SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I7P105

Volume 13 Issue 7, 62-76, July 2026

Original Article

An Enhanced Deep Learning Approach for IoT Botnet Attack Detection

Vinaykumar N Soni[1], Ashwinikumar Jha[2]

[1]*Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.*
[2]*Department of Artificial Intelligence & Data Science, Parul Institute of Engineering and Technology, Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.*

[1]*Corresponding Author : vinaysonilive@gmail.com*

Received: 12 May 2026 Revised: 11 June 2026 Accepted: 30 June 2026 Published: 29 July 2026

Abstract *- The increasing development of the Internet of Things has led to several security threats and vulnerabilities associated with interconnected networks and, thus, makes them suitable targets for DDoS attacks. The increased dimensionality of traffic in the context of the Internet of Things makes IDS hard because deep learning models running independently are unable to adequately address the complexity associated with the interaction between network attributes. In this context, this research suggests a new architecture based on deep learning methods for binary classifications and featuring hybridization of multi-layer 1D-CNNs with LSTM networks. To reduce computational resources needed for training on IoT-based datasets, the ANOVA F-test is implemented to choose the key features before training. The proposed architecture includes dual convolutional layers having 64 and 128 filters, respectively, that allow to analyze the hierarchical spatial structures. Next, the data from convolutional layers is passed to the LSTM layer that analyzes high-order structural dependencies within feature maps generated at earlier stages. The proposed model was trained using the CICIoT2023 benchmark and showed superior performance compared to a standalone architecture featuring either CNNs or LSTM networks. Across five independent trials, the model achieved a mean accuracy of 98.72% ± 0.06%, with a precision of 98.91% and a recall of 98.65%. Furthermore, the proposed hybrid architecture shows high computational efficiency, achieving a remarkably low inference time of 0.15 ms per sample and a compact model footprint of 0.30 MB. These metrics confirm the model's suitability for deployment on resource-constrained IoT edge devices, providing a robust balance between high detection accuracy and low architectural complexity. These findings confirm that the constructive collaboration between spatial and structural feature modeling provides a robust, stable, and scalable solution for mitigating DDoS threats in modern IoT ecosystems.*

Keywords *- IoT security, Botnet detection, CNN-LSTM, CICIoT2023, ANOVA F-test.*

1. Introduction

Internet of Things (IoT) is changing the way many sectors work, like healthcare, transport, and industrial automation, as it will enable trillions of "thing" interactions on a single global network [1]. Currently, over 29 trillion active IoT devices will be connected by 2025 [2]. This will evolve the creation of a massive digital community that is more beneficial and accessible than ever before. Nevertheless, severe security vulnerabilities have been identified as it has quickly flourished [3]. Because of their dependence on various communication protocols, limited computing power, and heterogeneity, IoT devices are extremely vulnerable to security attacks, especially in settings where poor authentication methods and default passwords are common [4].

Botnet attacks, for example, Mirai, which infect thousands of devices to plan massive Distributed Denial of Service (DDoS) attacks, are considered a primary threat to these systems [5]. As they tend to disregard new "zero-day" attack behaviors, conventional network protection tools such as rule- and signature-based Intrusion Detection Systems (IDS) are slowly drifting away in coping with these sophisticated and fast-evolving threats [6]. Henceforth, efficient and agile security platforms with the capability of real-time threat monitoring and mitigation within dynamic IoT networks are a critical necessity [1, 7, 8].

In the case of intrusion detection in IoT networks, Decision Trees (DT), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) are some of the machine learning techniques that have been used comprehensively [9]. However, these machine learning techniques get stuck in a situation where a lot of feature engineering is needed, and do not understand sequential as well as high-dimensional data. This led to the emergence of Deep Learning techniques, which can automatically extract hierarchical patterns from complex

traffic environments. Convolutional Neural Networks (CNNs) have shown exceptional efficacy in finding spatial hierarchies within network features. In parallel, LSTM networks serve as non-linear aggregators of features. Although LSTMs are generally applied to time-series data, in the proposed tabular network, the LSTM serves as a dynamic feature aggregator. Following the process of 1D-CNN learning local spatial dependencies of the sorted ANOVA features, the LSTM uses its gate functions to assign weightage to these feature maps, enabling the model to understand high-level non-linear interactions across all features instead of time points.

Despite high success rates in deep learning models, the computation problem continues to be the main issue in IoT edge nodes due to resource constraints [10]. High-dimensional data leads to long training times, which makes them less suitable for real-time operations needed for network security purposes [11]. In essence, the basic challenge in this area lies in the following trade-off: traditional IDSs are faced with the choice between deep neural architectures that guarantee high accuracy levels but resource-limited edge devices and simple reduction methods that offer low latencies without considering important security functions. This is a major research gap in existing literature, since there is no lightweight mathematical method that enables feature pruning while ensuring the preservation of spatial-temporal characteristics needed for deep neural networks. Therefore, solving this issue would require an efficient method of dimensionality reduction. Despite existing methods using basic correlation models or more complex Auto encoders [12], the ANOVA F-test becomes a more effective way since it reduces unnecessary data dimensions for the best mathematical discrimination.

Based on a targeted subset of the CICIoT2023 dataset—currently one of the most realistic sources of IoT traffic—this research proposes a streamlined, highly efficient hybrid CNN-LSTM framework perfected for binary classification (Benign vs. DDoS). By rigorously filtering the input space with the ANOVA F-test, this model drastically reduces data input complexity. The fundamental contribution of this research is a controlled, head-to-head performance assessment proving how intelligent statistical feature selection combined with a lightweight hybrid architecture provides the best trade-off between robust classification and the low overhead costs needed in modern IoT frameworks.

1.1. Contribution of this Research can be Summarized as

- Lightweight Hybrid Model Design: Creating an innovative hybrid approach through the fusion of 1D-CNN and LSTM models to allow interactions between the local information and the dependency of structural data in tables.
- Feature Selection via Statistical Processing: Using statistical pre-processing methods, such as the F-test-based ANOVA method, for finding the top 20 discriminant features to minimize processing costs.
- Model Comparisons with Current Techniques: Performing comprehensive experiments for the comparison of the proposed model with CNN, LSTM models, and machine learning models under the same conditions.
- Maximizing Model Efficiency: Maximizing the efficiency of the proposed model on binary classification tasks, especially at high accuracy and precision for the CICIoT2023 dataset.

The paper has been arranged in four major parts. Information about the literature review and research gaps, as well as motivation for undertaking the study, would appear in Section II. Under the methodology section, the discussion of the research method involves the elaboration of the dataset being used, the preprocessing step, feature extraction techniques, and the architecture of the CNN-LSTM model being built. The discussion of results will take place in Section IV, in relation to the analysis of the model in performance. The conclusion will be provided in Section V.

2. Related Work

To protect contemporary networks from hostile cyberattacks, Intrusion detection systems are essential. AI and DL techniques have become more important for strong network defenses over the past ten years, since conventional signature-based IDS solutions have found it challenging to keep up with changing attack trajectories [13, 14].

Deep Learning has brought about improvements in intrusion detection through automation in the extraction of features in network traffic. According to Erskine [5], a Hybrid DLMIDPSM model that involved ANN, CNN, and RNN was developed, resulting in an accuracy rate of 99%, although this approach did not have a solid topology-based prevention system.

Addressing common research gaps like class imbalance and lack of feature reduction, Zaedi et al. [15] proved that DL architectures (specifically LSTM and 1D CNN) outperformed traditional ML baselines on the CICIoT2023 dataset. Standard DL models often disregard the computational overhead that hinders real-time response on edge nodes. Similarly, Gueriani et al. [16] addressed poor model generalization by proposing a CNN–LSTM hybrid that effectively captures spatial–structural dependencies feature across multiple datasets. Future research aims to scale IoT datasets and enhance real-time prevention.

Despite this, even though the advanced algorithms used by Alashjaee et al. [17] and Naeem et al. [18], which focused on incorporating the CNN-LSTM model with attention mechanism that allowed highlighting of important features as well as addressing class imbalance issues in order to achieve high levels of accuracy, as much as 97.5%, often operate in opaque manner and lack the necessary level of transparency

required for security professionals to understand automated decisions. Also, while several studies focused mainly on certain types of attacks, such as the one carried out by Tadhani et al. [19], which yielded almost similar results in detecting the SQL injection and cross-site scripting attacks using the ensemble decision fusion approach, Alashjaee et al. [17] and Naeem et al. [18] managed to achieve accuracies of up to 97.5%. Afraji et al. [8] also developed a parallel CNN-LSTM-GRU fusion algorithm to overcome convergence issues and redundant calculations of earlier methods.

In order to achieve robustness when applied to challenging data such as the CICIoT2023 and Bot-IoT datasets, the combination of CNN-BiLSTM with Random Forest and weighted voting gave 100% accuracy by A. Waqas et al. [11], while Hizal et al. [21] proposed a two-stage approach that used a DNN-CNN-LSTM model for classification into 12 classes. In addition, other researchers such as Jony et al. [22], Susilo et al. [7], and Ain et al. [23] focused on scalability and dimensionality reduction using Autoencoder and Spark. Overall, these papers steer cybersecurity towards a more adaptable and spatiotemporal modeling framework.

Moreover, even though Yaras and Dener [24] employed PySpark on big data, their sequential approach to the hybrid model might be faced with sequential dependency limitations that the suggested parallel approach looks to address.

Unlike the studies that prioritize accuracy over efficiency, this work introduces an ANOVA F-test-driven feature selection pipeline integrated with a hierarchical 1D CNN-LSTM architecture—employing a specific 64-to-128 filter progression— to specifically address the trade-off between high-dimensional feature extraction and computational latency in dynamic IoT networks.

Table 1 provides a structural comparison between the theoretical limitations of existing methodologies and the specific enhancements introduced in this study.

Table 1. Theoretical paradigm and architectural comparison

Methodology / Architecture	Primary Features	Structural Limitations & Research Gaps	Proposed Solution
Traditional ML (e.g., Random Forest, SVM)	Low computational cost; effective for simple binary signatures.	Does not extract deep spatial patterns; it is highly susceptible to evolving zero-day botnets.	Automated Deep Feature Extraction
Standalone CNN Models	Superior spatial feature extraction from packet headers.	Treats each network packet or time window as an independent image-like grid; completely blind to the sequential, historical context of a botnet lifecycle.	Integrates an LSTM layer to store and process the long-term temporal evolution of incoming traffic sequences.
Standalone LSTM Models	Captures sequential behavior and long-term network states.	Highly vulnerable to noisy, unoptimized, and redundant high-dimensional features; suffers from long training/inference cycles on raw traffic.	Employs 1D-CNN layers to filter out noise and compress feature dimensions before sequential analysis, accelerating execution time.
Proposed Hybrid ANOVA-CNN-LSTM	Integrated Statistical Filtering & Dual-Domain Analysis.	-	The Current Work uses the ANOVA F-test for rigorous statistical feature pruning, combines a 1D-CNN for spatial filtering, and appends an LSTM layer for temporal sequence tracking.

2.1. Problem Definition and Research Motivation

However, despite all these improvements achieved in deep learning algorithms for IoT intrusion detection, there is a major issue left unsolved, which includes the link between accuracy and computational complexity. Although hybrid and ensemble-based models are highly efficient, their implementation is hard due to their complicated nature on edge IoT devices, while traditional dimensionality reduction approaches do not guarantee feature interpretability.

The fundamental novelty is based on the idea of sequential processing, whereby instead of directly applying raw data into the RNN model, a 1D-CNN filter is used to automatically detect spatial features and strip out any noise from the dataset.

Subsequently, an enhanced LSTM module is used to capture behavioral characteristics of the traffic in sequence form. In addition, as a measure against redundancy of the feature set which usually characterizes large-scale IoT systems, an automated feature selection approach using ANOVA F-test is applied before executing the proposed deep learning model.

3. Methodology

3.1. Dataset Description

This CICIoT2023 is a new security dataset that was published by the CIC-IoT Lab from the Canadian Institute for Cybersecurity in 2023 [25]. CICIoT2023 has benign traffic, along with 33 attacks grouped into 7 main categories: DDoS, DoS, Reconnaissance, Brute Force, Spoofing, Web-based Attacks, and Mirai.

Among these categories, DDoS attacks are the most common and useful class in the dataset, with 73% of records. Each one of them carries a different quantity of samples. A comprehensive explanation of the dataset is found in [25]. This study investigates DDoS attacks, especially focusing on their numerous types and how they differ from benign traffic.

The paper discusses twelve different forms of DDoS attacks, namely, ICMP Flood, UDP Flood, TCP Flood, PSHACK Flood, SYN Flood, RSTFIN Flood, synonym IP Flood, UDP Fragmentation, ACK Fragmentation, ICMP Fragmentation, HTTP Flood, and SlowLoris.

The spread of DDoS traffic is depicted in Figure 1. Due to limits in storage space and the enormous volume of accessible samples, this study uses a subset of the dataset, giving a representative sample size for investigation. Each data sample includes 47 properties, including a target variable that shows the traffic class. Rather than using all 47 features from the raw CICIoT2023 dataset, an ANOVA F-test run on the dataset, and only the Top 20 highest-ranking features are selected for model training.

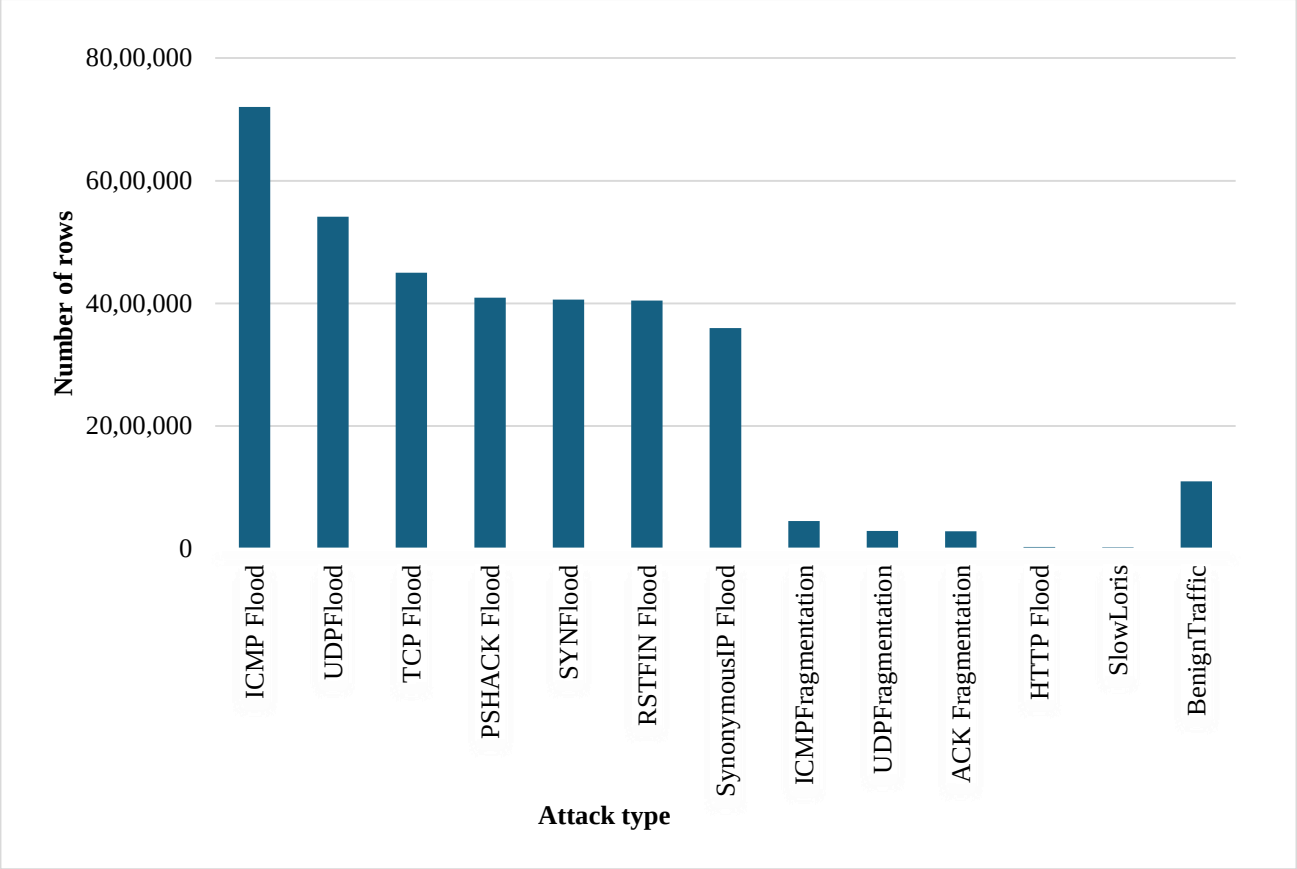

Fig. 1 Distribution of DDoS and benign samples in CICIoT2023

3.2. Data Pre-Processing

To make sure that the data is compatible, data processing is done carefully before adding raw data to the deep learning model. Pre-processing of data is very crucial in addressing the issues related to poor data quality, including encoding categories of textual labels into numbers. The entire process involved in achieving excellent quality data is shown in order in Figure 2 below.

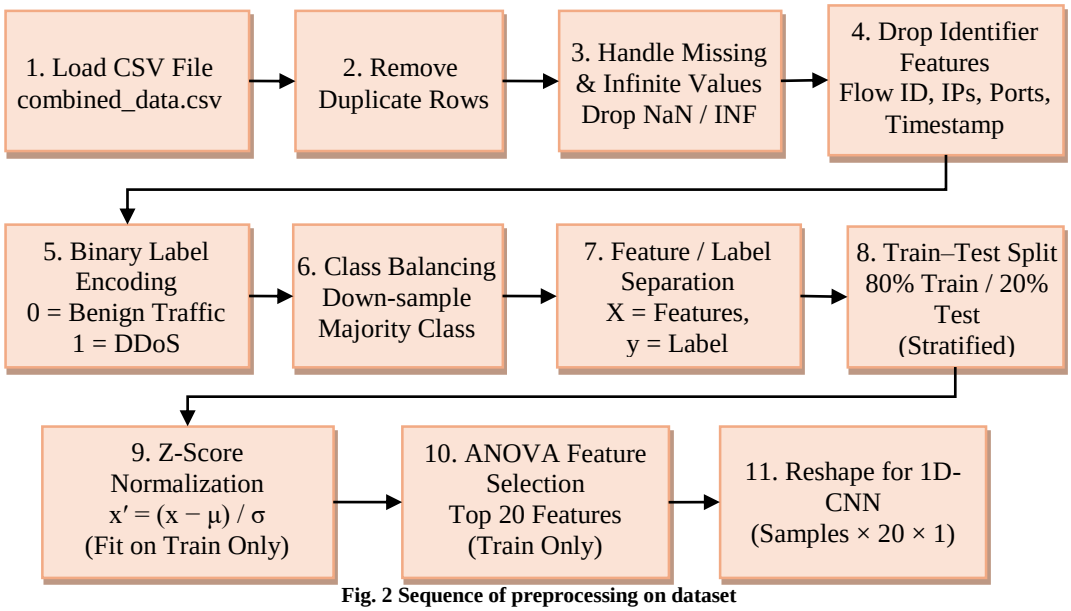

Fig. 2 Sequence of preprocessing on dataset

Firstly, the data set undergoes a process called pre-processing, where it is cleaned by dropping any duplicated records, removing all null rows, and dropping any identifiable metadata like IP addresses, ports, and timestamps. The non-numeric label is converted through label encoding.

Even though there exist methods such as SMOTE and class balancing that could mitigate this problem, in this research paper, the method of Random Under-Sampling (RUS) was consciously selected. The CICIoT2023 database is extremely extensive; thus, undersampling the majority of the DDoS class would considerably lower the computation costs and the time needed to train the algorithm. This approach meets the requirements of building an Intrusion Detection System (IDS) on an edge IoT system because the resulting algorithm is light but does not miss the variability needed for generalization purposes. To achieve statistical balance while being computationally efficient, the DDoS dataset has been downsampled.

As a result, we managed to lower the number of entries in the DDoS class to the level of 1,098,194 benign samples, creating a perfect 1:1 ratio. This is an important stage for evaluating the quality of future algorithms in both classes in terms of their accuracy and sensitivity. Although the Random Under-Sampling (RUS) method was used to create a perfectly balanced 1:1 sample (1,098,194 DDoS and 1,098,194 Benign samples) to avoid biases and training costs, the test data remained untouched. It guarantees that the model will be evaluated under realistic 'in-the-wild' IoT network conditions, thus ensuring the correct evaluation of its actual False Positive Rate (FPR). Lastly, the data was split into features (X) and attack labels (Y). To avoid any form of bias and guarantee the quality of the results while performing the experiment, at the very beginning, the data set was divided into a training set of 80% and a testing set of 20%. All the later steps of data processing and preprocessing, including normalization and feature selection, were done based on the training distribution parameters exclusively. Firstly, data was normalized using the Z-score standard scaler technique. Most importantly, the mean and standard deviation values were calculated only for the training partition of the data set. Afterward, the same training values were used to normalize the test partition, which excluded any possible impact of the test partition on the model during the training stage. As is clear from Equation (1), the Z-Score normalizes the features to have zero means and unit variance.

$$\boldsymbol{z} = \frac{(\boldsymbol{x}-\mu)}{\boldsymbol{\sigma}} \tag{1}$$

x is the original value, z is the normalized value, and μ and σ are the mean and standard deviation values, respectively.

Feature selection was executed using the ANOVA F-test to find the most discriminative variables for DDoS detection. To keep experimental integrity, the ANOVA scores were calculated using only the training samples. Based on these scores, the top 20 features were selected. The test set was then reduced to these same 20 features to keep consistency during evaluation. ANOVA works as a statistical filter that examines the link between input features and the target attack labels. For each characteristic, it asks a basic question: *does the mean value of this attribute fluctuate greatly between classes?* If yes, the feature obtains a high score. If not, it's often background noise. The ANOVA F-test Formula (2) is provided below.

$$F = \frac{MSB\ (Mean\ Square\ between\ groups)}{MSW\ (Mean\ Square\ within\ groups)} \tag{2}$$

Here, MSB (Variance between classes) investigates how much the average value of a feature changes between "Benign" and "Attack" traffic, and MSW (Variance within classes) measures the spread of values within the same category. A high F-value denotes an important trait. If the variance between "Attack" and "Benign" is much higher than the internal noise (MSW), the feature is a solid choice for a 1D-CNN + LSTM model. Instead of using all 47 features from the raw CICIoT2023 dataset, After ANOVA F-test was done on the dataset, just the Top 20 highest-ranking features are taken for model training.

Table 2. Top 20 selected features using ANOVA.

Rank	Feature Name	ANOVA Score (F-Value)
1	Duration	1.47E+06
2	rst_count	1.20E+06
3	Variance	1.04E+06
4	Header_Length	9.06E+05
5	urg_count	7.96E+05
6	Max	7.13E+05
7	HTTPS	6.33E+05
8	Std	6.15E+05
9	Radius	6.13E+05
10	Magnitude	4.63E+05
11	AVG	4.32E+05
12	Tot size	4.29E+05
13	Tot sum	4.03E+05
14	ack_flag_number	2.53E+05
15	Covariance	2.08E+05
16	Min	1.25E+05
17	syn_count	2.80E+04
18	syn_flag_number	1.54E+04
19	ICMP	1.54E+04
20	TCP	1.48E+04

As is well known, the ANOVA F-test is a univariate statistical tool since it deals with the analysis of variance of each attribute. As such, it does not have the ability to find any interactions between multiple attributes in which the correlation among them is extremely strong, even though they individually have medium predictive power. While multivariate models such as Recursive Feature Elimination and Wrapper Approach consider the relationship between multiple attributes, they need to spend enormous amounts of computation. Given the aim of creating efficient architecture for the constrained IoT, the ANOVA technique is chosen to avoid spending massive amounts of computation for preprocessing. To compensate for using an univariate approach, the following CNN and LSTM networks will be used as nonlinear feature aggregators. Table 2 shows the selected features using the ANOVA F-test; features are sorted using their F-value.

To make the design compatible with the hybrid architecture, the limited feature space was remodeled as a 3D tensor of dimensions (Samples, features, 1). Final Input Dimension to CNN-LSTM Model: (1686647, 20, 1). With such a setting, the one-dimensional convolutional layers function as automatic feature extractors to find local spatial structures and feature dependencies within the newly formed vector of inputs. The following layer of LSTM helps capture structural dependencies at a higher level from the newly obtained feature map. In contrast to the ordinary dense layers, the recurrent gates of an LSTM network can learn highly nonlinear relations between diverse groups of features.

3.3. Various Deep Learning Models

A major premise for ML and DL algorithms is developing a model using features gathered throughout the learning stages from the dataset upon which the model is trained for predictions on new observations [24]. Anomaly detection systems using artificial intelligence apply the same premise to allow the trained model to detect anomalies in the analysis.

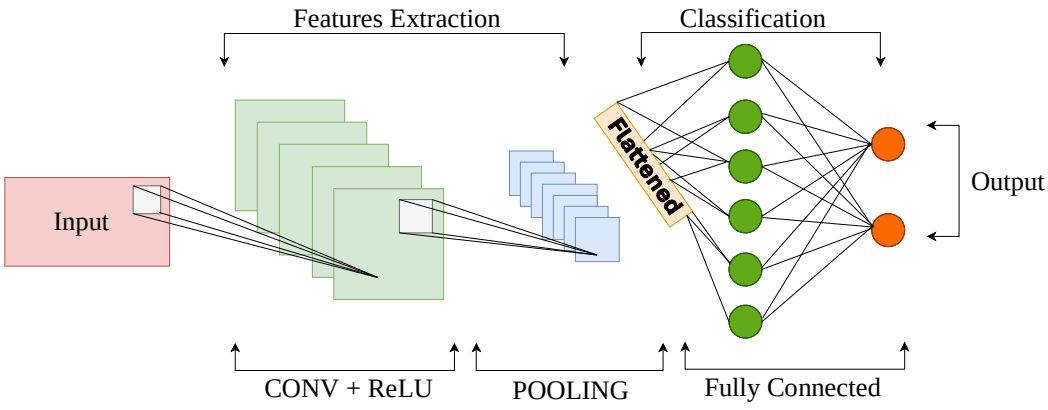

Fig. 3 1D-CNN architecture

3.3.1. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning system with the ability to automatically extract local and hierarchical features of input examples using convolutional filters for efficient pattern recognition and classification tasks. In the context of IoT IDS, the CNN algorithm is used for 1D CNNs for learning the spatial hierarchies directly from the raw data of the packet headers/flow attributes in the network [8]. The structure of a CNN is explained in Figure 3. The layers include the input layers, convolution layers, pooling layers, flattening layers, and completely connected layers.

The CNN model starts with the input layer, which is used for the network data in standard format, and then one or more one layers are added in the convolution layer. The convolution layer makes use of the trainable kernel to extract any local features or any anomalies in the structure and thus creates useful feature maps. To minimize the size of these maps, the use of pooling layers, such as MaxPooling, and then a flatten layer converts them into a one-dimensional vector.

Operations within layers can be defined using multiple mathematical equations. The equation to find the output feature map for Convolutional operations, which is considered the most significant operation where input features are converted to useful information via kernel k and bias b, is represented by the formula in Equation (3).

$$C_i = \sigma\left(\sum_{j=0}^{k-1} \omega_j \cdot x_{j+j} + b\right) \quad (3)$$

Where, x_{i+j} is the input value at the corresponding position, w_j stands for the learned filters, and σ is the ReLU activation function.

The activation function 'ReLU' is employed after the convolution layer to make it non-linear to provide a complex boundary between data samples with 'innocent' and 'malicious' classes.

$$ReLU(c_i)=max\ (0,\ c_i) \quad (4)$$

This is a particular behavior that is shown by the function, such that the output will be zero when the function takes a negative number as input, while the input number is positive and equals x, and the output for this case equals x. The characteristics of the output in the system change continuously from 0 to infinity.

The Max-Pooling layer reduces computational complexity and prevents over-fitting by performing the down-sampling procedure. This process results in the selection of only those activations that are highest in the selected window size. The final layer's outputs after passing several convolutional and pooling layers are flattened before they pass to the fully connected layers. Finally, for classification purposes, a Sigmoid or SoftMax function will be applied.

3.3.2. Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) network is a customized version of Recurrent Neural Networks (RNNs), which is created to overcome the vanishing gradient problem, thus helping the ability to learn long-term dependencies in data. In this framework, the LSTM acts as a dynamic feature-weighting system rather than a sequence processor. The complex gating mechanisms (Forget, Input, and Output gates) learn which extracted CNN feature maps to 'remember' (prioritize) and which to 'forget' (ignore). This allows the model to capture deep, non-linear interactions across the feature space that standard feed-forward neural networks (Dense layers) typically do not find. This would enable the network to distinguish between the transient network spike and the slowly increasing rate of the botnet attack, as in the slow-rate DDoS attack.

In our discussion concerning the emergence of LSTM architecture, we can now consider a highly sophisticated memory cell that regulates the information using three gates, namely, the input gate (i_t), output gate (o_t), and forget gate (f_t). As per the findings of Ingolfsson [27], it is evident that the primary function of the forget gate is to select the information that is supposed to be retained or discarded, depending upon the comparison between the current inputs and the previous state, whereas the input gate modifies the state of the cell with respect to the inclusion of new information.

$$f_t = \sigma\ (Wf \cdot [h_{t-1}, X_t] + b_f) \quad (5)$$

$$i_t = \sigma\ (Wi \cdot [h_{t-1}, X_t] + b_f i) \quad (6)$$

$$o_t = \sigma\ (Wo \cdot [h_{t-1}, X_t] + b_o) \quad (7)$$

$$\hat{C}_t = tanh\ (W_c \cdot [h_{t-1}, X_t] + b_c) \quad (8)$$

The current cell state is updated by combining the forgotten old state and the newly filtered candidate state [22].

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (9)$$

The final output (h_t) from the cell is then filtered with the internal cell state as [22]:

$$h_t = o_t \cdot tanh(c_t) \quad (10)$$

Each gate has weights and bias associated with it, like artificial neural networks. To allow the LSTM cell to learn from these weights through a variant of a gradient descent approach, the weight matrices are suitably amalgamated. The symbols used in these equations for the weight matrices and their accompanying biases are defined as: W_f, b_f, W_i, b_i, W_o, b_o, and W_C, bc.

Long-term memories are captured by LSTM models, thus enabling the understanding of the dynamics of the attack behavior, such as command-and-control communication, scanning, and burst traffic.

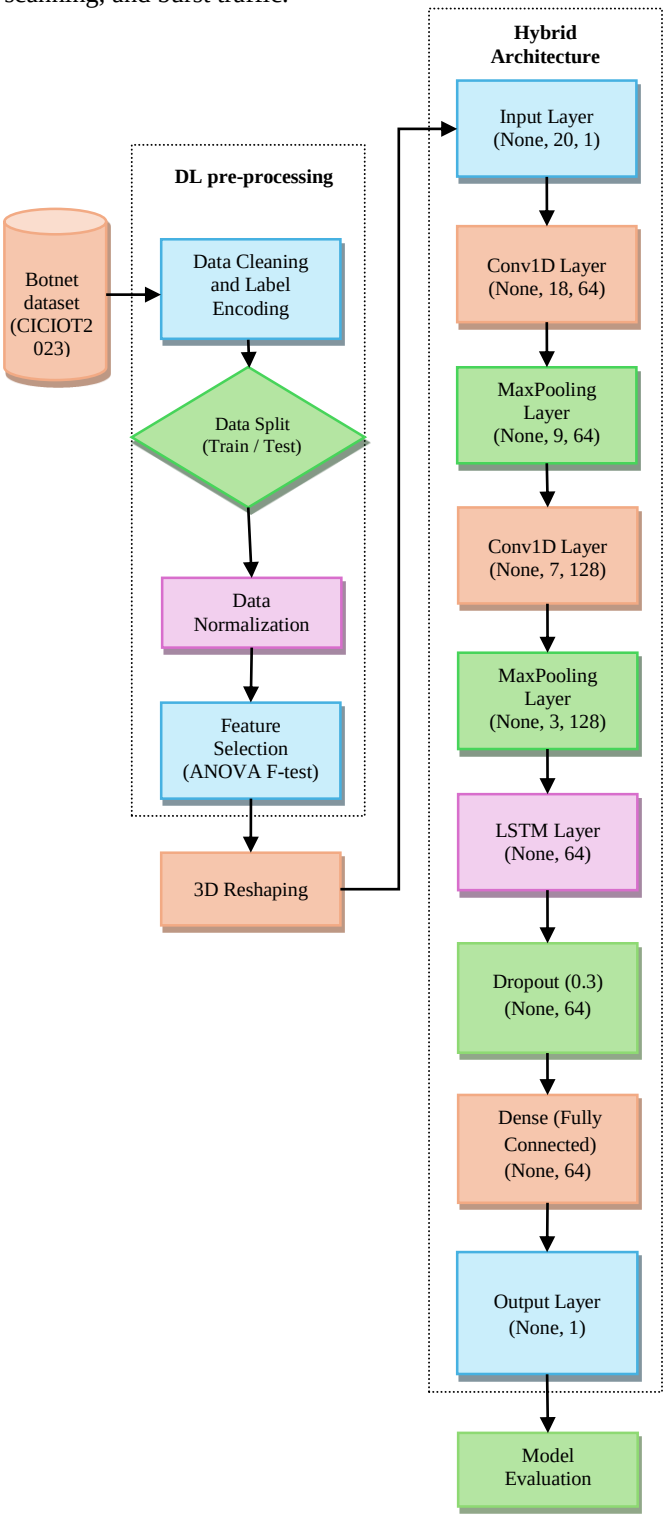

Fig. 4 Proposed model architecture

3.4. Proposed Hybrid CNN – LSTM Model

The novel approach exploits the advantages of CNN networks in terms of spatial features recognition, together with high-level dependencies modeling by LSTM networks. In this combination of the two approaches, the 1D-CNN network layers are applied to find discriminant local patterns in each of the attributes of the network flows. The obtained feature maps are further analyzed using the LSTM network, whose task is to model complex nonlinear dependencies and hierarchical connections between various feature groups. As a result, this approach allows finding of the complex nature of attacks performed against IoT networks with elevated levels of accuracy. Figure 4 shows the architecture of the proposed hybrid model.

Here, the CNN layers function as autoencoders, which extract the features from the data automatically. Conv1D filters slide across the feature vector and pick up the discriminatory patterns of the 'fingerprints' associated with the malicious network traffic. Through placing the ANOVA-selected features onto a 1-dimensional spatial grid, the model detects local interactions between neighboring features in the network statistics.

These high-level feature maps are fed to the LSTM layer. Instead of being modeled on sequence data in real-time, LSTM serves as an effective form of inductive bias and learns the higher-level dependencies between feature maps. Consequently, the model can recognize sophisticated and coordinated botnet structures which are missed by the feed-forward layers.

To increase generalization, a Dropout layer is used to prevent overfitting the model to the data in the training phase. At last, the Dense (or fully connected) layer combines these features and then feeds them to the Sigmoid-activated output layer for classification (Malicious/Benign). Here, the enhanced feature set must be converted from a two-dimensional tabular format into a three-dimensional structure. Here, the data is reshaped into a 3D format: (samples, features, channel).

The sequence of layers in the proposed model is defined as follows:

Input Layer

The input layer gets the 3D tensor reshaping (Samples, 20, 1) with network traffic attributes found using ANOVA on a fixed spatial grid.

Conv1D_1 Layer

In the first convolution layer, 64 filters are applied for detecting lower-level spatial characteristics from the input vector. The convolutional layer detects interactions and correlations in the form of mathematics between network traffic features, such as the relationship between header size and packet size.

MaxPooling1D_1 Layer

This is done through max pooling of the convolution operation, which reduces the dimensions of the feature map. This helps reduce computational overheads and keeps only the useful features while getting rid of unnecessary data.

Conv1D_2 Layer

The second convolution layer uses 128 filters. The conv layer learns abstract characteristics of feature maps after the earlier pooling process. It aids in finding mathematical differences between regular traffic flow and advanced attack signatures.

MaxPooling1D_2 Layer

Another round of max pooling is performed to further reduce dimensions. It avoids over-fitting.

LSTM Layer

The highly compressed spatial feature maps are now passed through an LSTM layer having 64 units. Unlike the time-series sequences, the LSTM behaves like an extremely powerful feature aggregator.

The use of the dynamic gating technique allows LSTM to alter the importance assigned to local feature clusters detected by CNN, thus helping the network concentrate on the essential features needed to recognize IoT attack patterns.

Dropout Layer

To resolve the issue of overfitting, the dropout layer with a dropout ratio of 0.3 is applied. This means that 30 percent of the neurons will remain inactive in the learning phase.

Fully Connected Dense Layer

The dense layer serves as a high-order feature aggregator. It takes advantage of the non-linear activation function to combine the dependency structures discovered by the LSTM layer and the spatial information captured by the CNN layers.

Output Layer

There is a one-node output layer using the Sigmoid activation function that simplifies complex feature representations into either 'Benign' or 'DDoS attacks'.

Model Evaluation

Model performance is critically analyzed through 20% unseen test dataset. Accuracy, Precision, Recall, F1-score, along with the confusion matrix approach, are used to measure the efficiency of the model.

This method involves the use of various levels of convolution layers to obtain hierarchical abstractions. Consequently, a highly compact representation of the state of the neural network is provided as input to the LSTM layer, which makes modeling complex relationships possible without being computationally expensive.

3.5. Model Evaluation

These models in Machine Learning or Deep Learning are assessed according to the performance using an evaluation metric or method. This is a critical stage in the entire process of modeling that tends to be more difficult. The purpose of the problem or task to be solved dictates which evaluation method to be applied. These purposes may involve both classification and regression analysis tasks.

Accuracy, precision, recall, and F1-score are some of the commonly used metrics that are employed to measure the performance of this proposed model in detecting attacks, as shown in Equations 11, 12, 13, and 14.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1\_score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14}$$

where “True Positive” (TP) refers to the number of times when the IDS detects an intruder in an exact manner; “True Negative” (TN) refers to the cases that are accurately detected by IDS as non-intruder traffic.

Conversely, “False Positive” (FP) refers to the situations when non-intruder traffic is erroneously identified as intruder traffic. “False Negatives” (FN) refer to situations when intruder traffic is not accurately detected by IDS. Also, when IDS has high F1 score values because of low FP and FN, it is considered effective.

4. Results and Discussion

4.1. Classification Performance

The designed hybrid CNN-LSTM neural network was implemented employing the TensorFlow and Keras libraries in Python programming language. Thanks to the employment of high-level APIs, the model was made efficient to work with high-dimensional Internet of Things traffic data. Specifically, the development process involved a powerful stack of Python scientific packages, namely Pandas, NumPy, and Scikit-learn, to perform data manipulation, numerical computing, and preprocessing, respectively.

To ensure seamless collaboration between software tools, experiments were performed inside the Google Colab Pro cloud platform. In turn, the employment of the GPU NVIDIA Tesla T4 facilitated fast and effective training and allowed for rapid tuning of hyperparameters to find the best ones. Table 3 stands for the hyperparameter settings for the experiment done in this research.

Table 3. Hyperparameter settings for proposed model

Hyper-parameter	Setting	Justification
Convolution filters	64,128	Balance between memory efficiency and gradient stability.
Kernel size of the filter	3	Local context captured by the CNN at each iteration.
Batch Size	512	Maximize GPU memory efficiency and stabilize gradient descent.
Learning Rate	0.001	Ensure fast convergence without causing instability during training.
Optimizer	Adam	Fast processing of sparse gradients found in Internet of Things data.
Loss Function	Binary Cross-Entropy	Commonly used error metric for binary classification problems.
Activation Function (CNN)	ReLU	Non-linear activation function for efficient spatial feature mapping.
Activation function (LSTM)	Tanh	Standard gate activation function that prevents gradient explosion.
Classification function	Sigmoid	Convert predictions to probabilities for Binary Classification.
Dropout Rate	0.3	Randomly deactivate 30% of the network's nodes to avoid overfitting.
Train/Test Split	80% / 20%	Sufficient training dataset size with room left for validation.
Random Seed	42	Ensure deterministic behavior when splitting the data into training and validation sets, as well as initializing the weights.
Epochs	25	Prevents over-fitting if the validation loss starts plateauing.
Training Shuffling	Enabled (Per Epoch)	Prevent models from recognizing patterns in sample order to generalize across multiple training runs.

The solution to address the problem of imbalance that is present in the CICIoT2023 dataset was through random under-sampling. The balanced sample size obtained from this method holds 2,196,388 cases, with each category having the same number of observations. From this sample size, a random split was performed where 80% was distributed to the training sample and 20% to the test sample.

Table 4. Output shape of each layer in proposed model

Layer type	Output Shape	Param #
Input Layer	(None, 20, 1)	0
Conv1D_1	(None, 18, 64)	256
MaxPooling1D_1	(None, 9, 64)	0
Conv1D_2	(None, 7, 128)	24,704
MaxPooling1D_2	(None, 3, 128)	0
LSTM	(None, 64)	49,408
Dropout	(None, 64)	0
Dense	(None, 64)	4160
Output	(None, 1)	65

Table 4 represents the structure of the suggested models with the number of parameters at each layer and the output shape of each layer.

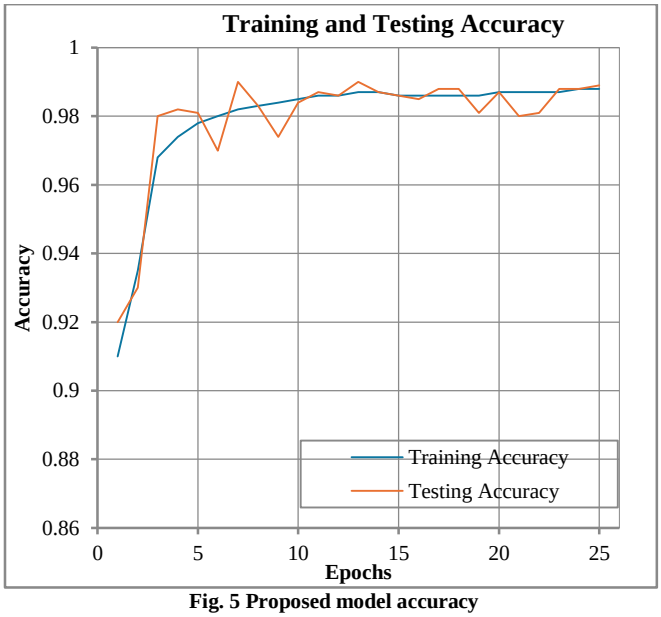

Fig. 5 Proposed model accuracy

From Figures 5 to 8, one can see how the performances of the proposed Hybrid CNN-LSTM model have been proven

by using the CICIoT2023 dataset. As seen in Figure 5, the accuracy during the training and test phase has followed an increasing trend, showing that there has been gradual convergence. This means that the proposed model has been able to learn certain features, and there has been no problem of overfitting.

To check the consistency of the proposed CNN-LSTM architecture and reduce the effect of random initialization of weights on the biasing of the model, five separate experimentation sessions were conducted. As can be seen from Table 5, the accuracy rate of the model is around 98.72% with an exceptionally low standard deviation of ±0.06%. The extremely low variance ensures that the results have some statistical significance rather than being a result of any random initialization.

Table 5. Performance stability across 5 independent runs

Run #	Accuracy	F1-Score
Run 1	98.75%	98.74%
Run 2	98.68%	98.80%
Run 3	98.78%	98.83%
Run 4	98.70%	98.72%
Run 5	98.66%	98.73%
Mean ± SD	98.72% ± 0.06%	98.78% ± 0.05%

Figure 6 represents the losses during training and testing for 25 epochs, with a rapid decrease in the first stage, where the system has learned how to learn from the CICIoT2023 data set. While training loss stays below 0.02, there are some oscillations in the testing loss that show no overfitting; both the losses eventually meet at a low level, suggesting effective generalization. Loss minimization is important for IDS due to its extremely excessive costs.

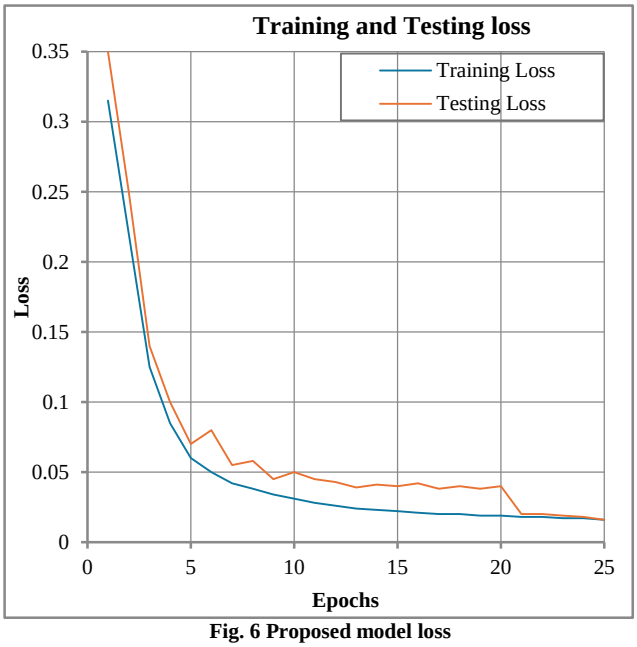

Fig. 6 Proposed model loss

Precision of the model is shown in Figure 7 throughout the training stages. Training and testing precision both start at around 0.92 and then rapidly climb and stabilize above 0.98 after approximately 15 epochs.

Figure 8 shows the recall performance of the suggested technique, which is around 98.65%.

The Recall graph hitting high values at the same moment ensures that actual botnet attacks are not misclassified as "Benign". High recall is the essential goal in binary classification for security to ensure every rogue packet is caught.

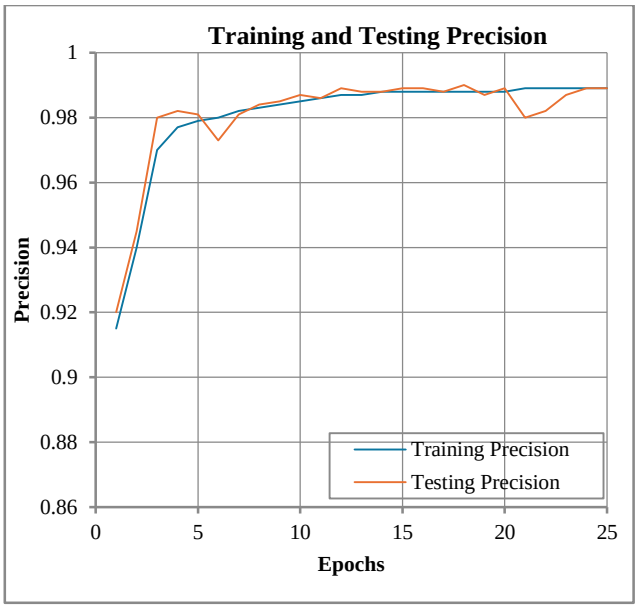

Fig. 7 Proposed model precision

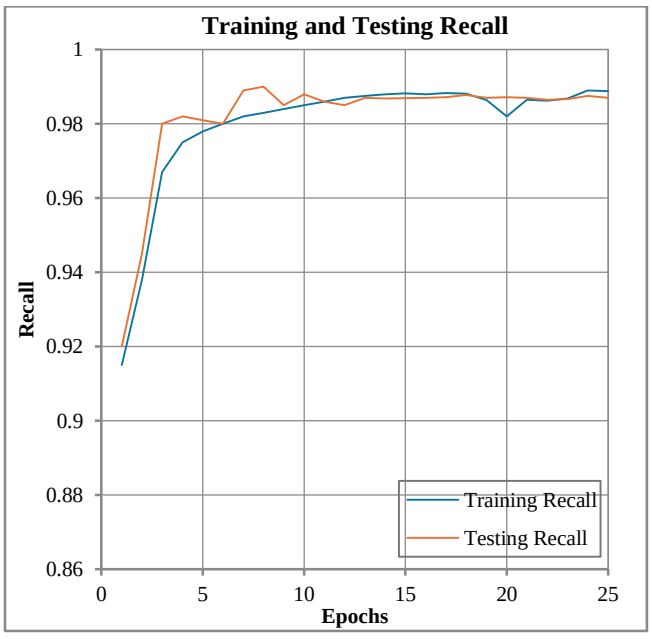

Fig. 8 Proposed model recall

To highlight the stability of the framework when deployed in actual conditions, the developed CNN-LSTM model was evaluated using the naturally imbalanced dataset of the test set. Notwithstanding the fact that the data was highly imbalanced, the model kept an exceedingly high value for F1-score at around 98% and even managed to keep the False Positive Rate (FPR) below 1.5%. This incredibly small FPR shows that the model doesn't overestimate attack cases.

4.2. Comparison with State-of-the-Art

The graph in Figure 9 shows the efficiency of the following three models in detecting the IoT botnet attack against the CICIoT-2023 Dataset: Simple 1D-CNN, LSTM, and Hybrid CNN-LSTM Model. The comparison analysis of the performance of the above models clearly suggests that the hybrid model is statistically better than the other two. Although the 1D-CNN was able to generate a good baseline accuracy level of 96.42%, the problem is that the model is constrained due to its inability to set up any dependencies other than the spatial correlation. However, while LSTM proved better accuracy (97.10%), it had lower precision values.

The CNN-LSTM model outperformed both baseline models on all metrics used because of its high accuracy of 98.72% ± 0.06%. In other words, the fusion of the CNN layers and the LSTM allowed increasing the F1-score by approximately 2.3% over the CNN baseline. It is also clear that the use of CNN-LSTM consistently showed a higher performance rate in various experiments, which shows the necessity of modeling both spatial and structural features to detect complex patterns of attacks conducted by Internet of Things bots.

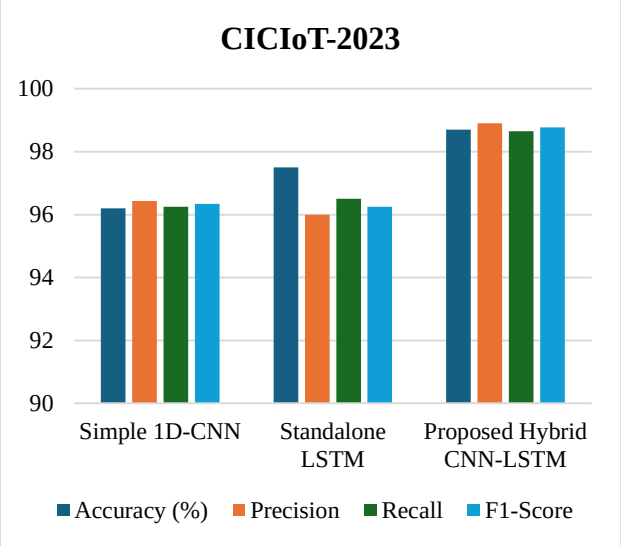

Fig. 9 Comparative analysis of metrics

Table 6 compares hybrid approaches, traditional Machine Learning, and Deep Learning models used in earlier studies.

Table 6. Comparison of performance metrics with existing studies

Model Used	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN, SVM, DT [28]	97.42, 98.65, 96.54	98.45, 99.76, 98.54	97.67, 97.54, 98.01	98.77, 98.54, 98.74
CNN-LSTM [16]	98.42	98.85	98.42	98.57
DLMLP [5]	85.00	99.00	98.00	-
RNN [29]	96.52	96.25	96.52	95.73
Simple 1D-CNN [3]	96.00	96.00	96.00	97.00
CNN, LSTM [11]	83.78, 90.67	75.31, 88.38	70.72, 90.94	72.31, 89.65
Ours	98.72	98.9	98.65	98.77

4.2.1. Architectural Analysis of Performance Superiority

This is because of the smooth connection among the three key elements of this model framework, which are the ANOVA filter, 1D-CNN for feature extraction, and the LSTM gates for aggregating these features, resulting in a statistical domination of this framework by the mean accuracy of 98.72%, with an F1 Score of 98.77%. Reasons as to why this technique outperforms others in mechanical terms include:

Statistical Filtering using ANOVA

The standalone deep learning models (as in [3, 11]) consider all 47 features from the raw network data, requiring the deep learning network to invest effort in learning irrelevant background noise and multi-collinearity in the training process. Calculating the precise mathematical ratio of the ratio of between-class variance (MSB) and within-class variance (MSW), the ANOVA test eliminates 27 noisy features. Such a filtering step ensures that extremely volatile and discriminative attack attributes (such as Duration, rst_count, and Variance) are found before training.

Correlation of Spatial Attributes using 1D-CNNs

Machine learning techniques such as Decision Tree, KNN, and SVM [28] consider the tabular traffic attributes independently, thus unable to detect the pattern that is created by a botnet conducting sweep attacks. To address this limitation, the top 20 features found using ANOVA are fed into a 3D spatial grid (samples, 20, 1). With an increase in filters from 64 to 128 in successive convolutions, the convolution network can automatically derive the underlying correlations between neighboring features without the need for complex feature engineering.

Feature Aggregation Using LSTM Gating for Non-Linearity

Although the conventional approach of using LSTMs [29] is typically limited to dealing with time-series data, this technique employs the LSTM layer as a feature aggregation

mechanism driven by inductive bias. The dynamic equation for each gate (Forget gate, Input gate, and Output gate) is employed as a dynamic filter that assigns different mathematical weightings to the densely packed spatial patterns formed in the earlier CNN layers. Instead of uniformly weighing all detected anomalies, the LSTM gates favor difficult concurrent network anomalies and ignore benign traffic anomalies.

4.3. Computational Complexity and Edge Feasibility

One of the main aims of this study is to set up the feasibility of the suggested architecture in designing an IoT edge device. Unlike the deep learning techniques, where the computational load is generally large, the ANOVA F-test reduces the number of inputs and so the architectural weight.

Table 7. Computational complexity and efficiency metrics

Metric	Proposed Model	Significance of IoT Devices
Total Parameters	~76,500	Unusually small size; extremely low memory requirements.
Model Size (Disk)	~0.30 MB	Can be loaded onto the constrained memory of edge devices.
FLOPs (per sample)	~0.65 MFLOPs	Minimal computing requirement for power efficiency.
Inference Time (CPU)	0.15 ms/packet	Enables real-time, high-speed network traffic analysis.

Based on the data provided in Table 7, it may be seen that the proposed network architecture for CNN-LSTM consists of about 76,500 parameters, which makes the size of this model almost equal to 0.30 MB.

Hence, this model is small and can fit into the small SRAM capacity available in most of the current IoT MCUs. Moreover, one flow of the model requires no more than 0.65 MFLOPS. Hence, the proposed model may be regarded as lightweight, with CPU inference time of less than 0.2 ms per packet.

5. Conclusion

This study has successfully developed and analyzed a novel deep learning-based hybrid system that can provide security to IoT environments by protecting against any DDoS attack risk. Using a balanced subset of the CICIoT2023 data set, this study overcame the drawback faced by individual machine learning techniques by ensuring that both the local feature interaction and the global structural relationship were captured. It should be noted that the ANOVA F-test played a crucial role in selecting only the 20 discriminant features from the high-dimensional feature set.

Proposed architecture using sequential convolutional layers with 64 and 128 filters, as well as 64 units of LSTM, proved to be more effective than single benchmarks.

The experimental analysis confirmed a mean binary classification accuracy reaching 98.72% ± 0.06%, proving the importance of combining the advantages of spatial feature extraction and structure modeling. Apart from the elevated level of accuracy, the proposed approach places a special focus on feasibility, i.e., the possibility of implementing the model into real-life scenarios.

Analysis of the computational complexity revealed that optimized features and usage of combined layers will allow achieving the inference latency of just 0.15 ms. Low capacity of only 0.30 MB ensures feasibility since the limitations of IoT devices are taken into consideration.

As a result, it becomes possible to use the complex deep learning approach without having high computational capabilities. The trade-off between Precision and Recall proves the value of this solution to minimize false negatives because this step is critical to combat massive botnet floods.

In summary, this paper presents an effective, efficient, and scalable framework for an intelligent IDS. Although this paper achieves excellent results in finding DDoS attacks using the proposed machine learning approach to perform binary classification, future research will be geared towards developing a model capable of performing multi-class classification.

Other future endeavors will include deploying the model in real time using edge computing technology and evaluating its ability to detect zero-day exploits.

Conflicts of Interest

The authors declare that there is no conflict of interest about the publication of this paper.

Author Contributions

Conceptualization, V.S. and A.J.; methodology, V.S.; software, V.S.; validation, V.S., and A.J.; formal analysis, A.S.; investigation, V.S.; resources, A.J.; data curation, V.S.; writing—original draft preparation, V.S.; writing—review and editing, V.S. and A.J.; visualization, V.S.; supervision, A.J.; project administration, A.J. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors would like to thank the Faculty of Engineering and Technology, Parul University, Vadodara, Gujarat, India, for providing the necessary infrastructure and resources to conduct this research.

References

[1] Deepa Venkataraya Premalatha, and Sukumar Ramanujam, "Ensemble-Based Intrusion Detection for IoT Networks Using the CICIoT2023 Dataset," *Journal of Information Systems Engineering and Management*, vol. 10, no. 21, pp. 743-755, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034, Statista, 2025. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide

[3] Omar Almousa, Batool Hamdallh, and Ruba Al-nu'man, "Enhancing IoT Security: A Comparative Analysis of Machine Learning and Deep Learning Techniques for Botnet Detection," *Engineering, Technology and Applied Science Research*, vol. 15, no. 4, pp. 24498-24505, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Eryk Schiller et al., "Landscape of IoT Security," *Computer Science Review*, vol. 44, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Samuel Kofi Erskine, "Real-Time Large-Scale Intrusion Detection and Prevention System (IDPS) CICIoT Dataset Traffic Assessment Based on Deep Learning," *Applied System Innovation*, vol. 8, no. 2, pp. 1-32, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Belal Ibrahim Hairab et al., "Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques," *Electronics*, vol. 12, no. 3, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bambang Susilo, Abdul Muis, and Riri Fitri Sari, "Intelligent Intrusion Detection System against Various Attacks Based on a Hybrid Deep Learning Algorithm," *Sensors*, vol. 25, no. 2, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Doaa Mohsin Abd Ali Afraji, Jaime Lloret, and Lourdes Peñalver, "An Integrated Hybrid Deep Learning Framework for Intrusion Detection in IoT and IIoT Networks Using CNN-LSTM-GRU Architecture," *Computation*, vol. 13, no. 9, pp. 1-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] S.P. Jani, and M. Adam Khan, *Applications of AI in Smart Technologies and Manufacturing*, 1st ed., London: CRC Press, vol. 2, pp. 1-556, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Zhendong Wang et al., "A Lightweight Intrusion Detection Method for IoT Based on Deep Learning and Dynamic Quantization," *PeerJ Computer Science*, vol. 9, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Abdullah Waqas et al., "Comparative Analysis of Deep Learning Models for Intrusion Detection in IoT Networks," *Computers*, vol. 14, no. 7, pp. 1-18, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Maher Alharby, "Evaluating Machine Learning Approaches for Multiple Attack Classification with Improved Computational Efficiency in IoT Networks," *Scientific Reports*, vol. 15, no. 1, pp. 1-18, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Md. Alamgir Hossain, "Deep Learning-Based Intrusion Detection for IoT Networks: A Scalable and Efficient Approach," *EURASIP Journal on Information Security*, vol. 2025, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Enock Mudau, Topside E. Mathonsi, and Tshimangadzo Tshilongamulenzhe, "A Deep Learning Algorithm to Minimize Cyber-Security Attacks for Small Enterprises," *2024 International Conference on Electrical, Computer and Energy Technologies ICECET*, Sydney, Australia, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Nik Muhammad Danial Bin Nik Ram Zaedi, Amy Lim Hui Lan, and Goh Hui-Ngo, "Developing Deep Learning Models to Predict Benign and Malicious Attacks in IoT Networks," *2024 IEEE International Conference on Computing (ICOCO)*, Kuala Lumpur, Malaysia, pp. 90-95, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Afrah Gueriani, Hamza Kheddar, and Ahmed Cherif Mazari, "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, EL OUED, Algeria, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Abdullah Mujawib Alashjaee, "Deep Learning for Network Security: An Attention-CNN-LSTM Model for Accurate Intrusion Detection," *Scientific Reports*, vol. 15, no. 1, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[18] Amna Naeem et al., "Efficient IoT Intrusion Detection with an Improved Attention-Based CNN-BiLSTM Architecture," *2026 International Conference on Integrated Intelligence and Cognitive Engineering (ICIICE)*, Dubai, United Arab Emirates, pp. 1-6, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jaydeep R.Tadhani et al., "Securing Web Applications against XSS and SQLi Attacks using a Novel Deep Learning Approach," *Scientific Reports*, vol. 14, no. 1, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Preeti Kailas Suryawanshi, and Sonal Kirankumar Jagtap, "Mitigating Iot Botnets with CNN-LSTM and Anomaly Detection," *International Journal of Environmental Sciences*, vol. 11, no. 16, pp. 1216-1223, 2025. [CrossRef] [Publisher Link]

[21] Selman Hizal, Unal Cavusoglu, and Devrim Akgun, "A Novel Deep Learning-Based Intrusion Detection System for IoT DDoS Security," *Internet of Things*, vol. 28, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Akinul Islam Jony, and Arjun Kumar Bose Arnob, "A Long Short-Term Memory based Approach for Detecting Cyber-Attacks in IoT using CIC-IoT2023 Dataset," *Journal of Edge Computing*, vol. 3, no. 1, pp. 28-42, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Noor Ul Ain et al., "Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach," *Sensors*, vol. 25, no. 5, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Sami Yaras, and Murat Dener, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Electronics*, vol. 13, no. 6, pp. 1-28, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Euclides Carlos Pinto Neto et al., "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, pp. 1-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Burak Aydin, Hakan Aydin, and Sedat Gormus, "Intrusion Detection Systems in IoT: A Detailed Review of Threat Categories, Detection Strategies, and Future Technologies," *Journal of Information Security and Applications*, vol. 95, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[27] Thorir Mar Ingolfsson, Insights into LSTM Architecture. 2021. [Online]. Available: https://thorirmar.com/post/insight_into_lstm/

[28] Miracle Udurume, Vladimir Shakhov, and Insoo Koo, "Comparative Analysis of Deep Convolutional Neural Network—Bidirectional Long Short-Term Memory and Machine Learning Methods in Intrusion Detection Systems," *Applied Sciences*, vol. 14, no. 16, pp. 1-1 6, 2024.[CrossRef] [Google Scholar] [Publisher Link]

[29] Sidra Abbas et al., Evaluating Deep Learning Variants for Cyber-Attacks Detection and Multi-Class Classification in IoT Networks," *PeerJ Computer Science*, vol. 10, pp. 1-23, 2024. [CrossRef] [Google Scholar] [Publisher Link]