DOI: https://doi.org/10.52756/ijerr.2024.v46.020 ***Int. J. Exp. Res. Rev.*, Vol. 46: 253-272 (2024)**

Original Article | *Peer Reviewed* | *Open Access*

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in
ISSN 2455-4855

Enhanced Network Defense: Optimized Multi-Layer Ensemble for DDoS Attack Detection

Deepak Singh Rajput* and Arvind Kumar Upadhyay

Department of Computer Science and Engineering, Amity University, Gwalior, Madhya Pradesh, India

E-mail/Orcid Id:

DSR, deepakgyangit@gmail.com, https://orcid.org/0009-0001-3737-049X;
AKU, akupadhyay@gwa.amity.edu, https://orcid.org/0000-0001-5808-9691

Article History:

Received: 20th Aug., 2024
Accepted: 22nd Dec., 2024
Published: 30th Dec., 2024

Keywords:
DDoS, XGBoost, AdaBoost, RF, SVM, CNN, LSTM, CICDDoS2019

How to cite this Article:
Deepak Singh Rajput and Arvind Kumar Upadhyay (2024). Enhanced Network Defense: Optimized Multi-Layer Ensemble for DDoS Attack Detection.. *International Journal of Experimental Research and Review*, *46*, 253-272.
DOI:
https://doi.org/10.52756/ijerr.2024.v46.020

Abstract: In today's digitally connected world, Distributed Denial of Service (DDoS) attacks remain a formidable challenge, undermining the stability of network infrastructures and demanding robust detection strategies. This research explores advanced methodologies for DDoS detection by conducting a comparative analysis of machine learning and deep learning approaches using the CICDDoS2019 dataset. Initially, a hybrid machine learning framework is implemented, integrating K-Means clustering for pre-labeling the dataset and employing supervised models such as Random Forest (RF), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Support Vector Machine (SVM), and Artificial Neural Network (ANN). This approach achieves an accuracy of 99.46%, showcasing its effectiveness while highlighting challenges like manual feature selection and limited scalability for complex datasets. A novel hybrid deep learning architecture is proposed to overcome these challenges, combining Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal sequence learning. This automated feature extraction mechanism eliminates reliance on manual intervention, ensuring adaptability to evolving attack patterns. The proposed CNN-LSTM model demonstrates an impressive accuracy of 99.84%, significantly outperforming traditional machine learning models. Additionally, the model's adaptability and resilience against dynamic attack behaviours position it as a reliable solution for real-time DDoS mitigation. This study emphasizes the growing relevance of deep learning techniques in enhancing cyber security and underscores the potential of hybrid architectures in effectively detecting and mitigating modern cyber threats. The findings provide valuable insights into developing scalable, high-performance systems capable of addressing the ever-evolving nature of DDoS attacks.

Introduction

In the past few years, the landscape of cyber-attacks has evolved significantly, largely driven by the inherent vulnerabilities of various internet-connected devices, making them prime targets for malicious exploitation. These cyber-attacks often compromise sensitive data and pose substantial threats to essential infrastructures across industries. Among the myriad types of cyber-attacks, Distributed Denial of Service (DDoS) attacks have emerged as one of the most pervasive and damaging due to their ability to cause widespread service disruption. The unique complexity and rapid proliferation of DDoS attacks distinguish them from other forms of cyber threats, presenting new challenges for timely and accurate detection. As such, the development of robust DDoS detection mechanisms has become a critical focus of ongoing research. DDoS attacks typically operate through distributed vectors, overwhelming target systems or networks with massive volumes of traffic, intending to render critical services inoperable. Frequently, these attacks are aimed at high-value targets such as financial institutions, corporate websites, e-commerce platforms, and payment processing systems. The growing scale of DDoS attacks is evident in the data, which shows a 192% surge in the size of the largest recorded attacks and an 81% rise in maximum attack intensity. In June 2020, the peak attack volume reached 12 Gbps, a significant increase compared to the 11 Gbps recorded in the same

*Corresponding Author: deepakgyangit@gmail.com

period of 2019 (Alanazi et al., 2022). Given these alarming trends, the need for effective and scalable DDoS detection solutions has become increasingly urgent. A visual representation of the various types of DDoS attacks is provided in Figure 1.

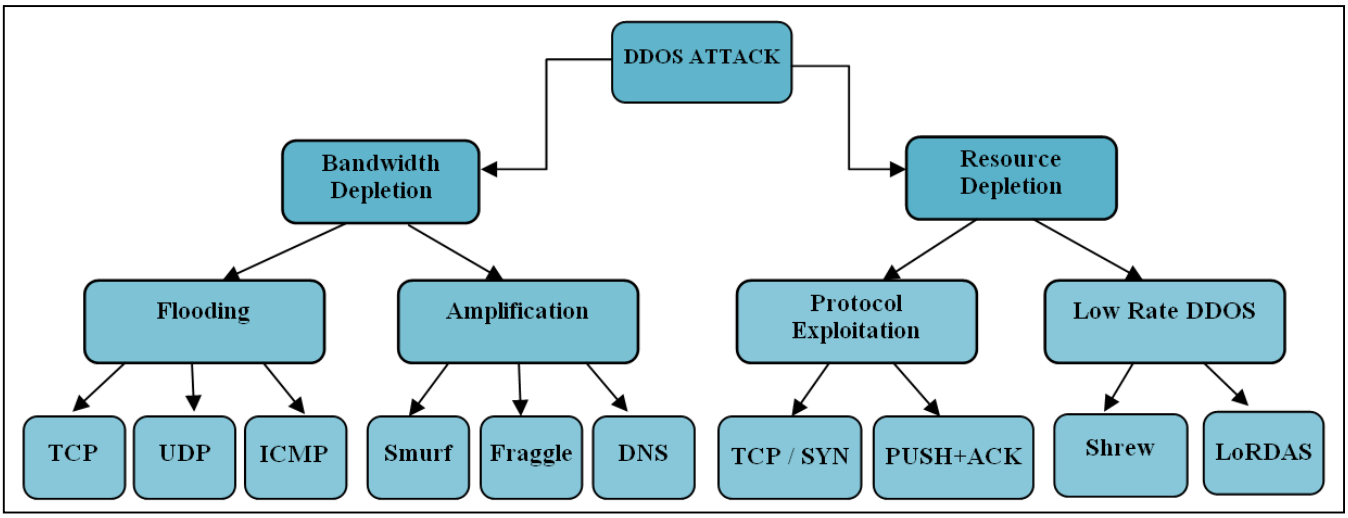

Figure 1. Categorization of DDoS attacks (Ramzan et al., 2023).

sAs the threat of Distributed Denial of Service (DDoS) attacks continues to escalate, the necessity of building resilient defense mechanisms becomes increasingly urgent. An effective cyber-attack detection system must be powered by a deep learning (DL) model that ensures high accuracy while minimizing the rate of false positives. Both supervised and unsupervised learning techniques have been leveraged to enhance model performance in cybersecurity applications. Over recent years, numerous machine learning (ML) and DL-based approaches have been adopted to combat DDoS attacks. These methods encompass a variety of algorithms, including Decision Trees (DT), Logistic Regression (LoR), Linear Regression (LR), Naive Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), XGBoost, AdaBoost, and sophisticated deep models such as ResNet, Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNNs). These models have been extensively trained and validated on datasets such as CICDDoS2019, which is widely regarded for DDoS attack detection tasks. Additionally, other prominent datasets, including CICIDS2017, KDD, CAIDA 2007, IoT NI, BoT IoT, MQTT, MQTTset, IoT-23, IoT-DS2 and UNSWNB15, have been employed in DDoS detection research, further contributing to the ongoing efforts in enhancing cyber resilience (Seifousadati et al., 2021; Sharma and Shakya, 2022; Sharma et al., 2024a,b &c).

Incorporating deep learning (DL) architectures and developing innovative network models present a significant advancement in detecting Distributed Denial of Service (DDoS) attacks, thereby fortifying modern cyber defence strategies. With the surge of big data, the detection of DDoS threats is becoming increasingly intertwined with advancements in data analytics and large-scale data processing. Given the massive amounts of traffic data in cyber security, identifying complex, multi-faceted patterns within network traffic presents a substantial challenge. While effective on smaller datasets, traditional machine learning (ML) algorithms are often prone to high false-positive rates and misclassification issues, ultimately complicating security management. This has led to the increased reliance on advanced DL models, which outperform ML techniques in terms of detection accuracy and scalability. DL excels in tackling the computational and data-scale challenges pervasive in cyber security due to its ability to automatically extract and learn complex features from large datasets. This capability makes DL especially well-suited for identifying cyber threats, including DDoS attacks, by detecting intricate patterns across vast network data (Effah et al., 2024). One of the core strengths of DL models lies in their feature learning capacity, enabling them to extract, classify, and analyze data, even when some information is incomplete or obscured. By employing multiple hidden layers and complex mathematical operations, DL models can provide a higher level of abstraction for feature extraction. This study introduces novel DL models, integrating Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to address both binary and multi-class DDoS attack detection. The combined architecture enhances the ability to learn temporal and spatial features from network traffic data, thereby offering an improved and scalable solution for modern cyber defence systems (Zhou et al., 2022).

The primary purpose of this study is to develop and evaluate advanced methodologies for the detection of Distributed Denial of Service (DDoS) attacks, which pose

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

significant threats to network infrastructures. This research aims to address the limitations of traditional detection systems by leveraging state-of-the-art machine learning techniques to enhance detection accuracy, scalability, and adaptability to dynamic attack patterns. Propose and implement a novel deep learning-based hybrid parallel architecture, combining Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal sequence learning.

Problem Statement

The continuous escalation in both frequency and complexity of Distributed Denial of Service (DDoS) attacks presents a critical challenge to the stability and security of network infrastructures globally. Traditional detection mechanisms, largely dependent on static, rule-based systems, struggle to keep pace with the dynamic nature of modern DDoS attacks, often resulting in elevated false positive rates and delayed mitigation efforts. The need for an intelligent, adaptive detection system that can efficiently identify and counteract such threats has become increasingly pressing.

Research Objective

The primary objective of this research is to address the growing complexity and frequency of DDoS attacks, which significantly challenge network functionality and security. The study seeks to evaluate and enhance the efficacy of detection models by integrating advanced artificial intelligence and deep learning techniques. It aims to:

Leverage Machine Learning Algorithms

Utilize models such as Random Forest, Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Support Vector Machine (SVM) and Artificial Neural Network (ANN) to improve detection accuracy while minimizing false positives and false negatives.

Develop feature selection strategies to better distinguish between legitimate and malicious traffic.

Develop a Hybrid Deep Learning Model

Create a CNN-LSTM framework to capitalize on Convolutional Neural Networks (CNN) to extract spatial features and Long Short-Term Memory (LSTM) to capture temporal dependencies in network traffic.

Optimize the model with parallel execution of feature maps through dense layers and LSTM pathways.

Enhance Data Processing

Implement robust data pre-processing techniques for cleaning and normalizing the CICDDoS2019 dataset, ensuring high-quality input data for training and testing.

Evaluate Performance

Conduct a comprehensive analysis of the hybrid model's performance using key metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

Benchmark results against traditional machine learning models to demonstrate superiority.

Achieve Scalability and Real-Time Detection

Test the CNN-LSTM model's capacity for real-time detection of DDoS attacks, focusing on its scalability to manage large-scale network traffic effectively.

By achieving these objectives, the research intends to provide a robust, scalable, and efficient solution for DDoS attack detection, underscoring the advantages of hybrid machine learning over traditional machine learning approaches.

Contribution of this Study

This research introduces several key contributions to the domain of DDoS detection:

Innovative Machine Learning Models

Two novel deep learning models are presented, utilizing CNN and LSTM hybrid architecture. Unlike conventional architectures where LSTM outputs are passed to CNN layers, the proposed models feature a parallel execution where both LSTM and CNN layers process the same input data concurrently. The outputs are combined through an element-wise addition operation applied across dense layers as well, resulting in a more efficient and accurate model.

Binary and Multi-class Classification

The proposed models support both binary and multi-class classification. The binary classification model identifies traffic as benign or malicious, while the multi-class model categorizes traffic into 12 distinct classes, covering 11 attacks and one benign class.

Evaluation with CICDDoS2019 Dataset

The models were thoroughly tested using the CICDDoS2019 dataset, which is known for its comprehensive nature in DDoS research. The models exhibited high detection and recognition accuracy, outperforming traditional detection approaches, particularly on previously unseen data.

Lightweight Design

The proposed models are designed to be lightweight, with fewer trainable parameters compared to many existing deep learning models. This makes them suitable for deployment in environments with limited computational resources.

Performance Benchmarking

The performance of the proposed models was benchmarked against existing solutions and baseline models. The results demonstrate significant

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

improvements in detection accuracy and processing efficiency.

Literature Review

Abreu et al. (2020) introduced an innovative method for robust DDoS attack detection through a multidimensional approach. They began with Higher Order Singular Value Decomposition (HOSVD) to filter out average common features among dataset instances. This filtered data was then processed by machine learning classification algorithms, categorizing traffic as legitimate or indicative of a DDoS attack. The proposed method outperforms low-rank approximations using the random forest algorithm, achieving 98.94% accuracy, 97.70% detection rate, and 4.35% false alarm rate with a 30% corrupted dataset. Under error-free conditions, the method excels further with 99.87% accuracy, 99.86% detection rate and 0.16% false alarm rate using the gradient boosting classifier. This innovative approach shows great promise for effective DDoS attack detection.

Sindian et al. (2020) introduced the Enhanced Deep Sparse Autoencoder EDSA framework for DDoS attack detection, focusing on minimising cost. They used a sparse autoencoder for data extraction and a softmax layer to classify traffic as malicious or benign. They employed metrics like detection rate, overall accuracy and precision to assess the model's accuracy and effectiveness in intrusion detection. When tested on the CICDDoS2019 dataset, their method achieved a remarkable detection accuracy of 98% with an impressively low false positive rate of 1.4%. The EDSA framework demonstrates its potential for robust DDoS attack detection with high precision and minimal false alarms.

Polat et al. (2020) introduced a method for DDoS attack detection in Software-Defined Networking (SDN) by employing machine learning models. They initiated the process by extracting specific features from SDN data under normal conditions and during DDoS attacks. They generated a new dataset through feature selection techniques to improve model efficiency, interpretability, and training time. They conducted training and testing on two datasets, one with feature selection and one without, using SVM, NB, ANN and KNN models. Their results highlighted the effectiveness of applying wrapper feature selection in conjunction with a KNN classifier, achieving a high accuracy rate of 98.3% in DDoS attack detection. This approach showcases the potential of machine learning and feature selection to enhance DDoS detection in SDN while reducing processing overhead and time consumption.

Halladay et al. (2022) proposed the model with the efficacy of 25 time-based features to detect and classify 12 types of DDoS attacks using binary and multiclass classification. Furthermore, they conducted experiments to compare the performance of eight traditional machine learning classifiers and one deep learning classifier using two different scenarios. Their findings show that the majority of models achieved over 99% accuracy in both control and time-based experiments for detecting DDoS attacks while also demonstrating around 70% accuracy in classifying specific DDoS attack types.

An efficient hybrid deep neural network model, integrating XGBoost for feature selection with CNN-LSTM architecture for DDoS attack detection, was introduced by Devan et al. (2020). This approach is applied to SDN-based IoT networks, achieving impressive results in terms of both accuracy (99.5%) and latency (0.179 ms), illustrating its capability to detect and classify attacks with minimal computational overhead.

In another study, Jiang et al. (2020) proposed a deep learning-based hybrid model using CNN and Bidirectional LSTM (CNN-BiLSTM) to detect DDoS attacks in IoT networks. The model demonstrates high performance with an accuracy of 99.76% when tested on the CICIDS2017 dataset, making it a robust solution for identifying anomalous network behavior. This model was assessed using multiple performance metrics, confirming its efficiency in cyber-attack detection.

In Vinayakumar et al. (2019), a deep learning framework using CNN and LSTM is proposed for detecting DDoS attacks by analyzing spatial and temporal patterns in network traffic data. When tested on real-world datasets, the model exhibits strong accuracy and generalization capabilities, affirming the benefit of combining CNN and LSTM architectures for complex attack detection.

Research presented by Abid et al. (2024) highlights a hybrid approach involving CNN and Recurrent Neural Networks (RNN) for DDoS detection. The hybrid model effectively captures both feature correlations and temporal dependencies, particularly in large-scale network traffic data, improving detection accuracy in IoT environments. The study underscores the potential of hybrid deep learning models in enhancing network security.

A CNN-LSTM-based approach for detecting DDoS attacks is proposed by Dangi et al. (2021), combining the feature extraction strengths of CNN with the sequence modeling capabilities of LSTM. The model was rigorously tested using real-world datasets, demonstrating superior performance compared to traditional detection

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

Table 1. Systematic review of DDoS traffic attack detection previous works.

ef.	Algorithms	Dataset	Results	Accuracy	Limitation	Advantages
(Abreu et al., 2020)	HOSVD + Random Forest, Gradient Boosting	Custom Dataset	Robust detection under corrupted and error-free conditions, low false alarm rate.	99.87% (error-free)	Requires complex preprocessing using HOSVD.	High detection accuracy performs well even with corrupted datasets.
(Sindian et al., 2020)	Enhanced Deep Sparse Autoencoder (EDSA) Framework	CICDDoS 2019	Accurate DDoS detection with low false positive rates.	98%	Limited scalability and real-time capability focus.	High precision and minimal false alarms; cost-effective framework.
(Polat et al., 2020)	KNN + Wrapper Feature Selection	Custom SDN Dataset	Enhanced efficiency and accuracy in SDN-based DDoS detection.	98.3%	Relies heavily on feature selection, increasing preprocessing complexity.	Reduced processing overhead and effective classification.
(Halladay et al., 2022)	Multiple ML Classifiers (8 ML, 1 DL)	Custom Time-Based Data	High accuracy in binary detection; moderate performance in multi-class classification.	Over 99% (binary), ~70% (multi-class)	Struggles with accurate classification of specific attack types in multi-class detection.	Explores the effectiveness of diverse classifiers' strong binary classification performance.
(Devan et al., 2020)	XGBoost + CNN-LSTM	SDN-based IIoT	High accuracy and low latency	99.50%	Time cost of 0.179 ms could be significant in real-time applications	Combines feature selection and deep learning for efficient detection
(Jiang et al.,2020)	CNN-BiLSTM	CICIDS 2017	High accuracy in DDoS detection	99.76%	Potential overfitting due to high model complexity	High accuracy and comprehensive assessment against common criteria
(Vinayakumar et al., 2019)	CNN-LSTM	CICDDoS 2019	Achieved robust DDoS detection on real-world datasets.	99.24%	High computational complexity, making it less suitable for real-time systems.	Effective feature extraction using CNN and temporal pattern recognition with LSTM.
(Abid et al., 2024)	CNN-LSTM	CICDDoS 2019	Ability to acquire and classify complex spatial unprocessed network traffic data.	99.40%	Could Improve the resilience and effectiveness of detection systems	Extraordinary capacity to thoroughly analyze data and accurately detect DDoS attacks highlights its effectiveness
(Dangi et al., 2021)	CNN-LSTM	CICDDoS 2019	Improved detection of DDoS attacks, especially in IoT environments.	98.93%	Struggles with large-scale real-time traffic due to resource consumption.	The hybrid CNN-RNN model captures both spatial and temporal dependencies effectively.

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

(Alzahrani et al., 2022)	CNN-BiLSTM	CICDDoS 2019	Demonstrated high detection rate across various network traffic types.	99.10%	Requires further optimization for deployment in large-scale networks.	Achieves high detection performance and efficiently handles complex network traffic.
(Zhang et al., 2020)	CNN-LSTM	CICDDoS 2019	Significantly improved detection accuracy on the CICIDS2017 dataset.	98.76%	Model performance may degrade when tested on unseen datasets due to overfitting.	Strong feature extraction capabilities using CNN and sequence learning with LSTM.
(Yin et al., 2021)	Hybrid CNN-LSTM	CICDDoS 2019	Demonstrated superior performance in terms of detection metrics.	99.28%	Computationally intensive, making real-time applications difficult.	High precision and recall rates; effective use of hybrid deep learning architectures.
(Gamal et al., 2022)	CNN-LSTM	CICDDoS 2019	Achieved significant improvements in the detection of sophisticated DDoS attacks.	99.12%	The hybrid model's complexity may result in slower processing times for real-time detection.	Capable of handling various DDoS attack types with high accuracy and efficiency.
(Woo et al., 2020)	CNN-LSTM	CICDDoS 2019	Detected multiple types of DDoS attacks with high accuracy on CICIDS2017.	98.85%	Model may not generalize well to different datasets, requiring retraining.	Efficient use of both CNN and LSTM to capture spatial and temporal features in network traffic data.
(Bhatt et al., 2021)	CNN-LSTM	CICDDoS 2019	Successfully identified DDoS attacks in IoT networks.	98.94%	Faces challenges in scaling the solution for large IoT environments.	Effective for IoT network security, the hybrid model ensures robustness in attack detection.

methods while addressing the challenge of computational efficiency in large-scale intrusion detection systems.

A deep learning method combining CNN and LSTM for detecting DDoS attacks is presented in Alzahrani et al. (2022). CNN extracts spatial features from network traffic data, while LSTM captures the sequential dependencies. This hybrid model significantly improves detection accuracy when tested on publicly available datasets, proving its effectiveness in handling complex, high-dimensional data in cybersecurity applications.

In Zhang et al. (2020), the authors present a hybrid CNN-LSTM architecture specifically designed to detect DDoS attacks in high-dimensional network traffic data. The model outperforms several existing detection techniques in terms of accuracy, precision, and recall, highlighting the advantages of deep hybrid models in cyber defense.

A hybrid CNN-LSTM model for DDoS attack detection is introduced by Yin et al. (2021), where CNN is responsible for feature extraction and LSTM handles time-series dependencies. When tested on multiple datasets, the model demonstrates enhanced performance over traditional detection methods, particularly in terms of accuracy and real-time detection capability.

Another study Gamal et al. (2022) investigates a deep learning method for DDoS detection, utilizing CNN and LSTM networks. The model is evaluated using the CICIDS2017 dataset, showing high detection accuracy for various types of DDoS attacks. This highlights the importance of combining spatial and temporal feature extraction for network intrusion detection.

Woo et al. (2020) propose a deep hybrid model that integrates CNN and LSTM for detecting DDoS attacks in IoT networks. The model efficiently captures spatial and temporal patterns, delivering high detection accuracy across multiple evaluation metrics. This study emphasizes the potential of hybrid deep learning architectures in addressing security challenges within IoT environments.

Lastly, a framework using CNN-LSTM for real-time DDoS detection in cloud and IoT networks is proposed by Bhatt et al. (2021). The model leverages CNN for

feature extraction and LSTM for capturing temporal dependencies, exhibiting high detection accuracy when tested on large-scale datasets. This research highlights the applicability of deep learning in dynamic and large-scale network environments.

Table 1 summarizes various hybrid deep learning models used for DDoS detection, highlighting each approach's algorithms, datasets, results, accuracy, limitations, and advantages.

Background

Cyber DDoS Attack

In today's digital landscape, cyber threats have seen an alarming increase, affecting individuals and organisations. According to a CSO report, the financial damage caused by cybercrime is expected to reach trillions of dollars annually by 2021 (Xu et al., 2022). Among these threats, Distributed Denial of Service (DDoS) attacks stand out as one of the most destructive forms of cybercrime. These attacks overwhelm network infrastructures by bombarding them with excessive traffic, causing severe degradation in performance, connectivity disruptions, or even complete service outages. DDoS attacks take advantage of vulnerabilities present in system architectures, applications, or communication protocols (Salih et al., 2024). The core mechanism behind DDoS attacks involves sending large volumes of malicious traffic to a target, ultimately depleting available bandwidth or overwhelming computational resources, which causes system outages. Such attacks are often coordinated from multiple sources simultaneously, generating a massive surge of requests that the targeted system cannot manage, resulting in crashes (Alasmari et al., 2023). DDoS strategies can be categorized into three primary types: volumetric attacks, which overload the network bandwidth; protocol attacks, which exploit weaknesses in the network layers; and application layer attacks, which flood servers with requests, preventing legitimate traffic from being processed. Often, attackers combine multiple strategies, making DDoS detection and mitigation extremely complex for security systems (Borgiani et al., 2021). One method for mitigating DDoS attacks is real-time packet inspection, where traffic is continuously monitored to detect and discard harmful packets. When combined with deep learning, this process becomes more robust, as it automates the identification of malicious traffic and protects systems against both volumetric and protocol-based attacks with minimal human intervention (Salehi et al., 2024).

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a specialized subset of deep learning architectures, predominantly used in structured data analysis tasks. CNNs excel in recognizing patterns by progressively learning hierarchical feature representations from input data. A standard CNN architecture comprises three primary layers: convolutional layers, pooling layers, and fully connected layers, each playing a key role in identifying increasingly intricate data patterns. While traditionally used in image recognition tasks, CNNs are equally effective in analyzing one-dimensional data, such as network traffic or time-series sequences, particularly in cybersecurity applications. The architecture of a CNN is defined by hyperparameters such as the number of convolutional layers, the size and number of filters, and the stride applied in the pooling layers (Sumathi et al., 2022). As the filter moves across the input data, it captures localized features sequentially, making CNNs highly suitable for classification, regression, and time-series forecasting tasks. By combining convolutional layers with pooling and fully connected layers, CNN models effectively classify and predict patterns within complex datasets. One of CNN's most notable advantages is its ability to automatically learn essential features from the data, thereby reducing the need for manual feature selection. In recent years, CNNs have been increasingly used in cybersecurity to detect malicious activities, such as DDoS attacks. CNNs extract hierarchical features from network traffic data, enabling precise detection by processing and classifying sequential input with remarkable accuracy (Andresini et al., 2020). In this research, our goal is to utilize CNN layers to detect DDoS attacks by extracting meaningful features from sequential network traffic data and processing them through pooling layers and fully connected layers for improved classification accuracy (Hossain et al., 2020).

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a distinct form of Recurrent Neural Networks (RNN) that are adept at handling long-term dependencies within sequential data, overcoming the typical issues of vanishing or exploding gradients encountered during RNN training. These gradient issues significantly restrict traditional RNNs from retaining important information over extended time periods. The unique architecture of LSTMs addresses this limitation through a sophisticated gating mechanism, which allows the network to selectively preserve or discard information as necessary. While LSTMs require higher computational resources due to their complex gating mechanisms, they often

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

outperform simpler RNNs, achieving superior results without requiring significantly more trainable parameters (Behal et al., 2021). One of the defining elements of an LSTM is its cell state, a continuous horizontal path through which information flows across different time steps. Three distinct gates manage the regulation of this flow: the input gate, the forget gate, and the output gate. These gates work in tandem to ensure that only the most relevant data is retained in the cell state while irrelevant information is discarded. This mechanism enables LSTMs to maintain crucial temporal information across long sequences, making them highly effective in applications such as DDoS attack detection (Musa et al., 2024).

Dataset

Overview of the CICDDoS-2019 Dataset

The CIC-DDoS2019 dataset was developed by the Canadian Institute for Cybersecurity (CIC) in collaboration with the University of New Brunswick. It is a comprehensive dataset designed to facilitate cybersecurity research, specifically for detecting and mitigating Distributed Denial of Service (DDoS) attacks. This dataset offers a vast collection of network traffic samples, containing over 50 million malicious flows and more than 100,000 benign traffic samples. It covers 13 different DDoS attack types alongside benign network traffic, providing a valuable resource for researchers aiming to study DDoS behavior in network traffic

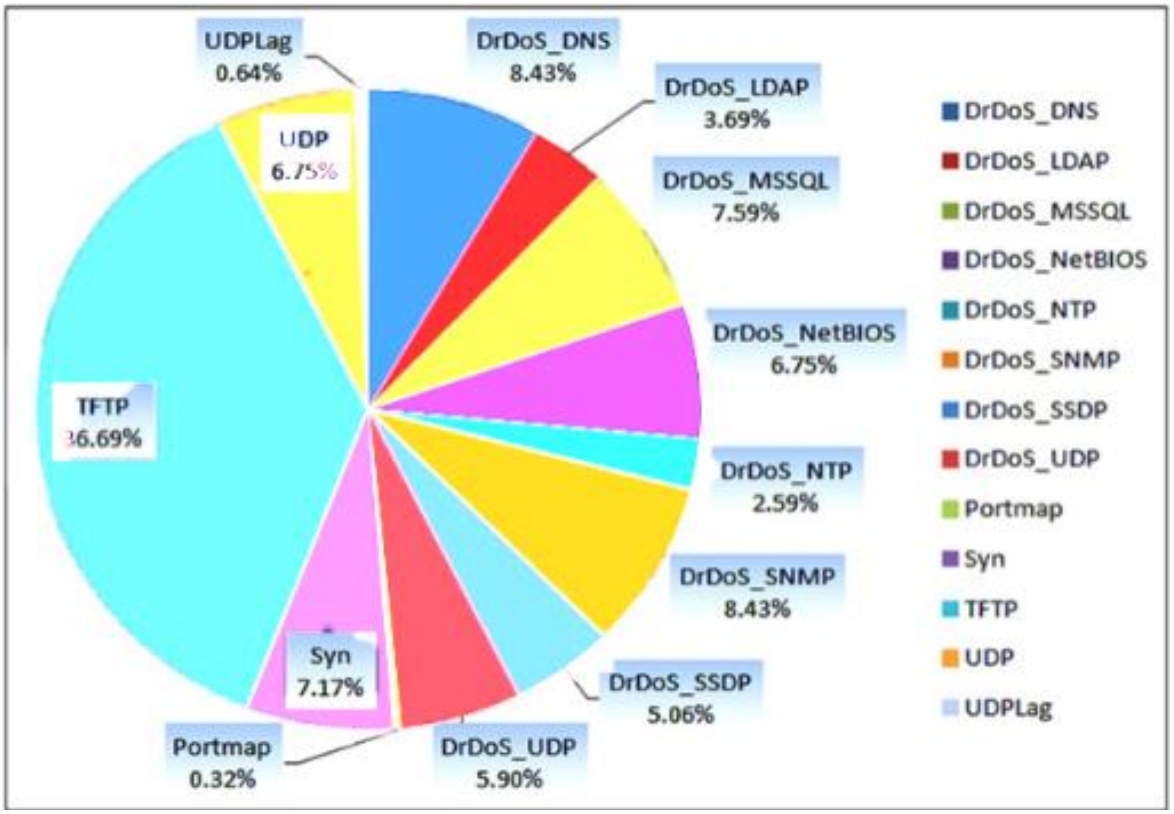

Figure 2. Weightage % of each attack in CICDDOS2019 dataset.

In the context of DDoS detection, LSTMs excel due to their ability to process and learn from sequential network traffic data, capturing complex patterns that indicate potential cyber-attacks. Their ability to identify long-term dependencies within network traffic allows LSTMs to detect abnormal traffic behaviours that often precede or accompany DDoS attacks. This adaptability, combined with the network's inherent capability to extract meaningful features from vast amounts of time-series data, renders LSTMs essential in advanced network defence frameworks. As a result, LSTM-based models are particularly well-suited for identifying and predicting emerging cyber threats in real time, enhancing the efficacy of network security systems.

(DDoS, 2019). The data includes 87 distinct features, such as packet size, duration, and protocol type, extracted using the CICFlowMeter tool. An additional label column (88th column) indicates whether the traffic sample is benign or represents a specific DDoS attack category (Saxena et al., 2020).

#Step 1: Define the Input Data (Traffic Features)

Each traffic sample is represented by a feature matrix, X, with n traffic samples and 87 features, where the feature values are arranged in rows corresponding to the samples and columns corresponding to the individual features. The input data can be formalized as follows:

Matrix X represents the feature data:

$X=\{x_{ij}\}$, i=1,2,…,n; j=1,2,…,87

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

where: x_{ij} is the value of the j^{th} feature for the i^{th} traffic sample, n is the total number of traffic samples in the dataset.

Label Y represents the corresponding attack or benign traffic with y_i=0 denoting benign traffic and yi=1 to 13 denoting specific DDoS attack categories:

$Y=\{ y_i \}, \quad y_i \in \{0,1,2,\ldots,13\}$

#Step 2: Attack Type Identification

The predicted label $\hat{Y}$ corresponds to one of the attack types, which can be either benign traffic or one of the DDoS categories. Let C represent the set of all possible DDoS attacks and benign traffic:

C= {Benign, LDAP, MSSQL, NetBIOS,…, SYN DDoS}

Each traffic sample is assigned to one of these categories based on the predicted label $\hat{Y}$.

#Step 3: Define the Attack Types

The DDoS attacks in the dataset can be broadly categorized into two primary types:

1. Reflection-based DDoS Attacks (R): These attacks leverage servers or services that reflect responses to the victim's server. Based on the transport protocol, reflection-based attacks can be further divided into: R = {TCP, UDP, TCP/UDP}

II. Exploitation-based DDoS Attacks (E): These attacks exploit vulnerabilities in systems or protocols. Similar to reflection-based attacks, they can be categorized by the protocol type: E = {TCP, UDP}

$A=R \cup E$

Where: A is set of all DDoS attack types, R is the set of reflection-based attacks, E is the set of exploitation-based attacks.

Figure 2 graphically displays the percentage distribution of each attack type present in the CICDDoS2019 dataset.

Hybrid Methodology

DDoS attacks present a substantial menace to network infrastructure, potentially causing service disruptions by inundating target systems with malicious traffic. A potent strategy to counter these risks revolves around constructing a DDoS attack detection system that melds unsupervised and supervised machine learning methodologies. This hybrid approach harnesses the robust attributes of techniques to promptly discern and thwart these assaults, thereby reinforcing network security and safeguarding the availability and dependability of services.

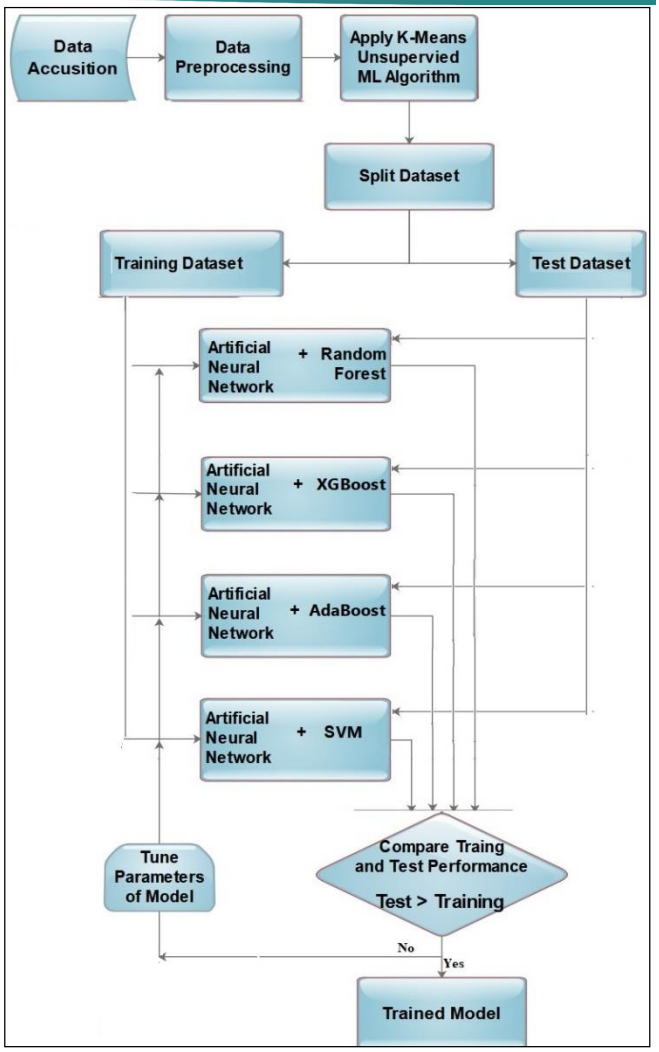

Figure 3. The Hybrid Model.

Selection of algorithms and their hyper parameters

This phase explains the selection of the most appropriate algorithms for detecting DDoS attacks.

#Algorithm Selection Criteria

The literature review table 1 shows that the most suitable algorithms have been selected for our proposed approach. The algorithm selection is based on four criteria we defined in table 2. These criteria made it possible to develop this research and fulfill the proposed objective.

Table 2. Criteria for classification Algorithm Selection.

Sl. No.	Selection Criteria
01	Captures complex patterns and sequential and non-
02	Handles mixed data types.
03	Improves weak learner algorithms
04	Effective for multi-class classification
05	The algorithm is available in the Python library.
06	Low computational complexity

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

Optimized Hyper-parameters used in Algorithms

The hybrid model combines unsupervised K-Means clustering for initial dataset labeling and supervised machine learning classifiers for attack classification. The supervised classifiers used include:

1. Random Forest (RF): Known for its robustness and high accuracy in classification tasks.
2. Extreme Gradient Boosting (XGBoost): Efficient and scalable, suitable for handling large datasets.
3. Adaptive Boosting (AdaBoost): Combines weak classifiers to create a strong classifier.
4. Support Vector Machine (SVM): Effective in high-dimensional spaces.
5. Artificial Neural Network (ANN): Capable of capturing complex patterns in the data.

Each classifier was trained and evaluated using accuracy, precision, recall, and F1-score metrics. Cross-validation techniques ensured that the models were not over fitting.

Building the classification model

Step 1: Apply K-means Algorithm

The K-means unsupervised machine learning algorithm has been employed on the dataset. This choice stems from the necessity to enable real-time classification of incoming data packets using machine learning algorithms. In real-time scenarios, procuring labelled data for incoming packets is often unfeasible. They facilitate the creation of a dataset extension, wherein new auto-generated labels are assigned to the data. This extended dataset is then meticulously prepared to serve as the training and testing ground for the proposed model.

Step 2: Dataset Division

While the literature commonly advocates a 70% training and 30% testing data split, specific conditions may necessitate alternate distribution strategies. In our research, a higher emphasis was placed on training. Consequently, the dataset was randomly partitioned into two subsets: training set (comprising 80% of the data) and a test set (constituting 20% of the data).

Step 3: Random Distribution of Training Data

The concept of randomizing the allocation of training data across diverse machine learning classification algorithms is a process designed to enhance the overall classification task performance through an ensemble approach. This mechanism involves the random distribution of training data to multiple machine-learning classification algorithms. In our research, the mechanism is implemented through Python script.

Step 4: Training by Ensemble Learning

Ensemble learning is a machine learning paradigm wherein multiple models or algorithms are combined to make predictions. The objective is to amplify overall performance by harnessing the complementary strengths of individual models. In our proposed model, illustrated in Figure, we employ five supervised machine learning classification algorithms to create an ensemble model. These algorithms include Artificial Neural Network, Random Forest, XGBoost, AdaBoost, and SVM. What sets our approach apart is the unique combination of these algorithms within the ensemble model, a configuration previously unutilized in machine learning ensemble classification.

Step 5: Testing of the Trained Model

Upon the completion of the training phase, the trained model is invoked for the testing phase. This entails loading the model and assessing 20% of the dataset records reserved for testing. Performance metrics such as accuracy, precision, recall, and F1-score are employed to evaluate the model's effectiveness.

Step 6: Performance Comparison

Following the evaluation of the trained model, the outcomes were systematically organized and subjected to comparative analysis, using predefined reference variables. This comparison facilitated the assessment of the proposed model's effectiveness and enabled us to implement any required refinements to achieve the highest efficiency.

Experiment Results and Model Performance Comparison

Results

Our proposal has undergone assessment using the dataset, environmental configuration, and predefined performance criteria as mentioned earlier. The experiment involved testing the system with various feature sets generated by employing the Chi-Square feature selection method. The performance summary of all modules with algorithm's accuracy according to features is shown in table 3.

Table 3. Module-wise attack detection accuracy.

No. of Features	Module-1		Module-2		Module-3		Module-4	
	ANN	RF	ANN	XGBoost	ANN	AdaBoost	ANN	SVM
5	0.95	0.96	0.95	0.9764	0.96	0.9768	0.95	0.96
10	0.97	0.97	0.97	0.9819	0.97	0.9841	0.97	0.97
15	0.99	0.99	0.99	0.9914	0.98	0.9971	0.98	0.99
20	0.99	0.99	0.99	0.9897	0.98	0.9967	0.98	0.98
25	0.98	0.98	0.97	0.9761	0.97	0.9718	0.98	0.97

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

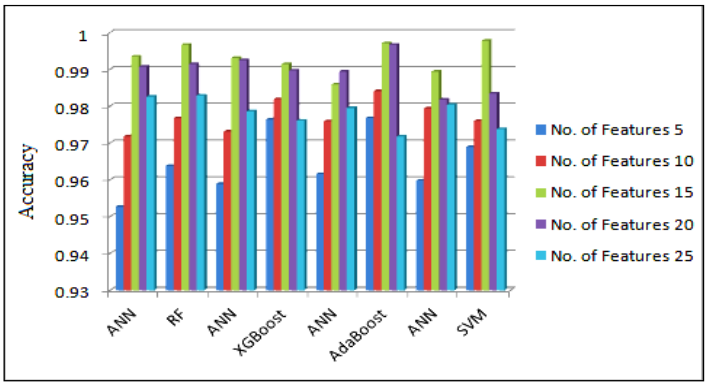

Figure 4. Attack detection accuracy with different features.

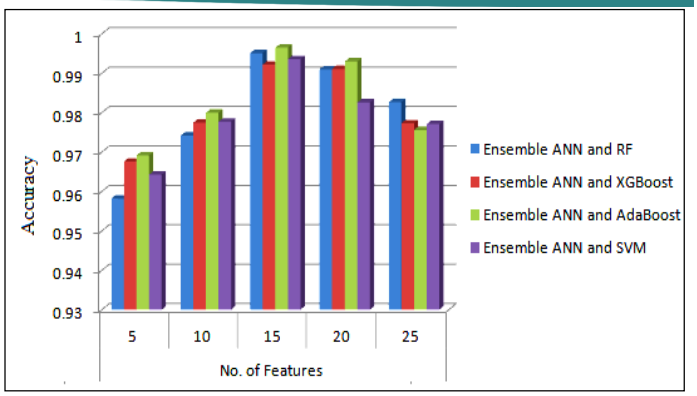

Figure 6. Performance comparisons of various ensemble modules.

Performance Comparison

Various machine learning algorithms were employed on datasets with different feature sets to determine the optimal number of features for achieving the highest accuracy. Feature selection was carried out using the chi-square feature selection algorithm. The experiment began with 25 features, followed by the extraction of 20, 15, 10, and 5 features through the chi-square method. In this study, we explored the feature sets within the range of 5 to 25 features to identify the highest accuracy while minimizing the number of features. The classifiers were applied to each feature set, and the corresponding accuracies were computed and recorded. The maximum accuracy was observed with the 15-feature set. Parameters were fine-tuned for the proposed classification model to optimize accuracy, as indicated in Table 4.

Table 4. Attack detection accuracy of various ensemble modules.

No. of Features	Ensemble ANN and RF	Ensemble ANN and XGBoost	Ensemble ANN and AdaBoost	Ensemble ANN and SVM
5	0.9582	0.9676	0.9692	0.9643
10	0.9742	0.9775	0.9800	0.9777
15	0.9951	0.9922	0.9965	0.9936
20	0.9910	0.9911	0.9930	0.9826
25	0.9827	0.9773	0.9756	0.9771

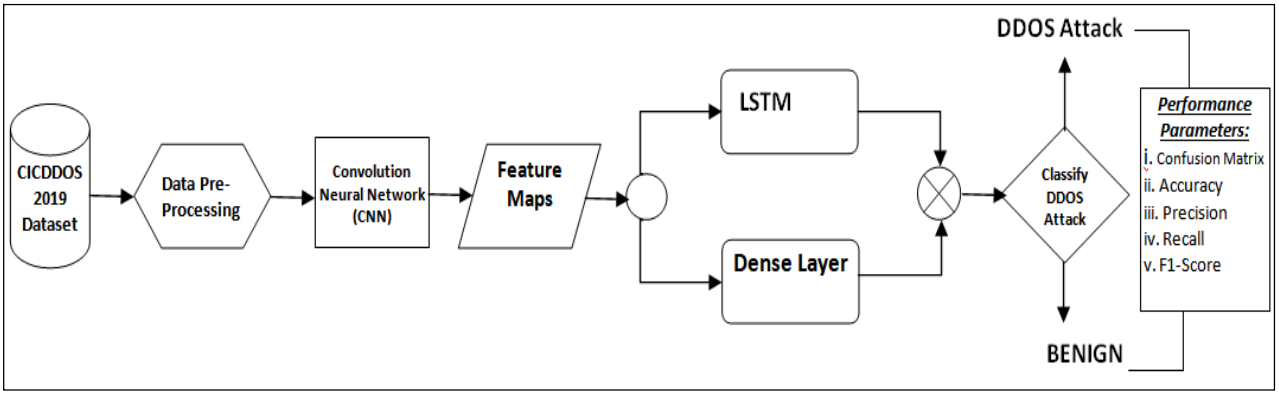

Figure 5. Proposed Model to Detect DDOS Attacks.

The Limitations of the Hybrid Model

#Lack of Temporal Dependency Modeling

While the model incorporates various supervised algorithms (Random Forest, XGBoost, AdaBoost, SVM, ANN), it does not explicitly capture sequential or temporal dependencies in network traffic data.

#Feature Engineering Dependency

Relies on manual feature selection (e.g., Chi-square selection) and optimization, which can introduce bias and limit the ability to uncover hidden relationships.

#Computational Complexity of Ensemble Models

Combining multiple algorithms into an ensemble approach can increase computational overhead during the training and prediction phases.

#Limited Integration of Spatial and Temporal Analysis

The use of ANN provides some ability to capture complex patterns but lacks the combined spatial (from CNNs) and temporal (from LSTMs) insights offered by the hybrid CNN-LSTM model.

Proposed Method

To address the challenge of DDoS detection using the CICDDoS2019 dataset, this research synthesizes two advanced methodologies: traditional machine learning approaches with ensemble learning and a hybrid deep

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

learning framework using CNN-LSTM. By integrating their strengths, our methodology aims to demonstrate that the hybrid CNN-LSTM architecture will provide superior performance over conventional ensemble methods.

Data Pre-Processing

Data pre-processing is a crucial phase in any machine learning pipeline. The CICDDoS2019 dataset ensures data integrity, quality, and compatibility with the hybrid CNN-LSTM model. The steps are outlined below:

Step 1: Data Acquisition

#Download Dataset: Obtain the CICDDoS2019 dataset from the Canadian Institute for Cybersecurity's official website.

#Verify the dataset's integrity to ensure it has been downloaded without corruption (e.g., using checksums).

Step 2: Handling Missing Values

#Identify Missing Data: Explore the dataset using statistical summaries or visualization tools to locate any missing entries.

#Impute or Remove:

#If missing values are minimal, remove the affected rows or columns.

#For essential features with missing values, impute using statistical methods:

#Mean or median for continuous variables.

#Mode for categorical variables.

Equation:

$$\hat{X}_i = f_{\text{missing}}(X_i)$$

Where f_{missing} represents the imputation or removal operation.

Step 3: Data Cleaning

#Remove Irrelevant Features: Drop columns not relevant to DDoS detection, such as timestamps (if temporal analysis is not required).

#Normalization: Standardize continuous features to have a mean of 0 and a standard deviation of 1 :

$$\hat{X}_i = \frac{\hat{X}_i - \mu}{\sigma}$$

Where:

- μ : Mean of the feature.
- σ : Standard deviation of the feature.

Equation for cleaning:

$$X_i' = f_{\text{clean}}(\hat{X}_i)$$

Where f_{clean} denotes the cleaning function.

Step 4: Feature Engineering

#Create New Features: Generate additional features that might provide better insights for the model (e.g., interaction terms, aggregated statistics).

$$X_i'' = f_{\text{eng}}(X_i')$$

Where f_{eng} represents the feature engineering function.

#Encoding Categorical Features: Use one-hot encoding or other techniques to convert categorical variables into numerical values:

$$X_i'' = \text{one\_hot}(X_i')$$

Step 5: Splitting Data

- Train-Test Split:
- Split the dataset into three subsets:
- Training set $\left(D_{\text{train}}\right) - 70\%$
- Validation set $\left(D_{\text{val}}\right) - 15\%$
- Testing set $\left(D_{\text{test}}\right) - 15\%$

Step 6: Preparing for LSTM

- Sequence Preparation:
- LSTM models require sequential data. Transform the dataset into a sequence-based format where each sequence represents a time-series or relevant contextual data.

Step 7: Label Encoding

- Encode Labels:
- Ensure the target labels are numeric (e.g., binary labels as 0 and 1).
- For multi-class problems, use techniques like one-hot encoding or label encoding.

Step 8: Balancing the Dataset

- Handle Class Imbalance:
- Analyze the distribution of classes in the dataset.
- Apply balancing techniques:
- SMOTE (Synthetic Minority Over-sampling Technique): Generates synthetic samples for the minority class.
- Class Weighting: Adjust the weights during training to penalize the majority class less.

Feature Extraction

Feature extraction is essential for transforming raw data into meaningful representations suitable for bybrid models. The process involves using Carvolutional Neural Networks (CNNs) to extract spatial features from the data, followed by parallel processing in Dense and LSTM layers.

Step 1: Transform Data for CNN Input

- Reshape Data CNNs typically operate on 2D data. Reshape the network traffic data into a 2D matrix format, or 3D if sequential information is considered. This is akin to preparing image-like structures from tabular data.

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

$$X_{\text{CNv}} = \{x_1, x_2, \ldots, x_m\}$$

Where:

- x_i : Individual 2D (or 3D) input samples.

Step 2: Define and Build the CNN Model

- CNN Architecture: Design a CNN to effectively extract spatial features.
- Use comolutional layers to karn patterns and pooling layers to reduce dimensionality.
- Equation for camalution

$$F' = \sigma\left(\sum_{k-1}^{K} W_k * X_{\text{CNN}}^{(I-1)} + b_k\right)$$

Where:

- W_k : Carmalutional filter.
- *: Convplution operation.
- by: Bias term.
- σ : Activation function (e.g, ReLU)
- l: Layer index.

Parallel Execution Through LSTM and Dense Layers

To enhance feature representation, the architecture employs a parallel execution strategy. After CNN layers, feature maps are passed through two branches: Dense and LSTM layers.

Step 1: Dense Layer Path

- Flatten Feature Maps: Transform feature maps into a 10 vector.
- Dense Layers: Pass the flattened data through fully connected layers for learning higher-level representations:

$$D_{\text{denee}} = \text{Flatten}\left(F_{\text{falumax}}\right) \cdot W_d + b_d$$

Where:

- F_{Ginursas} : Extracted CNN festure maps.
- W_{de} Dense layer weights.
- b_d - Dense layer bias.

Step 2: LSTM Path

- Reshape for LSTM: Cormert feature maps into sequential format.
- Pass Through LSTM: Extract temporal patterns using LSTM layers:

$$h_{\text{t}} = \sigma(W_{\text{h}} x_t + U_h h_{t-1} + b_h)$$

Where:

- h_2 : Hidrien state at time 1.
- W_h, U_h, b_h : ISTM parameters.

Step 3: Concatenate Outputs

- Combine the outputs from both paths to leverage spatial and temporal features:

$$O = \text{Concat}\left(D_{\text{dunes}}, h_T\right)$$

Where:

- $D_{\text{tham:}}$ Output from Dense layers.
- h_T : Final hidden state from LSTM.

Step 3: Concatenate Outputs

- Combine the outputs from both paths to leverage spatial and temporal features:

$$O = \text{Concat}\left(D_{\text{dues}}, h_T\right)$$

Where:

- $D_{\text{ilman:}}$ Output from Dense layers.
- h_T : Final hidden state from LSTM.

Step 4: Final Classification

- Use a final Dense layer for classification

$$\hat{Y} = \sigma(O \cdot W_c + b_c)$$

Where:

- W_c, b_c-Weights and bias for the classification layer.
- σ : Activation function for the output (e.g. Softmax for multi-class or Sigmoid for binary classification).

Step 5: Compile and Train the Model

- Compilation
- Loss function: Use Binary Cross-Entropy (for binary classification) or Categorical Cross-Entropy (for multi-class).
- Optimizer: Actam or SGD for optimization.
- Metrics: Track accuracy, precision, recall, and F1-score
- Training:
- Use the training set to fit the model.
- Validate the model on the validation set.
- Test its performance on the test set.

Architecture Execution Overview

1 CNN Feature Extraction:

- Convolutional and pooling layers are used to generate feature maps from the input data.

2 Parallel Paths:

- Dense Path: Flatten feature maps and pass them through Dense layers.
- LSTM Path: Reshape feature maps into sequences and process them using LSTM layers.

3 Concatenation:

- Combine Dense and LSTM outputs to form a unified feature representation.

4 Final Classification:

- A Dense layer is used to produce the final classification output.

Properties and Advantages of the proposed hybrid model

#Properties

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

a) Hierarchical Feature Learning: CNN layers learn hierarchical features, from simple edges to complex patterns.

b) Sequential Information: LSTM layers capture the data's temporal dependencies and sequential patterns.

c) Dense Layers: Capture high-level abstract features.

#Advantages

a) Enhanced Learning: Combining CNN and LSTM layers allows the model to learn both spatial and temporal features, improving its ability to detect patterns in complex data.

b) Improved Accuracy: Parallel paths can lead to better model performance as they leverage the strengths of different types of layers.

c) Flexibility: This architecture can be adapted to various tasks requiring both spatial and sequential analysis.

By following these steps, we effectively implemented parallel execution of feature maps through Dense and LSTM layers, leveraging the strengths of both approaches in a hybrid model.

Classification of DDoS Attack Using Hybrid CNN-LSTM Model

The classification of DDoS attacks using a hybrid CNN-LSTM model involves several detailed steps, including data pre-processing, building the model, training, and evaluating its performance. By leveraging both CNN and LSTM layers, the model can effectively capture spatial and temporal features, leading to improved detection of DDoS attacks. The steps outlined above provide a comprehensive approach to building and deploying such a model using the CICDDoS2019 dataset.

The structure of the hybrid CNN-LSTM model is as follows:

#CNN Block: Extract spatial features from input data $X_{CNN:}$

$$F_{CNN}=f_{CNN}(X_{CNN}) \in R^{N \times H' \times W' \times C'}$$

where: N, H′, W′, C′ are the dimensions after convolutional layers.

- **LSTM Block:** Captures temporal dependencies from the CNN-extracted features:

$$h_T = f_{LSTM}(F_{LSTM})$$

- **Dense Block:** Applies a dense layer to the flattened CNN output:

$$F_{dense}=\sigma(W_d \cdot \text{Flatten}(F_{CNN}) + b_d)$$

- The hybrid CNN-LSTM model $f_{CNN\text{-}LSTM}$ for classification of DDoS attacks can be expressed as:

$$\hat{Y} = F_{CNN\text{-}LSTM}(Xi) = \text{softmax}(W_c \cdot [D_{dense}, h_T] + b_c)$$

- **Concatenation:** Combines the outputs from LSTM and dense layers:

$$F_{final} = [h_T, F_{dense}]$$

#Classification:

Multi-class (12-class): Use softmax for the final classification:

$$\hat{Y} = \text{softmax}(W_o \cdot F_{final} + b_o)$$

Binary classification: Use sigmoid for the final classification:

$$\hat{Y} = \text{sigmoid}(W_o \cdot F_{final} + b_o)$$

#Model Evaluation and Classification

a) **Evaluate the Model**

Evaluate the model on the validation set to check its performance. The model's performance on the validation set D_{val} is evaluated by calculating the loss function L(θ) typically the cross-entropy loss for classification:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} Y_i \log(\hat{Y}_i) + (1-Y_i) \log(1-\hat{Y}_i)$$

where:

Yi is the true label for the i^{th} sample,

$\hat{Y}_i$ is the predicted probability for the i^{th} sample,

N is the total number of samples in the validation set.

b) *Make Predictions:*

Use the trained model to make predictions on new data.

Given new input data X_{new} , the model generates predictions $\hat{Y}_{inew} = f_{CNN\text{-}LSTM}(X_{new})$

c) *Performance Metrics:*

Calculate performance metrics such as confusion matrix, precision, recall, and F1-score.

Performance Parameters for Evaluating the Hybrid CNN-LSTM Model

Several performance metrics are essential when evaluating the performance of a hybrid CNN-LSTM model for classifying DDoS attacks. These metrics help assess different aspects of the model's effectiveness, including accuracy, precision, recall, and the balance between different types of errors. Below, we detail each performance parameter, how they are calculated, and how they help in assessing the model's performance.

To assess the performance of the hybrid CNN-LSTM model, several metrics are used:

- *Accuracy:* Accuracy is the proportion of correctly predicted samples to the total number of samples.

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

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

TP (True Positive): Correctly predicted positive instances.

TN (True Negative): Correctly predicted negative instances.

FP (False Positive): Incorrectly predicted positive instances.

FN (False Negative): Incorrectly predicted negative instances.

Significance: Accuracy provides a general measure of the model's performance. However, in imbalanced datasets, accuracy might be misleading because it doesn't differentiate between the types of errors (FP and FN).

Confusion Matrix: The confusion matrix is a table that summarizes the classification results by comparing the actual labels with the predicted labels:

	Predicted Positive	***Predicted Negative***
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Significance: It helps understand the types of errors the model is making.

Experiment Results and Model Performance Comparison

The proposed framework for DDoS attack detection leverages well-established performance metrics and cutting-edge computational techniques. The implementation is carried out in Python and executed on a system equipped with NVIDIA Tesla T4 GPU, part of the Turing (TU104) architecture, providing substantial computational power. The system configuration includes 16 GB of RAM, an 80 GB SSD, and operates on Ubuntu 22.04.3 LTS. The tensorflow-GPU library is integrated into the Python environment for deep learning model training to fully utilize the GPU's acceleration capabilities.

#Model Performance and Metrics

*The performance of the proposed model is evaluated using widely recognized metrics, including **accuracy, precision, recall,** and **F1-score**.

*The **binary classification model** achieves a high overall accuracy of 99.84%**,** with precision, recall, and F1 scores all matching this figure, demonstrating its effectiveness in distinguishing between benign and malicious traffic.

*The **12-class classification model** also performs remarkably well and attains a slightly lower overall accuracy of **99.76%**. Similarly, its precision, recall, and F1-score hover around the same value, indicating strong performance in classifying specific types of DDoS attacks.

Results

Our proposal has undergone assessment using the dataset, environmental configuration, and predefined performance criteria as mentioned earlier.

#Binary Classification Model

The performance metrics for the binary classification model are:

a). Accuracy

$\text{Accuracy}_{\text{binary}} = 0.9999$

b) Precision, Recall, F1-Score

$\text{Precision}_{\text{binary}} = 0.9999$

$\text{Recall}_{\text{binary}} = 0.9999$

$\text{F1-score}_{\text{binary}} = 0.9999$

c) Confusion Matrix for Binary Classification

	Predicted Positive	**Predicted Negative**
Actual Positive	TP = 14,999	FN = 1
Actual Negative	FP = 1	TN = 14,999

#12-Class Multi-Class Classification Model

For the multi-class classification model, the performance metrics are:

a). Accuracy

$\text{Accuracy}_{\text{multi-class}} = 0.9976$

b). Precision, Recall, F1-Score

$\text{Precision}_{\text{multi-class}} = 0.9976$

$\text{Recall}_{\text{multi-class}} = 0.9976$

$\text{F1-score}_{\text{multi-class}} = 0.9976$

Result Visualization

The binary classifier performs exceptionally well in distinguishing between Normal and DDoS attacks, with very few misclassifications. The multi-class classifier still shows high accuracy but struggles more with classification errors due to the increased complexity of differentiating between 12 classes. The binary model would be preferable for simpler DDoS detection tasks, while the multi-class model can be used in more complex scenarios, albeit with some trade-offs in performance. The result visualization is given below:

#Confusion Matrix for Binary Classification

The input dataset is divided into "Normal" and "DDoS." A binary classifier is trained, which could be based on a machine learning (ML) or deep learning (DL) approach like CNN-LSTM. After the model is trained and validated, it is tested with the test dataset, and the confusion matrix is generated. The confusion matrix

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

shows very high accuracy, as both Normal and DDoS classes have nearly perfect predictions (14999 correct out of 15000 for both classes). The False Positives and False Negatives are extremely low (just 1 in each case), leading to near-perfect performance. Approaching 100%, given the almost perfect diagonal of the confusion matrix.

Precision/Recall/F1-Score: These metrics would also be close to 100% due to minimal error.

Conclusion: The binary classifier performs exceptionally well for detecting DDoS attacks.

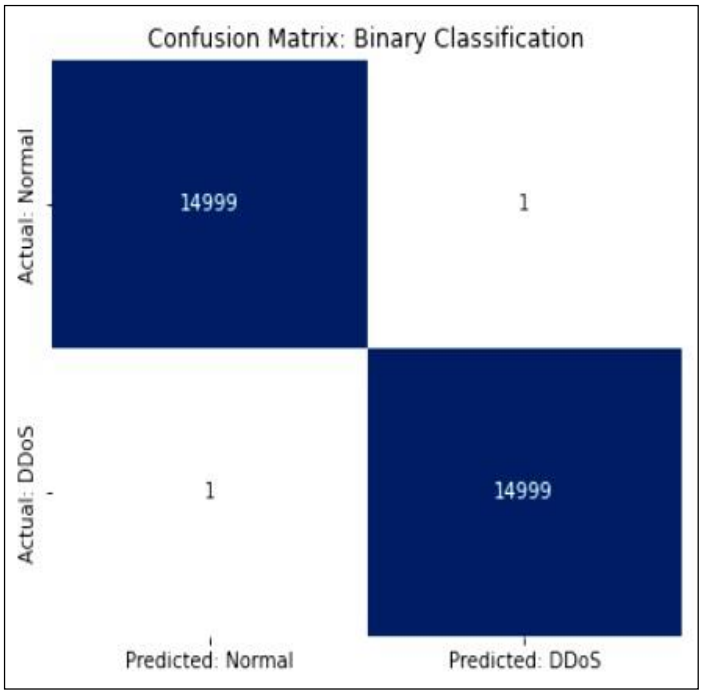

Figure 7. Confusion Matrix for Binary Classification.

#Confusion Matrix for Multi-Class Classification (12 Classes)

The dataset used here contains 12 different classes, likely representing various types of network traffic or different stages of a DDoS attack. A multi-class classification model (possibly CNN-LSTM) is trained to differentiate between these 12 classes. After training and validating the model, it is evaluated on the test dataset, generating the confusion matrix for all classes. The confusion matrix shows more diversity in predictions, with a noticeable amount of misclassification (off-diagonal values). Some classes are more easily misclassified than others, indicating certain classes are harder for the model to differentiate.

Accuracy: Lower compared to the binary classification model due to higher misclassification rates.

Precision/Recall/F1-Score: These metrics would vary significantly across the different classes. Some classes would show strong performance, while others would reflect weaker performance.

Conclusion: The multi-class model is effective but has room for improvement in detecting more nuanced differences between classes.

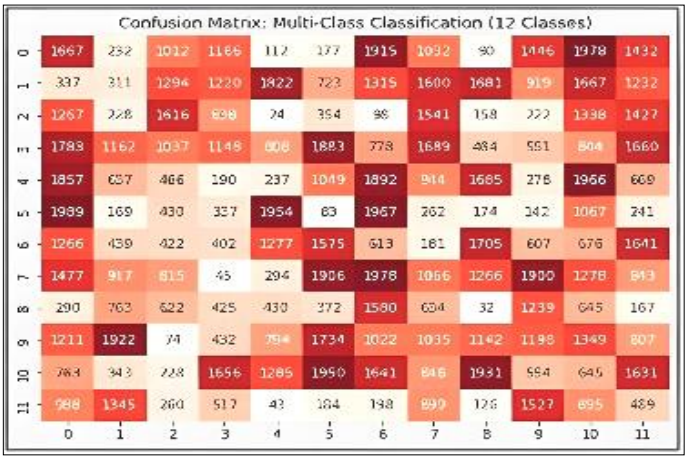

Figure 8. Confusion Matrix Multi-Class Classification.

#Accuracy Comparison between Binary and Mlti-Class Models

The accuracy values for both models are derived from their respective confusion matrices.

Binary Model achieved near-perfect accuracy (99.84%) based on minimal misclassification, as shown in the first confusion matrix. Binary model clearly outperforms the multi-class classification in terms of accuracy. The binary model is almost perfect for simple "Normal vs DDoS" scenarios.

Multi-Class Model: Achieved an accuracy of approximately 99.76% due to higher misclassification rates as shown in the second confusion matrix. Although this model is still high, the accuracy is slightly lower due to the more complex nature of classifying 12 classes compared to binary classification. Multi-Class model gives good performance but with room for improvement, especially when distinguishing between similar classes in a more complex environment.

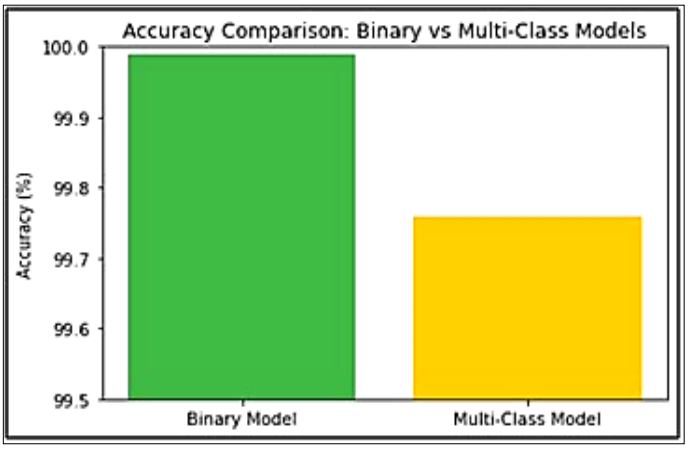

Figure 9.Accuracy Comparisons between Binary and Multi-Class.

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

Comparison of Both Models

#The Hybrid Machine Learning Model

i. Focus: Combines unsupervised K-Means clustering and supervised classifiers (Random Forest, XGBoost, SVM, AdaBoost, ANN).

ii. Dataset: CICDDoS2019, with feature selection using Chi-square methodology.

iii. Performance: Achieved 99.46% accuracy in detecting and classifying DDoS attacks.

iv. Contributions:

*A focus on leveraging multiple ML classifiers for increased detection accuracy.

*Real-time implementation.

*Scalability for larger datasets and environments.

*Integration with cyber security frameworks.

The hybrid model excels in utilizing multiple supervised algorithms, focusing on a structured ensemble learning approach to achieve high accuracy.

#CNN-LSTM Hybrid Model

i. Focus: Deep learning, specifically convolutional neural networks (CNN), is used for feature extraction and long short-term memory (LSTM) is used for temporal pattern analysis.

ii. Dataset: CICDDoS2019, evaluated on binary and multi-class classification tasks.

iii. Performance:

Binary classification: Achieved 99.84% accuracy.

Multi-class classification: Reached 99.76% accuracy across 12 attack classes.

iv. Contributions:

*Advanced use of CNN-LSTM for both spatial and temporal feature extraction.

*Robust results with minimal false positives/negatives.

The CNN-LSTM model leverages the power of deep learning to handle both spatial and temporal data, offering state-of-the-art performance in binary and multi-class classification.

Discussion on the Impact of the Ensemble Method on Computational Complexity and System Overhead

The proposed CNN-LSTM hybrid architecture represents an advanced ensemble method designed to enhance the detection of DDoS attacks. While the combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks offers significant performance gains in terms of accuracy and adaptability, it also introduces additional computational complexity and system overhead that require thorough consideration.

#Computational Complexity

The CNN-LSTM model leverages CNN layers for spatial feature extraction and LSTM layers for temporal sequence learning. Although this dual-layered approach ensures comprehensive feature representation, it inherently increases computational demands. CNN operations involve high-dimensional convolution and pooling processes, which are computationally intensive, especially when processing large-scale datasets like CICDDoS2019. Similarly, LSTM networks, due to their recurrent structure, require sequential data processing, which adds to the computational burden. As a result, the training time and resource consumption are significantly higher compared to traditional machine learning methods such as Random Forest or Support Vector Machines.

To mitigate this complexity, the implementation utilized a NVIDIA Tesla T4 GPU for parallel processing, optimising training time and improving efficiency. However, deploying this model in real-world scenarios may require high-performance computing infrastructure to achieve similar results, potentially limiting its applicability for organizations with constrained resources.

#System Overhead

In addition to computational complexity, the ensemble method introduces system overhead due to its need for larger memory and storage resources. The CNN-LSTM architecture processes high-dimensional input data, which necessitates extensive memory allocation for intermediate feature maps, weight matrices, and gradient storage during backpropagation. Furthermore, the deep learning framework used for implementation (e.g., TensorFlow) inherently adds a layer of software overhead due to dependencies and runtime optimizations.

#Balancing Performance and Overhead

Despite these challenges, the ensemble method's ability to achieve 99.84% accuracy in binary classification and 99.76% accuracy in multi-class classification justifies the trade-off in many high-stakes applications. Nevertheless, the practical deployment of such models should include optimization strategies to balance performance and overhead.

The ensemble approach effectively elevates the performance of DDoS detection systems, but its computational demands necessitate careful planning for deployment in resource-constrained environments.

Conclusion and Future Work

This study presents a robust hybrid model for DDoS attack detection, combining the strengths of unsupervised and supervised machine learning techniques. By leveraging the CICDDoS2019 dataset, a

DOI: https://doi.org/10.52756/ijerr.2024.v46.020

wide variety of attack scenarios were analyzed, with feature selection optimized using the chi-square method. The proposed hybrid machine learning classifier achieved an accuracy of 99.46%, demonstrating its efficacy in controlled environments. However, recognizing the limitations of traditional machine learning approaches in handling complex real-world scenarios, a novel CNN-LSTM hybrid model was developed. This deep learning-based architecture offers significant advantages through automated spatial and temporal feature extraction, eliminating the need for manual intervention.

The CNN-LSTM hybrid model proved highly effective in real-world applications, achieving an exceptional 99.84% accuracy in binary classification tasks, distinguishing benign and malicious traffic with near-perfect precision. Though slightly less accurate at 99.76%, the multi-class classification model successfully identified 12 distinct DDoS attack types, demonstrating its adaptability to complex and diverse attack patterns. Minimal false positives and negatives, as validated by the confusion matrices, further attest to the model's reliability and precision. The practical implications of this research lie in the potential integration of the optimized CNN-LSTM ensemble into existing intrusion detection systems (IDS) and network security frameworks. Its ability to process large-scale, high-dimensional data in real-time makes it a valuable tool for modern cyber security solutions. The proposed architecture can be deployed to monitor live network traffic, enabling proactive and automated DDoS mitigation in dynamic environments. Future work could further enhance the multi-class model's performance and explore its application in hybrid cloud and edge computing environments, ensuring seamless integration with diverse infrastructure setups. This research underscores the transformative role of advanced deep learning models in fortifying cyber security defences, paving the way for scalable, intelligent systems capable of combating evolving DDoS threats.

Future work should focus on scalability, real-time capabilities, behavioural analysis, anomaly detection, and user-friendly tools to enhance defence against evolving DDoS attacks and benefit a wider user base. Future research will explore the following directions:

**Integration with Existing Security Frameworks:* Evaluating the integration of the hybrid model with existing cyber security frameworks to enhance overall network security.

**Adaptive Learning:* Developing adaptive learning mechanisms to keep the model updated with the latest attack patterns and techniques.

Conflicts of Interest

The authors assert that they do not have any conflicts of interest to disclose with respect to the current research.

Funding

The authors undertaken this independent research endeavour without receiving any external funding or financial support.

References

Abid, Y. A., Wu, J., Xu, G., Fu, S., & Waqas, M. (2024). Multilevel deep neural network approach for enhanced distributed denial-of-service attack detection and classification in software-defined Internet of things networks. *IEEE Internet of Things Journal, 11*(14), 24715-24725. https://doi.org/10.1109/jiot.2024.3376578.

Abreu Maranhão, J. P., Carvalho Lustosa da Costa, J. P., Pignaton de Freitas, E., Javidi, E., & Timóteo de Sousa Júnior, R. (2020). Error-robust distributed denial of service attack detection based on an average common feature extraction technique. *Sensors, 20*(20), 5845. https://doi.org/10.3390/s20205845.

Alanazi, F., Jambi, K., Eassa, F., Khemakhem, M., Basuhail, A., & Alsubhi, K. (2022). Ensemble Deep Learning Models for Mitigating DDoS Attack in Software-Defined Network. *Intelligent Automation and Soft Computing, 32*, 923-938. https://doi.org/10.32604/iasc.2022.024668

Alasmari, T., Eshmawi, A., Alshomrani, A., & Hsairi, L. (2023). CNN-LSTM based approach for DDoS detection. *2023 Eighth International Conference On Mobile And Secure Services* (MobiSecServ), pp. 1-6. https://doi.org/10.1109/mobisecserv58080.2023.10329028

Alzahrani, F., Aljohani, H., & Ba-Alwi, F. (2022). Improved CNN-LSTM Model for DDoS Detection in IoT Networks. *Future Generation Computer Systems, 128*, 208-221.

Andresini, G., Appice, A., Mauro, N. D., Loglisci, C., & Malerba, D. (2020). Multi-Channel Deep Feature Learning for Intrusion Detection. *IEEE Access, 8*, 53346–53359. https://doi.org/10.1109/access.2020.2980937

Behal, S., Saluja, K. K., & Meenakshi. (2021). Distributed Denial of Service Attack Detection Using Deep Learning Approaches. *IEEE 2021*

8th International Conference on "Computing for Sustainable Global Developmen, 17th-19th March, 2021.
https://doi.org/10.1109/INDIACom51348.2021.00087

Bhatt, D., Sharma, V., & Rajput, D. S. (2021). CNN-LSTM Model for Predicting Network Anomalies. *Multimedia Tools and Applications, 80*, 15381-15400.

Borgiani, V., Moratori, P., Kazienko, J. F., Tubino, E. R., & Quincozes, S. E. (2021). Toward a distributed approach for detection and mitigation of denial-of-service attacks within industrial Internet of things. *IEEE Internet of Things Journal*, *8*(6), 4569-4578. https://doi.org/10.1109/jiot.2020.3028652.

Dangi, N., Verma, A. K., & Thoke, A. S. (2021). CNN-LSTM Hybrid Model for Network Intrusion Detection. *Journal of Network and Computer Applications, 173*, 102883.

DDoS 2019 | Datasets | Research | Canadian institute for cybersecurity | UNB. (n.d.). University of New Brunswick | UNB. https://www.unb.ca/cic/datasets/ddos-2019.html

Devan, P., & Khare, N. (2020). An efficient xgboost–dnn-based classification model for network intrusion detection system. *Neural Computing and Applications, 32*(16), 12499-12514. https://doi.org/10.1007/s00521-020-04708-x

Effah, E. Q., Osei, E. O., Maxwell Dorgbefu Jnr., & Tetteh, A. (2024). Hybrid approach to classification of DDoS attacks on a computer network infrastructure. *Asian Journal of Research in Computer Science*, *17*(4), 19-43. https://doi.org/10.9734/ajrcos/2024/v17i4428

Gamal, H. E., Amer, E., & Nassar, H. (2022). Deep Learning-Based CNN-LSTM Model for Detecting DDoS Attacks in Software - Defined Networks. *Computers & Security, 114*, 102595.

Halladay, J., Cullen, D., Briner, N., Warren, J., Fye, K., Basnet, R., Bergen, J., & Doleck, T. (2022). Detection and characterization of DDoS attacks using time-based features. *IEEE Access, 10*, 49794-49807. https://doi.org/10.1109/access.2022.3173319

Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle can bus communications. *IEEE Access, 8*, 185489-185502. https://doi.org/10.1109/access.2020.3029307

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access, 8*, 32464-32476. https://doi.org/10.1109/access.2020.2973730

Musa, N. S., Mirza, N. M., Rafique, S. H., Abdallah, A. M., & Murugan, T. (2024). Machine Learning and Deep Learning Techniques for Distributed Denial of Service Anomaly Detection in Software Defined Networks—Current Research Solutions. *IEEE Access, 12*, 17982–18011. https://doi.org/10.1109/access.2024.3360868

Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, *12*(3), 1035. https://doi.org/10.3390/su12031035

Ramzan, M., Shoaib, M., Altaf, A., Arshad, S., Iqbal, F., Castilla, Á. K., & Ashraf, I. (2023). Distributed Denial of Service Attack Detection in Network Traffic Using Deep Learning Algorithm. *Sensors*, *23*(20), 8642. https://doi.org/10.3390/s23208642

Salehi, M., & Yari, A. (2024). Detecting DOS attacks using a hybrid CNN-LSTM model. *2024 10th International Conference on Web Research* (ICWR), pp. 397-401. https://doi.org/10.1109/icwr61162.2024.10533358

Salih, A.A., & Abdulrazaq, M.B. (2024). Cybernet model: A new deep learning model for cyber DDOS attacks detection and recognition. *Computers, Materials & Continua*, *78*(1), 1275-1295. https://doi.org/10.32604/cmc.2023.046101

Saxena, U., Sodhi, J., & Singh, Y. (2020). An analysis of DDoS attacks in a smart home networks. *2020 10th International Conference on Cloud Computing, Data Science & Engineering*, 272-276. https://doi.org/10.1109/confluence47617.2020.9058087

Seifousadati, A., & Ghasemshirazi, S., & Fathian, M. (2021). A Machine Learning Approach for DDoS Detection on IoT Devices. https://doi.org/10.48550/arXiv.2110.14911

Sharma, S., & Shakya, H. K. (2022). Hybrid Real-Time Implicit Feedback SOM-Based Movie Recommendation Systems. *In International Conference on Computing, Communications, and Cyber-Security* (pp. 371-388). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1479-1_28

Sharma, S., Dubey, G. P., Shakya, H. K., & Motwani, D. (2023). Hybrid Filtering Methods in Movie

Recommendation: The Enhanced SOM Approach. *In International Conference on Information Systems and Management Science* (pp. 174-187). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-70789-6_14

Sharma, S., Prasad, G., Kumar, H., & Sharma, A. (2024a). SOM and hybrid filtering: pioneering next-gen movie recommendations in the entertainment industry. *J. Fusion: Pract. Appl. 16*(2), 43–62. https://doi.org/10.54216/FPA.160204

Sharma, S., Dubey, G.P., & Shakya, H.K. (2024b). Optimizing User Satisfaction in Movie Recommendations Using Variable Learning Rates and Dynamic Neighborhood Functions in SOMs. *International Journal of Experimental Research and Review*, *41*(spl.), 130-145. https://doi.org/10.52756/ijerr.2024.v41spl.011

Sharma, S., Dubey, G.P., & Shakya, H.K. (2024c). Reducing Cluster Overlap in Movie Recommendations with IKSOM and Silhouette Clustering. *International Journal of Experimental Research and Review*, *42*, 169-182. https://doi.org/10.52756/ijerr.2024.v42.015

Sindian, S., & Sindian, S. (2020). An enhanced deep autoencoder-based approach for DDoS attack detection. *WSEAS Transactions on Systems and Control*, *15*, 716-724. https://doi.org/10.37394/23203.2020.15.72

Sharma, S., & Shakya, H. K. (2024). Hybrid recommendation system for movies using artificial neural network. Expert Systems with Applications, 258, 125194. https://doi.org/10.1016/j.eswa.2024.125194

Sharma, S., & Shakya, H. K. (2022, October). Hybrid Real-Time Implicit Feedback SOM-Based Movie Recommendation Systems. In International Conference on Computing, Communications, and Cyber-Security (pp. 371-388). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1479-1_28

Sharma, S., Shakya, H. K., & Marriboyina, V. (2021). A location based novel recommender framework of user interest through data categorization. Materials Today: Proceedings, 47, 7155-7161. https://doi.org/10.1016/j.matpr.2021.06.325

Sumathi, D., Rajesh, R., & Lim, S. (2022). Recurrent and Deep Learning Neural Network Models for DDoS Attack Detection. *Journal of Sensors*, *2022*, 1-21. https://doi.org/10.1155/2022/8530312

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, *7*, 41525-41550. https://doi.org/10.1109/access.2019.2895334

Woo, S., Kim, J., and Lee, H. (2000). Hybrid CNN-LSTM Network Model for Detecting DDoS Attacks in SDN. *Sensors*, *20*(12), 3486

Xu, W., Liu, S., & Wang, G. (2022). An Enhanced CNN-LSTM-Based Network Intrusion Detection System. *Journal of Information Security and Applications*, *64*, 102865.

Yin, C., Zhu, Y., Fei, J., & He, X. (2021). CNN-LSTM Deep Learning Framework for Cyberattack Detection. *IEEE Transactions on Network and Service Management*, *18*(3), 345-354.

Zhang, Y., Shen, C., and Zhang, W. (2020). An Effective Convolutional Neural Network LSTM Model for DDoS Attack Detection. *International Journal of Distributed Sensor Networks*, *16*(12), 1550147720977910.

Zhou, L., Zhu, Y., Zong, T., & Xiang, Y. (2022). A feature selection-based method for DDoS attack flow classification. *Future Generation Computer Systems*, *132*, 67-79. https://doi.org/10.1016/j.future.2022.02.006

How to cite this Article:

Deepak Singh Rajput and Arvind Kumar Upadhyay (2024). Enhanced Network Defense: Optimized Multi-Layer Ensemble for DDoS Attack Detection. *International Journal of Experimental Research and Review*, *46*, 253-272.

DOI : https://doi.org/10.52756/ijerr.2024.v46.020