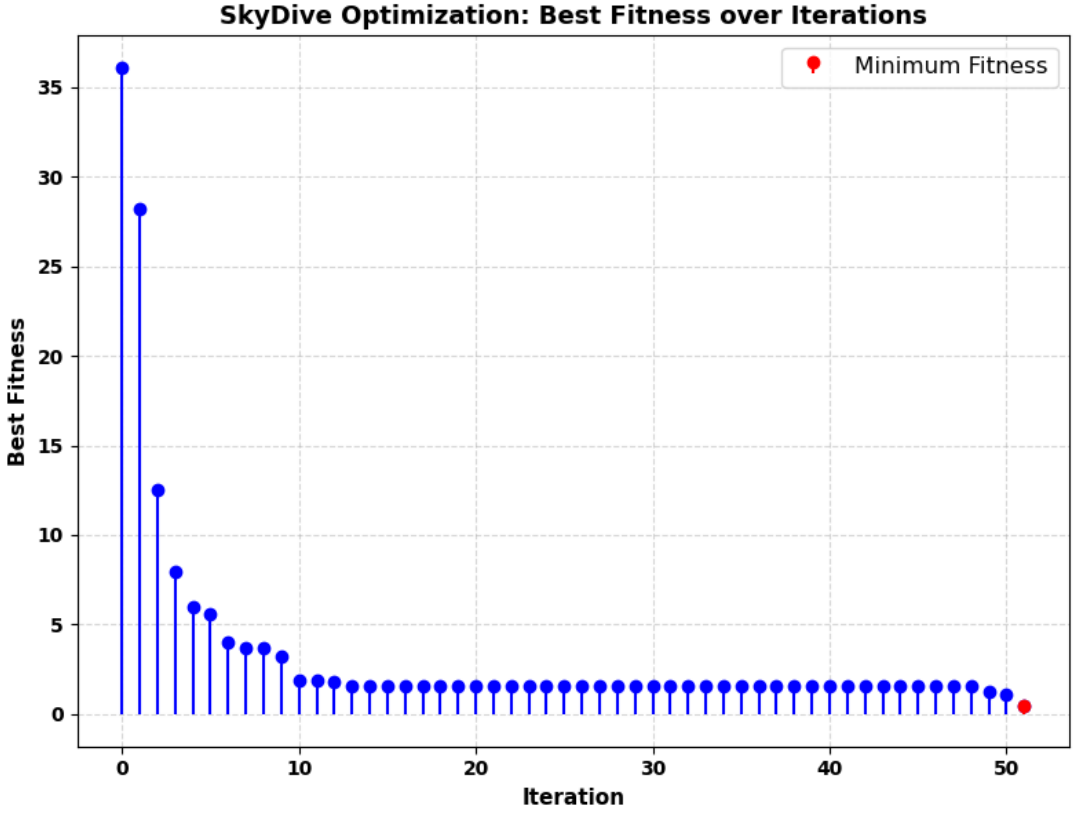


Figure 9: Convergence Curve of SkyDive Optimization Showing Best Fitness Across Iterations

Figure 9 illustrates the convergence behavior of the SkyDive Optimization algorithm over 50 iterations. The curve demonstrates a steep drop in the fitness value during the initial iterations, indicating rapid exploration of the search space. As the iterations progress, the fitness value gradually stabilizes near the global minimum, highlighting the algorithm's ability to balance exploration and exploitation effectively. The final minimum fitness (marked in red) confirms successful convergence to an optimal solution. This efficient optimization process ensures that the proposed DeepFocusGuard framework is fine-tuned for robust performance in detecting DDoS attacks.

4.6.1 Ablation Study: Component-wise Analysis of Optimization Strategies

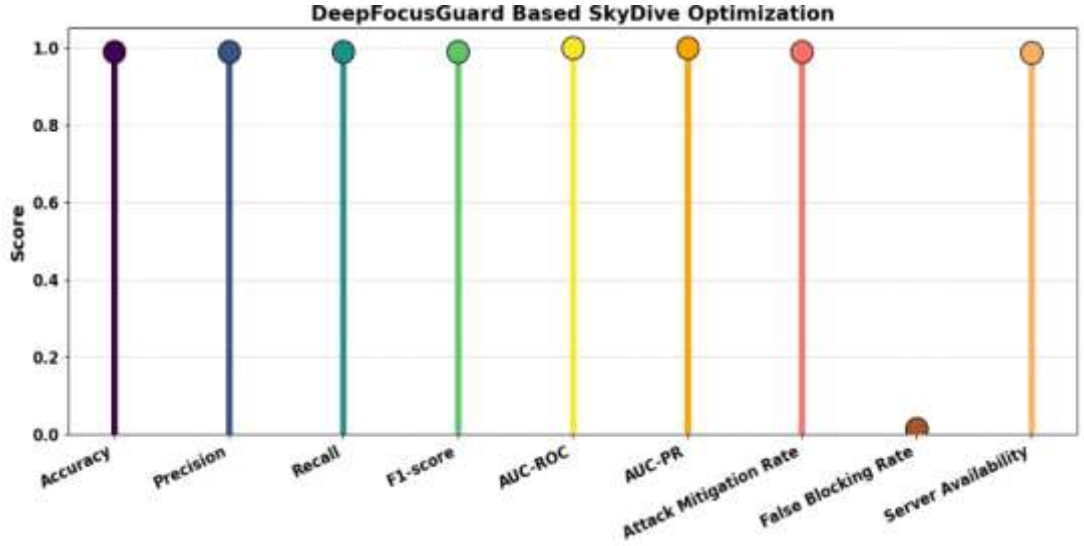
To better understand the contribution of individual components within the proposed DeepFocusGuard Based SkyDive Optimization framework, we conducted an ablation

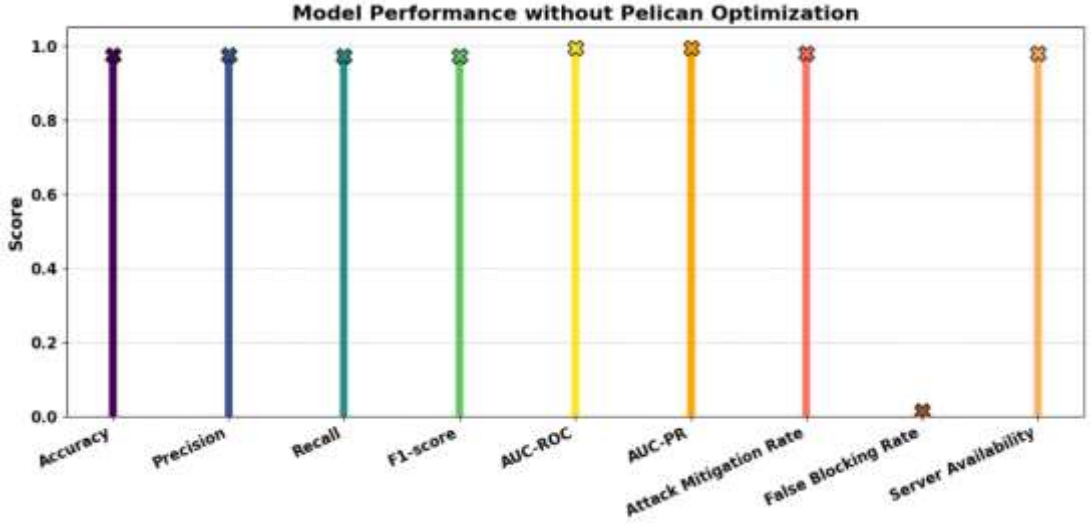
study.

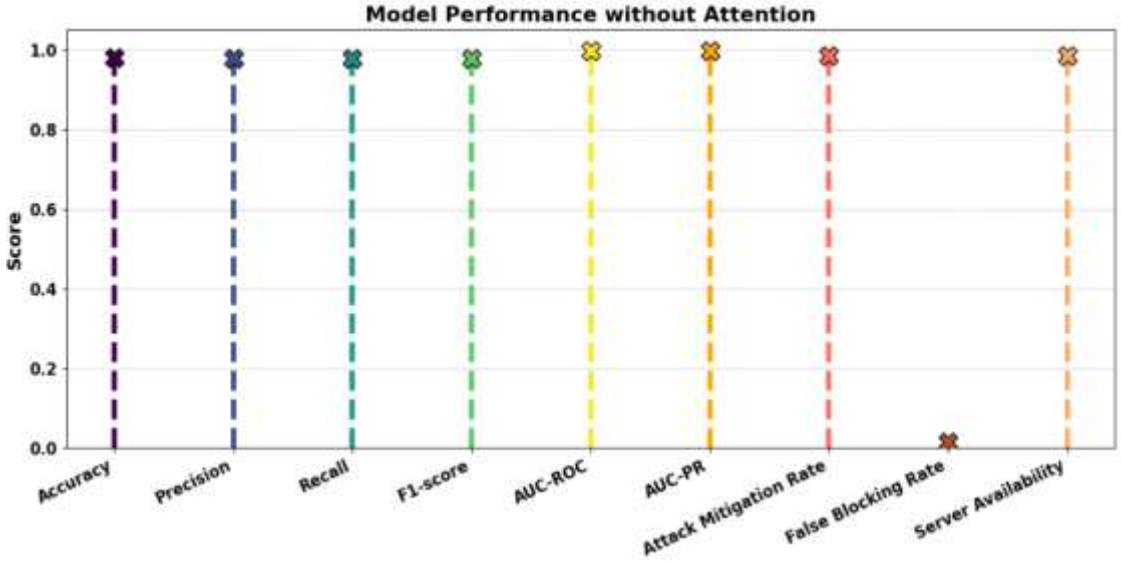
Four variants were evaluated

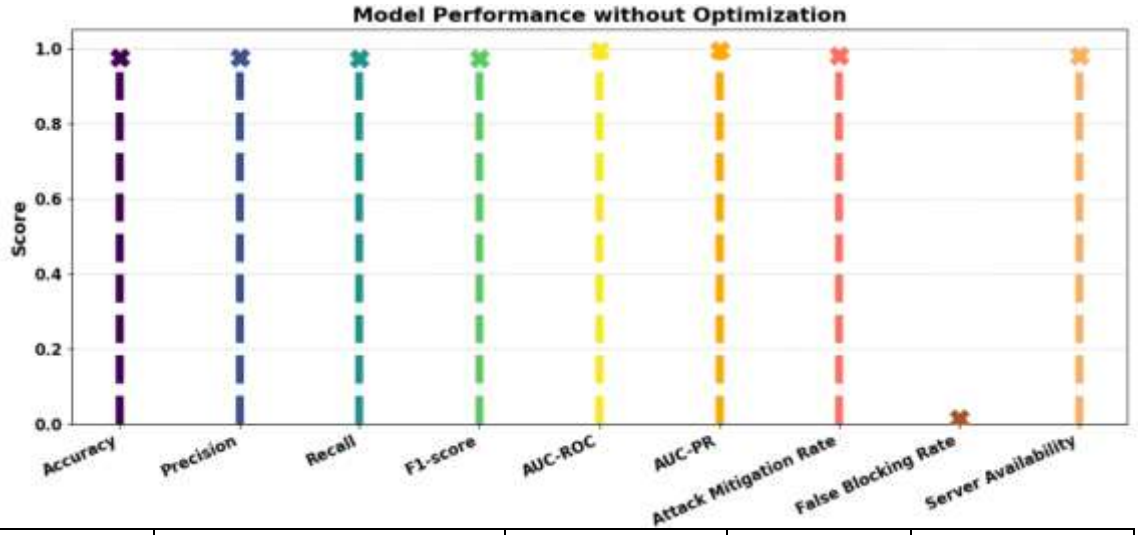
- The full DeepFocusGuard model with all optimization strategies
- The model without PO
- The model without the Attention mechanism
- The model without any optimization techniques

The graphs below present a comparative analysis of their performance across key metrics such as PDR, energy efficiency, and latency. These results highlight how each component influences the overall robustness and efficiency of the model.









Metric	Overall Performance(Optimized)	Without Pelican Optimization	Without Attention	Without Optimization
Accuracy	0.9888	0.975	0.98	0.975
Precision	0.9888	0.974	0.978	0.974
Recall	0.9888	0.973	0.977	0.973
F1-score	0.9888	0.973	0.977	0.973
AUC-ROC	0.9995	0.995	0.997	0.995
AUC-PR	0.9995	0.995	0.997	0.995
Attack Mitigation Rate	0.9905	0.98	0.985	0.98
False Blocking Rate	0.013	0.015	0.014	0.015
Server Availability	0.987	0.98	0.985	0.98

4.7 Confusion Matrix and ROC Analysis

For a comprehensive evaluation of the classification performance of the Proposed DeepFocusGuard Model Based SkyDive Optimization, a confusion matrix and ROC curve were constructed, as illustrated in Figure 10 and Figure 11, respectively.

The confusion matrix in Figure 10 provides a comprehensive view of the model's predictive capability across two traffic classes, normal and DDoS. The results indicate excellent classification performance, with 12,532 normal instances and 12,919 DDoS instances correctly identified. Misclassifications were minimal, consisting of only 103 normal flows predicted as attacks and 158 attack flows predicted as normal. This achieves a very high true positive rate and true negative rate, while keeping false positives and false negatives at a negligible level. Overall, the confusion matrix confirms the model's strong discriminative ability in separating malicious traffic from legitimate requests in web server environments.

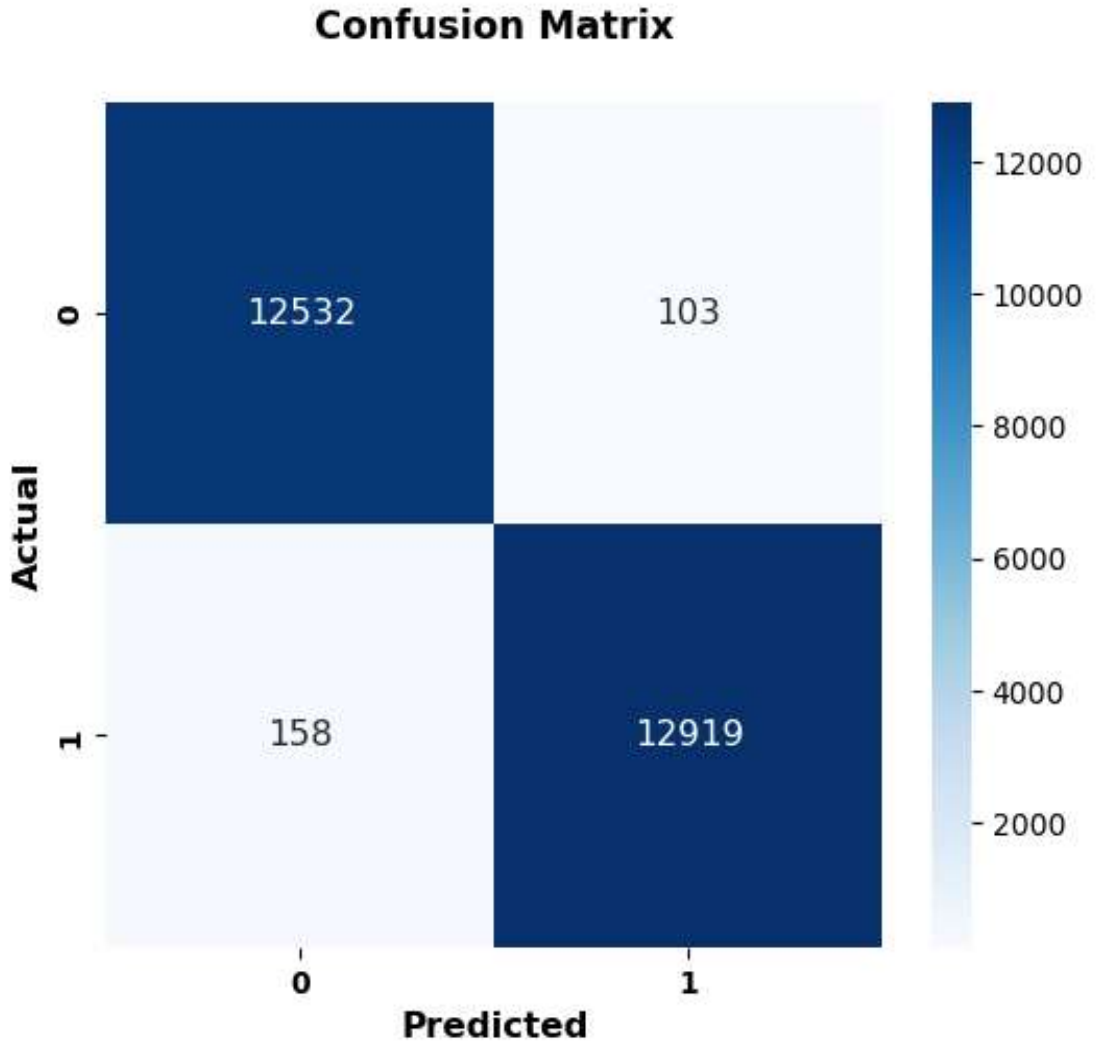


Figure 10: Confusion matrix showing true vs. predicted class labels for normal and DDoS traffic

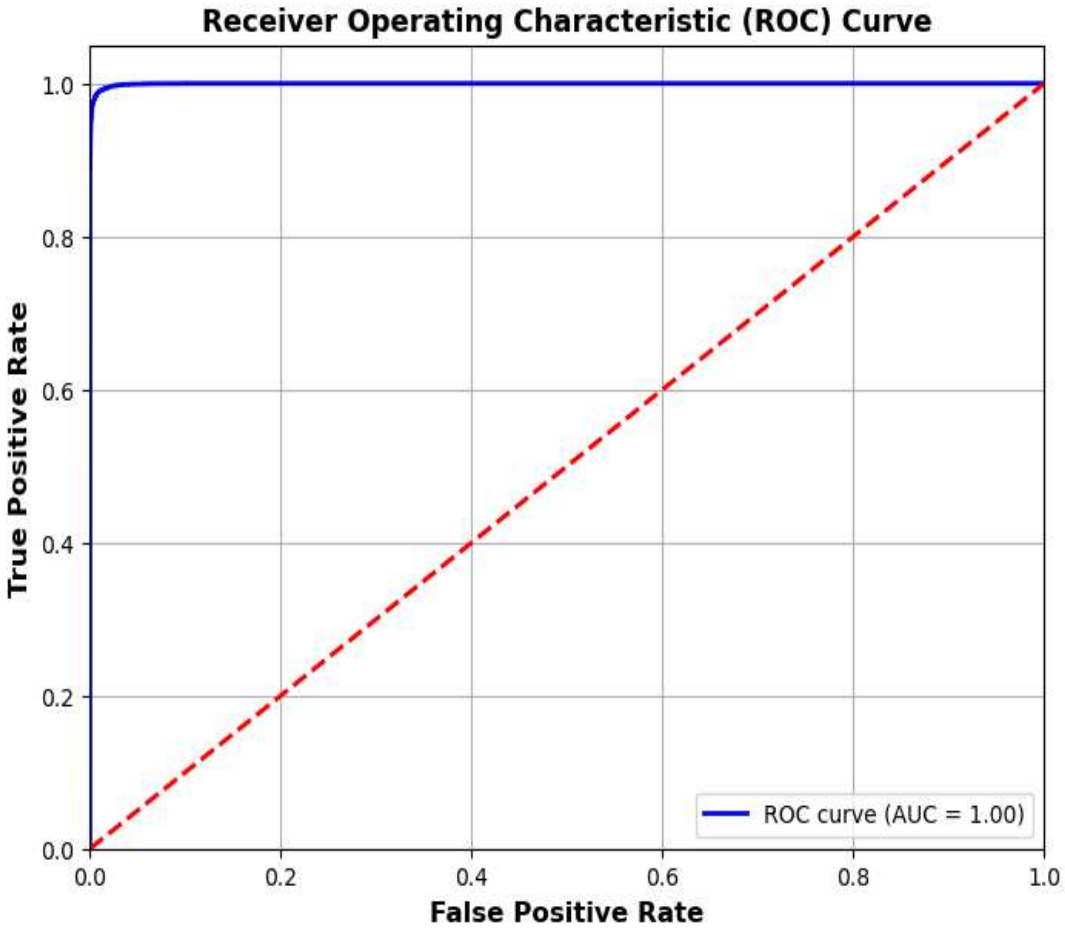


Figure 11: ROC curve with AUC = 1.00, indicating near-perfect separability between normal and DDoS traffic.

This performance is further substantiated by the ROC curve shown in Figure 11, which achieves an Area Under the Curve (AUC) score of 1.0000, reflecting optimal discrimination capability. The ROC plot maps the True Positive Rate (TPR) against the False Positive Rate (FPR), and the steep rise toward the top-left corner demonstrates that the proposed DeepFocusGuard Based SkyDive Optimization model maintains exceptionally high sensitivity. The perfect AUC score confirms the model's strong generalization across diverse traffic conditions, ensuring reliable detection of DDoS attacks with minimal misclassifications. These findings validate the robustness and reliability of the proposed framework, establishing its effectiveness for real-time web server protection and prevention against evolving DDoS strategies.

4.8 Class-wise performance evaluation

To further validate the effectiveness of the proposed DeepFocusGuard Based SkyDive Optimization, an integrated evaluation of performance and defense metrics was conducted. These metrics are accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, attack mitigation rate, false blocking rate, and server availability these are provide a holistic assessment of how well the model not only classifies malicious traffic but also sustains reliable defense in real-time web server environments.

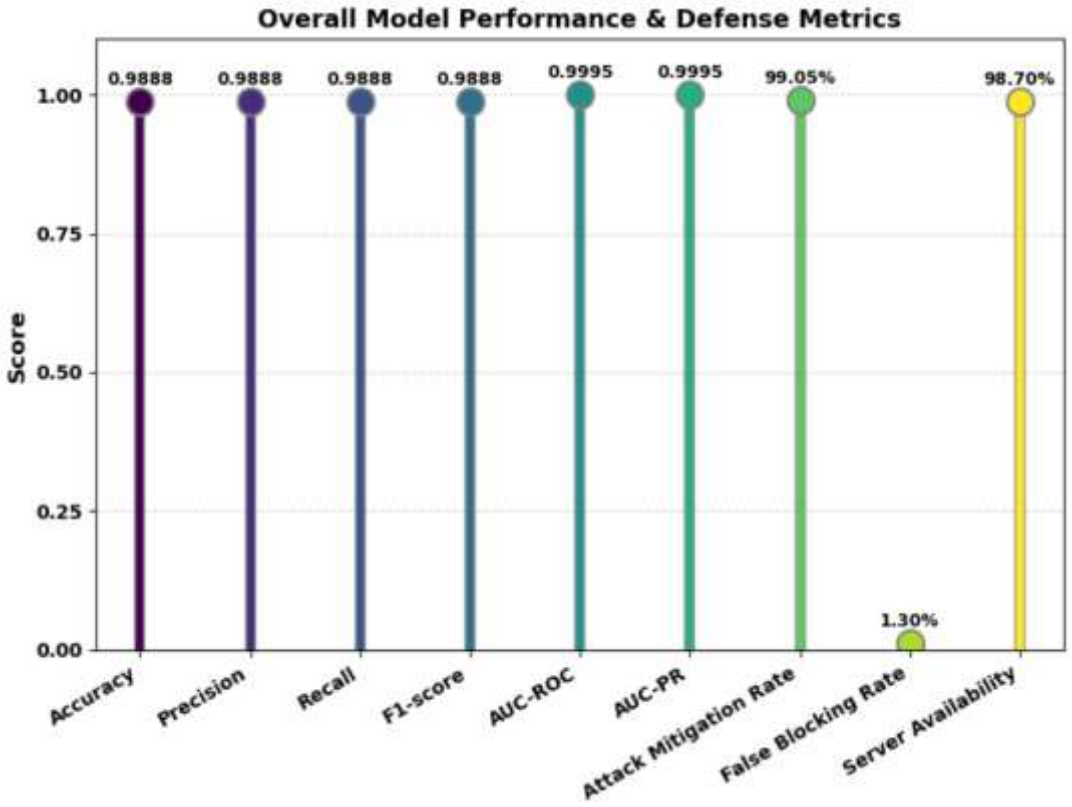


Figure 12: Class-wise Comparison of all metrics Demonstrating the Balanced Performance of the Proposed DeepFocusGuard Based SkyDive Optimization

Figure 12 presents the consolidated evaluation outcomes of the proposed system. The classification metrics demonstrate exceptional performance, with accuracy, precision, recall, and F1-score all reported at 0.9888, confirming balanced and consistent detection capabilities. Both AUC-ROC (0.9995) and AUC-PR (0.9995) indicate near-perfect separability between normal and attack traffic, further validating the robustness of the model. In the defense evaluation, the framework achieved an attack mitigation rate of 99.05%, while maintaining a very low false blocking rate of 1.30%, thereby minimizing unnecessary disruption to legitimate users. Additionally, the server availability of 98.70% highlights the system's ability to sustain continuous service operation even under adversarial conditions.

This analysis confirms that the proposed DeepFocusGuard framework not only excels in precise DDoS detection but also strengthens overall web server resilience. By combining high classification accuracy with reliable prevention metrics, the model ensures robust defense while preserving service quality, making it highly suitable for deployment in real-world, high-traffic web environments.

4.9 Comparative Evaluation

To validate the performance of the proposed DeepFocusGuard Based SkyDive Optimization (DFG-SDO) framework, a comparative evaluation was conducted against four baseline deep learning models CNN, BiLSTM, Attention, and ANN. The evaluation employed multiple performance metrics including accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, attack mitigation, false blocking, and server availability.

As illustrated in Figure 13 and Table 2, the proposed DFG-SDO consistently outperforms all baseline models across key performance metrics. It achieves superior recall and attack mitigation, demonstrating its strong capability to accurately detect

DDoS traffic while effectively blocking attacks and minimizing service disruption. In comparison, the CNN baseline shows the weakest performance, with very low recall and attack mitigation, indicating its limited ability to capture temporal and contextual traffic dependencies. BiLSTM performs better in recall but suffers from a higher false blocking rate, impacting server availability. The Attention model achieves moderate recall and attack mitigation, though still below DeepFocusGuard, while the ANN model demonstrates reasonable precision but lags in recall and attack mitigation, reducing its reliability. The improvement margin of DFG-SDO is most evident in attack mitigation and false blocking rates. While BiLSTM, Attention, and ANN achieved mitigation a rate between 59% and 89%, DFG-SDO reaches near-perfect attack mitigation with minimal false blocking, highlighting its adaptive and robust protection against DDoS attacks. Overall, these results confirm that the proposed DFG-SDO, ensures high detection accuracy, effective attack mitigation, and strong server availability, outperforming traditional deep learning models.

Table 2: Comparative Evaluation of Deep Learning Models on Various Metrics

Metric	CNN	BiLSTM	Attention	ANN	**Proposed DFG Based SDO**
Accuracy (%)	57.88	84.77	81.93	75.27	98.42
Precision (%)	92.09	81.92	83.11	87.83	98.88
Recall (%)	18.62	89.82	80.58	59.52	98.88
F1-score (%)	30.98	85.69	81.83	70.96	98.88
AUC-ROC	62.99	93.95	91.98	84.69	99.95
AUC-PR	61.50	94.14	92.30	86.61	99.95
Attack Mitigation	18.62	89.82	80.58	59.52	99.05
False Blocking	1.65	20.44	16.88	8.50	1.30
Server Availability	98.35	79.56	83.12	91.50	98.70

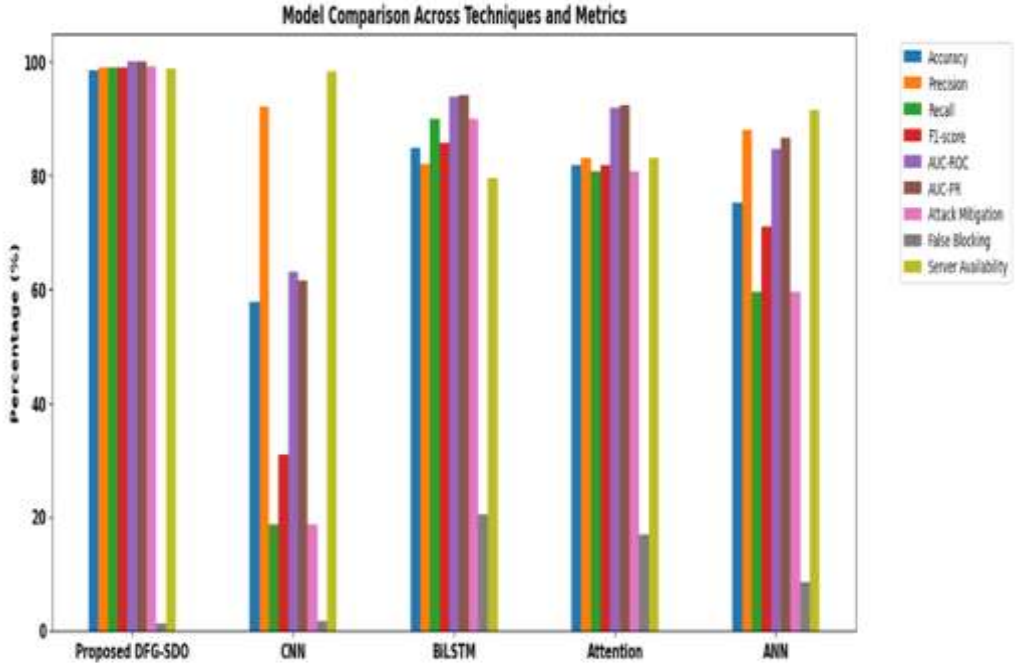


Figure 13: Model Comparison Across different Techniques and various Metrics

4.10 Prevention Phase

The prevention phase of the DeepFocusGuard Based SkyDive Optimization involves multiple strategies to mitigate malicious traffic in real time. These strategies include Blocking that temporary blacklisting of identified malicious IP addresses, Rate Limiting is for reducing the request frequency from flagged sources to limit attack impact, Traffic Redirection is used for diverting suspicious traffic to honeypot or analysis servers for further inspection, and Adaptive Policies that dynamically updating firewall and SDN rules to strengthen overall defense. The effectiveness of each prevention step is evaluated by measuring the reduction in malicious traffic counts.

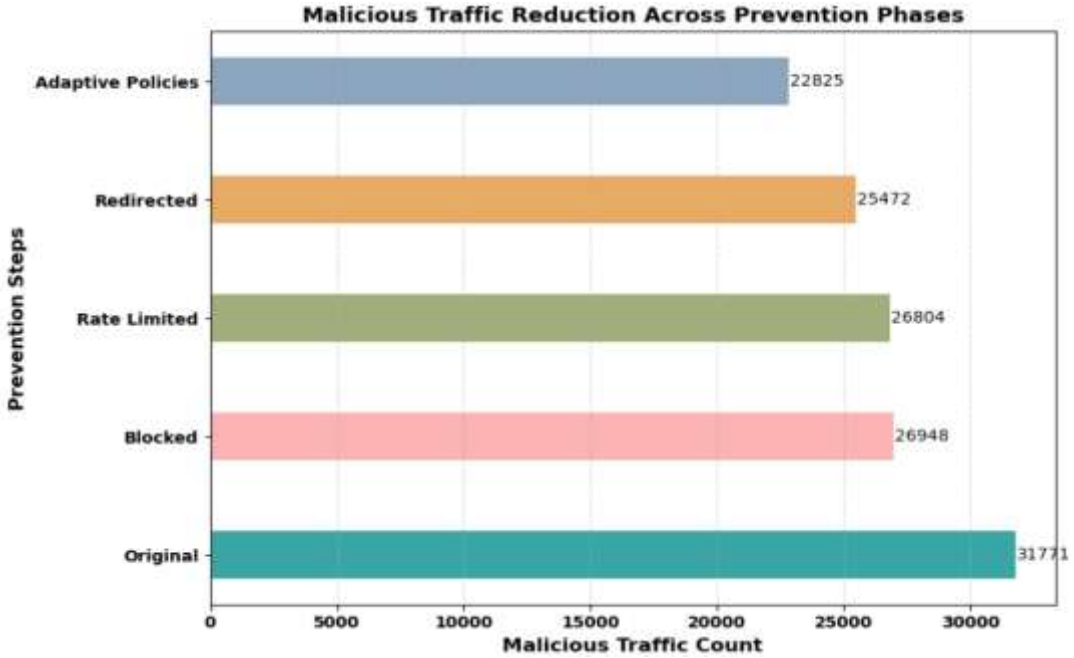


Figure 14: Malicious Traffic Reduction Across Different Prevention Steps

Figure 14 illustrates the reduction in malicious traffic counts at each stage of the prevention phase. Starting from an original count of 31,771, the Blocking step reduced it to 26,948, followed by Rate Limiting to 26,804. Subsequent Traffic Redirection lowered the count further to 25,472, and finally, Adaptive Policies achieved the most reduction, bringing malicious traffic down to 22,825. This step-wise mitigation demonstrates the cumulative effectiveness of the integrated prevention strategies in maintaining web server security.

5. Conclusion and Future Work

In this research, an advanced AI-enabled detection and prevention framework named DeepFocusGuard, Based SkyDive Optimization, and was proposed for securing web servers against DDoS attacks. The architecture integrates CNN, BiLSTM, and Attention layers in a sequential manner to capture spatial correlations in traffic features, temporal dependencies across sequences, and critical discriminative patterns that contribute most to attack identification. To ensure enhanced performance and reliable generalization, the SkyDive Optimization algorithm was employed, inspired by dynamic free-fall and gliding mechanisms to efficiently refine hyperparameters. This optimization significantly improved detection accuracy, reduced false positives, and provided robust adaptability to varying attack intensities and evolving traffic patterns. Experimental evaluation using the DDoS-SDN dataset validated the effectiveness of the proposed framework, demonstrating superior results across key metrics such as accuracy, precision, recall, F1-score AUC-ROC, AUC-PR, Attack Mitigation, False Blocking, and Server Availability when compared with traditional and existing deep learning models. By enabling early detection and rapid prevention, the proposed model enhances resilience against large-scale DDoS threats while ensuring stable web service continuity.

For future work, the framework can be extended by integrating federated learning to support privacy-preserving collaborative training across distributed web servers. In addition, adaptive online learning mechanisms can be incorporated to dynamically update the model as new attack vectors emerge, ensuring resilience against zero-day DDoS threats. Finally, large-scale evaluation across diverse real-world server environments and hybrid attack scenarios will further validate the scalability, robustness, and deployment readiness of the proposed model.

References

[1] Ahmadi, S. (2024). AI in the Detection and Prevention of Distributed Denial of Service (DDoS) Attacks. International journal of advanced computer science and applications (IJACSA), 15(10).

[2] Alemayehu, M., Ghanem, M. C., Ouazzane, K., Kheddar, H., & Lacerda, M. J. (2025). A Systematic Analysis on the Use of AI Techniques in Industrial IoT DDoS Attacks Detection, Mitigation and Prevention.

[3] Setitra, M. A., Fan, M., Agbley, B. L. Y., & Bensalem, Z. E. A. (2023). Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment. Network, 3(4), 538-562.

[4] Raza, A., Ali, A.K.S., & Hussain, A.A.(2024). AI-DRIVEN APPROACHES TO CYBER AND INFORMATION SECURITY: MACHINE LEARNING ALGORITHMS FOR THREAT PREDICTION AND ANOMALY DETECTION. Spectrum of Engineering Sciences, 2(4), 565-573.

[5] Mehmood, S., Amin, R., Mustafa, J., Hussain, M., Alsubaei, F. S., & Zakaria, M. D. (2025). Distributed denial of services (DDoS) attack detection in SDN using Optimizer-equipped CNN-MLP. PloS one, 20(1), e0312425.

[6] Zhao, J., Liu, Y., Zhang, Q., & Zheng, X. (2023). CNN-AttBiLSTM mechanism: A DDoS

attack detection method based on attention mechanism and CNN-BiLSTM. IEEE Access, 11, 136308-136317.

[7] Zhang, J., Zhang, X., Liu, Z., Fu, F., Jiao, Y., & Xu, F. (2023). A network intrusion detection model based on BiLSTM with multi-head attention mechanism. Electronics, 12(19), 4170.

[8] Singh, B. S. H. S., Fathima, M., Mahesh, T., Sameer, M., Anguraj, D. K., & Kuppan, P. (2024). A constructive model for Cyber-attack Prediction using efficient weighted bi-directional learning approaches. International Journal of Communication Networks and Information Security, 16(1), 100-116.

[9] Wang, H., Yang, X., & Jia, N. (2025). DDoS attack detection method based on improved convolutional long short-term memory and three-way decision in SDN. PLoS One, 20(5), e0322839.

[10] Sharma, H. S., & Singh, K. J. (2025). A Conv-BiLSTM Sandwich Network for DDoS Attack Detection. Procedia Computer Science, 258, 3324-3332.

[11] Xu, H., Sun, L., Fan, G., Li, W., & Kuang, G. (2023). A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism. IEEE Access, 11, 66212-66226.

[12] Kanthimathi, S., Venkatraman, S., Jayasankar, K. S., Jiljith, T. P., & Jashwanth, R. (2024). A Novel self-attention-enabled weighted ensemble-based convolutional neural network framework for distributed denial of service attack classification. IEEE Access.

[13] Dai, W., Li, X., Ji, W., & He, S. (2024). Network intrusion detection method based on CNN-BiLSTM-attention model. IEEE Access, 12, 53099-53111.

[14] Elshewey, A. M., Abbas, S., Osman, A. M., Aldakheel, E. A., & Fouad, Y. (2025). DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network. Scientific Reports, 15(1), 29122.

[15] Golduzian, A. (2023). Predict and prevent ddos attacks using machine learning and statistical algorithms. arXiv preprint arXiv:2308.15674.

[16] Girubagari, N., Ravi, T. N., & Arokiaraj, S. P. (2024). Hybrid Intelligent Anomaly Detection System Using Attention based Deep Learning Approach for Cyber Attacks Prevention. Indian Journal Of Science And Technology, 17, 38.

[17] Shou, D., Li, C., Wang, Z., Cheng, S., Hu, X., Zhang, K., ... & Wang, Y. (2024). An intrusion detection method based on attention mechanism to improve CNN-BILSTM model. The Computer Journal, 67(5), 1851-1865.

[18] Zhou, H., & Ling, J. (2023, September). A Cooperative Detection of DDoS attacks based on CNN-BiLSTM in SDN. In Journal of Physics: Conference Series (Vol. 2589, No. 1, p. 012001). IOP Publishing.

[19] Zhang, C., Li, J., Wang, N., & Zhang, D. (2025). Research on Intrusion Detection Method Based on Transformer and CNN-BiLSTM in Internet of Things. Sensors, 25(9), 2725.

[20] Yin, X., Fang, W., Liu, Z., & Liu, D. (2024). A novel multi-scale CNN and Bi-LSTM arbitration dense network model for low-rate DDoS attack detection. Scientific Reports, 14(1), 5111.

[21] Jebril, I., Premkumar, M. A. U. A. G. M., Abdulsahib, G. M., Ashokkumar, S. R., Dhanasekaran, S., Khalaf, O. I., & Algburi, S. (2024). Deep Learning based DDoS Attack Detection in Internet of Things: An Optimized CNN-BiLSTM Architecture with Transfer Learning and Regularization Techniques. Infocommunications Journal, 16(1).

[22] Zhang, Y., Liu, Y., Guo, X., Liu, Z., Zhang, X., & Liang, K. (2022). A BiLSTM-based DDoS attack detection method for edge computing. Energies, 15(21), 7882.

[23] Li, B., Li, J., & Jia, M. (2025). ADFCNN-BiLSTM: A Deep Neural Network Based on Attention and Deformable Convolution for Network Intrusion Detection. Sensors, 25(5), 1382.

[24]DDoS SDN dataset is taken from https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-datasetBalance 2 [12] Natha, S., Siraj, M., Ahmed, F., Altamimi, M., & Syed, M. (2025). An Integrated CNN-BiLSTM-Transformer Framework for Improved Anomaly Detection using Surveillance Videos. IEEE Access.

[25] Yang, Y., Tu, S., Ali, R. H., Alasmary, H., Waqas, M., & Amjad, M. N. (2023). Intrusion detection based on bidirectional long short-term memory with attention mechanism.

Author Details



Mr. R. Raja is an Assistant Professor in the Department of Computer Science and Engineering at Velammal Institute of Technology, Chennai. He holds a B.Tech in Information Technology from K.S.Rangasamy College of Technology and M.E. in Computer Science and Engineering from Annai Mathammal Sheela Engineering College. Currently, he is pursuing his Ph.D. in the Department of Information and Communication at Anna University. With over 18+ years of teaching experience in various engineering colleges, he has established himself as a dedicated educator. He has completed 7 NPTEL courses in core subjects and received the Discipline Star Award. His research interests include Machine Learning, Natural Language Processing, Computer Vision, and Deep Learning. He has published 2 papers in International Journals and presented 2 papers in International Conferences, in addition to attending workshops and Faculty Development programs.E-Mail: rajavaibavi0609@gmail.com



Dr. Venkadesh.R completed his B.E. Computer Science and Engineering from University of Madras in 1999, M.Tech Electronics and Communication Engineering from Pondicherry University (Central University) in 2004 and Ph.D Electronics and Communication Engineering in 2013. He has 25 years of teaching experience in engineering colleges. He has published book titled Advanced Computer Architecture and Organization, 21 international Journals and Conferences, filed UK Patent and granted during his service. Interested area is Network security, Proxy servers, Artificial intelligence and Data science, Deep Learning, Machine Learning, Communication and Networking. At present he is working as Professor and HOD in the Department of

Computer Science and Engineering, Velammal Institute of Technology, Chennai-601204, India.



Dr. C. Ezhilazhagan received B.E. degree in Electronics and Communication Engineering, M.E. degree in Optical Communication in Anna University, Chennai and Ph.D in the area of Wireless Communication, Anna University, Chennai.. His areas of interests wireless communication, Mobile ad hoc Networks and Sensor Networks. He has a total experience in teaching of more than 18 years. He has published a book titled Optical Communication and Networking and has multiple patents published in the IoT and Communication domains. He has published 8 international journals, 9 international conferences and 3 national conferences. He is currently working as Associate Professor in the Department of Electronics and Communication Engineering, VelTech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India.



A. M. Sarravanaprabhu is an Assistant Professor in the Department of Computer Science and Engineering at Velammal Institute of Technology, Panchetti, Thiruvallur, Tamil Nadu, India. He holds an M.Tech in Information Technology from Anna University, Coimbatore; an MBA and M.Phil from Bharathiar University; and a Master's degree in Mathematics from Sri Ramakrishna Mission Vidyalaya CAS, Coimbatore. He has completed his doctoral research at the University of Madras and is currently awaiting his Ph.D. viva.With over 16 years of teaching experience at both undergraduate and postgraduate levels, he has actively contributed to academia by presenting numerous papers at national and international conferences and seminars. His research interests include Networking, Data Analytics, Cybersecurity, Mathematical Biology, Differential Equations and Graph Theory. Prof. A.M. Sarravanaprabhu has published several research papers in UGC CARE, Scopus and Web of Science indexed journals.