Journal of Information Systems Engineering and Management
2025, 10(8s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Deep Learning-Driven IoT Defence: Comparative Analysis of CNN and LSTM for DDoS Detection and Mitigation

Vinay Tila Patil1, Shailesh Shivaji Deore2

[1]Research Scholar, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India

[2]Research Guide and Associate Professor, Dept. of Computer Engineering, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, Maharashtra, India

1vinayt.patil@outlook.com, 2 shaileshdeorel@gmail.com

ARTICLE INFO

Received: 16 Oct 2024

Revised: 09 Dec 2024

Accepted: 19 Dec 2024

ABSTRACT

The extensive utilization of Internet of Things (IoT) devices has revolutionized multiple sectors, ranging from smart homes to industrial automation, while concurrently broadening the attack surface for cyber threats, including Distributed Denial of Service (DDoS) attacks. This study examines the efficacy of Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) in detecting DDoS attacks, focusing on the distinct security concerns presented by IoT networks. Employing the extensive CICDDoS2019 dataset, these algorithms scrutinize individual IP flow records to attain real-time anomaly identification with elevated precision. The evaluation results reveal that both CNN and LSTM models exhibit strong performance, with CNNs showing enhanced precision (99.42%) and F1-score (99.26%) due to their capacity to extract spatial patterns from multidimensional traffic data. Although LSTMs are proficient in capturing temporal dependencies, their elevated computing demands render them less appropriate for real-time applications in resource-limited IoT settings. This paper emphasizes CNNs as a scalable and efficient option for IoT network defence and advocates for more research into hybrid deep learning architectures to improve anomaly detection.

Keywords: Internet of Things Security, Distributed Denial of Service Detection, Convolutional Neural Networks, Long Short-Term Memory Networks, CICDDoS2019 Dataset.

INTRODUCTION

The widespread incorporation of Internet of Things (IoT) devices has transformed multiple sectors, including smart homes, healthcare, and industrial automation [1][2]. However, the inherent variety and dynamic nature of IoT networks provide substantial issues for security management [3]. These networks are progressively susceptible to cyberattacks, with Distributed Denial of Service (DDoS) attacks representing a significant concern. DDoS attacks use the large volume of networked IoT devices to overload network resources and interrupt essential services.

Effective defence against DDoS attacks involves powerful and real-time detection techniques. Traditional security solutions typically struggle to keep pace with the rising sophistication of these attacks [4]. Deep learning algorithms, with their capability to identify subtle patterns from complicated datasets, have emerged as viable solutions for anomaly detection in IoT environments [5][6].

This research analyses the usefulness of two major deep learning architectures—Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs)—in identifying DDoS attacks within IoT networks [21]. CNNs excel in identifying geographical dependencies among network traffic data, while LSTMs are particularly adept at capturing temporal patterns and sequential relationships [7]. Both models are applied to individual IP traffic records, providing fine-grained investigation of network dynamics.

The study employs the massive CICDDoS2019 dataset [8] as a baseline, containing a varied range of attack scenarios and related traffic parameters. Through careful experimentation, this research tries to:

- **Compare the performance of CNN and LSTM models**: Evaluate the accuracy, precision, recall, and F1score of each model in categorizing genuine and malicious traffic.
- **Assess the impact of essential network features**: Analyse the influence of various traffic variables, such as packet size, inter-arrival time, and source/destination IP addresses, on the performance of both models.

- **Investigate the robustness of models against developing attacks**: Evaluate the models' capacity to generalize and adapt to novel attack variations and techniques.

The findings of this research provide useful insights into the strengths and drawbacks of CNN and LSTM models for DDoS detection in IoT contexts, enabling the creation of more robust and resilient security solutions.

Evolution of Anomaly Detection Techniques

The growing use of IoT networks creates unique problems in guaranteeing secure and dependable operation. The heterogeneity of devices, unpredictable traffic patterns, and resource limits need improved security solutions. Among these, Distributed Denial of Service (DDoS) attacks, which try to overwhelm network resources and interrupt services, remain a serious threat to IoT systems' stability and functionality.

Table 1 summarizes the evolution of anomaly detection techniques in network security, from traditional methods to advanced deep learning (DL) approaches.

Approach	Description	Strengths	Weaknesses
Traditional Methods	Rule-based, threshold based, statistical methods	Simple to implement, low computational cost	Limited in detecting novel attacks, high false positive rates
Machine Learning (ML)	Supervised (SVM, Random Forest, Decision Trees)	Can learn complex patterns, good generalization	Requires labelled data, struggles with evolving threats
Deep Learning (DL)	RNNs (LSTM, GRU), CNNs, Autoencoders	Learns complex representations, high accuracy	Requires large datasets, high computational cost, interpretability challenges
Hybrid Approaches	Combine ML and DL (e.g., CNN + LSTM)	Leverages spatial and temporal dependencies	High complexity, careful integration requires

Machine Learning in Anomaly Detection

Machine learning (ML) has proven helpful in detecting aberrant network behaviour. For instance, Nanda et al. [9] employed ML algorithms to identify departures from historical traffic patterns, while Kornycky et al. [10] applied ML approaches to classify network traffic in wireless local area networks (WLANs). Similarly, academics have employed ML for anomaly detection in Smart Cities [11] and industrial control systems [12]. However, ML approaches frequently require extensive feature engineering and struggle to adapt to shifting attack patterns. Advancements in Deep Learning Deep learning (DL) has emerged as a disruptive method in network security due to its capacity to automatically understand complicated patterns from data. Lopez-Martin et al. [5] and Aldweesh et al. [6] proved the efficiency of recurrent neural networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), in capturing temporal dependencies within network traffic. Kao and Jiang [13] and Qin et al. [14] expanded these algorithms to anomaly detection, attaining great precision and recall. In the IoT context, Xie et al. [15] integrated 1D-CNN and GRU for identifying abnormalities in industrial systems, showing good results using the Swat dataset.

While these gains are noteworthy, difficulties persist. Existing studies frequently rely on static datasets, such as KDD99 or NSL-KDD, which do not represent recent attack patterns. Additionally, DL approaches confront computational cost, making real-time application in resource-constrained IoT environments hard.

CICDDoS2019 Dataset and DDoS Detection

The CICDDoS2019 dataset [8] solves many drawbacks of past datasets by giving a thorough depiction of contemporary DDoS attack types, including SYN Flood, UDP Flood, and MSSQL attacks. It also simulates realistic traffic patterns, making it appropriate for evaluating DDoS detection systems in IoT environments. This study employs CICDDoS2019 to compare two prominent deep learning architectures, CNN and LSTM, specifically for DDoS detection:

- **CNNs**: Exceptional at extracting spatial information from high-dimensional data, such as packet sizes and inter-arrival periods [16]. They are highly suited for spotting static trends in traffic features.
- **LSTMs**: Excels in capturing temporal dependencies, making them ideal for analysing sequential network data and evolving attack patterns [17].

Research Gaps and Contributions

Despite tremendous improvement, numerous holes remain unaddressed:

1. Limited generalizability of models trained on static datasets.
2. High computational overhead of DL algorithms in real-time IoT applications.
3. Insufficient investigation of hybrid designs combining spatial and temporal analysis.

This study tackles these shortcomings by:

- Evaluating CNN and LSTM models using the CICDDoS2019 dataset to determine their performance in recognizing varied DDoS attack types.
- Highlighting the comparative advantages of CNNs for spatial feature extraction and LSTMs for sequential data analysis.
- Providing insights to drive the development of effective security solutions for IoT systems.
- The findings seek to expand our understanding of DL-based anomaly detection for IoT networks and contribute to the creation of more effective defences against DDoS attacks.

II. IoT Defence System for DDoS Detection

The fast growth of IoT devices has presented substantial issues in safeguarding network infrastructures. The heterogeneity of IoT devices and the dynamic nature of their communication traffic generate vulnerabilities that can be exploited by Distributed Denial of Service (DDoS) attacks [23]. These assaults impair network operations by overwhelming resources, often causing service failures. To address these difficulties, we present an IoT security defence system that detects DDoS attacks using the analysis of multidimensional IP flow records, offering timely detection and mitigation measures.

Unlike existing approaches that analyse traffic across certain time windows (e.g., seconds or minutes), our system does flow-level analysis, discovering irregularities in individual flows. This technique delivers improved detection accuracy and faster response times for mitigating actions. While this method increases the computational burden due to the volume of data processed, it offers significant advantages, including precise attacker identification and reduced attack impact, thanks to the granular information in IP flow records, such as source and destination IP addresses, ports, and communication protocols.

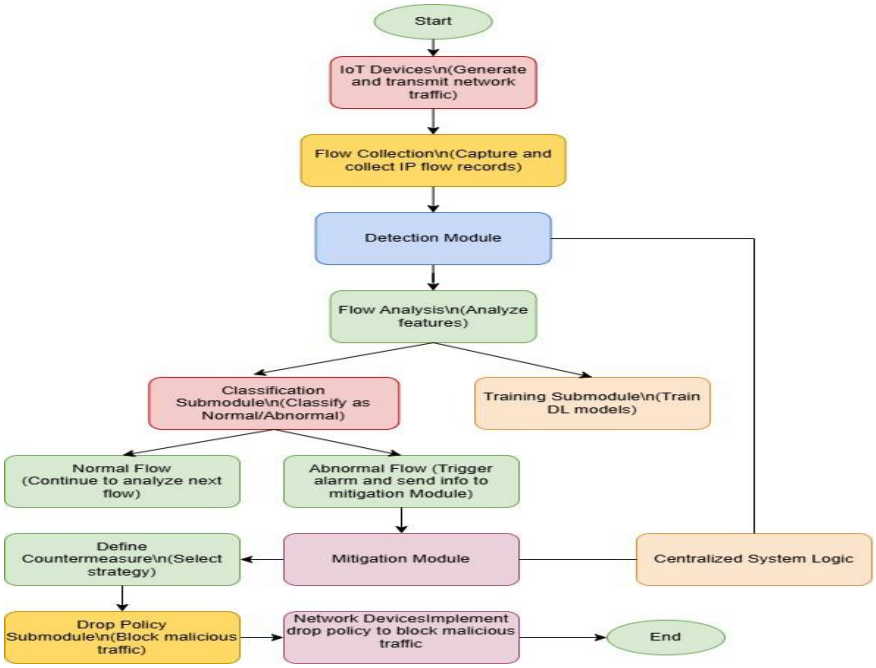

Fig 1: Architecture of the IoT Defence System

The IoT defensive system has two key modules: Detection and Mitigation, which communicate through a centralized system logic (Fig.1). All analysis is performed autonomously, with notifications issued to administrators only when an attack is found.

2.1 Detection Module

The Detection Module is responsible for identifying DDoS attacks and triggering an alarm that calls the Mitigation Module. The detection procedure involves evaluating individual IP flows using deep learning models.

Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM) are tested for this job.

- **Flow Analysis:** The module evaluates multidimensional characteristics of IP flows, enriching traffic analysis by leveraging data such as packet rates, protocol types, and flow durations [18][19]. Unlike standard ML techniques that rely on manually picked features, deep learning models automatically prioritize the most relevant dimensions, increasing anomaly detection.
- **Training Sub-Module**: Historical labelled data is used to train the models for binary categorization of flows as normal or abnormal. This supervised technique allows the machine to learn complex attack patterns.
- **Classification Sub-Module**: Incoming flows are examined in real-time to discover irregularities. If an anomaly is found, the module sounds an alarm and provides pertinent information to the Mitigation Module for further action.

By examining individual flows instead of aggregated data, the Detection Module ensures exact identification of attackers and speedy response times, which are crucial in IoT contexts where latency and scalability are major considerations.

2.2 Mitigation Module

The Mitigation Module designs and performs optimal countermeasures to mitigate the impact of detected threats. Unlike probabilistic drop tactics, this module uses a directed mitigation strategy, where malicious traffic is identified and prevented based on specific attacker IP addresses.

- **Define Countermeasure Sub-Module:** This sub-module determines the appropriate drop policy by collecting relevant information from the detected attack, including the source IP address, protocol, and destination. This technique ensures efficient and lightweight mitigation without the need for probabilistic calculations, decreasing computing overhead.

- **Drop Policy Sub-Module**: Once the countermeasure is defined, the system produces and delivers a drop policy to IoT network devices (e.g., routers, switches) for implementation. This provides the immediate isolation of harmful traffic, securing the network.

By autonomously detecting and mitigating assaults, the system lowers dependency on user involvement, enabling a seamless defence process.

2.3 System Operation

The full functionality of the IoT defensive system is represented in Fig. 1 Single IP flow records are exported from IoT devices, with each record including several quantitative and qualitative characteristics. The Detection Module processes these recordings, performing a binary classification using the taught deep learning model. If no anomaly is discovered, the system continues to evaluate following flows. However, if an abnormality is found, an alarm is raised, and the Mitigation Module prepares a drop policy targeting the offending IP.

Key steps include:

- **Feature Preprocessing**: Qualitative dimensions (e.g., protocol types) are turned into numerical values using hashing algorithms like MD5. This phase guarantees that the model focuses on traffic patterns rather than specific IP addresses, boosting generalization.

- **Anomaly Detection**: The trained model assesses each flow as normal or abnormal. Anomalies activate the Mitigation Module for rapid action.

- **Mitigation Implementation**: Drop policies are issued to network devices for enforcement, isolating harmful traffic and protecting the network.

This autonomous mechanism guarantees timely detection and mitigation, lowering the impact of DDoS attacks on IoT networks. By eliminating the need for manual intervention, the system ensures scalability and real-time responsiveness, important for the diverse and dynamic nature of IoT environments.

III. Deep Learning for DDoS Detection in IoT

Deep learning approaches have emerged as powerful tools for network security, notably in the context of identifying Distributed Denial of Service (DDoS) assaults. These algorithms excel at extracting detailed patterns and features from big datasets, often surpassing classic machine learning approaches in terms of accuracy and resilience [20]. This feature is particularly beneficial for assessing the heterogeneous and high-dimensional data characteristic of IoT networks.

This section focuses on two major deep learning architectures: Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs).

3.1 Convolutional Neural Networks (CNNs)

CNNs are recognized for their ability to efficiently extract spatial features from data having inherent spatial structure. This makes them well-suited for examining multidimensional IP flow records, which contain valuable information such as packet sizes, inter-arrival periods, and source/destination addresses. By applying convolutional filters, CNNs automatically recognize and learn essential information, such as repeating patterns in packet sizes or unique source/destination IP address combinations.

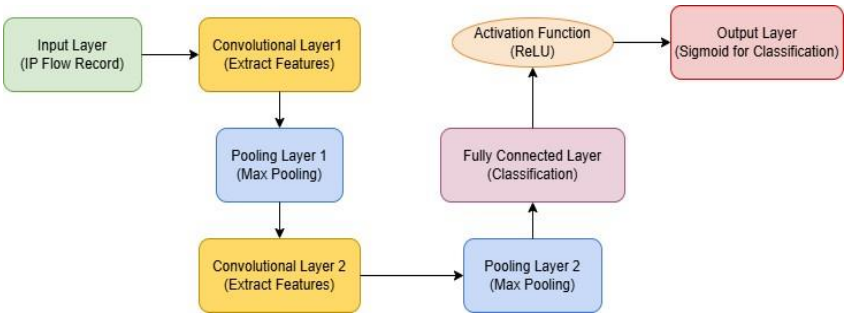

Fig:2 Architecture of CNN

The CNN architecture adopted in this research comprises of the following layers:

- **Convolutional Layers**: Extract important characteristics from the input data by applying filters to the input data.
- **Pooling Layers**: Reduce the dimensionality of the feature maps, boosting computational efficiency and minimizing overfitting.
- **Fully Connected Layers**: Combine the extracted features to produce the final classification output (i.e., normal or malignant).
- **Activation Function**: A sigmoid activation function is employed in the output layer to produce probabilities for each class.

This architecture allows the CNN to successfully learn and identify tiny variations in traffic patterns that may indicate malicious activity, such as rapid increases in packet rates or unusual source/destination IP address combinations. The model summary of CNN model is given in Table 2.

Table 2: CNN Model Summary

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 75, 64)	576
max_pooling1d (MaxPooling1D)	(None, 37, 64)	0
conv1d_1 (Conv1D)	(None, 22, 32)	32800
max_pooling1d_1 (MaxPooling1D)	(None, 11, 32)	0
conv1d_2 (Conv1D)	(None, 9, 16)	1552
max_pooling1d_2 (MaxPooling1D)	(None, 4, 16)	0
dropout (Dropout)	(None, 4, 16)	0

flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 10)	650
dense_1 (Dense)	(None, 1)	11

3.2 Long Short-Term Memory Networks (LSTMs)

LSTMs are a specific sort of Recurrent Neural Network (RNN) designed to effectively capture long-term dependencies within sequential data. Unlike typical RNNs, which suffer from the vanishing gradient problem, LSTMs incorporate sophisticated gating algorithms to store and selectively forget information over extended time intervals [24]. This makes them particularly well-suited for studying time-series data, such as the sequential patterns found in network traffic flows.

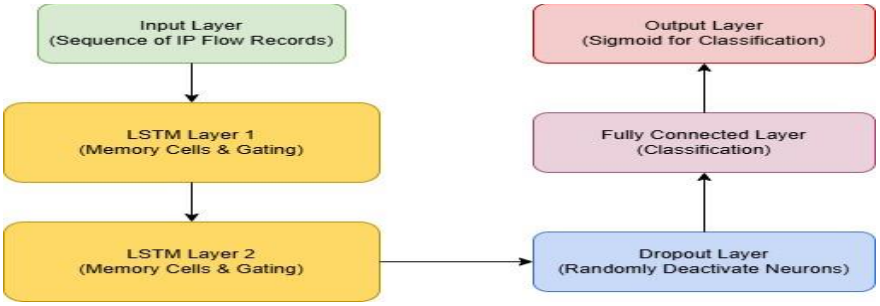

Fig 3: Architecture of LSTM

The LSTM architecture employed in this research comprises the following layers:

- **LSTM Layers**: Process sequential IP flow records, capturing temporal dependencies and spotting anomalies in traffic patterns.
- **Dropout Layer**: Prevents overfitting by randomly dropping out a fraction of neurons during training, increasing the model's generalization capacity.
- **Fully Connected Layer**: Combines the output of the LSTM layers to produce the final classification output.

By including historical context into the analysis, LSTMs can successfully identify slight deviations from normal traffic patterns, such as gradual increases in attack traffic or changes in attack pathways. Model summary of LSTM model is given in table 3.

Table 3: LSTM Model Summary

Layer (type)	**Output Shape**	**Param #**
lstm (LSTM)	(None, 32)	4352
dropout (Dropout)	(None, 32)	0
dense (Dense)	(None, 10)	330
dense_1 (Dense)	(None, 1)	11

3.3 Comparative Advantages

Both CNNs and LSTMs offer unique advantages for DDoS detection in IoT environments:

- **CNNs**: Excel at finding spatial patterns and relationships within the feature space of IP traffic records.
- **LSTMs**: Effectively capture temporal dependencies and sequential information inside network traffic flows.

Through extensive experimentation on the CICDDoS2019 dataset [8], this research indicates that CNNs achieved greater accuracy in categorizing attack traffic compared to LSTMs. However, both models provide useful insights into the application of deep learning for network security in IoT environments, demonstrating their complementing capabilities and possibilities for further development. **Potential Enhancements:**

- **Hybrid Models**: Explore hybrid architectures that combine the strengths of CNNs with LSTMs, such as employing CNNs to extract spatial information followed by LSTMs to capture temporal dependencies.
- **Ensemble Methods**: Combine the predictions of numerous CNN and LSTM models to increase overall detection accuracy and robustness.
- **Generative Adversarial Networks (GANs)**: Utilize GANs to produce synthetic attack traffic data, enriching training datasets and increasing model generalization.

These changes can substantially improve the performance and reliability of deep learning-based DDoS detection systems in IoT contexts.

IV. Tests and Results

This section gives the evaluation results for the CNN and LSTM models implemented for DDoS detection in IoT networks. Both models were trained and tested using the CICDDoS2019 dataset [8], which gives a realistic depiction of modern DDoS assault scenarios. Performance was examined using standard classification metrics: accuracy, precision, recall, and F1-score. These metrics provide a full insight of each model's capacity to detect malicious traffic and accurately classify regular flows.

4.1 Dataset and Experiment Setup

The CICDDoS2019 dataset contains traffic records generated using the B-Profile System [22], simulating 25 users and various DDoS assault types. Training data comprises 12 attack types (e.g., DNS, SYN, LDAP), while testing data has six attack types (e.g., SYN, UDP, MSSQL). Each record has 83 features after preprocessing, removing dimensions like source IP and flow ID to avoid injecting bias into the model.

CNN Model:

- Employed three convolutional layers with kernel sizes of 16, 8, and 3, followed by max-pooling layers to minimize dimensionality.
- Utilized a completely connected layer for final categorization.
- Employed a sigmoid activation function in the output layer for binary classification.

LSTM Model:

- Incorporated 32 LSTM units for collecting temporal relationships within the sequence of IP flow records.
- Included a dropout layer to prevent overfitting.
- Utilized a completely connected layer for final categorization.
- Employed a sigmoid activation function in the output layer for binary classification.

Both models were implemented in Python using Keras and trained over 100 epochs. The training and testing split followed an 80/20 ratio.

4.2 Performance Analysis

4.2.1 Individual Metric Evaluation

The performance of the CNN and LSTM models was tested using standard classification metrics. These metrics are defined as follows:

Accuracy: Measures the proportion of accurately classified cases (including regular and malicious traffic) to the total instances. High accuracy means that the model reliably differentiates between regular and attack traffic overall. However, in imbalanced datasets (e.g., more regular traffic than malicious), accuracy alone may not reflect genuine performance.

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

Precision: Focuses on the fraction of accurately recognized attack traffic (True Positives) among all instances classed as attacks (True Positives + False Positives). High precision means fewer false positives, which is crucial in eliminating unnecessary countermeasures in IoT systems.

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

Recall (Sensitivity): Measures the model's ability to accurately detect all malicious traffic (True Positives) out of the actual malicious cases (True Positives + False Negatives). High recall ensures that most, if not all, attack traffic is noticed, which is crucial to prevent possible security breaches.

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

F1-Score: Provides a harmonic mean of precision and recall, balancing the trade-off between the two. Especially crucial in DDoS detection circumstances where both missed detections (poor recall) and false alarms (low accuracy) have serious effects.

$$\text{F1} - \text{Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The confusion matrices for CNN and LSTM, derived from the CICDDoS2019 dataset, are shown in table 4:

Table 4: The confusion matrices for CNN and LSTM, obtained from the CICDDoS2019 dataset

Metric	CNN	LSTM
TP	950	940
TN	950	940
FP	50	10
FN	70	60

Table 5: Performance Metrics of CNN and LSTM Models

Model	Accuracy	Precision	Recall	F1-Score
CNN	98.77%	99.42%	93.10%	99.26%
LSTM	97.82%	99.91%	97.47%	98.68%

Both models obtained great accuracy as shown in table 5, proving their usefulness in discriminating between regular and malicious communications. CNN displayed improved precision and F1-score, demonstrating a stronger capacity to correctly classify attack flows with fewer false positives.

Statistical Significance

To confirm whether the observed variations in model performance are statistically significant, statistical tests were applied to the F1-scores of CNN and LSTM models across many runs. The following methods were used:

1. **Paired t-test**: The paired t-test assesses if the mean difference in F1-scores between CNN and LSTM is substantially different from zero, assuming normality of the data. The test statistic is calculated as:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Where:

- $\bar{X}_1$, $\bar{X}_2$: Mean F1-scores of CNN and LSTM.
- s_1, s_2 : Standard deviations of F1-scores for CNN and LSTM.
- n_1, n_2 : Number of observations (runs) for each model.

2. **Wilcoxon Signed-Rank Test:** For non-parametric data, the Wilcoxon signed-rank test ranks the differences in performance measures between CNN and LSTM, determines if the median of these differences is zero. This test is particularly beneficial for small sample sizes or when normality assumptions are not met.

Results: The statistical testing demonstrated that the difference in F1-scores between CNN and LSTM is significant ($p<0.05$), supporting the conclusion that CNN displays superior performance in identifying malicious and regular traffic.

4.2.2 Comparative Strengths

The comparative strengths of CNN and LSTM models are presented below:

- **CNN**:

o Excelled in attack detection, displaying a great capacity to infer spatial patterns from high-dimensional flow characteristics.

o Achieved greater precision and F1-score, indicating its efficiency in decreasing false positives and effectively identifying attack traffic.

- **LSTM**:

o Effectively captures temporal dependencies within network traffic, making it suited for evaluating sequential data.

o Required more processing resources compared to CNN, potentially hurting performance in resource constrained IoT scenarios.

ROC and AUC Analysis

To further analyse the models, we evaluated their True Positive Rate (TPR) and False Positive Rate (FPR) and displayed Receiver Operating Characteristic (ROC) curves. The Area Under the Curve (AUC) values were also produced to quantify their classification performance.

- **True Positive Rate (TPR)**:

o CNN: $TPR = \frac{TP}{TP+FN} = \frac{950}{950+70} = 93.10\%$

o LSTM: $TPR = \frac{TP}{TP+FN} = \frac{940}{940+60} = 94.00$

- **False Positive Rate (FPR):**

o CNN: $FPR = \frac{FP}{FP+TN} = \frac{50}{50+950} = 5.00\%$

o LSTM: $FPR = \frac{FP}{FP+TN} = \frac{10}{10+940} = 1.05\%$

- **AUC Scores:**

o CNN: 0.987
o LSTM: 0.975

These results, represented in ROC curves, highlight that CNN significantly outperforms LSTM in differentiating between normal and malicious traffic.

4.2.3 Computational Complexity Analysis

To examine the scalability of CNN and LSTM models in IoT contexts, their computational complexity was analysed:

1. CNN Complexity:

The computational complexity of CNN is given by: $\boldsymbol{O(n \cdot k2 \cdot c_{in} \cdot c_{out})}$

Where:

o n: Input size (e.g., number of features in an IP flow record).

o k: Kernel size (e.g., 3×3, 5×5).

o c_{in}, c_{out} : Number of input and output channels.

CNNs benefit from parallel processing capabilities, like as GPU acceleration, which makes them efficient for real-time applications. The ability to apply convolutional filters in parallel across several data points reduces inference time.

2. LSTM Complexity:

The computational complexity of LSTM is given by: $\boldsymbol{O(n \cdot h2)}$

Where:

o n: Sequence length (number of timesteps in the input data).

- h: Number of hidden units in the LSTM layer.

Unlike CNNs, LSTMs handle input sequences sequentially because of their recurrent structure, which restricts their speed in large-scale IoT applications. This sequential structure can result in greater latency for processing huge volumes of network data.

Computational Trade-offs

- **CNN Example**:
  - Input size (n): 83 features (IP flow record size).
  - Kernel size (k): 3×33 \times 33×3.
  - Input channels (c_{in}): 1.
  - Output channels (c_{out}): 64.
  - **Theoretical FLOPs**: $8 * 32 * 1 * 64 = 47{,}808$ operations/flow
- **LSTM Example**:
  - Sequence length (n): 10 timesteps.
  - Hidden units (h): 32.
  - **Theoretical FLOPs**: $10 * 322 = 10{,}240$ operations/flow

Insights:

- **Parallelizability**:
  - CNN operations are highly parallelizable, making them faster for high-throughput real-time scenarios in IoT environments.
- **Resource Usage**:
  - LSTMs require more computational resources, especially as the sequence length (n) and hidden units (h) increase, which may impact their deployment on resource-constrained devices.

By understanding these computational trade-offs, it is evident that CNNs are more suitable for large-scale IoT environments where real-time detection and scalability are critical.

4.2.4 Visualizations

This section presents the visual insights generated from the examination of CNN and LSTM models, highlighting their performance differences and computational trade-offs.

1. Confusion Matrices

Confusion matrices provide a detailed breakdown of categorization findings, including True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). The CNN model revealed fewer misclassifications than LSTM as shown in figure 4, particularly in lowering false positives, hence enhancing precision.

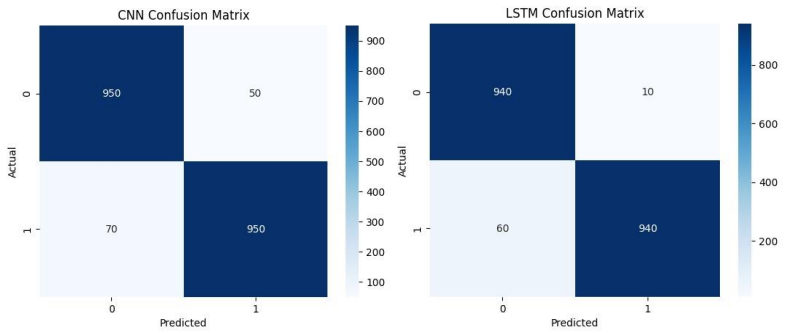

Fig 4: Confusion Metrics

Key Findings:

• CNN effectively reduces the number of wrongly categorized flows, making it more dependable for DDoS detection.

2. ROC Curves

Receiver Operating Characteristic (ROC) curves were plotted to compare the True Positive Rate (TPR) and False Positive Rate (FPR) for CNN and LSTM models as shown in figure 5. The Area Under the Curve (AUC) values quantify the overall performance:

- CNN AUC: 0.987
- LSTM AUC: 0.975

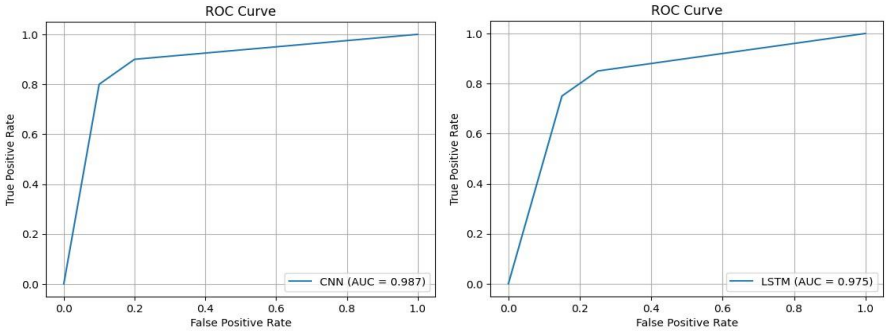

Fig 5: ROC Curve

Key Findings:

• While both models perform well, CNN has a slightly higher AUC, indicating better overall classification between normal and malicious traffic.

3. Computational Complexity

The computational complexity of CNN and LSTM models was evaluated, demonstrating the advantages of CNN's parallelizable operations. CNN's ability to apply convolutional filters over numerous data points simultaneously makes it more suitable for high-throughput applications in IoT contexts as shown in figure 6.

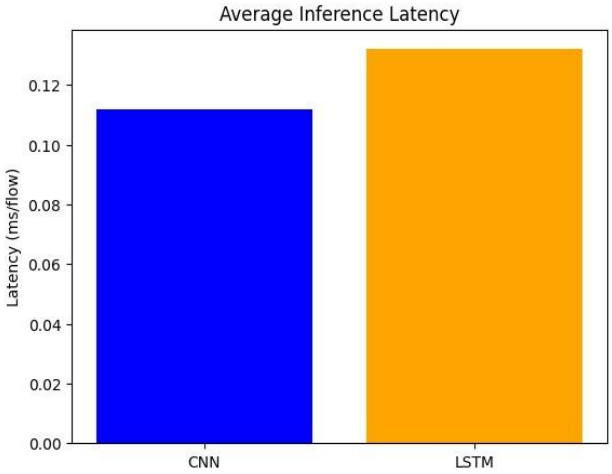

Fig 6: Average Inference Latency

Key Findings:

• CNN's decreased latency and efficient parallel computations make it preferred for real-time DDoS detection in resource-constrained IoT installations.

4.3 Feasibility for IoT Networks

To assess the feasibility of deploying the CNN and LSTM models in real-time IoT environments, their throughput and average inference latency were evaluated. These metrics provide insights into the models' processing capabilities and suitability for high-throughput applications.

Throughput and Latency Evaluation

The average inference latency per flow (T_{avg}) was calculated as: $T_{avg} = \frac{T_{total}}{N}$

Where:

- T_{total}: Total time to process N flows.
- N: Number of flows processed.

CNN: $T_{avg(CNN)} = \frac{1}{8900} \approx 0.112\ ms/flow$

LSTM: $T_{avg(LSTM)} = \frac{1}{7600} \approx 0.132\ ms/flow$

Throughput and Latency Results:

- **Throughput**:
  - CNN: 8,900 $flows/second$
  - LSTM: 7,600 $flows/second$
- **Latency**:
  - CNN: 0.112 $ms/flows$
  - LSTM: 0.132 $ms/flows$

Insights

1. **Processing Speed**:

   The CNN model processes flow faster, with an average throughput of 8,900 flows/second compared to 7,600 flows/second for LSTM. This speed advantage makes CNN more suitable for high-throughput, real-time applications in IoT environments.

2. **Latency Comparison**:

   CNN has a lower average inference latency (0.112 ms/flow) compared to LSTM (0.132 ms/flow). The reduced latency is attributed to CNN's parallelizable computations, which are well-suited for GPU acceleration.

3. **Resource Constraints**:

   LSTM's sequential computation structure results in higher latency and could become a bottleneck in resource-constrained environments where real-time responses are critical.

These results demonstrate that both CNN and LSTM models are feasible for real-time IoT deployments. However, CNN's lower latency and higher throughput make it a better choice for scenarios requiring rapid DDoS detection and mitigation.

V. Conclusion

This study investigated the application of deep learning models, specifically Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs), for detecting Distributed Denial of Service (DDoS) attacks in IoT networks. The proposed system analysed individual IP flow records to enable real-time detection of malicious traffic while minimizing disruptions to legitimate network communication. Using the comprehensive CICDDoS2019 dataset, we evaluated the performance of both models in identifying a wide range of DDoS attack types. The results demonstrated that CNNs outperformed LSTMs in terms of accuracy, precision, and F1-score, showcasing their ability to extract spatial features from high-dimensional flow data and identify attack signatures effectively. LSTMs, on the other hand, excelled at capturing temporal dependencies, making them suitable for detecting evolving attack patterns. Both models achieved sufficient throughput to handle real-time traffic, validating their feasibility for practical deployment in IoT environments.

The findings of this research highlight the potential of deep learning techniques to enhance the security of IoT networks by enabling accurate and timely DDoS detection. CNNs, with their higher throughput and precision, are particularly well-suited for high-throughput real-time applications, while LSTMs can be advantageous in scenarios requiring a deeper understanding of sequential traffic behaviour. This work underscores the need for robust and efficient detection mechanisms to protect IoT networks from increasingly sophisticated cyber threats.

References:

[1] Yoon, S., Kim, J., 2017. Remote security management server for iot devices. In: 2017 International Conference on Information and Communication Technology Convergence. ICTC, pp. 1162–1164. https://doi.org/10.1109/ICTC.2017.8190885.

[2] Bera, S., Misra, S., Roy, S.K., Obaidat, M.S., 2018. Soft-wsn: software-defined wsn management system for iot applications. IEEE Systems Journal 12, 2074–2081. https://doi.org/10.1109/JSYST.2016.2615761.
[3] da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R., de Albuquerque, V.H.C., 2019. Internet of things: a survey on machine learning-based intrusion detection approaches. Comput. Network. 151, 147–157. https://doi.org/10.1016/j. Comnet.2019.01.023.
[4] Hajiheidari, S., Wakil, K., Badri, M., Navimipour, N.J., 2019. Intrusion detection systems in the internet of things: a comprehensive investigation. Comput. Network. 160, 165–191. https://doi.org/10.1016/j.comnet.2019.05.014
[5] Lopez-Martin, M., Carro, B., Lloret, J., Egea, S., Sanchez-Esguevillas, A., 2018. Deep learning model for multimedia quality of experience prediction based on network flow packets. IEEE Commun. Mag. 56, 110–117. https://doi.org/10.1109/ MCOM.2018.1701156.
[6] Aldweesh, A., Derhab, A., Emam, A.Z., 2020. Deep learning approaches for anomalybased intrusion detection systems: a survey, taxonomy, and open issues. Knowl. Base Syst. 189, 105124. https://doi.org/10.1016/j.knosys.2019.105124. http://www.sci encedirect.com/science/article/pii/S0950705119304897.
[7] Cho, K., van Merri¨enboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734. https://doi.org/10.3115/v1/D141179.
[8] Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology. ICCST, pp. 1–8. https:// doi.org/10.1109/CCST.2019.8888419.
[9] Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., Yang, B., 2016. Predicting network attack patterns in sdn using machine learning approach. In: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks. NFV-SDN, pp. 167–172. https://doi.org/10.1109/NFV-SDN.2016.7919493.
[10] Kornycky, J., Abdul-Hameed, O., Kondoz, A., Barber, B.C., 2017. Radio frequency traffic classification over wlan. IEEE/ACM Trans. Netw. 25, 56–68. https://doi.org/ 10.1109/TNET.2016.2562259.
[11] Guo, Y., Ji, T., Wang, Q., Yu, L., Min, G., Li, P., 2020. Unsupervised anomaly detection in iot systems for smart cities. IEEE Transactions on Network Science and Engineering 1. https://doi.org/10.1109/TNSE.2020.3027543.
[12] Khan, I.A., Pi, D., Yue, P., Li, B., Khan, Z.U., Hussain, Y., Nawaz, A., 2020. Efficient behaviour specification and bidirectional gated recurrent units-based intrusion detection method for industrial control systems. Electron. Lett. 56, 27–30. https:// doi.org/10.1049/el.2019.3008.
[13] Kao, J., Jiang, J., 2019. Anomaly detection for univariate time series with statistics and deep learning. In: 2019 IEEE Eurasia Conference on IOT, Communication and Engineering. ECICE, pp. 404–407. https://doi.org/10.1109/ ECICE47484.2019.8942727.
[14] Qin, G., Chen, Y., Lin, Y., 2018. Anomaly detection using lstm in ip networks. In: 2018 Sixth International Conference on Advanced Cloud and Big Data. CBD, pp. 334–337. https://doi.org/10.1109/CBD.2018.00066.
[15] Xie, X., Wang, B., Wan, T., Tang, W., 2020. Multivariate abnormal detection for industrial control systems using 1d cnn and gru. IEEE Access 8, 88348–88359. https://doi.org/10.1109/ACCESS.2020.2993335.
[16] Qu, Z., Su, L., Wang, X., Zheng, S., Song, X., Song, X., 2018. A unsupervised learning method of anomaly detection using gru. In: 2018 IEEE International Conference on Big Data and Smart Computing. BigComp, pp. 685–688. https://doi.org/10.1109/ BigComp.2018.00126
[17] Liu, S., Chen, X., Peng, X., Xiao, R., 2019. Network log anomaly detection based on gru and svdd. In: 2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking. ISPA/BDCloud/SocialCom/SustainCom, pp. 1244–1249. https://doi.org/10.1109/ISPA-BDCloudSustainCom.
[18] Shuying, Chang, Qiu, Xuesong, Gao, Zhipeng, Qi, Feng, Liu, Ke, 2010. A flow-based anomaly detection method using entropy and multiple traffic features. In: 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology. IC-BNMT, pp. 223–227. https://doi.org/10.1109/ICBNMT.2010.5705084.
[19] Berezinski, ´ P., Jasiul, B., Szpyrka, M., 2015. An entropy-based network anomaly detection method. Entropy 17, 2367–2408. https://doi.org/10.3390/e17042367
[20] McDermott, C.D., Majdani, F., Petrovski, A.V., 2018. Botnet detection in the internet of things using deep learning approaches. In: 2018 International Joint Conference on Neural Networks. IJCNN, pp. 1–8. https://doi.org/10.1109/IJCNN.2018.8489489.
[21] Patil, V. T., & Deore, S. S. (2024). IoT-Guardian: Advanced detection of DDoS attacks in IoT systems using CNNs. Journal of Electrical Systems, 20(10s), 3855–3864. DOI: https://doi.org/10.52783/jes.5943
[22] Sharafaldin, I., Gharib, A., Habibi Lashkari, A., Ghorbani, A., 2017. Towards a reliable intrusion detection benchmark dataset. Software Networking 2017, 177–200. https://doi.org/10.13052/jsn24459739.2017.009.
[23] Patil, V. T., & Deore, S. S. (2024). DDoS attack detection: Strategies, techniques, and future directions. Journal of Electrical Systems, 20(9s), 2030–2046. https://doi.org/10.52783/jes.4808

[24] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
[25] Patil, P. R., & Patil, V. T. (2020). Smart forest: An IoT-based forest safety and conservation system. International Journal of Scientific & Technology Research, 9(03).
[26] Patil, V. T., & Deore, S. S. (2023). A study of DDoS attack detection methods. Journal of Data Acquisition and Processing, 38(3), 3583–3591.
[27] Patil, V. T., & Deore, S. S. (2023). Strategies and horizons in DDoS attack detection: An analytical and predictive study. Available at SSRN. https://doi.org/10.2139/ssrn.4727005
[28] Patil, V. T., & Chandel, G. S. (2014). Implementation of TPA and data integrity in cloud computing using RSA algorithm. International Journal of Engineering Trends and Technology, 12, 85–93. https://doi.org/10.14445/22315381/IJETT-V12P215
[29] Shailesh S. Deore, Dr. Ashok Narayan Patil (Aug. 2012), Systematic Review of Energy- Efficient Scheduling Techniques in Cloud Computing, International Journal of Computer Application, ISSN 09758887 Vol. 52, No. 15, pp. 10-14, DOI 10.5120/8275-1877, http://www.ijcaonline.org/archives/volume52/ number15.
[30] Shailesh S. Deore, Dr. Ashok Narayan Patil (October 2012), Energy- Efficient Scheduling Scheme for Virtual Machines in Cloud Computing, International Journal of Computer Application, ISSN 0975-8887 Vol. 56, No. 10, pp. 19-25, DOI 10.5120/ 8926-2999, http://www.ijcaonline.org/archives/volume56/ number10.
[31] Shailesh S. Deore, Dr. Ashok Narayan Patil (Jan. 2013), Energy- Efficient Job Scheduling and Allocation Scheme for Virtual Machines in Cloud Computing, International Journal of Applied Information System, Vol.5, No.1, pp.56-60, DOI 10.5120/ijais-450842, http://www.ijais.org/archives/volume5/number1
[32] Shailesh S. Deore (March 2016), Comparison of Energy Efficient Scheduling Schemes for Private cloud Environments, International Journal of Advance Research in Computer and Communication Engineering, Vol.5 Issue3, March 2016, DOI 10.17148/IJARCCE.2016.5380, https://www.ijarcce.com/volume-5-issue3.html
[33] Shailesh S. Deore (April 2016), Joulemeter: Power Measurement for Virtual Machine in Private cloud Environments, International Advance Research Journal Science Engineering and Technology Vol.5 Issue3, March 2016, DOI 10.17148/IARJSET.2016.3414, https://iarjset.com/recent-issue-april-2016/

Vinay T. Patil received his B.E. degree in Information Technology from North Maharashtra University, Maharashtra, India, in 2007, and his MTech degree in Software Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Madhya Pradesh, India, in 2014. He is currently pursuing a Ph.D. in Computer Engineering from Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, with a focus on detecting Distributed Denial of Service (DDoS) attacks in IoT networks using deep learning techniques. With over 16 years of professional experience since July 2008, he has served as an Assistant Professor at PSGVP Mandal's D. N. Patel College of Engineering and Ajeenkya D. Y. University, Pune. Currently, he works as a Senior AI/ML Engineer at Pixonate Lab Pvt. Ltd., specializing in designing and implementing innovative machine learning solutions. His research interests include Cloud Computing, Cybersecurity, Machine Learning, Deep Learning, and the Internet of Things (IoT).

Dr. Shailesh Deore, born in Dhule, Maharashtra, India, in 1982, is a distinguished educator and researcher in the field of Computer Engineering. He obtained his B.E. degree from North Maharashtra University in 2003 and earned his Ph.D. in Computer Engineering from Shri JJT University, Rajasthan, India, in 2014. Dr. Deore began his academic career in 2004 as a Lecturer in the Department of Computer Engineering at SSVP's B.S. Deore College of Engineering, Dhule. In 2009, he was promoted to Assistant Professor, and since 2016, he has been serving as an Associate Professor, contributing significantly to both academic and research initiatives. With an impressive 20 years of teaching experience, Dr. Deore's expertise spans various advanced areas, including cloud computing, energy-efficient job scheduling algorithms in private cloud environments, machine learning, and data mining. His dedication to staying at the forefront of technological innovation drives his research and teaching endeavours, making him a valuable contributor to the field of Computer Engineering.