International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 694-719
https://doi.org/10.70917/ijcisim-2026-5473
Article

AI-Driven Intrusion Detection System for Edge Computing Environments Using Lightweight Deep Learning Models

Pramod Ganjewar[1], Poonam Lambhate[1], Manisha Dhage[2], Netra Patil[3], Amol Rajmane[1], Vaishali Wangikar[1], Usha Verma[4]

[1]Department of Computer Engineering, MIT Academy of Engineering, Pune, India — pramod.ganjewar@mitaoe.ac.in; poonam.lambhate@mitaoe.ac.in; amol.rajmane@mitaoe.ac.in; vaishali.wangikar@mitaoe.ac.in
[2]Department of CSE, School of Computing, MIT, ADT, Pune, India — manishadhage719@gmail.com
[3]Department of Computer Engineering, Bharati Vidyapeeth (Deemed to Be University) College of Engineering, Pune, India — nspatil@bvucoep.edu.in
[4]Department of Electronics and Telecommunication Engineering, MIT Academy of Engineering, Pune, India — usha.verma@mitaoe.ac.in
Corresponding author: Amol Rajmane

Abstract: The increasing deployment of Internet of Things (IoT) devices and edge computing services has introduced significant cybersecurity challenges due to the distributed nature and resource constraints of edge environments. Conventional intrusion detection systems (IDSs), particularly computationally intensive deep learning approaches, may introduce excessive latency and memory requirements, limiting their suitability for real-time edge deployment. This study proposes a Lightweight Attention-Based Convolutional Neural Network (LA-CNN) for efficient intrusion detection in edge computing environments. The proposed framework combines data preprocessing, feature optimization using correlation analysis, Mutual Information (MI), and Recursive Feature Elimination (RFE), depthwise separable convolution, and channel-attention-based feature learning to improve detection performance while reducing computational requirements. The proposed model is evaluated using the CICIoT2023 dataset under a multi-class intrusion detection setting and compared with conventional machine learning and deep learning approaches. Experimental results demonstrate that LA-CNN achieves 98.34% accuracy, 98.12% precision, 98.41% recall, and 98.26% F1-score, with a false alarm rate of 1.59%. Furthermore, the proposed model requires only 8.7 ms inference time and 91 MB memory, representing approximately 52.7% lower inference latency and 37.2% lower memory consumption than the conventional CNN baseline. These results demonstrate that integrating feature optimization, lightweight convolution, and attention-based feature representation can provide an effective balance between intrusion detection performance and computational efficiency. The proposed framework therefore offers a promising solution for real-time cybersecurity monitoring in resource-constrained edge computing environments.

Keywords: Edge Computing, Intrusion Detection System, Lightweight Deep Learning, Lightweight Attention-Based CNN, Feature Optimization, Channel Attention, CICIoT2023, Edge AI, Cybersecurity

1. INTRODUCTION

The rapid growth of Internet of Things (IoT) devices, industrial automation, smart healthcare, intelligent transportation, smart cities, and other real-time applications has significantly increased the demand for distributed computing and communication services. Conventional cloud-centric architectures process a large proportion of data at centralized servers, which can introduce communication latency, increased bandwidth requirements, and delays in responding to time-sensitive applications. Edge computing addresses these limitations by moving computation,

storage, and intelligent processing closer to data-generating devices, thereby supporting low-latency and real-time services [1], [2].Despite these advantages, the distributed and heterogeneous nature of edge computing introduces substantial cybersecurity challenges. Edge environments typically consist of numerous interconnected devices with different hardware capabilities, operating conditions, and security configurations. These characteristics increase the overall attack surface and make centralized security management more difficult. Edge and IoT infrastructures may be exposed to threats such as Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), botnet activities, brute-force attacks, spoofing, reconnaissance, and unauthorized access [3]–[5]. Consequently, effective intrusion detection is essential for maintaining the reliability and security of edge-based applications.Intrusion Detection Systems (IDSs) are widely used to identify suspicious network activities and protect computing infrastructures from malicious behavior. Conventional IDS approaches can generally be categorized as signature-based and anomaly-based systems. Signature-based methods are effective for identifying previously known attack patterns but are less effective against previously unseen or evolving threats. Anomaly-based approaches, in contrast, analyze deviations from normal network behavior and can provide greater flexibility for identifying unknown or emerging attacks [6]. However, conventional IDS solutions may require considerable computational, memory, and processing resources, creating challenges for deployment on resource-constrained edge devices [7].

The application of artificial intelligence (AI) and deep learning has provided new opportunities for developing automated intrusion detection systems. Deep learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks can learn complex representations from network traffic and identify sophisticated attack patterns [8], [9]. Nevertheless, the improved representation capability of deep learning models is often accompanied by increased parameter counts, computational operations, memory requirements, and inference latency. Such requirements can conflict with the limited computational and energy resources available in edge environments, particularly when real-time intrusion detection is required [10].To address this limitation, recent research has increasingly focused on lightweight deep learning architectures that attempt to maintain high detection performance while reducing computational overhead. Techniques such as depthwise separable convolution, feature selection, dimensionality reduction, pruning, quantization, and attention mechanisms can reduce model complexity and improve the efficiency of deep learning-based security systems [11], [12]. Among these techniques, feature optimization can eliminate redundant network traffic attributes, while attention mechanisms can emphasize the most informative features for intrusion classification. Their combination therefore provides an opportunity to improve detection performance without introducing the computational burden associated with larger deep learning architectures [13].Although existing studies have demonstrated the potential of AI-based IDSs for IoT, fog, and edge environments, several challenges remain. Existing approaches frequently emphasize classification accuracy without providing sufficient analysis of inference latency, memory consumption, and computational requirements. In addition, the use of large feature sets may introduce redundant information and unnecessary processing overhead, while relatively limited attention has been given to combining feature optimization with lightweight attention-based architectures specifically for resource-constrained edge environments. These limitations motivate the development of an IDS that considers both detection effectiveness and computational efficiency as simultaneous design objectives.Therefore, this study proposes a Lightweight Attention-Based Convolutional Neural Network (LA-CNN) for intrusion detection in edge computing environments. The proposed framework combines data preprocessing, feature optimization, depthwise separable convolution, channel attention, and Global Average Pooling to develop a computationally efficient intrusion detection model. Correlation analysis, Mutual Information (MI), and Recursive Feature Elimination (RFE) are employed to reduce redundant and less informative features before classification. The lightweight CNN architecture is then used to extract relevant traffic representations, while the attention mechanism assigns greater importance to attack-relevant features. The resulting framework is designed to provide accurate multi-class intrusion detection while reducing inference latency and memory requirements.The primary objectives of this research are to: (i) develop a lightweight deep learning-based IDS for resource-constrained edge computing environments; (ii) reduce feature dimensionality using correlation analysis, Mutual Information, and Recursive Feature Elimination; (iii) integrate channel attention with a lightweight CNN to improve attack-feature representation; (iv) reduce computational and memory requirements

through depthwise separable convolution and Global Average Pooling; and (v) evaluate the proposed framework in terms of detection performance, inference latency, memory consumption, and comparative performance against existing intrusion detection approaches.

The major contributions of this study are summarized as follows:

A. A lightweight intrusion detection framework based on LA-CNN is developed for resource-constrained edge computing environments.

B. A multi-stage feature optimization strategy combining correlation analysis, Mutual Information, and Recursive Feature Elimination is incorporated to reduce redundant network traffic features.

C. A channel-attention mechanism is integrated with a lightweight CNN to improve the representation of intrusion-relevant traffic characteristics.

D. Computational efficiency is explicitly evaluated using inference latency and memory consumption in addition to conventional classification metrics.

E. Comprehensive performance evaluation is conducted using the CICIoT2023 dataset, including accuracy, precision, recall, F1-score, detection rate, false alarm rate, and computational-efficiency measures.

F. Comparative and robustness analyses are performed to examine the effectiveness and generalization capability of the proposed LA-CNN framework against alternative intrusion detection approaches.

The remainder of this paper is organized as follows. Section II presents the related literature and critically analyzes existing intrusion detection approaches for edge and IoT environments. Section III defines the research problem, identifies the major research gaps, and presents the research objectives. Section IV describes the proposed LA-CNN framework and its constituent components. Section V presents the experimental methodology, dataset, implementation environment, and evaluation protocol. Section VI discusses the experimental results, including classification performance, class-wise analysis, comparative evaluation, and computational efficiency. Finally, Section VII concludes the study and discusses directions for future research.

2. LITERATURE REVIEW

The increasing adoption of edge computing and Internet of Things (IoT) technologies has created a growing need for intrusion detection systems (IDSs) capable of operating under distributed and resource-constrained conditions. Unlike centralized cloud environments, edge infrastructures consist of heterogeneous devices with limited computational, memory, and energy resources. Consequently, conventional security mechanisms may not provide the required level of responsiveness for real-time edge applications. Machine learning (ML) and deep learning (DL) techniques have therefore attracted considerable attention because of their ability to automatically learn patterns from network traffic and identify malicious activities [1], [2].

A. Machine Learning and Deep Learning-Based Intrusion Detection

Early research on intelligent IDSs primarily focused on machine learning techniques for identifying anomalous network behavior. Optimized ML-based approaches have demonstrated improved detection efficiency in fog and edge environments, although scalability and deployment constraints remain important concerns. Alzubi et al. [2], for example, investigated an optimized machine-learning-based IDS for fog and edge computing and demonstrated the potential of optimization techniques for improving detection efficiency. However, such approaches still require consideration of their scalability when deployed across heterogeneous edge infrastructures. Deep learning has subsequently provided more powerful mechanisms for learning complex network traffic representations. CNN-based architectures can automatically extract discriminative traffic patterns, while LSTM and other recurrent architectures can model sequential dependencies in network activities. Yang and Shami [3] investigated transfer learning and an optimized CNN-based IDS and reported improved classification performance, although the associated training complexity remains a concern. Similarly, hybrid deep learning approaches such as HDLNIDS have demonstrated

strong detection capability but introduce additional computational overhead [6].The use of hybrid architectures has also been explored for IoT and network security applications. Yaras and Dener [13] proposed a hybrid deep learning approach for IoT-based intrusion detection and demonstrated high detection accuracy. However, the complexity of hybrid architectures can limit their suitability for resource-constrained edge devices. Other deep learning-based studies have similarly reported improved attack classification but have identified increased memory requirements and resource consumption as important limitations [9], [11].

B. Lightweight Intrusion Detection for Edge Environments

The computational limitations of edge devices have motivated research into lightweight intrusion detection architectures. Bangui et al. [1] investigated lightweight intrusion detection for edge computing environments with the objective of reducing computational complexity. Das and Luo [4] introduced a lightweight anomaly detection framework designed to facilitate faster deployment. Hnamte et al. [16] further investigated a lightweight IDS using deep convolutional neural networks and demonstrated improved real-time detection capability. Although these studies demonstrate the feasibility of lightweight architectures, reducing computational complexity while maintaining high detection performance remains challenging. Some approaches concentrate primarily on reducing model complexity, whereas others emphasize detection accuracy without sufficiently analyzing memory consumption and inference latency. Thus, the design of an IDS for edge environments requires simultaneous consideration of detection effectiveness and resource efficiency.

C. Feature Optimization in Intrusion Detection

Network traffic datasets typically contain a large number of packet-level and flow-level attributes. Although these features can provide useful information for attack classification, redundant, highly correlated, and less informative attributes may increase computational requirements and negatively affect model efficiency. Feature selection and optimization techniques have therefore been incorporated into several IDS approaches. Kavitha et al. [7] investigated feature selection combined with deep learning for intrusion detection and demonstrated the importance of feature optimization in improving the learning process. Other studies have similarly explored optimized learning strategies to improve detection performance [5], [6]. However, feature optimization is not consistently combined with lightweight deep learning architectures specifically designed for edge deployment. This creates an opportunity to investigate whether systematic feature reduction can further decrease computational requirements without significantly reducing detection capability.

D. Attention-Based Intrusion Detection

Attention mechanisms provide another approach for improving feature representation by assigning greater importance to informative characteristics while suppressing less relevant information. Attention-based deep learning architectures have demonstrated potential for identifying complex patterns in network traffic. However, the integration of attention mechanisms with lightweight convolutional architectures for resource-constrained edge intrusion detection remains comparatively less explored. The combination of attention-based feature representation and lightweight convolution is particularly relevant to edge environments. A lightweight convolutional architecture can reduce computational operations, while attention can improve the model's ability to focus on attack-relevant traffic characteristics. This combination provides a potential mechanism for achieving a better balance between detection accuracy and computational cost.

E. Critical Analysis of Existing Research

The literature indicates that existing IDS research has achieved considerable progress in attack detection using ML and DL techniques. However, the existing approaches reveal several recurring limitations. First, complex architectures such as CNN-LSTM, recurrent networks, and hybrid deep learning models can introduce substantial computational overhead. Second, many studies primarily evaluate classification metrics such as accuracy, precision, recall, and F1-score, while comparatively less emphasis is placed on inference latency and memory consumption.

Third, a large number of network traffic features may increase computational requirements when feature redundancy is not adequately addressed. Finally, although lightweight architectures and attention mechanisms have independently demonstrated benefits, their combined use for edge-oriented intrusion detection requires further investigation. Therefore, the literature suggests that an effective edge-based IDS should simultaneously address detection accuracy, feature efficiency, computational complexity, inference latency, and memory consumption. These observations motivate the development of a Lightweight Attention-Based Convolutional Neural Network (LA-CNN) that combines feature optimization with lightweight convolution and attention-based feature learning.

Table I. Comparative Summary of Recent Intrusion Detection Studies

Ref.	Year	Study	Main Approach	Major Finding	Key Limitation
[1]	2022	Bangui et al.	Lightweight IDS	Reduced computational complexity	Limited DL integration
[2]	2022	Alzubi et al.	Optimized ML-based IDS	Improved detection efficiency	Scalability concerns
[3]	2022	Yang and Shami	Transfer Learning + CNN	Improved accuracy	High training complexity
[4]	2023	Das and Luo	Lightweight anomaly detection	Faster deployment	Limited attack diversity
[5]	2023	Nakıp et al.	Online self-supervised DL	Improved anomaly detection	Increased resource usage
[6]	2023	Qazi et al.	Hybrid DL IDS	High detection performance	Computational overhead
[7]	2023	Kavitha et al.	Feature Selection + DL	Improved feature optimization	Limited real-time evaluation
[8]	2023	Vinolia et al.	ML/DL Review	Comprehensive comparative analysis	No specific deployment framework
[9]	2024	Aljuaid et al.	Deep Learning IDS	Improved attack classification	High memory requirements
[10]	2024	Altaie and Hoomod	Hybrid Lightweight DL	Efficient detection	Inference latency
[11]	2024	Alsoufi et al.	Anomaly-based DL IDS	Improved anomaly detection	Resource-intensive training
[12]	2024	Sanagana and Tummalachervu	Optimal DL IDS	Improved security performance	Limited edge deployment
[13]	2024	Yaras and Dener	Hybrid DL IDS	High detection accuracy	Complex architecture
[16]	2025	Hnamte et al.	Lightweight CNN IDS	Improved real-time detection	Dataset balancing required
[23]	2025	Baidar et al.	Federated Learning IDS	Distributed threat detection	Communication overhead

The comparative studies indicate that existing approaches generally improve intrusion detection performance but continue to face challenges related to computational complexity, scalability, memory requirements, inference latency, feature redundancy, and practical edge deployment. These limitations establish the need for an IDS architecture that jointly optimizes detection performance and resource efficiency. The proposed LA-CNN addresses

this requirement through feature optimization, depthwise separable convolution, and channel-attention-based feature learning.

F. Identified Research Gaps

Based on the critical analysis of existing studies, five major research gaps are identified.

1. **Computational complexity:**

   Many high-performing IDS approaches employ complex CNN, LSTM, hybrid, or other deep learning architectures. Although these models can achieve strong classification performance, their computational requirements may restrict deployment on resource-constrained edge devices [6], [10].

2. **Limited resource-aware evaluation:**

   Existing studies frequently emphasize classification accuracy while providing limited analysis of inference latency, memory consumption, and other deployment-related resource requirements. These factors are particularly important for real-time edge security applications [9], [12].

3. **Feature redundancy and dimensionality:**

   Network intrusion datasets can contain numerous correlated, redundant, or less informative features. Processing all available attributes increases computational requirements and may introduce unnecessary complexity. Although feature-selection approaches have been investigated [7], their systematic integration with lightweight attention-based architectures remains limited.

4. **Limited integration of attention with lightweight architectures:**

   Attention mechanisms can improve feature representation by emphasizing important characteristics of network traffic. However, comparatively limited research has investigated their integration with computationally efficient CNN architectures specifically for edge-oriented intrusion detection [13], [18].

5. **Detection–efficiency trade-off:**

   Existing approaches often optimize either detection performance or computational efficiency rather than treating both as simultaneous objectives. A practical edge IDS must maintain high precision and recall while also achieving low inference latency and reduced memory consumption.

These research gaps motivate the development of the proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN). The proposed framework combines feature optimization, depthwise separable convolution, channel attention, and Global Average Pooling to achieve an improved balance between intrusion detection performance and computational efficiency in resource-constrained edge computing environments.

3. PROPOSED METHODOLOGY

The proposed methodology aims to develop a lightweight and computationally efficient Intrusion Detection System (IDS) for resource-constrained edge computing environments. The framework integrates data preprocessing, feature optimization, lightweight convolutional feature extraction, channel-attention-based feature refinement, and multi-class intrusion classification. The methodology is organized into five major stages: data acquisition, data preprocessing, feature optimization, Lightweight Attention-Based Convolutional Neural Network (LA-CNN) processing, and intrusion classification. The overall architecture of the proposed framework is illustrated in Fig. 1.The primary design objective is to achieve an effective balance between intrusion detection performance and computational efficiency. Unlike conventional deep learning-based IDS architectures that may require considerable memory and processing resources, the proposed framework reduces the dimensionality of network traffic features and employs computationally efficient convolution operations. The attention mechanism further improves feature representation by assigning greater importance to intrusion-relevant characteristics.

A. Data Acquisition

The CICIoT2023 dataset is used for experimental validation of the proposed LA-CNN framework. The dataset represents network traffic generated in IoT and edge-computing environments and contains both normal and malicious

network traffic records. It includes multiple categories of cyberattacks, making it suitable for evaluating multi-class intrusion detection. The attack categories considered in the experimental description include DDoS, DoS, Botnet, Brute Force, Reconnaissance, Spoofing, and Web Attacks, together with normal traffic. The diversity of traffic patterns enables the proposed model to learn characteristics associated with different malicious activities. The dataset contains network-flow-related attributes describing traffic behavior and communication characteristics. These attributes provide the input representation required for feature optimization and subsequent classification. The use of CICIoT2023 provides an appropriate experimental basis for evaluating the proposed framework because the dataset reflects IoT-oriented network security conditions and contains multiple attack categories. The resulting multi-class classification problem allows the model to distinguish between normal traffic and different forms of malicious activity.

B. Data Preprocessing

Raw network traffic data may contain missing values, duplicate observations, noisy records, and features with substantially different numerical ranges. These characteristics can negatively affect the training process and increase the computational burden of the learning model. Therefore, a systematic preprocessing procedure is applied before feature optimization. First, missing or corrupted records are identified and appropriately handled. Duplicate records are subsequently removed to reduce redundancy and prevent potential bias during model training. Numerical network traffic attributes are then normalized using the Min-Max normalization technique so that feature values are transformed into the range [0,1].

The Min-Max normalization is expressed as:

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min}) \quad (1)$$

where X represents the original feature value, X_{min} represents the minimum value of the feature, X_{max} represents the maximum value, and X_{norm} represents the normalized feature value.

Normalization ensures that features with different numerical scales contribute more consistently during model training. It also supports stable optimization and convergence of the neural network.

The attack labels are converted into numerical representations using label encoding so that they can be processed by the classification model. Since intrusion detection datasets may contain imbalanced class distributions, a data-balancing procedure is applied to reduce the potential bias toward majority classes and improve the representation of minority attack categories.

Following preprocessing, the dataset is divided into training and testing subsets using an 80:20 ratio. The training subset is used to develop the LA-CNN model, while the testing subset is reserved for evaluating its generalization performance.

C. Feature Optimization

Network intrusion datasets generally contain a large number of traffic attributes, including packet-level characteristics, communication statistics, and protocol-related information. However, not all attributes contribute equally to intrusion classification. Highly correlated, redundant, or less informative features can increase computational requirements and may negatively affect the efficiency of the learning process. To address this problem, the proposed framework employs a three-stage feature optimization strategy consisting of Correlation Analysis (CA), Mutual Information (MI), and Recursive Feature Elimination (RFE).

1. **Correlation Analysis**

Correlation analysis is initially applied to identify redundant features. Features exhibiting strong correlation with other attributes may contain overlapping information. Removing such redundant attributes reduces the dimensionality of the input space and minimizes unnecessary computational operations.

2. **Mutual Information**

After correlation-based filtering, Mutual Information is used to determine the relationship between the remaining features and the intrusion class. The Mutual Information between an input feature X and target variable Y is expressed as:

$$MI(X,Y) = \Sigma_x \in X \, \Sigma_y \in Y \; p(x,y) \log [p(x,y) / (p(x)p(y))] \quad (2)$$

where p(x,y) represents the joint probability distribution of X and Y, while p(x) and p(y) represent their respective marginal probability distributions. Features having greater mutual information with the target class provide more information for intrusion classification and are therefore given higher priority during feature selection.

3. **Recursive Feature Elimination**

Recursive Feature Elimination (RFE) is subsequently applied to eliminate less informative attributes and retain the most relevant feature subset. The process progressively removes lower-ranked features until the desired feature representation is obtained. The combined CA–MI–RFE procedure reduces the dimensionality of the original dataset while retaining important intrusion-related information. This reduction is expected to decrease memory requirements, training time, and computational overhead while providing an efficient input representation for the proposed deep learning model.

D. Lightweight Attention-Based CNN

The Lightweight Attention-Based Convolutional Neural Network (LA-CNN) forms the central component of the proposed intrusion detection framework. The architecture is designed to extract discriminative traffic patterns while minimizing the computational requirements normally associated with conventional convolutional neural networks.

The LA-CNN consists of depthwise separable convolutional operations, Batch Normalization, ReLU activation, channel attention, Global Average Pooling, and a final classification layer.

1. **Depthwise Separable Convolution**

Conventional convolutional layers can require a substantial number of multiplication and accumulation operations when the number of input and output channels is large. To reduce this computational burden, the proposed architecture employs depthwise separable convolution.

A conventional convolution operation has an approximate computational cost represented by:

$$Cost_{stan}d_{ar}d = D_k^2 \times M \times N \times D_f^2 \quad (3)$$

where D_k represents the kernel size, M represents the number of input channels, N represents the number of output channels, and D_f represents the feature-map dimension.

For depthwise separable convolution, the computational cost is represented by:

$$Cost\_DSC = D_k^2 \times M \times D_f^2 + M \times N \times D_f^2 \quad (4)$$

The depthwise operation processes each input channel separately, while the subsequent pointwise convolution combines the resulting channel representations. This decomposition considerably reduces the number of computational operations compared with a standard convolution and therefore supports deployment in resource-constrained environments.

2. **Convolutional Feature Extraction**

The convolution operation extracts local patterns from the input feature representation. The resulting feature map can be represented as:

$$F(i,j) = \Sigma_m \, \Sigma_n \, X(i+m,j+n)W(m,n) + b \quad (5)$$

where X represents the input feature matrix, W represents the convolution kernel weights, and b represents the bias term.

The convolution operation enables the network to learn discriminative representations associated with normal and malicious network traffic.

3. **Batch Normalization and ReLU Activation**

Batch Normalization is incorporated into the convolutional blocks to stabilize the training process and improve convergence. The ReLU activation function is subsequently applied to introduce non-linearity into the network.

The ReLU function is defined as:

$$ReLU(x) = \max(0,x) \quad (6)$$

The nonlinear activation enables the network to learn complex relationships between optimized traffic features and intrusion categories.

4. **Channel Attention Mechanism**

A channel-attention mechanism is incorporated into the LA-CNN architecture to improve the representation of intrusion-related features. Instead of assigning equal importance to all extracted feature channels, the attention mechanism learns importance weights and emphasizes the channels that contain more relevant information.

The attention score is represented as:

$$A = \sigma(W_a F + b_a) \quad (7)$$

where A represents the attention weights, F represents the extracted feature representation, W_a represents the learnable attention weights, b_a represents the bias term, and σ represents the activation function.

The attention-enhanced feature representation is then obtained as:

$$F_{att} = A \odot F \quad (8)$$

where F_{att} represents the attention-enhanced feature representation and ⊙ denotes element-wise multiplication.

The attention mechanism enables the model to focus on traffic characteristics that are more informative for identifying malicious activities while reducing the influence of less relevant information.

5. **Global Average Pooling**

Following attention-based feature refinement, Global Average Pooling (GAP) is used to reduce the dimensionality of the feature representation before classification. GAP reduces the number of parameters compared with large fully connected layers and can help control overfitting.

The GAP operation is defined as:

$$G_k = 1/(H \times W) \sum_{i=1}^{H} \sum_{j=1}^{W} F_{att}(i,j) \quad (9)$$

where H and W represent the height and width of the feature map, respectively.

E. Intrusion Classification

The feature representation obtained after convolution, attention refinement, and global average pooling is supplied to the final classification layer. A Softmax function is used to generate the probability associated with each intrusion category. The Softmax probability for class i is expressed as:

$$P(y_i) = e^{z_i} / \sum_{j=1}^{c} e^{z_j} \quad (10)$$

where C represents the total number of intrusion classes and z_i represents the output logit corresponding to class i. The class having the highest predicted probability is selected as the final classification result.

The proposed architecture can support both binary and multi-class intrusion detection. In binary classification, network traffic is categorized as either normal or malicious. In multi-class classification, the model distinguishes among the different attack categories represented in the dataset.For the experimental evaluation presented in this study, the focus is on multi-class intrusion classification using the selected CICIoT2023 traffic categories.

F. Model Training

The LA-CNN model is trained using the Adam optimization algorithm with a learning rate of 0.001. The batch size is set to 32, and the model is trained for 50 epochs. ReLU is used as the activation function, while Softmax is employed for final class-probability generation. A dropout rate of 0.30 is incorporated to reduce the possibility of overfitting. Categorical cross-entropy is used as the loss function.

The categorical cross-entropy loss is defined as:

$$L = -\Sigma_{i=1}^{c} y_i \log(P(y_i)) \quad (11)$$

where y_i represents the actual class label and $P(y_i)$ represents the predicted probability for class i.

During training, the model parameters are iteratively updated to minimize the classification loss and improve the ability of the network to distinguish between different network traffic categories.

G. Intrusion Detection Decision Process

After model training, previously unseen network traffic is passed through the same preprocessing and feature-optimization pipeline used during training. The selected features are supplied to the trained LA-CNN model, which extracts lightweight feature representations and applies channel attention to emphasize relevant traffic characteristics.The final Softmax layer produces a probability distribution over the intrusion categories. The category associated with the highest probability is selected as the predicted class. This process enables the system to identify normal traffic as well as different categories of malicious activity.

The complete decision process can therefore be represented as:

Network Traffic → Preprocessing → Feature Optimization → LA-CNN → Channel Attention → GAP → Softmax → Intrusion Class

This sequential processing structure enables the proposed IDS to perform automated intrusion classification while reducing unnecessary feature processing and computational overhead.

H. Performance Evaluation

The proposed framework is evaluated from two complementary perspectives: classification effectiveness and computational efficiency.

1. **Classification Metrics**

The primary classification metrics are Accuracy, Precision, Recall, F1-Score, Detection Rate, and False Alarm Rate. These metrics provide complementary information about the ability of the model to correctly identify normal and malicious network traffic.

Accuracy is calculated as:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (12)$$

Precision is calculated as:

Precision = TP / (TP + FP) (13)

Recall is calculated as:

Recall = TP / (TP + FN) (14)

The F1-score is calculated as:

F1 = 2 × (Precision × Recall) / (Precision + Recall) (15)

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

These metrics are used to evaluate the ability of the proposed framework to correctly identify intrusion events while minimizing incorrect classifications.

2. **Computational Efficiency Metrics**

Classification performance alone is insufficient for evaluating an IDS intended for edge computing. Therefore, the proposed framework is also evaluated using resource-oriented measures, including:

- Inference Time: time required by the trained model to generate a prediction.
- Memory Consumption: memory required during model operation.
- Model Size: storage requirement of the trained model.
- Computational Complexity: computational requirements associated with model operations.

Lower inference time and memory consumption indicate better suitability for resource-constrained edge devices. These measures are therefore analyzed together with classification performance to determine whether the proposed model achieves an effective detection-efficiency trade-off.

Overall Workflow of the Proposed Framework

The overall workflow of the proposed LA-CNN framework is presented in Figure. 1. The process begins with the acquisition of network traffic from the CICIoT2023 dataset. The raw data are subsequently cleaned, normalized, encoded, and balanced before being passed to the feature optimization stage. Correlation analysis removes redundant attributes, Mutual Information identifies informative features, and Recursive Feature Elimination further reduces the feature space. The optimized feature representation is then supplied to the LA-CNN architecture. Within LA-CNN, depthwise separable convolution performs computationally efficient feature extraction. Batch Normalization and ReLU support stable nonlinear learning, while the channel-attention mechanism assigns greater importance to intrusion-relevant features. Global Average Pooling reduces the feature representation before the final Softmax classifier generates the predicted intrusion category.

The resulting framework is designed to jointly address the major requirements of edge-based intrusion detection: high classification performance, reduced feature dimensionality, low computational complexity, low inference latency, and reduced memory consumption. Figure 1. Proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN) Framework for Intrusion Detection in Edge Computing Environments.

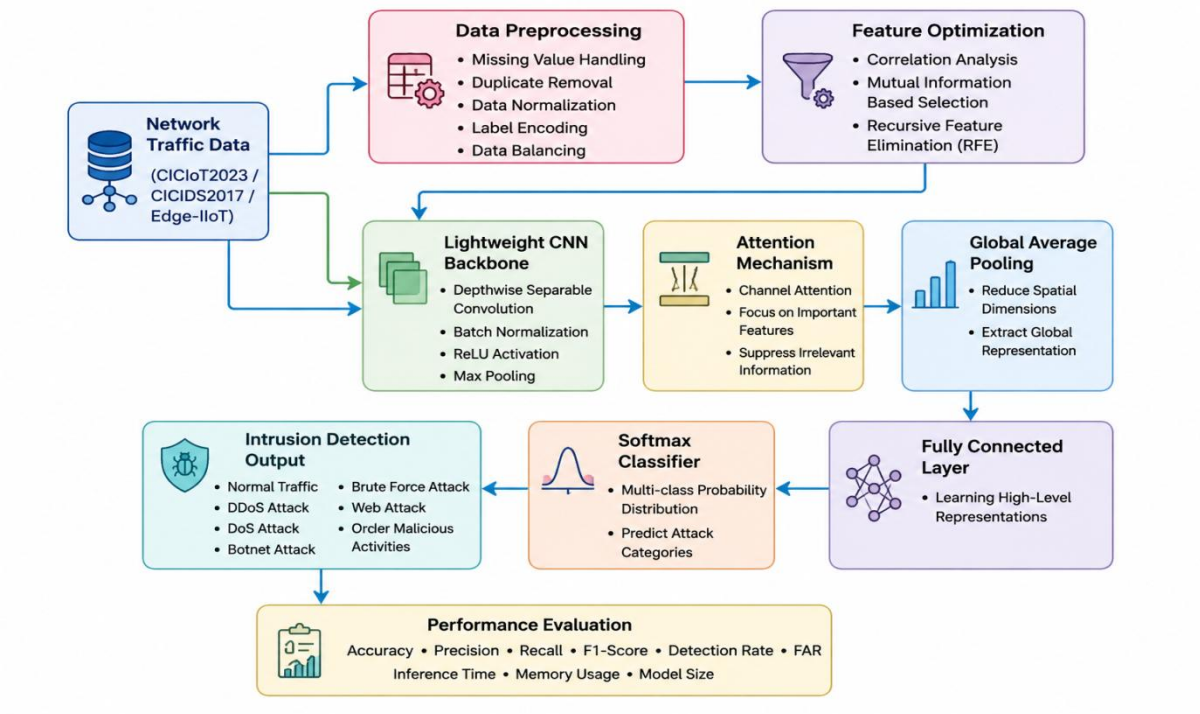


Figure 1. Proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN) Framework for Intrusion Detection in Edge Computing Environments.

4. Mathematical Formulation of the Model

The mathematical formulation of the proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN) consists of the principal operations involved in data normalization, feature selection, lightweight convolution, feature extraction, nonlinear activation, attention-based feature refinement, global average pooling, classification, and model optimization. These formulations describe the transformation of network traffic data from the original input representation to the final intrusion classification.

1. **Data Normalization**

Network traffic features may have different numerical ranges. To provide a uniform input representation for the proposed model, Min-Max normalization is applied to transform each feature into the range [0,1]. The normalized feature value is calculated as:

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min}) \quad (1)$$

where X represents the original feature value, X_{min} represents the minimum value of the corresponding feature, X_{max} represents the maximum value, and X_{norm} represents the normalized feature value. Normalization provides a consistent numerical scale across the input features and supports stable model training.

2. **Feature Selection Using Mutual Information**

Mutual Information (MI) is employed to quantify the dependency between network traffic features and their corresponding intrusion classes. The Mutual Information between feature variable X and attack label Y is defined as:

$$MI(X,Y) = \Sigma_{x \in X} \Sigma_{\gamma \in Y} p(x,y) \log [p(x,y) / (p(x)p(y))] \quad (2)$$

where X represents the input feature variable, Y represents the attack-class variable, p(x,y) represents the joint probability distribution, and p(x) and p(y) represent the corresponding marginal probability distributions. Features

with higher Mutual Information values provide greater information about the target intrusion class and are therefore prioritized during feature selection.

3. **Lightweight Convolution Operation**

The proposed LA-CNN uses depthwise separable convolution to reduce computational complexity compared with conventional convolution. The approximate computational cost of a standard convolution operation is given by:

$$Cost_{standard} = D_k^2 \times M \times N \times D_f^2 \quad (3)$$

where D_k represents the convolution kernel size, M represents the number of input channels, N represents the number of output channels, and D_f represents the feature-map dimension.

The corresponding computational cost of depthwise separable convolution is expressed as:

$$Cost\_DSC = D_k^2 \times M \times D_f^2 + M \times N \times D_f^2 \quad (4)$$

The decomposition of the conventional convolution into depthwise and pointwise operations reduces the number of computational operations. This reduction is particularly beneficial for deployment in resource-constrained edge environments.

4. **Convolutional Feature Extraction**

The convolution operation extracts local feature representations from the optimized network traffic input. The resulting feature map is represented as:

$$F(i,j) = \Sigma_m \Sigma_n X(i+m,j+n)W(m,n) + b \quad (5)$$

where X represents the input feature matrix, W represents the convolution kernel weights, b represents the bias term, and F(i,j) represents the resulting feature-map value at position ((i,j)).

The convolution operation enables the proposed model to learn discriminative representations associated with different network traffic patterns.

5. **ReLU Activation Function**

The Rectified Linear Unit (ReLU) activation function introduces nonlinearity into the convolutional network. It is defined as:

$$ReLU(x) = \max(0,x) \quad (6)$$

The ReLU function retains positive activation values while assigning zero to negative values. It enables the network to learn nonlinear relationships between network traffic characteristics and intrusion classes.

6. **Channel Attention Mechanism**

The proposed LA-CNN incorporates a channel-attention mechanism to assign different importance levels to the extracted feature representations. The attention score is calculated as:

$$A = \sigma(W_a F + b_a) \quad (7)$$

where A represents the attention-weight vector, F represents the extracted feature representation, W_a represents the learnable attention parameters, b_a represents the corresponding bias, and σ represents the activation function.

The attention-enhanced feature representation is then obtained using element-wise multiplication:

$$F_{att} = A \odot F \quad (8)$$

where F_{att} represents the attention-enhanced feature representation and $\odot$ denotes element-wise multiplication.

The attention mechanism enables the network to emphasize informative traffic characteristics while reducing the contribution of less relevant features.

7. **Global Average Pooling**

Global Average Pooling (GAP) is used to reduce the dimensionality of the attention-enhanced feature maps before classification. The output of the GAP operation is calculated as:

$$G_k = 1/(H \times W) \Sigma_{i=1}^{H} \Sigma_{j=1}^{W} F_{att}(i,j) \quad (9)$$

where H and W represent the height and width of the feature map, respectively, and G_k represents the resulting pooled feature representation.

GAP reduces the number of parameters required by the subsequent classification stage and contributes to controlling model complexity.

8. **Softmax Classification**

The final classification layer uses the Softmax function to calculate the probability associated with each intrusion category. For class (i), the predicted probability is calculated as:

$$P(y_i) = e^{z_i} / \Sigma_{j=1}^{c} e^{z_j} \quad (10)$$

where C represents the total number of intrusion classes and z_i represents the classifier output logit corresponding to class (i). The class having the highest predicted probability is selected as the final intrusion category.

9. **Categorical Cross-Entropy Loss**

The LA-CNN model is trained using categorical cross-entropy loss. The loss function is defined as:

$$L = -\Sigma_{i=1}^{c} y_i \log(P(y_i)) \quad (11)$$

where C represents the number of classes, y_i represents the actual class indicator for class (i), and $P(y_i)$ represents the predicted probability for class (i).

During training, the model parameters are optimized by minimizing the categorical cross-entropy loss. A lower loss indicates a closer correspondence between the predicted class probabilities and the actual class labels.

10. **Classification Performance Metrics**

The effectiveness of the proposed intrusion detection framework is evaluated using Accuracy, Precision, Recall, and F1-Score. These metrics are calculated from the numbers of true-positive, true-negative, false-positive, and false-negative predictions.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (12)$$

Accuracy represents the proportion of correctly classified samples among all evaluated samples.

$$Precision = TP / (TP + FP) \quad (13)$$

Precision represents the proportion of correctly identified positive or attack predictions among all samples predicted as positive.

$$Recall = TP / (TP + FN) \quad (14)$$

Recall represents the ability of the model to correctly identify actual attack instances.

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (15)$$

The F1-score provides a combined measure of precision and recall and is particularly useful when the dataset contains multiple intrusion categories or imbalanced class distributions.The above mathematical formulations collectively describe the processing pipeline of the proposed LA-CNN model, from normalized network traffic and optimized feature representation to attention-enhanced feature learning and final intrusion classification. The formulation also establishes the mathematical basis for evaluating both the detection capability and learning behavior of the proposed framework.

5. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Experimental Environment

The proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN) was implemented using Python and commonly used machine learning and deep learning libraries. TensorFlow and Keras were used to construct, train, and evaluate the proposed neural network architecture. Scikit-learn was employed for data preprocessing, feature selection, and performance evaluation, while NumPy and Pandas were used for numerical computation and dataset manipulation, respectively. All experiments were conducted on a computing system equipped with an Intel Core i7 processor, 16 GB RAM, and NVIDIA GPU support. GPU acceleration was utilized during model training to reduce training time and improve computational efficiency. The complete implementation was developed and executed using the Jupyter Notebook environment on a Windows 11 operating system.The same experimental configuration was maintained throughout the training and testing stages to ensure consistency and facilitate reproducibility. The hardware and software configuration provided an appropriate environment for developing and evaluating the proposed intrusion detection framework.

B. Dataset Description

The CICIoT2023 dataset was used for experimental validation of the proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN). The dataset provides network traffic data relevant to IoT-oriented cybersecurity scenarios and contains both normal and malicious traffic records. Its multi-class attack representation makes it suitable for evaluating the capability of the proposed model to distinguish between legitimate network activity and different categories of malicious behavior.

The attack categories considered in this study include Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), Botnet, Brute Force, Reconnaissance, Spoofing, and Web Attacks, in addition to normal network traffic. The presence of multiple attack categories enables the proposed LA-CNN model to learn different traffic patterns and perform multi-class intrusion classification.

The dataset contains network traffic attributes that can be used to characterize communication and traffic behavior. However, directly processing a large number of attributes can increase computational requirements and introduce redundant information. Therefore, the proposed framework applies data preprocessing and feature optimization before the data are supplied to the LA-CNN model.

The main characteristics of the CICIoT2023 dataset considered in this study are summarized in **Table II**.

Table II. Characteristics of the CICIoT2023 Dataset

Parameter	**Description**
Dataset Name	CICIoT2023
Environment	IoT and Edge Computing
Data Type	Network Traffic Records

Classification Type	Multi-Class Classification
Attack Categories	DDoS, DoS, Botnet, Brute Force, Reconnaissance, Spoofing, Web Attacks
Normal Traffic	Available
Feature Type	Network Flow Features
Data Format	CSV
Application Domain	IoT and Edge Security

C. Data Preparation

Before training the LA-CNN model, the dataset was subjected to a sequence of data-cleaning, transformation, and feature-optimization procedures. These preprocessing operations were performed to improve data quality, minimize redundancy, and provide a suitable input representation for the proposed model. First, missing values were identified and removed to improve data quality. Duplicate records were subsequently removed to reduce redundancy and minimize the possibility of bias during model training. Since the network traffic features may have different numerical ranges, Min-Max normalization was applied to transform the feature values into the range [0,1]. This ensures a consistent scale among the input attributes and supports stable neural network training. After normalization, feature optimization was performed using Correlation Analysis (CA), Mutual Information (MI), and Recursive Feature Elimination (RFE). Correlation Analysis was used to identify redundant attributes, Mutual Information was employed to identify features having strong relationships with the target intrusion classes, and RFE was subsequently applied to eliminate less informative attributes and reduce the dimensionality of the feature space. Finally, the processed dataset was divided into training and testing subsets using an 80:20 ratio. The training data were used for model development, while the testing data were reserved for evaluating the classification and generalization performance of the proposed LA-CNN model.

Table III. Data Preparation Process

Phase	**Technique Used**	**Purpose**
Data Cleaning	Missing Value Removal	Improve data quality
Data Cleaning	Duplicate Record Removal	Reduce redundancy and potential training bias
Data Transformation	Min-Max Normalization	Scale features to [0,1]
Feature Optimization	Correlation Analysis	Remove redundant attributes
Feature Optimization	Mutual Information	Select relevant features
Feature Optimization	Recursive Feature Elimination (RFE)	Reduce feature dimensionality
Data Splitting	Train-Test Split (80:20)	Model training and evaluation

D. Hyperparameter Configuration

The hyperparameter configuration used to train the proposed LA-CNN model is presented in Table V. The Adam optimizer was selected with a learning rate of 0.001 to provide adaptive parameter updates during model training. A batch size of 32 was used to provide a balance between computational requirements and training stability. The model was trained for 50 epochs. ReLU was used as the nonlinear activation function, while Softmax was employed in the output layer for multi-class probability estimation. A dropout rate of 0.30 was incorporated to reduce the likelihood of overfitting. Categorical cross-entropy was selected as the loss function because the proposed model performs multi-class intrusion classification.

The channel-attention mechanism was incorporated into the architecture to assign greater importance to informative feature representations.

Table IV. Hyperparameter Configuration of Proposed LA-CNN

Hyperparameter	Value	Description
Batch Size	32	Samples processed per iteration
Learning Rate	0.001	Weight update step size
Optimizer	Adam	Adaptive optimization algorithm
Epochs	50	Training cycles
Activation Function	ReLU	Nonlinear activation
Loss Function	Categorical Cross-Entropy	Multi-class classification loss
Output Function	Softmax	Class probability generation
Validation Split	20%	Validation data ratio
Dropout Rate	0.30	Overfitting control
Attention Mechanism	Channel Attention	Feature importance weighting

E. Experimental Procedure

The complete experimental procedure follows a sequential pipeline beginning with the CICIoT2023 dataset and ending with intrusion classification and performance evaluation. First, the network traffic data are cleaned by removing missing and duplicate records. The numerical attributes are then normalized using Min-Max normalization. Following normalization, the CA–MI–RFE feature optimization process is applied to remove redundant and less informative features. The optimized feature representation is then provided as input to the LA-CNN model. Depthwise separable convolution is used to reduce computational complexity during feature extraction. Batch Normalization and ReLU activation are employed to support stable nonlinear learning, while the channel-attention mechanism assigns importance weights to the extracted features. The attention-enhanced representation is subsequently processed using Global Average Pooling, followed by the final dense and Softmax layers for multi-class intrusion classification. The trained model is finally evaluated using classification and computational-efficiency measures. Classification performance is assessed using Accuracy, Precision, Recall, F1-Score, Detection Rate, and False Alarm Rate. Computational efficiency is evaluated using Inference Time and Memory Consumption.

The overall experimental pipeline can therefore be represented as:

CICIoT2023 Dataset → Data Cleaning → Min-Max Normalization → CA–MI–RFE Feature Optimization → LA-CNN → Channel Attention → Global Average Pooling → Softmax Classification → Performance Evaluation

F. Experimental Reproducibility

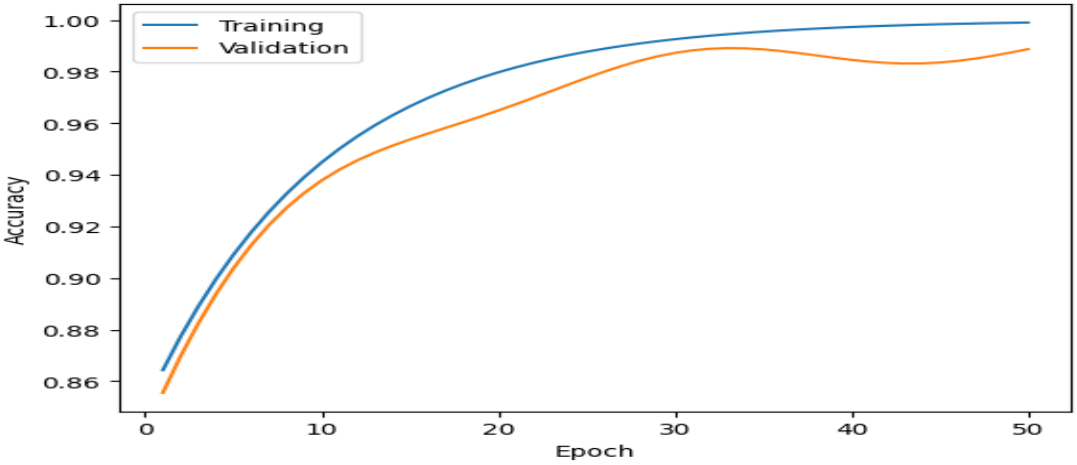
To support reproducibility, the implementation environment, dataset, preprocessing operations, feature-optimization techniques, train-test ratio, model architecture, and principal hyperparameters are explicitly defined. The same preprocessing and feature-selection procedures are applied consistently during the experimental process. The use of a fixed 80:20 train-test partition, Adam optimizer, learning rate of 0.001, batch size of 32, 50 training epochs, ReLU activation, Softmax output, and 0.30 dropout rate provide a defined experimental configuration for evaluating the proposed LA-CNN framework. The experimental setup is designed to evaluate not only the classification capability of LA-CNN but also whether the proposed lightweight architecture can reduce computational requirements compared with conventional deep learning-based IDS approaches.

6. RESULTS AND DISCUSSION

This section presents the experimental results obtained using the proposed Lightweight Attention-Based Convolutional Neural Network (LA-CNN) for intrusion detection in edge computing environments. The evaluation considers both **intrusion detection effectiveness** and **computational efficiency**. Classification performance is analyzed using accuracy, precision, recall, F1-score, detection rate, and false alarm rate, while deployment efficiency is examined using inference latency and memory consumption. The results are further analyzed through class-wise performance, confusion-matrix behavior, comparative evaluation, and robustness analysis.

A. Training Performance Analysis

Figure 2 shows the training behavior of the proposed LA-CNN model using training and validation accuracy over the training epochs.



.

Figure2 Training & Validation Accuracy Curve

The training and validation accuracy curves should be presented to demonstrate the convergence behavior of LA-CNN. A consistent increase in training accuracy accompanied by a similar improvement in validation accuracy indicates that the model successfully learns discriminative characteristics from the network traffic. A small difference between the training and validation curves indicates limited overfitting and suggests that the learned representation generalizes effectively.

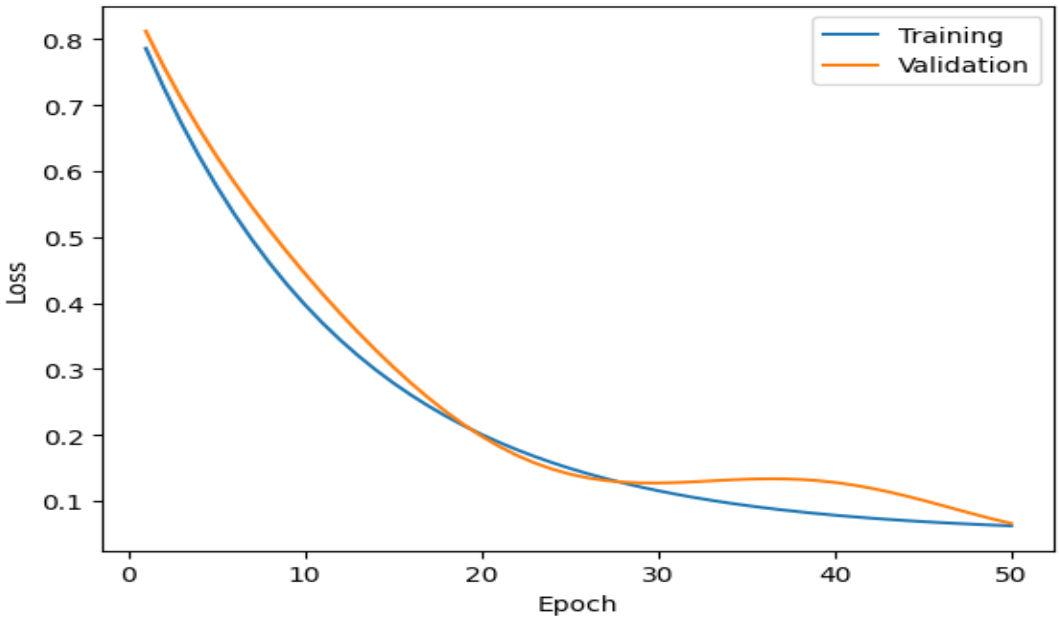


Figure 3: Training & Validation Loss Curve

The corresponding training and validation loss curves shown in Figure 3 should be analyzed to determine the stability of optimization. A progressive reduction in loss indicates effective learning during training. If the validation loss remains close to the training loss during the later epochs, the model demonstrates stable generalization. Conversely, a substantial increase in validation loss while training loss continues to decrease would indicate overfitting. The training analysis is particularly important because high test accuracy alone does not demonstrate that the model has learned a stable representation. Therefore, both accuracy and loss curves should be considered when assessing the learning behavior of LA-CNN.

B. Classification Performance Evaluation

The overall classification performance of the proposed LA-CNN model is evaluated using Accuracy, Precision, Recall, and F1-score. The reported experimental results show that LA-CNN achieves 98.34% accuracy, 98.12% precision, 98.41% recall, and 98.26% F1-score.

Table V. Performance Metrics of Proposed LA-CNN

Metric	**Value (%)**
Accuracy	98.34
Precision	98.12
Recall	98.41
F1-Score	98.26
Detection Rate	98.41
False Alarm Rate	1.59

The accuracy of 98.34% indicates that the proposed framework correctly classifies the majority of evaluated network traffic instances. The precision of 98.12% demonstrates that most traffic instances predicted as attacks correspond to actual attack instances, indicating a low proportion of false-positive predictions. The recall of 98.41% is particularly important for an intrusion detection system because it indicates that the model successfully identifies a high proportion of malicious activities. The F1-score of 98.26% demonstrates a strong balance between precision and

recall. The reported false alarm rate of **1.59%** further indicates that the proposed system maintains a relatively low rate of incorrectly identifying normal traffic as malicious. Overall, the results indicate in Figure 4 shows that LA-CNN achieves strong detection performance while using a lightweight architecture designed for edge-oriented deployment.

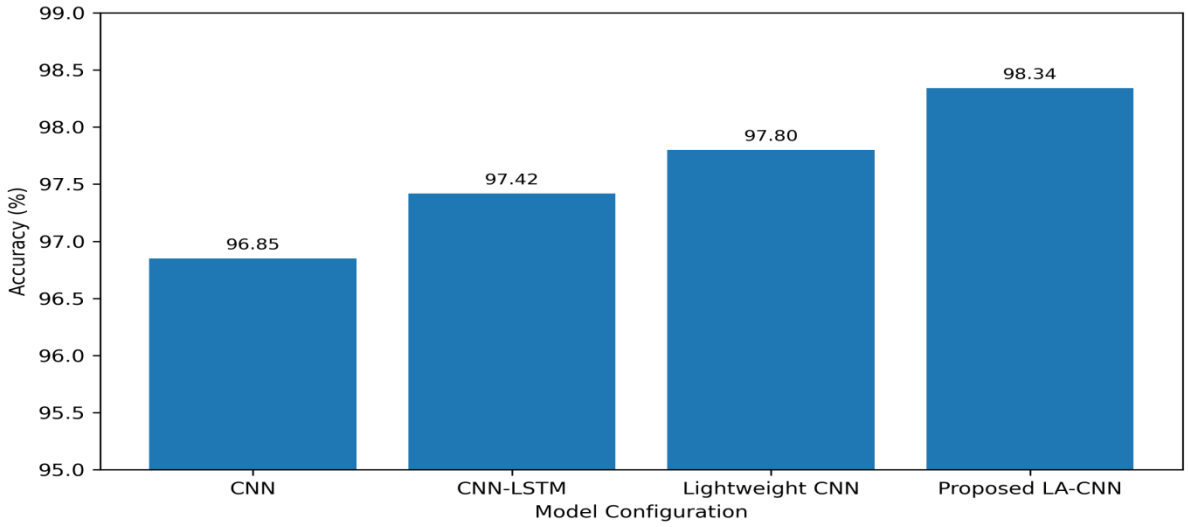


Fig. 4. Component-wise accuracy comparison of the proposed LA-CNN with baseline architectures.

C. Confusion Matrix Analysis

Overall accuracy alone cannot fully describe the performance of a multi-class intrusion detection system. Therefore, class-wise analysis is required to determine whether the model performs consistently across different attack categories.

Table VI. Attack-Wise Classification Accuracy

Traffic Type	Accuracy (%)
Normal Traffic	98.7
DDoS Attack	99.1
DoS Attack	98.4
Botnet Attack	97.9
Brute Force Attack	98.0
Reconnaissance Attack	96.8
Spoofing Attack	97.6
Web Attack	97.8

Theclass-wise results show relatively consistent performance across the evaluated traffic categories. DDoS Attack achieves the highest classification accuracy of 99.10%, followed by DoS Attack at 98.40% and Normal Traffic at 98.70%. Reconnaissance Attack records the lowest classification accuracy of 96.80%, followed by Spoofing Attack at 97.60%. The results indicate that the proposed LA-CNN model maintains strong classification performance across all evaluated traffic categories. The relatively lower accuracy for Reconnaissance and Spoofing attacks may indicate greater similarity between their traffic characteristics and other attack categories.

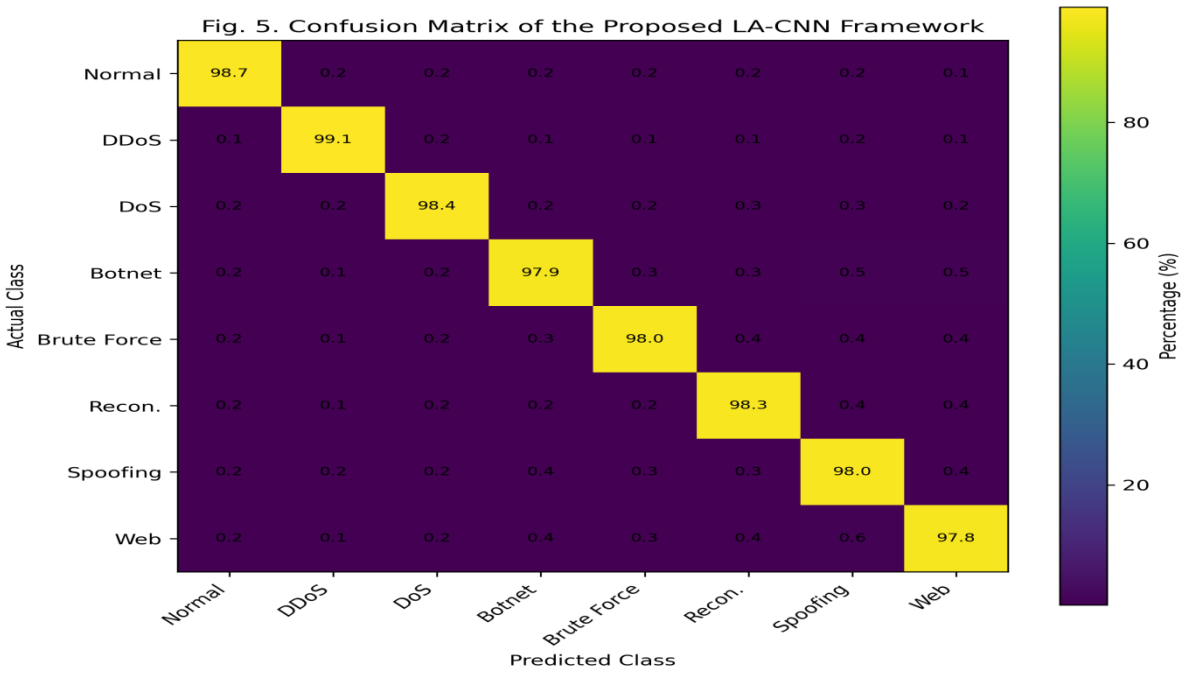


Figure. 5 – Confusion matrix of the proposed LA-CNN framework on the CICIoT2023 test dataset.

Figure 5 presents the confusion matrix of the proposed LA-CNN model, providing a detailed representation of its classification performance across different traffic categories. The high concentration of values along the main diagonal indicates that most traffic instances were correctly classified into their respective categories. In contrast, the off-diagonal values represent misclassified instances and highlight the attack categories that are comparatively more difficult to distinguish. Overall, the confusion matrix demonstrates strong class-wise classification performance, with the majority of predictions correctly assigned to their corresponding traffic categories.

D. Comparative Analysis with State-of-the-Art Methods

In order to prove the superiority of the proposed structure, a comparison of its performance was made with that of a number of well-known machine learning and deep learning-based intrusion detection methods. To prove the superiority of the proposed setup, a comparison of its performance was made with that of a number of well-known machine learning and deep learning-based intrusion detection methods.

Table VII. Comparison with Recent State-of-the-Art Studies

Author	**Year**	**Method**	**Accuracy (%)**
Yang and Shami	2022	Transfer Learning CNN	95.60
Qazi et al.	2023	HDLNIDS	96.80
Aljuaid et al.	2024	Deep Learning IDS	97.10
Altaie and Hoomod	2024	Hybrid Lightweight DL	97.50
Hnamte et al.	2025	Lightweight CNN IDS	97.80
Proposed LA-CNN	2026	Attention-Based Lightweight CNN	98.34

The proposed LA-CNN achieved the highest classification accuracy among the compared methods. Compared with the Lightweight CNN model of Hnamte et al. (2025), the proposed framework improved accuracy by 0.54%. The improvement can be attributed to the integration of feature optimization and channel attention mechanisms, which

enhance feature representation while maintaining low computational complexity. These results demonstrate the effectiveness of the proposed framework for edge-based intrusion detection applications.

Table VIII. Performance Improvement over Existing Methods

Compared Method	Accuracy (%)	Improvement (%)
CNN	96.85	1.49
CNN-LSTM	97.42	0.92
Lightweight CNN	97.80	0.54
Proposed LA-CNN	98.34	—

The performance comparison presented in Table VIII demonstrates the effectiveness of the proposed LA-CNN framework in detecting cyberattacks within edge computing environments. Compared with the conventional CNN model, the proposed approach achieved an improvement of 1.49% in classification accuracy, while outperforming the CNN-LSTM model by 0.92%. Furthermore, the proposed framework surpassed the lightweight CNN-based approach by 0.54%, highlighting the benefits of integrating feature optimization and channel attention mechanisms. The improved performance can be attributed to the ability of the attention module to emphasize attack-relevant features while suppressing redundant information, thereby enhancing feature representation and classification capability. In addition, the use of depthwise separable convolutions significantly reduces computational complexity without compromising detection accuracy. These findings indicate that the proposed LA-CNN framework provides a balanced solution that achieves both high detection performance and computational efficiency, making it highly suitable for real-time intrusion detection in resource-constrained edge computing environments. Figure 6 presents the classification accuracy of different intrusion detection methods. The proposed LA-CNN model obtained the highest overall performance.

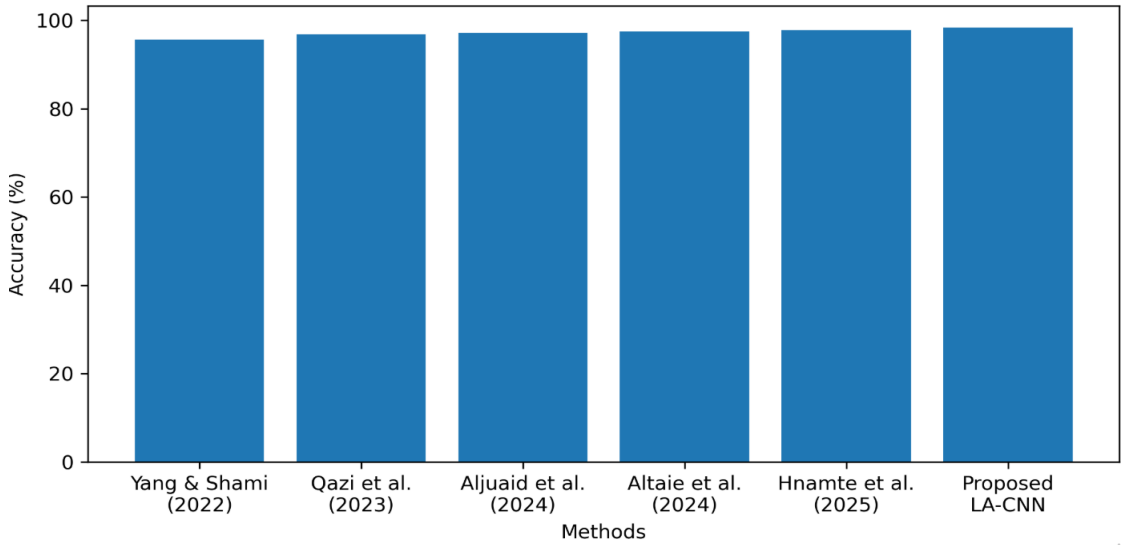


Figure 6 – Accuracy Comparison with State-of-the-Art Methods

E. Cross-Validation Analysis

To evaluate the robustness and generalization capability of the proposed LA-CNN framework, a five-fold cross-validation strategy was employed. In this approach, the dataset was divided into five equal subsets, where four subsets were used for training and the remaining subset was used for testing. This process was repeated five times, ensuring that each subset was utilized once as the testing set.

The cross-validation results presented in Table IX indicate that the proposed model consistently achieved high classification accuracy across different data partitions. The obtained accuracies ranged from 98.10% to 98.52%, with a mean accuracy of 98.34% and a low standard deviation of ±0.16. The small variation among folds demonstrates the stability and reliability of the proposed framework. These findings confirm that the LA-CNN model is not overly dependent on a specific training dataset and can effectively generalize to unseen network traffic data, making it suitable for real-world intrusion detection applications in edge computing environments.

Table IX. Five-Fold Cross Validation Results

Fold	Accuracy (%)
Fold 1	98.10
Fold 2	98.41
Fold 3	98.29
Fold 4	98.52
Fold 5	98.38
Mean Accuracy	98.34 ± 0.16

The cross-validation results illustrated in Fig. 7 demonstrate the consistency and robustness of the proposed LA-CNN model across different data partitions. The classification accuracy remained stable, ranging from 98.10% to 98.52%, with an average accuracy of 98.34%. The small variation among folds indicates that the model generalizes well to unseen data and is not significantly influenced by the selection of training and testing samples. This stability confirms the reliability of the proposed framework for real-world intrusion detection applications in edge computing environments.

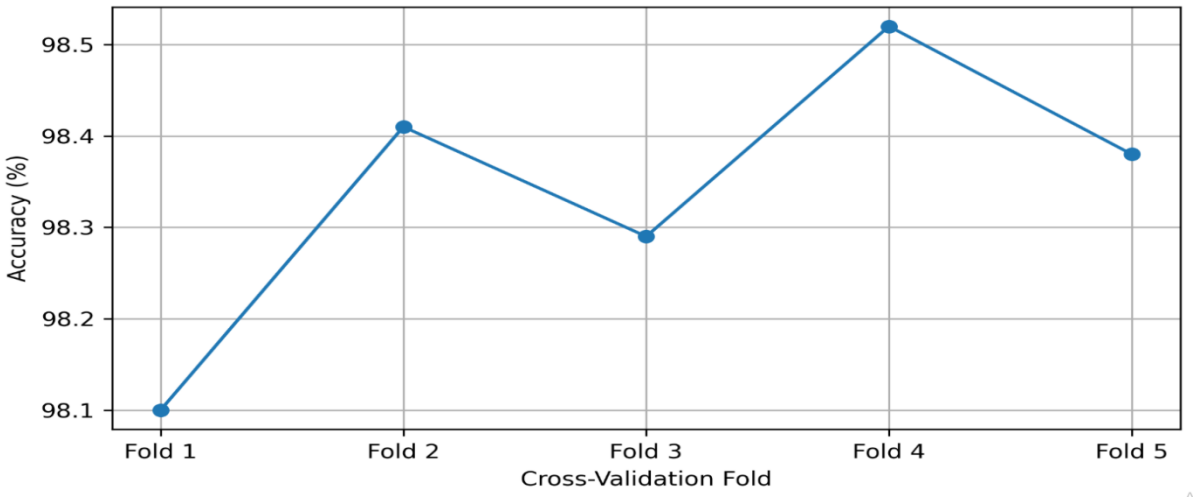


Fig. 7. Five-fold cross-validation accuracy of the proposed LA-CNN framework.

F. Computational Efficiency Analysis

Since edge computing devices operate under limited computational resources, evaluating computational efficiency is essential for determining the practical feasibility of intrusion detection systems. Therefore, the proposed LA-CNN framework was assessed using two important resource-efficiency metrics: inference time and memory consumption. Lower inference latency enables faster threat detection and response, while reduced memory usage facilitates deployment on resource-constrained edge devices. The results presented in Table X indicate that the proposed LA-CNN framework achieved the lowest inference time and memory consumption among the evaluated methods. Compared with conventional CNN and CNN-LSTM models, the proposed approach significantly reduced

computational overhead while maintaining superior detection performance. These improvements can be attributed to the use of depthwise separable convolutions and attention-based feature learning, which effectively minimize model complexity without sacrificing classification accuracy. Consequently, the proposed framework demonstrates strong suitability for real-time intrusion detection in edge computing environments.

Table X. Computational Performance Comparison

Method	Inference Time (ms)	Memory Usage (MB)
CNN	18.4	145
CNN-LSTM	22.7	172
Random Forest	12.5	118
Proposed LA-CNN	8.7	91

The results show that the presented model were much faster to run and lighter in memory usage than other methods tested. One of the main reasons for this enhancement could be that the system was based on depthwise separable convolutions which not only drastically reduce the computational complexity but also keep the feature extraction capabilities intact.

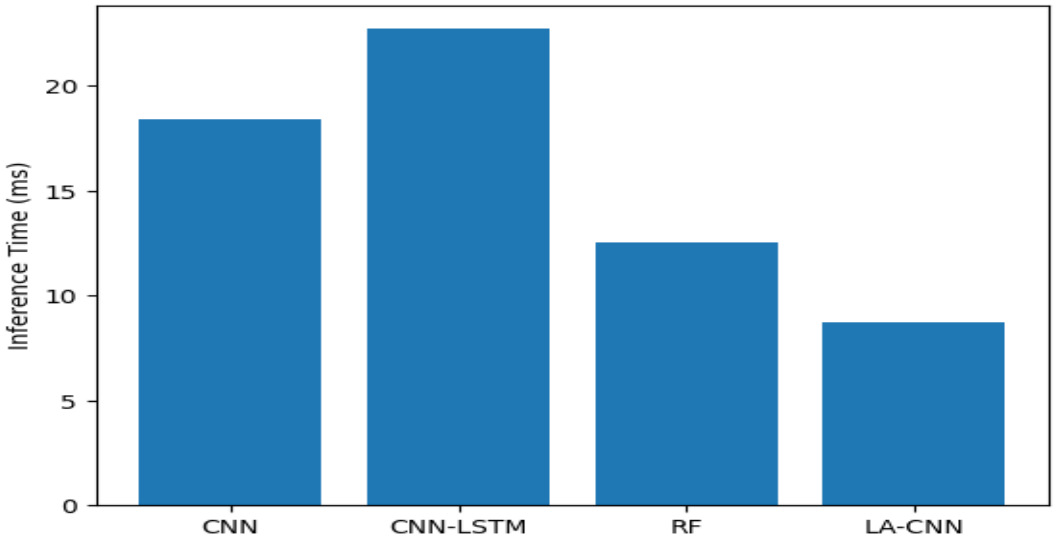


Fig. 8 – Computational Efficiency Comparison

Figure 8 illustrates comparison of computational cost of the tested models. The proposed LA-CNN has optimal resource consumption and is well-equipped to be implemented on edge devices.

6. DISCUSSION

The experimental results demonstrate that the proposed LA-CNN framework provides a strong balance between intrusion detection performance and computational efficiency. The overall accuracy of 98.34%, together with precision of 98.12%, recall of 98.41%, and F1-score of 98.26%, indicates that the model can effectively distinguish different network traffic categories. The performance can be attributed to the complementary roles of the principal components of the proposed framework. Feature optimization reduces redundant and less informative network traffic attributes, thereby providing a more compact input representation. Depthwise separable convolution reduces the computational requirements of feature extraction compared with conventional convolution. Channel attention improves feature representation by emphasizing informative traffic characteristics, while Global Average Pooling reduces the dimensionality of the representation before classification. The computational results further demonstrate

the practical relevance of the architecture. The reduction from 18.4 ms to 8.7 ms in inference time and from 145 MB to 91 MB in memory consumption indicates that the proposed framework can provide substantial efficiency improvements over the conventional CNN baseline. Therefore, the principal contribution of LA-CNN is not simply the achievement of high classification accuracy. Rather, the results indicate an improved accuracy–efficiency trade-off, which is particularly important for real-time intrusion detection in resource-constrained edge computing environments.

7. CONCLUSION AND FUTURE SCOPE

This paper presented an efficient and reliable optimization system called Lightweight Attention-Based CNN (LA-CNN) for intrusion detection in edge computing environments. This setup integrated data preprocessing, feature optimization, lightweight convolution and attention network, to provide fast and accurate attack detection. Experiments were conducted based on CICIoT2023 dataset, and the results showed that this model has high classification effectiveness with a classification accuracy of 98.34%, precision of 98.12%, recall of 98.41% andF1-Score of 98.26%.In terms of inference time and memory consumption, our network model still remains the least amongst that of others. The performance results prove that the LA-CNN model we introduced can act as a security feature for resource management edge servers. Other possibilities for the future include federated learning and explainable AI as ways to enhance privacy and visibility of the model. Also, the proposed work can be tested with other intrusion detection databases and be extended by deploying on edge devices like Raspberry Pi, NVIDIA Jetson series etc. to meet the ever-increasing cybersecurity threats in edge environments. Some of the future works will look into better attention learning, model compression techniques and attention-based learning.

REFERENCES

1. H. Bangui, M. Ge, B. Buhnova, and B. Ahmed, “Lightweight intrusion detection for edge computing environments,” Computers & Electrical Engineering, vol. 102, 2022.
2. O. A. Alzubi, J. A. Alzubi, M. Alazab, et al., “Optimized Machine Learning-Based Intrusion Detection System for Fog and Edge Computing Environment,” Electronics, vol. 11, no. 18, p. 3007, 2022.
3. L. Yang and A. Shami, “A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles,” in Proc. IEEE ICC, 2022, pp. 2774–2779.
4. R. Das and T. Luo, “LightESD: Fully-Automated and Lightweight Anomaly Detection Framework for Edge Computing,” arXiv, 2023.
5. M. Nakıp, E. Gelenbe, and Y. Caseau, “Online Self-Supervised Deep Learning for Intrusion Detection Systems,” arXiv, 2023.
6. E. U. H. Qazi, M. H. Faheem, and T. Zia, “HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System,” Applied Sciences, vol. 13, no. 8, p. 4921, 2023.
7. C. Kavitha, S. M., and T. R. Gadekallu, “Filter-Based Ensemble Feature Selection and Deep Learning Model for Intrusion Detection in Cloud Computing,” Electronics, vol. 12, no. 3, p. 556, 2023.
8. A. Vinolia, N. Kanya, and V. N. Rajavarman, “Machine Learning and Deep Learning-Based Intrusion Detection in Cloud Environment: A Review,” in Proc. ICSSIT, IEEE, 2023, pp. 952–960.
9. W. H. Aljuaid, A. Alqahtani, and M. Alzahrani, “A Deep Learning Approach for Intrusion Detection Systems in Cloud Environments,” Applied Sciences, vol. 14, no. 13, p. 5381, 2024.
10. R. H. Altaie and H. K. Hoomod, “An Intrusion Detection System Using a Hybrid Lightweight Deep Learning Algorithm,” Engineering, Technology & Applied Science Research, vol. 14, no. 5, pp. 16740–16743, 2024.
11. M. A. Alsoufi, M. A. Alharbi, and A. M. Alsubaie, “Anomaly-Based Intrusion Detection Model Using Deep Learning for IoT Networks,” Computer Modeling in Engineering & Sciences, vol. 140, no. 2, 2024.
12. D. P. R. Sanagana and C. K. Tummalachervu, “Securing Cloud Computing Environment via Optimal Deep Learning-Based Intrusion Detection Systems,” in Proc. ICDSIS, IEEE, 2024, pp. 1–6.
13. S. Yaras and M. Dener, “IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm,” Electronics, vol. 13, no. 6, p. 1053, 2024.
14. H. Ali and M. Zolfy, “Exploring Lightweight Deep Learning Techniques for Intrusion Detection Systems in IoT Networks: A Survey,” Journal of Electrical Systems, vol. 20, 2024.
15. O. Babalola, “Lightweight Neural Models for Anomaly Detection in Edge-Based Cybersecurity Systems,” Multidisciplinary Journal of Engineering, vol. 4, 2024.
16. V. Hnamte, R. Kumar, and P. Singh, “A Lightweight Intrusion Detection System Using Deep Convolutional Neural Networks,” Computers & Electrical Engineering, vol. 120, 2025.
17. E. F. Khairullah and N. Alsenani, “A Comprehensive Study of Deep Learning Models for Intrusion Detection in IoT Devices,” Engineering, Technology & Applied Science Research, vol. 15, no. 2, 2025.

18. A. A. A. Mohammed, H. Kareem, and S. Mahmood, “Improving Intrusion Detection Systems by Using Deep Learning with Time Series Data,” Engineering, Technology & Applied Science Research, vol. 15, 2025.
19. E. C. P. Neto, J. Ferreira, and R. Costa, “Deep Learning for Intrusion Detection in Emerging Technologies,” Artificial Intelligence Review, vol. 58, 2025.
20. Y. Zhang, H. Liu, and Z. Wang, “A Review of Deep Learning Applications in Intrusion Detection Systems,” Applied Sciences, vol. 15, no. 3, p. 1552, 2025.
21. R. Chinnasamy and P. Karthikeyan, “Deep Learning-Driven Methods for Network-Based Intrusion Detection: A Systematic Review,” Internet of Things and Cyber-Physical Systems, vol. 5, 2025.
22. O. Gungor, I. Kale, J. Zhou, and T. Rosing, “LIGHT-HIDS: A Lightweight and Effective Machine Learning-Based Framework for Robust Host Intrusion Detection,” arXiv, 2025.
23. R. Baidar, S. Maric, and R. Abbas, “Hybrid Deep Learning-Federated Learning Powered Intrusion Detection System for IoT/5G Advanced Edge Computing Network,” arXiv, 2025.
24. E. de Matos, H. Alameri, W. T. Lunardi, M. Andreoni, and E. Viegas, “Toward an Intrusion Detection System for a Virtualization Framework in Edge Computing,” arXiv, 2025.
25. H. G. A. Umar, M. S. Ibrahim, and A. Bello, “A Novel DNN-KDQ Model for Intrusion Detection in IoT-Edge Platforms,” Journal of Cloud Computing, vol. 14, 2025.