International Journal of Innovative Information Systems & Technology Research 14(3):176-189, July-Sept, 2026

© SEAHI PUBLICATIONS, 2026 www.seahipublications.org **ISSN: 2467-8562**

DOI:10.5281/zenodo.21841327

Development Of Machine Learning-Based Model For DDoS Attack Classification In IoT Networks

Sheidu, Nanahawa Oyiza; Aimufua, G.I.O; Chaku, S.E.; Idris. M. & Paul Owoicho Gideon

Centre for Cyberspace Studies, Nasarawa State University Keffi, Nasarawa State, Nigeria.
Owoicho.paul@fuoye.edu.ng, sheidunanahawa@gmail.com

ABSTRACT

Cybersecurity has become a critical concern in the era of the Internet of Things (IoT), where interconnected devices are increasingly vulnerable to distributed denial-of-service (DDoS) attacks. The growing sophistication and frequency of these attacks pose serious challenges to the reliability, availability, and security of IoT networks. Traditional detection techniques often struggle to adapt to dynamic attack patterns and typically exhibit low detection accuracy and high false positive rates. Although several machine learning-based approaches have been proposed, many existing models rely on single classifiers or limited hybrid combinations, which are insufficient to effectively capture the complexity of DDoS attack patterns in IoT environments. To address these limitations, this study presents the development of a machine learning-based model for DDoS attack classification in IoT networks using an ensemble learning approach. The proposed model utilizes a comprehensive framework that integrates multiple machine learning algorithms, including Random Forest, Decision Tree, K-Nearest Neighbor, XGBoost, LightGBM, Logistic Regression, and Naïve Bayes, combined using Bagging techniques to improve predictive performance and generalization. The model was trained and evaluated using a benchmark dataset containing diverse network traffic features and DDoS attack scenarios relevant to IoT environments. A robust data preprocessing pipeline, including data cleaning, normalization, and feature selection, was employed to enhance data quality and model efficiency. Experimental results demonstrate that the proposed ensemble model achieved high performance, with an accuracy of 93.5%, recall of 96.8%, and improved precision and F1-score, indicating its effectiveness in accurately classifying DDoS attacks while minimizing false alarms. The model was implemented using Python and the scikit-learn library, showcasing its practical applicability for real-time IoT network security. This study contributes to the field of cybersecurity by providing a reliable and scalable solution for DDoS attack detection, thereby enhancing the resilience and security of IoT infrastructures.

Keywords: machine learning, classification, DDoS, IoT networks

1. INTRODUCTION

The rapid growth of interconnected digital systems has revolutionized modern society, driving innovation across finance, healthcare, education, and critical infrastructure. These networks provide seamless communication, efficient resource sharing, and improved operational effectiveness (Ezeigweneme et al., 2024). However, their increasing dependence on connectivity has also heightened exposure to sophisticated cyber threats (Adewuyi et al., 2024). Malicious actors exploit vulnerabilities through tactics such as malware distribution, phishing, denial-of-service (DoS) attacks, and zero-day exploits to disrupt

operations or exfiltrate sensitive data (Cliford et al., 2024; Ciuriak, 2021). Among these threats, Distributed Denial-of-Service (DDoS) attacks have emerged as one of the most destructive, overwhelming network resources by flooding them with malicious traffic and rendering critical services inaccessible. The rising scale and sophistication of DDoS attacks underscore the inadequacy of traditional security systems and the pressing need for advanced detection techniques (Sladkova et al., 2024; Filkins et al., 2016).

Conventional Intrusion Detection Systems (IDS) rely primarily on signature-based methods that match incoming traffic with predefined patterns of known attacks (Safitra, Lubis & Fakhrurroja, 2023). While effective against previously encountered threats, these systems are unable to detect novel or evolving DDoS variants where no signatures exist (Alsoufi et al., 2024). Anomaly-based detection approaches, which monitor deviations from normal traffic behavior, offer a promising alternative (Diro et al., 2024). However, these methods face limitations such as high false-positive rates and difficulty distinguishing between legitimate network surges (e.g., flash crowds) and actual DDoS attempts (Kumar, Gupta, Singh & Singh, 2023). This creates a gap that necessitates more adaptive and intelligent detection mechanisms capable of learning from dynamic traffic patterns.

Machine learning (ML) has emerged as a transformative approach in cybersecurity for its ability to process vast volumes of network data, uncover hidden relationships, and classify malicious traffic in real time (Khaleel et al., 2024; Paramesha, Rane & Rane, 2024). Ensemble learning, a subset of ML techniques, has gained prominence due to its effectiveness in improving accuracy and robustness by combining multiple classifiers (Mian et al., 2024; Jemili, Meddeb & Korbaa, 2024). Bagging (Bootstrap Aggregating), in particular, enhances model stability by reducing variance and mitigating overfitting, thereby ensuring more reliable detection outcomes (Rane, Choudhary & Rane, 2024). This makes bagging-based models highly suitable for addressing the dynamic and evolving nature of DDoS threats (Elmas, 2023).

In this study, a machine learning-based predictive model for DDoS attack classification in IoT networks is proposed using a bagging ensemble framework. The model integrates seven diverse algorithms—Random Forest, Decision Tree, Artificial Neural Network (ANN), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN). Each algorithm contributes unique strengths: tree-based models excel at interpretability, neural networks at capturing complex non-linear patterns, and deep learning methods at processing high-dimensional and temporally dynamic data (Sakib, Mustajab & Alam, 2025). By aggregating these strengths, the proposed model aims to enhance detection accuracy, reduce false positives, and improve adaptability against evolving DDoS threats in IoT environments. Ultimately, this approach seeks to deliver a scalable, intelligent, and real-time solution for safeguarding critical IoT infrastructures.

2. Literature Review

There is a clear knowledge, methodological, and practical gap in existing research concerning the application of machine learning models for Network Intrusion Detection Systems in wireless network (NIDS). Although significant efforts have been made to detect and mitigate network threats using artificial intelligence and machine learning approaches, several limitations persist:

i. Numerous studies have employed machine learning algorithms such as Decision Tree, Random Forest, Artificial Neural Networks (ANN), and Support Vector Machines (SVM) for intrusion detection, yielding relatively high detection accuracies (Khraisat *et al*., 2020; Raza *et al*., 2023; Coscia *et al.,* 2024;, De-Carvalho *et al.* (2023).

ii. Recent advances have also explored deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to improve classification performance in identifying sophisticated cyberattacks (Dalal *et al*., 2023; Musthafa *et al*., 2024).

iii. Ensemble methods like voting and boosting have been investigated to combine the strengths of individual classifiers and improve detection capability (Khraisat *et al*., 2020; Raza *et al.,* 2023).

Despite these advancements, key research gaps remain. At the knowledge level, existing models often underperform due to high false positive and false negative rates, which compromise the reliability of

NIDS in wireless network. False positives generate excessive alerts that overwhelm security personnel, while false negatives allow undetected attacks, posing serious threats to network systems (Khraisat *et al*., 2020; Raza *et al*., 2023).

At the methodological level, while many studies emphasize accuracy, they often neglect crucial metrics such as recall, precision, F1-score, training time and prediction especially when evaluating models on imbalanced datasets (Coscia *et al.,* 2024). Moreover, existing research lacks comparative evaluations across diverse datasets and model types, leading to challenges in model generalization across different attack categories and network environments.

Table 2. 1: Summary of Literature Review

S/N	Authors/Year	Problem	Methodology	Findings	Limitations
1	Raza *et al., (*2023)	Network attack detection	Class Probability Random Forest (CPRF) using the CICIDS2017 dataset	Achieved 99.9% accuracy; outperforming existing methods with a high-performance accuracy.	May need validation across diverse attack types and real-world environments.
2	**De-Carvalho *et al.* (2023)**	Generalizing intrusion detection across heterogeneous networks	Stacked-unsupervised federated learning for a flow-based NIDS; utilizes a deep autoencoder and energy flow classifier	Achieved Accuracy: 0.98, Recall: 0.88, Precision: 0.91, F1-Score: 0.90	the false positive and false negative rates need to be reduced to enhance precision and recall rates
3	**Bouke *et al.* (2023)**	DDoS attack detection in IoT and intelligent devices	Decision Tree algorithm with enhanced Gini index feature selection; evaluated on the UNSW-NB15 dataset	Achieved 98% accuracy, reduced dimensionality from 45 to 13 features, and only 2% misclassification rate	Need to improve accuracy to further reduce false positives
4	Karthikeyan *et al.* (2023)	Limitations of traditional IDS in dynamic network environments	Proposed a hybrid intrusion detection system using decision trees and ensemble learning techniques; evaluated on diverse datasets	Achieved an overall accuracy of 93.33%, demonstrating robustness across varying attack scenarios	Despite high accuracy, false positive and false negative rates need to be reduced for increased reliability and effectiveness
5	Khraisat *et al.,* (2020)	Intrusion detection with single classifiers	Hybrid IDS combining C5 Decision Tree and One-Class SVM; evaluated with NSL-KDD and ADFA	High detection accuracy of 83.24% with low false alarm rates; improved performance over single	Limited exploration of varied attack types and real-world scenarios; possible performance

			datasets	classifiers.	variations.
6	Kumar *et al.,* (2020)	Network attack detection using unsupervised methods	MeanShift algorithm applied to KDD 99 dataset	Achieved 81.2% accuracy and 79.1% detection rate; foundational work for MeanShift in network attack detection.	Performance could vary with different attack types and datasets; future work is needed for real-time applications.
7	Yang *et al.,* (2022)	Anomaly-based network intrusion detection	Systematic literature review of techniques and datasets	Identified current techniques, challenges, and future research directions in anomaly-based intrusion detection.	Lack of focus on specific methodologies and detailed analysis of new techniques.

This study addresses the identified knowledge, methodological, and practical gaps by proposing a Bagging Ensemble-Based Predictive Model for network intrusion detection in wireless networks that integrates seven diverse classifiers: Decision Tree, Random Forest, Support Vector Machine, Artificial Neural Network, Long Short-Term Memory, Multi-Layer Perceptron, and Convolutional Neural Network. By combining multiple models using a bagging ensemble framework, this approach aims to balance detection accuracy and robustness while significantly reducing false alarm rates. In doing so, the proposed system offers a more reliable, scalable, and operationally efficient solution for detecting evolving and complex network intrusions.

3. METHODOLOGY

3.1 Framing Research Questions

Crafting well-defined research questions is a challenging task, particularly in the realm of cybersecurity, where the scope of Development of machine learning-based model for DDOS attacks classification in IoT Networks can be diverse and complex. To ensure a thorough examination of the literature on IoT attacks, the questions were framed to identify research articles that effectively employ ML. This study is aimed at answering the following questions:

i. How can a machine learning-based model be formulated to effectively classify DDoS attacks in IoT networks?
ii. How can an algorithm be developed to accurately classify DDoS attacks based on IoT network traffic patterns?
iii. How does the proposed model perform in terms of accuracy, precision, recall, and computational efficiency when evaluated with standard performance metrics?
iv. Development of machine learning-based model for DDoS attack classification in IoT networks

3.2 Research Design

This study employs the Design and Creation Research (DCR) methodology, a derivative of Design Science Research (DSR), which is widely recognized in the development of computational models and information systems (Oates, 2006). DSR emphasizes the design and construction of artifacts that address practical challenges, blending engineering, scientific, and organizational principles. Although its definitions and concepts remain subject to ongoing debate due to its interdisciplinary nature (Johannesson et al., 2013), DSR is generally acknowledged as a methodology that unifies human, organizational, and technological dimensions to solve complex, real-world problems.

In the context of this research, DCR provides a systematic framework for the development of a Machine Learning-Based Model for DDoS Attack Classification in IoT Networks. The model leverages an

ensemble-based strategy, specifically Bagging, to combine the predictive power of multiple algorithms, including Random Forest, Decision Tree, Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN). By integrating these techniques, the model aims to enhance detection accuracy, reduce false positives and false negatives, and ensure resilience against the evolving nature of DDoS attacks.

The iterative nature of DCR aligns well with this objective, as it facilitates design, implementation, and evaluation cycles that progressively refine the model based on empirical validation and feedback. This ensures that the artifact is not only theoretically sound but also practically effective in real-world IoT network environments. By combining insights from machine learning, cybersecurity, and systems engineering, the methodology supports the creation of a robust and scalable intrusion detection system capable of mitigating DDoS threats in IoT ecosystems.

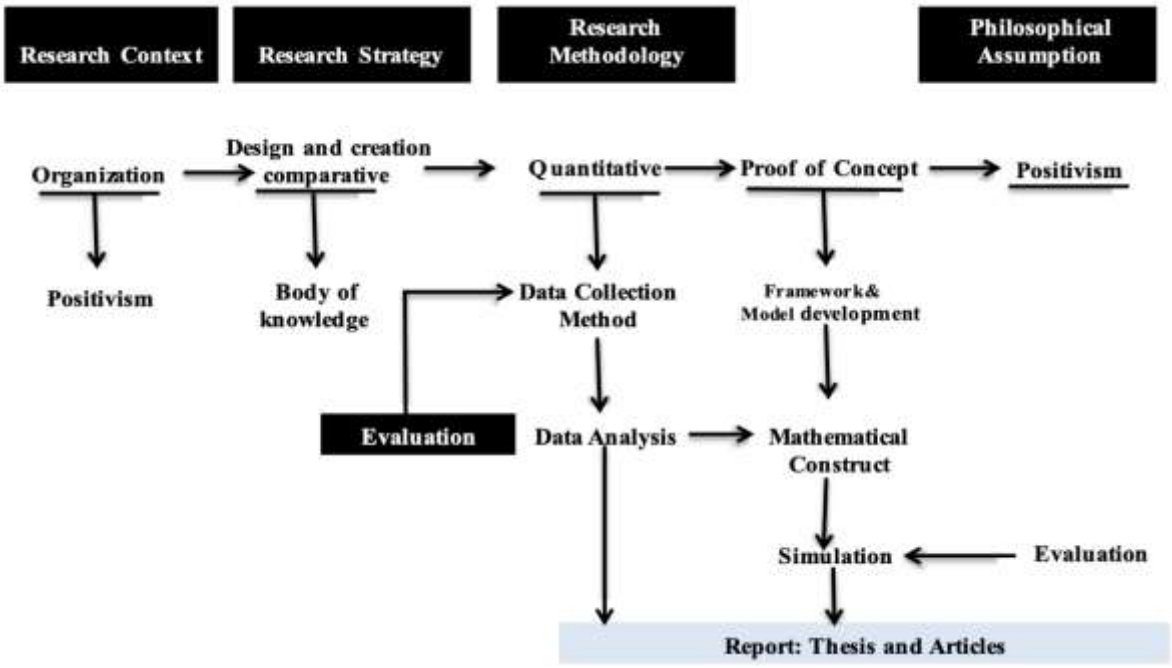

Figure 3. 1: Design Science Research (DSR) Approach

Figure 3.1 illustrates the Design and Creation Research Strategy, highlighting how it aligns with the objectives of this study. The iterative nature of DSR supports the continual improvement of the model, ensuring it addresses the complexities of network security. The model's development and evaluation will be guided by the principles of DSR, allowing for a comprehensive approach to problem-solving that incorporates both theoretical foundations and real-world applications. Through this approach, the study aims to contribute not only to the advancement of intrusion detection techniques but also to the broader understanding of how design science can be applied to the development of practical, technology-driven solutions.

Data Collection Method

Data for this research are drawn from benchmark intrusion detection datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15, which provide comprehensive representations of both legitimate and malicious network traffic. Data preprocessing, including cleaning, feature selection, and transformation, is applied to ensure quality and compatibility with machine learning algorithms. This process is essential for preparing accurate and balanced datasets for training and evaluation of the model.

3.4 Technique for Data Analysis and Model Specification

This section outlines the framework, algorithm, and performance evaluation metrics of the ensemble-based predictive model for Bagging Ensemble-Based Predictive Model for Network Intrusion Detection.

3.4.1 Formulation of Machine Learning-Based Model for DDoS Attack Classification in IoT Networks

Let duration, protocol type, service, source byte, destination byte, flag and count be represented as $D, PT, S, SB, DB, F \text{ and } C$ respectively, therefore, Applying the concept of set theory, we have:

$$D = \{d_1, d_2, \ldots, d_n\} \quad (3.1)$$

$$PT = \{pt_1, pt_2, \ldots, pt_n\} \quad (3.2)$$

$$S = \{s_1, s_2, \ldots, s_n\} \quad (3.3)$$

$$SB = \{sb_1, sb_2, \ldots, sb_n\} \quad (3.4)$$

$$DB = \{db_1, db_2, \ldots, db_n\} \quad (3.5)$$

$$F = \{f_1, f_2, \ldots, f_n\} \quad (3.6)$$

$$C = \{c_1, c_2, \ldots, c_n\} \quad (3.7)$$

Hence, the combination of the values of $D, PT, S, SB, DB, F \text{ and } C$ will produce the dataset Q for network intrusion detection.

$$Q = \sum_{i=1}^{n} d_i + \sum_{i=1}^{n} pt_i + \sum_{i=1}^{n} s_i + \sum_{i=1}^{n} sb_i + \sum_{i=1}^{n} db_i + \sum_{i=1}^{n} f_i + \sum_{i=1}^{n} c_i \quad (3.8)$$

Let n represent the skLearn data cleaning model, the function of n on Q will therefore be $n(Q)$ producing dataset$(d1)$.

$$dataset(d1) = n(Q) \quad (3.9)$$

Models Definition:
Let the seven base models be denoted as:
RF: Random Forest, SVM: Support Vector Machine, XGBoost: Extreme Grdaient Boost, DT:Decision Tree, CNN: Convolutional Neural Network, LSTM: Long Short-Term Memory (LSTM) and MLP: Multi-Layer Perceptron.
where d_i is the input feature vector, and y_i is the corresponding target value.

Base Learners: Each model M_j(where $j \in \{1,2,3,\dots,7\}$) produces a prediction $\hat{y}_i^{(j)}$ for each input sample x_i:

$$\hat{y}_i^{(1)} = rf(d_i) \quad (3.10)$$

$$\hat{y}_i^{(2)} = svm(d_i) \quad (3.11)$$

$$\hat{y}_i^{(3)} = xgboost(d_i) \quad (3.12)$$

$$\hat{y}_i^{(4)} = dt(d_i) \quad (3.13)$$

$$\hat{y}_i^{(5)} = cnn(d_i), \quad (3.14)$$

$$\hat{y}_i^{(6)} = lstm(d_i) \quad (3.15)$$

$$\hat{y}_i^{(7)} = mlp(d_i) \quad (3.16)$$

Aggregation: After obtaining the individual predictions from all base learners, we aggregate these predictions to obtain the final prediction $\hat{y}_i^{(final)}$.the aggregation can be done as follows

For classification (Majority Voting):

$$\hat{y}_i^{(final)} = mode(\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \dots, \hat{y}_i^{(7)}) \quad (3.17)$$

For regression (Averaging):

$$\hat{y}_i^{(final)} = \frac{1}{7}\sum_{j=1}^{7}\hat{y}_i^{(j)} = \frac{1}{7}(\hat{y}_i^{(1)} + \hat{y}_i^{(2)} + \hat{y}_i^{(3)} + \hat{y}_i^{(4)} + \hat{y}_i^{(5)} + \hat{y}_i^{(6)} + \hat{y}_i^{(7)}$$

(3.18)

Final Prediction: The final prediction for a given input sample d_i is obtained from the aggregated result as:

$$\hat{y}_i^{(final)} = f(\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \ldots, \hat{y}_i^{(7)}) \quad (3.19)$$

Where f is the aggregation function (mode for classification, mean for regression).

Thus, the ensemble model can be described as:

$$\hat{y}_i^{(final)} = (rf(d_i) + svm(d_i) + xgboost(d_i) + dt(d_i) + cnn(d_i) + lstm(d_i) + mlp(d_i)) \quad (3.20)$$

This machine learning approach combines the strengths of multiple base learners, improving the robustness and accuracy of the model compared to individual models, especially in terms of handling different types of data distributions and reducing the risk of overfitting.

4. RESULTS AND DISCUSSION

4.1 Data Presentation

The Python libraries Numpy, Pandas, Matplotlib, and Seaborn were imported into Jupyter Notebook. Numpy facilitates efficient vectorized computing and broadcasting over arrays with several dimensions (Stančin & Jović, 2019). Pandas is an advanced and user-friendly open-source software for analyzing and manipulating data. It is built on the Python programming language (Subasi, 2020). Matplotlib and Seaborn are Python packages specifically designed for visualizing data. They provide a user-friendly interface for creating visually stunning and useful graphs. Seaborn is built on the foundation of matplotlib and offers somewhat less features compared to matplotlib (Pintor et al., 2019).

```
#!pip install nbformat

#!pip install graphviz

#!pip install dtreeviz

import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

sns.set_context('notebook')
sns.set_style('white')

import dtreeviz
import os
from sklearnex import patch_sklearn
patch_sklearn()
os.environ["PATH"] += os.pathsep + 'C:/program files/graphviz/bin'
```

Figure 4. 1: Importing the Libraries

4.1.1 Dataset Description

The *describe ()* method provides a concise summary of a dataset containing 257672 records and 44 columns, all of which include numerical values. Table 4.3 offers statistical data for several factors, including dur. Proto, service, state, spkts, sbytes, dbytes, dloss, sinpkt, dinpk etc.

```
df.describe()
```

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl
count	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	2.576730e+05	2.576730e+05	2.576730e+05	257673.000000
mean	1.246715	110.136025	1.555902	4.341468	19.777144	18.514703	8.572952e+03	1.438729e+04	9.125381e+04	180.000931
std	5.974305	21.349900	2.245945	0.887420	135.947152	111.985965	1.737739e+05	1.461993e+05	1.603446e+05	102.488268
min	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	2.400000e+01	0.000000e+00	0.000000e+00	0.000000
25%	0.000008	113.000000	0.000000	4.000000	2.000000	0.000000	1.140000e+02	0.000000e+00	3.078928e+01	62.000000
50%	0.004285	113.000000	0.000000	4.000000	4.000000	2.000000	5.280000e+02	1.780000e+02	2.955665e+03	254.000000
75%	0.685777	119.000000	2.000000	5.000000	12.000000	10.000000	1.362000e+03	1.064000e+03	1.250000e+05	254.000000
max	59.999989	132.000000	12.000000	10.000000	10646.000000	11018.000000	1.435577e+07	1.465753e+07	1.000000e+06	255.000000

sttl	...	ct_src_dport_ltm	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	label
257673.000000	...	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000	257673.000000
180.000931	...	5.238271	4.032677	8.322964	0.012819	0.012850	0.132905	6.800045	9.121049	0.014274	0.639077
102.488268	...	8.160622	5.831515	11.120754	0.116091	0.116421	0.681854	8.396266	10.874752	0.118618	0.480269
0.000000	...	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000
62.000000	...	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	2.000000	2.000000	0.000000	0.000000
254.000000	...	1.000000	1.000000	3.000000	0.000000	0.000000	0.000000	3.000000	4.000000	0.000000	1.000000
254.000000	...	4.000000	3.000000	8.000000	0.000000	0.000000	0.000000	8.000000	11.000000	0.000000	1.000000
255.000000	...	59.000000	46.000000	63.000000	4.000000	4.000000	30.000000	60.000000	62.000000	1.000000	1.000000

Figure 4. 1: Dataset Description

4.2 Data Frame

The dataset was loaded into the Pandas data frame for easy analysis, model development, and prediction. Table 4.1 depicts the Pandas Data Frame comprising 257672 rows and 44 columns representing different types of attacks.

Table 4. 2: Dataset Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 257673 entries, 0 to 257672
Data columns (total 44 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   dur                 257673 non-null  float64
 1   proto               257673 non-null  object
 2   service             257673 non-null  object
 3   state               257673 non-null  object
 4   spkts               257673 non-null  int64
 5   dpkts               257673 non-null  int64
 6   sbytes              257673 non-null  int64
 7   dbytes              257673 non-null  int64
 8   rate                257673 non-null  float64
 9   sttl                257673 non-null  int64
 10  dttl                257673 non-null  int64
 11  sload               257673 non-null  float64
 12  dload               257673 non-null  float64
 13  sloss               257673 non-null  int64
 14  dloss               257673 non-null  int64
 15  sinpkt              257673 non-null  float64
 16  dinpkt              257673 non-null  float64
 17  sjit                257673 non-null  float64
 18  djit                257673 non-null  float64
 19  swin                257673 non-null  int64
 20  stcpb               257673 non-null  int64
 21  dtcpb               257673 non-null  int64
 22  dwin                257673 non-null  int64
 23  tcprtt              257673 non-null  float64
 24  synack              257673 non-null  float64
 25  ackdat              257673 non-null  float64
 26  smean               257673 non-null  int64
 27  dmean               257673 non-null  int64
 28  trans_depth         257673 non-null  int64
 29  response_body_len   257673 non-null  int64
 30  ct_srv_src          257673 non-null  int64
 31  ct_state_ttl        257673 non-null  int64
 32  ct_dst_ltm          257673 non-null  int64
 33  ct_src_dport_ltm    257673 non-null  int64
 34  ct_dst_sport_ltm    257673 non-null  int64
 35  ct_dst_src_ltm      257673 non-null  int64
 36  is_ftp_login        257673 non-null  int64
 37  ct_ftp_cmd          257673 non-null  int64
 38  ct_flw_http_mthd    257673 non-null  int64
 39  ct_src_ltm          257673 non-null  int64
 40  ct_srv_dst          257673 non-null  int64
 41  is_sm_ips_ports     257673 non-null  int64
 42  attack_cat          257673 non-null  object
 43  label               257673 non-null  int64
dtypes: float64(11), int64(29), object(4)
memory usage: 86.5+ MB
```

4.2.2 Data Frame after Encoding

The transformed Data Frame depicted in Table 4.3 shows the first few rows with encoded categorical columns: proto, service, and state are converted to numerical codes (e.g., 119, 0, and 5), making them suitable for machine learning models. The attack_cat column retains its categorical labels (like 'Normal') but is optimized as a categorical data type for memory efficiency. Other numerical columns, such as dur, spkts, and dpkts, remain unchanged, representing various network traffic features. The label column indicates the presence or type of attack, with 0 likely representing 'Normal' or 'No Attack'.

Table 4. 1: Data Frame after Encoding

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	...	ct_dst_sport_ltm
0	0.000011	119	0	5	2	0	496	0	90909.0902	254	...	1
1	0.000008	119	0	5	2	0	1762	0	125000.0003	254	...	1
2	0.000005	119	0	5	2	0	1068	0	200000.0051	254	...	1
3	0.000006	119	0	5	2	0	900	0	166666.6608	254	...	1
4	0.000010	119	0	5	2	0	2126	0	100000.0025	254	...	1

ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	attack_cat	label
2	0	0	0	1	2	0	Normal	0
2	0	0	0	1	2	0	Normal	0
3	0	0	0	1	3	0	Normal	0
3	0	0	0	2	3	0	Normal	0
3	0	0	0	2	3	0	Normal	0

Confusion Matrix for the Random Forest

The confusion matrix for the Random Forest Classifier depicted in Figure 4.13 provides a detailed breakdown of the model's performance in classifying data as either normal traffic (class '0') or attack traffic (class '1'). The matrix contains four key components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In this scenario, the matrix shows that there are 8,196 true negatives, indicating that the model correctly predicted 8,196 instances as normal traffic. The true positives, totalling 47,396, reflect the instances where the model accurately identified attack traffic. These high numbers of true positives and true negatives suggest that the model is effective at correctly identifying both attack and normal instances.

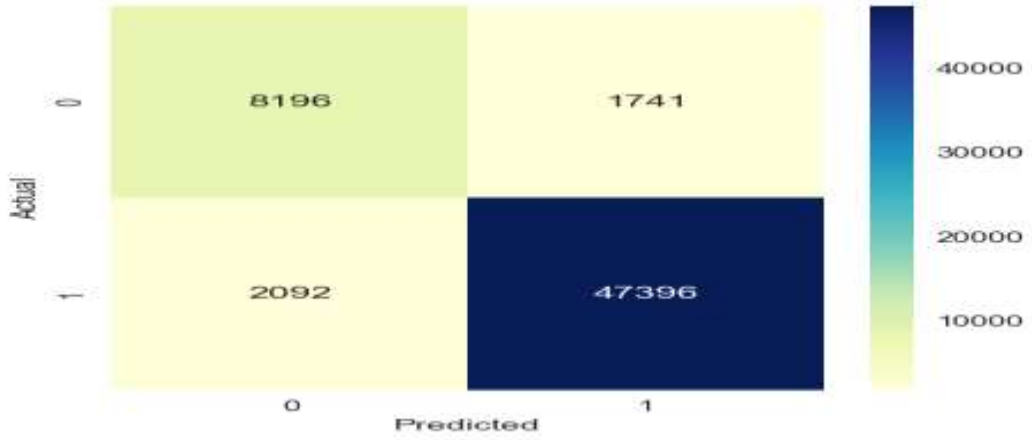

Figure 4. 2: Confusion Matrix for the Random Forest Classifier

However, the matrix also reveals areas for potential improvement. There are 1,741 false positives, where the model mistakenly classified normal traffic as an attack. Similarly, there are 2,092 false negatives, where the model incorrectly predicted attack traffic as normal. These misclassifications highlight the trade-off the model faces between sensitivity (recall) and specificity. While the high number of true

positives demonstrates the model's strong ability to detect attacks, the presence of false positives and false negatives indicates there is still a possibility of error in the predictions.

ROC Curves for Random Forest Classifier

The ROC curve illustrates the model's ability by plotting the true positive rate (TPR or sensitivity) against the false positive rate (FPR) at various threshold settings. A model with perfect discrimination would have a curve passing through the top left corner of the plot, representing a TPR of 1 and an FPR of 0. In this case, the ROC curves for both class 0 (normal traffic) and class 1 (attack traffic) depicted in Figure 4.14 are almost perfect, hugging the top left corner and indicating that the RandomForestClassifier is highly effective at distinguishing between these two classes.

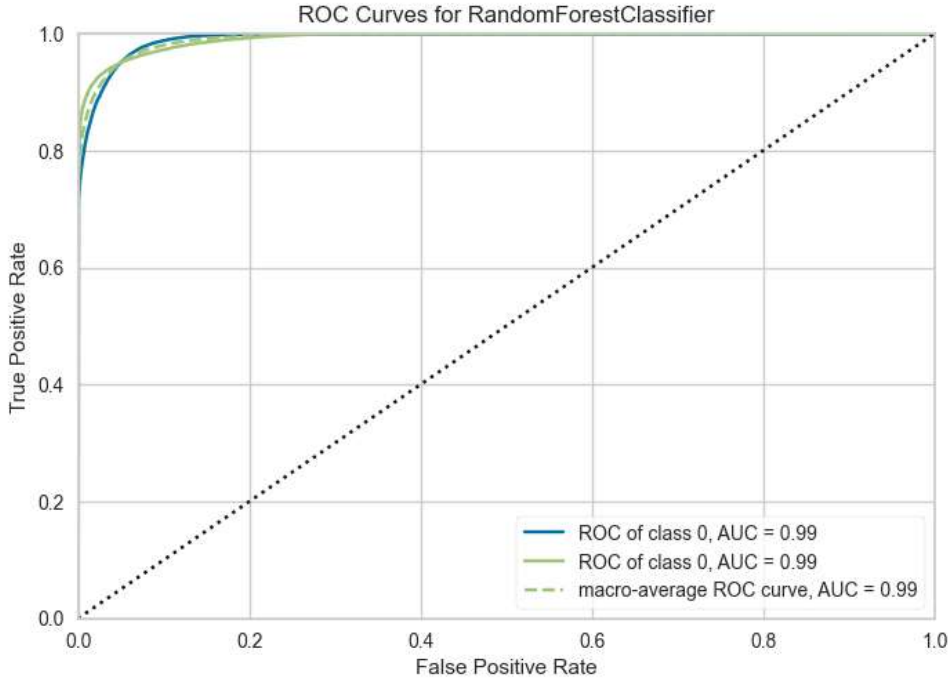

Figure 4.4: ROC Curves for Random Forest Classifier

4.3 Comparison with Existing Studies

able 4.10 presents a comparative analysis of the performance of various machine learning models applied to DDoS attack detection and classification, highlighting the improvements achieved by the proposed study in relation to existing research. The comparison focuses primarily on accuracy scores, providing a clear evaluation of how different models perform across similar cybersecurity tasks.

Table 4. 2: A comparison with the literature.

S/N	Author/Year	Machine Learning Model	Accuracy (%)
1.	Shukla & Sengupta (2022)	Ensemble Machine Learning Using Threshold and Aggregation Methods	93%
2.	Inuwa & Das (2024)	Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Tree (DT), Logistic Regression (LR), and k-Nearest Neighbours (k-NN).	69.33–92.13

3.	Lai *et al.*(2023)	Binary and Multi-Cat	86 –89
4.	This Study (2024)	Random Forest, Decision Tree, K-Nearest Neighbor, XG Boost, Light DM, Logistic Regression, Naïve Bayes and Stacked Model	72.1-93.8

The table evaluates the effectiveness of different machine learning approaches for detecting and classifying DDoS attacks within network environments, including IoT systems. Shukla and Sengupta (2022) employed an ensemble learning technique based on thresholding and aggregation, achieving an accuracy of 93%, which highlights the strength of combining multiple models to improve detection performance.

Inuwa and Das (2024) utilized a variety of machine learning algorithms, including SVM, ANN, Decision Tree, Logistic Regression, and k-NN, with accuracy results ranging from 69.33% to 92.13%. This variation reflects differences in model suitability, dataset characteristics, and parameter optimization, indicating that no single model consistently outperforms others across all scenarios.

Lai et al. (2023) focused on both binary and multi-class classification approaches, achieving accuracy levels between 86% and 89%. While these results demonstrate reasonable performance, they suggest limitations in handling complex multi-class intrusion detection scenarios such as diverse DDoS attack patterns in IoT environments.

The proposed study (2024) adopts a comprehensive and diversified approach by integrating multiple machine learning models, including Random Forest, Decision Tree, K-Nearest Neighbors, XGBoost, LightGBM, Logistic Regression, Naïve Bayes, and a stacked ensemble model. The study achieved accuracy values ranging from 72.1% to 93.8%, with the highest performance attributed to the ensemble and stacked approaches. This demonstrates the effectiveness of combining multiple models to leverage their individual strengths in improving classification accuracy.

Overall, the comparison highlights that ensemble and hybrid machine learning techniques offer superior performance in DDoS attack classification, particularly in complex and dynamic IoT network environments. The findings reinforce the importance of model diversity, proper feature engineering, and optimization in developing efficient and reliable intrusion detection systems.

5. CONCLUSION

The primary objective of this study was to develop an efficient and intelligent machine learning-based model for the classification of Distributed Denial of Service (DDoS) attacks in IoT networks. To achieve this, a comprehensive dataset containing diverse network traffic patterns and attack categories, including Denial of Service (DoS) and related intrusion types, was utilized. The dataset underwent extensive preprocessing using Python, which involved data cleaning, handling missing values, removing noise and inconsistencies, and analyzing feature correlations to ensure high data quality and reliability.

To improve model performance, data normalization was carried out using the Min-Max scaling technique, which enhanced feature representation and ensured uniformity across input variables. Irrelevant and redundant features were eliminated during the preprocessing stage to optimize the learning process and reduce computational complexity, which is particularly important in resource-constrained IoT environments.

The classification model was developed using multiple machine learning algorithms, including Random Forest, Decision Tree, K-Nearest Neighbors (KNN), XGBoost, LightGBM, Logistic Regression, and

Naïve Bayes. These models were trained, tested, and evaluated using the processed dataset to determine their effectiveness in accurately detecting and classifying DDoS attacks.

The results demonstrated that the proposed model achieved high performance, with an accuracy of 93.5%, a recall rate of 96.8%, and strong precision and F1-scores. These results indicate the model's robustness in correctly identifying DDoS attacks while minimizing false negatives, which is critical for maintaining the availability and security of IoT networks.

Overall, this study contributes significantly to the field of IoT cybersecurity by presenting a reliable and scalable machine learning-based approach for DDoS attack classification. The findings highlight the effectiveness of integrating advanced machine learning techniques in enhancing real-time threat detection and improving the resilience of IoT systems against cyber-attacks.

REFERENCES

Adewuyi, A., Oladele, A. A., Enyiorji, P. U., Ajayi, O. O., Tsambatare, T. E., Oloke, K., & Abijo, I. (2024). The convergence of cybersecurity, Internet of Things (IoT), and data analytics: Safeguarding smart ecosystems. *World Journal of Advanced Research and Reviews*, *23*(1), 379-394.

Agarwal, L., Jaint, B., & Mandpura, A. K. (2024). Reducing overfitting in deep learning intrusion detection for power systems with CTGAN. Chaos, Solitons & Fractals, 188, 115603.

Aggarwal, D., Sharma, D., & Saxena, A. B. (2023). Role of AI in cyber security through Anomaly detection and Predictive analysis. *Journal of Informatics Education and Research*, *3*(2).

Ahmed, M. R., Shatabda, S., Islam, A. M., & Robin, M. T. I. (2023). Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques--A Comprehensive Survey. *Authorea Preprints*.

Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detectiona comparative study. *Future Internet*, *15*(7), 243.

Al-Ghuwairi, A. R., Sharrab, Y., Al-Fraihat, D., AlElaimat, M., Alsarhan, A., & Algarni, A. (2023). Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. *Journal of Cloud Computing*, *12*(1), 127.

Al-Hashem, N., & Saidi, A. (2023). The psychological aspect of cybersecurity: understanding cyber threat perception and decision-making. *International Journal of Applied Machine Learning and Computational Intelligence*, *13*(8), 11-22.

AL-Hawamleh, A. (2024). Cyber resilience framework: Strengthening defenses and enhancing continuity in business security. *International Journal of Computing and Digital Systems*, *15*(1), 1315-1331.

Al-Hoorie, A. H., Hiver, P., Larsen-Freeman, D., & Lowie, W. (2023). From replication to substantiation: A complexity theory perspective. *Language Teaching*, *56*(2), 276-291.

Al-Muntaser, B., Mohamed, M. A., & Tuama, A. Y. (2023). Real-Time Intrusion Detection of Insider Threats in Industrial Control System Workstations Through File Integrity Monitoring. *International Journal of Advanced Computer Science and Applications*, *14*(6).

Alam, S., Israr, J., & Kumar, A. (2024). Artificial Intelligence and Machine Learning in Bioinformatics. In *Advances in Bioinformatics* (pp. 321-345). Singapore: Springer Nature Singapore.

Aldallal, A. and Alisa, F., 2021. Effective intrusion detection system to secure data in cloud using machine learning. Symmetry, 13(12), p.2306.

Alhasawi, Y., & Alghamdi, S. (2024). Federated Learning for Decentralized DDoS Attack Detection in IoT Networks. *IEEE Access*, *12*, 42357-42368.