Research Article

HİBRİT MAKİNE ÖĞRENME TEKNİKLERİ YOLUYLA GELİŞTİRİLMİŞ DDoS SALDIRISI TESPİTİ

Feraidoon FARAHMANDNIA[†] **Serhat ÖZEKES** [††]

[†] Üsküdar University, Institute Of Science, Istanbul, Türkiye

[††] Marmara University, Faculty of Technology, Department of Computer Engineering, İstanbul, Türkiye

Farahmandnia.fr@gmail.com, serhat.ozekes@marmara.edu.tr

0009-0004-9516-6122, 0000-0002-7432-0272

Atıf/Citation: FARAHMANDIA, F., ÖZEKES, S. (2025). Hibrit Makine Öğrenme Teknikleri Yoluyla Geliştirilmiş Ddos Saldirisi Tespiti, Journal of Technology and Applied Sciences 7(2) s.275-307, DOI: 10.56809/icujtas.1513881

ÖZET

Bu çalışma, makine öğrenimi tekniklerini kullanarak Dağıtılmış Hizmet Reddi (DDoS) saldırılarını tespit etmek için güçlü mekanizmaların geliştirilmesini araştırmaktadır. Araştırmanın temel amacı, bir metaklasifikatör yığma modeli ve transfer öğrenme modeli olmak üzere iki farklı yaklaşımı keşfederek DDoS tespit doğruluğunu artırmaktır. Bu modelleri eğitmek ve değerlendirmek için CICDDoS2019 ve CICIDS2017 veri setleri kullanılmıştır. İlk yaklaşımda, K-En Yakın Komşu, Destek Vektör Makinesi ve Rastgele Orman algoritmaları bir lojistik regresyon metaklasifikatörü kullanılarak birleştirilmiştir. Bu topluluk yöntemi, her bir algoritmanın güçlü yönlerinden yararlanarak doğruluk, kesinlik, geri çağırma ve F1-skora gibi performans ölçümlerinde iyileşme sağlamıştır. Yığma modeli %99.94 doğruluk elde etmiştir. İkinci yaklaşımda ise, CICIDS2017 üzerinde önceden eğitilmiş bir Yapay Sinir Ağı modeli, CICDDoS2019 veri seti kullanılarak ince ayar yapılarak transfer öğrenme uygulanmıştır. Bu yöntem, bilgi aktarımının avantajlarını göstererek %99.81 doğruluk ve önemli ölçüde azaltılmış eğitim süresi ile yüksek tespit performansı elde etmiştir. Bulgular, her iki yaklaşımın da DDoS tespitini önemli ölçüde iyileştirdiğini ve metaklasifikatör yaklaşımının biraz daha yüksek performans sağladığını, ancak daha fazla hesaplama gücü gerektirdiğini göstermektedir. Transfer öğrenme yaklaşımı, performans ve verimlilik arasında pratik bir denge sunarak hızlı model dağıtımı gerektiren senaryolar için uygun hale gelmektedir. Sonuç olarak, araştırma, gelişmiş makine öğrenimi tekniklerinin etkili DDoS tespit sistemlerinin geliştirilmesinde taşıdığı potansiyeli vurgulamaktadır.

Anahtar Kelimeler: DDoS Tespiti, Makine Öğrenimi, Metaklasifikasyon, Transfer Öğrenme, Siber Güvenlik

Geliş/Received	:	**17.07.2024**
Gözden Geçirme/Revised	:	**02.09.2024**
Kabul/Accepted	:	**17.10.2024**

ENHANCED DDoS ATTACK DETECTION THROUGH HYBRID MACHINE LEARNING TECHNIQUES

ABSTRACT

This study investigates the development of robust detection mechanisms for Distributed Denial of Service (DDoS) attacks using machine learning techniques. The primary objective of the research is to enhance DDoS detection accuracy by exploring two distinct approaches: a meta-classifier stacking model and a transfer learning model. The CICDDoS2019 and CICIDS2017 datasets are utilized to train and evaluate these models. In the first approach, the K-Nearest Neighbors, Support Vector Machine, and Random Forest algorithms are combined using a logistic regression metaclassifier. This ensemble method leverages the strengths of each individual algorithm, resulting in improved performance metrics such as accuracy, precision, recall, and F1-score. The stacking model achieved an accuracy of 99.94%. The second approach employs transfer learning, where a pre-trained Artificial Neural Network model on the CICIDS2017 is fine-tuned using the CICDDoS2019 dataset. This method demonstrates the advantages of knowledge transfer, achieving high detection performance with an accuracy of 99.81% and significantly reduced training time. The findings indicate that both approaches significantly improve DDoS detection. The metaclassifier approach achieves higher performance metrics but is more computationally intensive. The transfer learning approach offers a practical balance between performance and efficiency, making it suitable for scenarios requiring rapid model deployment. In conclusion, the research highlights the potential of advanced machine learning techniques in developing effective DDoS detection systems.

Keywords: DDoS Detection, Machine Learning, Metaclassifier, Transfer Learning, Cybersecurity

1. INTRODUCTION

The widespread use of digital technology and the substantial incorporation of Internet-based services have revolutionized the manner in which individuals, enterprises, and institutions engage and carry out their activities. Despite the many advantages of the digital era, there remains a notable danger to network security and stability through cyber threats, particularly Distributed Denial-of-Service (DDoS) assaults (Merkebaiuly, 2024). With the increasing reliance on digital infrastructure for critical functions such as communication, commerce, and information dissemination, cybercriminals have capitalized on vulnerabilities within network protocols and systems to orchestrate malicious attacks. Among these threats, DDoS attacks have become a favored tactic due to their potential for causing widespread disruption with minimal effort (Gupta, 2008; Cheema et al., 2022).

DDoS attacks involve the coordinated attack on specific servers or networks by flooding them with a large amount of traffic from various sources, overwhelming their capacity to function properly in response to legitimate requests (Cheema et al., 2022). These assaults can manifest in different forms, including volumetric attacks, protocol attacks, and application layer attacks, each presenting unique challenges for identification and mitigation (Rafsanjani and Kazeminejad, 2014; Jia et al., 2020). The repercussions of DDoS attacks extend far beyond temporary service outages, often leading to monetary losses, harm to an organization's image, and a decrease in customer confidence (Shahzad and Mateen, 2021). The growing prevalence and complexity of DDoS assaults underscore the urgent need for resilient security systems capable of protecting against evolving threats (Gupta and Dahiya, 2021).

Traditional methods for detecting and mitigating DDoS attacks, such as rate limiting, blacklisting, and traffic filtering, have proven inadequate in handling the scale and complexity of modern attacks (Parekh and Sathwara, 2017). These rule-based approaches often struggle to differentiate between legitimate and malicious traffic, resulting in false positives, false negatives, and service degradation (Chahal et al., 2019). Machine learning techniques have emerged as a promising solution for enhancing DDoS detection capabilities (Tiwari et al., 2018). By leveraging computational intelligence and data-driven algorithms, machine learning models can learn intricate patterns and anomalies from large-scale network traffic data, enabling more effective and accurate differentiation between malicious activity and benign communications (Li et al., 2008; Herrera et al., 2015).

The dynamic and adaptable nature of DDoS assaults necessitates advanced defense mechanisms capable of recognizing and neutralizing new threats in real-time or near-real-time scenarios (Gopinaath et al., 2022). By harnessing machine learning capabilities, organizations can augment their existing security infrastructure and fortify defenses against evolving cyber threats. In light of these considerations, this research aims to address the challenge of DDoS detection by exploring and implementing innovative machine learning techniques. By leveraging insights from the latest research in cybersecurity and machine learning, this study aims to build models that are both resilient and adaptable, with the ability to reliably identify and mitigate DDoS assaults, thereby improving the resilience and security of network infrastructures in the face of emerging cyber threats (Li and Castagna, 2004; Mishina et al., 2015).

To address the challenge of Distributed Denial-of-Service (DDoS) detection, this study makes several key contributions. First, it develops a novel hybrid detection model that enhances DDoS detection accuracy by combining machine learning models through a meta-classifier stacking approach (Sultana and Islam, 2019; Rajendran and Vincent, 2021). Second, the study introduces a transfer learning approach using pre-trained Artificial Neural Network (ANN) weights to reduce training time while maintaining high detection performance on the CICDDoS2019 dataset (Gurjar and Voditel, 2022; Islam, 2024).

A comprehensive comparative analysis is conducted between the meta-classifier stacking and transfer learning approaches, demonstrating their respective advantages in terms of detection accuracy, computational complexity, and efficiency (Huang and Zhou, 2021). Additionally, this research employs extensive data preprocessing techniques, including feature normalization, transformation, and handling of missing values, to improve the model's robustness and reliability using the CICIDS2017 and CICDDoS2019 datasets (Cheema et al., 2022).

The paper is structured as follows. The next section presents a literature review of existing DDoS detection methods and advancements in machine learning for cybersecurity. The subsequent section discusses the materials and methods used, detailing the datasets, data preprocessing steps, and model design, including both the meta-classifier stacking approach and transfer learning model. Following that, the results section provides a thorough performance evaluation of the proposed models using metrics like accuracy, precision, recall, and F1-score. The discussion section then analyzes the findings, comparing them with previous studies and identifying potential areas for further research. Finally, the conclusion summarizes the key findings and suggests future directions for enhancing DDoS detection using advanced machine learning techniques.

2. LITERATURE REVIEW

2.1. Overview

This study offers a thorough analysis of previous research on Distributed Denial of Service (DDoS) attacks through a comprehensive literature review. The review examines what constitutes a DDoS incident, how these attacks have evolved over time, and the common methods used by attackers. It also explores both traditional and innovative approaches for detecting and mitigating DDoS attacks, including Software-Defined Networking (SDN), which allows network administrators to manage network services by abstracting lower-level functionality, as well as machine learning techniques. By analyzing recent publications, this review aims to provide insights into the development of effective DDoS defense strategies.

2.2. Explanation of DDoS Attacks

Attacks known as DDoS are intentional attempts to stop a server, service, or network's normal operation by flooding it or the infrastructure around it with an excessive amount of Internet traffic (Singh et al., 2022). Historically, DDoS attacks have evolved from simple nuisances carried out by amateur hackers to sophisticated disruptions orchestrated by well-funded adversaries, with the magnitude of attacks escalating from mere megabytes to hundreds of terabytes of data (Kasture, 2023). The range of attack methods and the growing complexity of the Internet environment, which includes the emergence of SDN and Internet of Things (IoT) devices, make it difficult to mitigate DDoS attacks. (Bhushan et al., 2022).

Interestingly, while DDoS attacks have become more complex, the strategies for their mitigation have also advanced. Stochastic Gradient Descent and Support Vector Machine are two examples of machine learning techniques that have been used to achieve great accuracy in attack detection.(Umamaheswari et al., 2023). Additionally, the programmability and central management features of SDN have been leveraged to develop mitigation strategies across different planes of the network architecture (Li & Wang, 2022; Shakya & Karnani, 2022). However, the effectiveness of these solutions can be inconsistent due to the dynamic nature of DDoS attack patterns and the continuous evolution of attack methods (Kasture, 2023).

In summary, DDoS attacks present a significant challenge to network security, with their definition rooted in the intent to disrupt service availability through traffic overload. The historical progression of these attacks reflects a trend towards growing in scale and complexity. Mitigating DDoS assaults remains a challenge due to the diversity of attack methods and the need for solutions that can adapt to evolving threats. Research continues to focus on developing more effective detection and mitigation techniques, with machine learning and SDN-based strategies showing promise in recent studies (Bhushan et al., 2022; Kasture, 2023; Li & Wang, 2022; Shakya & Karnani, 2022; Umamaheswari et al., 2023).

2.3. Traditional Methods for DDoS Detection and Mitigation

Conventional approaches to detecting and mitigating Distributed Denial-of-Service (DDoS) attacks have mostly concentrated on identifying and dealing with the overwhelming volume of network traffic associated with these attacks. These methods include a variety of strategies, Examples of intrusion detection techniques include anomaly-based methods, which detect unusual patterns of network traffic, as well as preventative strategies like as packet filtering and rate restriction, which reduce the impact of an attack (Ojha et al., 2023). Additionally, DDoS defense strategies have been categorized into detection, defense, and mitigation, with an emphasis on the importance of understanding different attack types and methodologies (Singh et al., 2022).

However, these traditional methods have faced challenges given the dynamic nature of DDoS assaults, which have grown in complexity and scale. For instance, the traditional internet architecture's susceptibility to DDoS attacks has been a significant concern, leading to the development of new defense techniques. Moreover, the effectiveness of traditional methods can be limited by the increasing variety of vulnerable hosts and the sophistication of attack networks or botnets (Singh et al., 2022; Suhag & Daniel, 2022).

In summary, while traditional methods for DDoS detection and mitigation have provided a foundation for defending against such attacks, the dynamic and sophisticated nature of modern DDoS threats necessitates continuous improvement and adaptation of these strategies. Research indicates that integrating sophisticated methodologies, including machine learning and artificial intelligence, can enhance the effectiveness of DDoS defense solutions (Chong et al., 2023; Suhag & Daniel, 2022). Therefore, it is crucial to integrate traditional methods with newer, more advanced approaches to develop robust and efficient DDoS mitigation strategies.

2.4. Machine Learning Techniques for DDoS Detection

Machine learning (ML) approaches have gained considerable attention in recent years due to their effectiveness in detecting DDoS assaults. These approaches leverage the ability of ML algorithms to analyze large volumes of network data and identify complex patterns indicative of malicious activity. Various ML techniques, such as Random Forest (RF) , Decision Trees, Support Vector Machines (SVM), and Neural Networks have shown notable success in DDoS detection tasks (T et al., 2023).

SVM, for instance, has been widely utilized in DDoS detection due to its effectiveness in classifying data into different categories based on a set of input features (T et al., 2023). By constructing hyperplanes that effectively separate normal and attack traffic, SVM-based classifiers can accurately identify instances of DDoS attacks.

Decision trees, such as the C4.5 algorithm, have also found application in DDoS detection owing to their interpretability and efficiency in classifying network traffic(Li & Wang, 2022). These models segment the feature space into hierarchical decision rules, allowing for the detection of attack patterns based on input feature values.

Random Forest, that is an ensemble learning method consisting of several decision trees, has demonstrated promise in DDoS detection tasks by aggregating the outputs of individual trees to enhance detection accuracy (Wang et al., 2017). This approach improves resilience against noise and outliers in the data, leading to more robust detection capabilities.

Furthermore, recent breakthroughs in the field of deep learning have focused on two specific types of neural networks, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). have facilitated the creation of advanced DDoS detection systems that can learn analyzing intricate patterns derived from unprocessed network traffic data. (Khan, 2021). These deep learning architectures offer enhanced capabilities for detecting subtle and evolving DDoS attack patterns.

2.5. Review of Relevant Research Studies and Approaches

Multiple research papers have made major contributions to the subject of recognizing DDoS attacks by utilizing diverse methodology and tactics to minimize the impact of these assaults.

Seifousadati et al. (2021) conducted a study titled "A Machine Learning Approach for DDoS Detection on IoT Devices" to address the increasing threat of DDoS attacks on IoT technology. The study highlights the effectiveness of AdaBoost and XGBoost algorithms, achieving 100% accuracy and an F1-Score of 1 on the CICDDoS2019 dataset. It also identifies the top 10 features crucial for predicting network traffic classes, emphasizing the importance of feature selection in improving DDoS detection systems. However, the study notes the limitations of the Naïve Bayes algorithm due to its suboptimal performance in terms of F1-Score and False Positive Percentage.(Seifousadati et al., 2021)

Halladay et al. (2022) conducted a study, published in IEEE Access, focusing on the detection and analysis of DDoS attacks using time-based attributes. The experimental setup included two scenarios (A and B) and employed classifiers such as LightGBM, XGBoost, Adaptive Boosting, and Deep Neural Networks (DNN). The study highlighted the importance of time-based features in identifying unique traffic flow signatures. Scenario A showed exceptional performance with accuracies between 98-99%, while Scenario B experienced a slight degradation in accuracy and F1-score but improved training time (Halladay et al., 2022).

Elsayed et al. (2020) explored the use of Deep Learning (DL) techniques, specifically RNN, for enhancing DDoS attack detection. Their study highlights DL's ability to autonomously extract features from raw data, improving detection accuracy without human intervention. The research emphasizes the importance of data partitioning and using Receiver Operating Characteristic (ROC) curves to evaluate model performance. They found optimal performance with 70% of data used for training and underscored the significance of the Area Under the Curve (AUC) metric in assessing the model's discriminative ability (Elsayed et al., 2020).

Gopinaath et al. (2022) assessed four categorization models: Random Forest, K-Nearest Neighbors (KNN), Decision Tree, and ANN. They evaluated these models using Accuracy, Recall, Precision, and F1-score metrics. The ExtraTreesClassifier was used for feature selection, optimizing computational efficiency by selecting the top 15 features for training. Using the CICDDoS2019 dataset, which includes diverse DDoS attack classes, the models were trained for binary classification to differentiate between benign and malicious traffic (Gopinaath et al., 2022).

Gaur and Kumar (2022) investigated the performance of various machine learning algorithms, including KNN, Decision Tree, Random Forest, and ANN, for detecting DDoS attacks on IoT devices using the CICDDoS2019 dataset. Their study highlighted the ANN model's exceptional accuracy of 99.95% in identifying malicious IP addresses, enhanced by using the ExtraTreesClassifier for feature selection and focusing on the top 15 features. The research emphasized minimizing false positives and ensuring robust model training by pre-processing data, removing outliers, and encoding labels. Their findings underscore the ANN model's effectiveness in improving network security, while also noting the need for further exploration into scalability, feature selection strategies, and integrating technologies like blockchain for enhanced IoT security (Gaur & Kumar, 2022).

Md. Alamgir Hossain's(2023) study, "Enhanced Ensemble-Based DDoS Attack Detection with Novel Feature Selection: A Robust Cybersecurity Approach," introduces innovative methodologies to improve DDoS attack detection. The research emphasizes the importance of feature selection, exploring metrics such as “Total Length of Bwd Packets,” “Active Mean,” and “Flow IAT Std.” Hossain employs techniques like correlation analysis and principal component analysis (PCA) to identify crucial features. The study proposes a model development pipeline, including dataset preparation, ensemble approach selection (e.g., Random Forest), and model evaluation using precision, recall, accuracy, and F1-score. Robustness analysis ensures model effectiveness against variations. The model consistently achieves high accuracy across evaluation metrics and demonstrates strong alignment with real data, as indicated by a high Cohen's Kappa score and a favorable Precision-Recall curve (Hossain, 2023).

Table 1 provides a comprehensive summary of recent research studies focused on DDoS attack detection using machine learning techniques. All the studies summarized in this table utilize the CICDDoS2019 dataset as the primary data source. These studies explore various algorithms, feature selection methods, and techniques to achieve high accuracy in identifying malicious network traffic.

Table 1. Summary of researches review

Title	Authors, Year	Algorithms	Features	Accuracy
A Machine Learning Approach for DDoS Detection on IoT Devices	Seifousadati, Ghasemshirazi, Fathian, 2021	Naïve Bayes, SVM, AdaBoost, XGBoost, KNN, Random Forest	Top 10 important features	100%
Detection and Characterization of DDoS Attacks Using Time-Based Features	Halladay, Cullen, 2022	LightGBM, XGBoost, Adaptive Boosting, DNN	Time-based features(25 features)	98-99%
DDoSNet: A Deep-Learning Model for Detecting Network Attacks	Elsayed, Le-Khac, Dev, Jurcut, 2022	RNN, Deep Learning techniques	77 features	99%
DDoS Detection using Machine Learning Techniques	Gopinaath, Amrish, Kumar, Bavapriyan, 2022	KNN, Decision Tree, Random Forest, ANN	Top 15 features by ExtraTrees	99.95%
Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices	Kumar Ranjeesh, Gaur Vimal ,2022	KNN, Decision Tree, Random Forest, ANN	Top 15 features from	99.95%
Enhanced Ensemble-Based DDoS Attack Detection	Md. Alamgir Hossain, 2023	Ensemble Method (e.g., Random Forest)	--	100%

3. MATERIALS AND METHODS

3.1. Overview

This study employs two distinct approaches to enhance the analysis of DDoS attacks: the Metaclassifier Approach and the Transfer Learning Approach.

- The Metaclassifier Approach entailed training multiple machine learning algorithms, specifically SVM, RF, and KNN, on the CICDDoS2019 dataset. These algorithms were then combined using a logistic regression-based meta-classifier as shown in Figure 1.

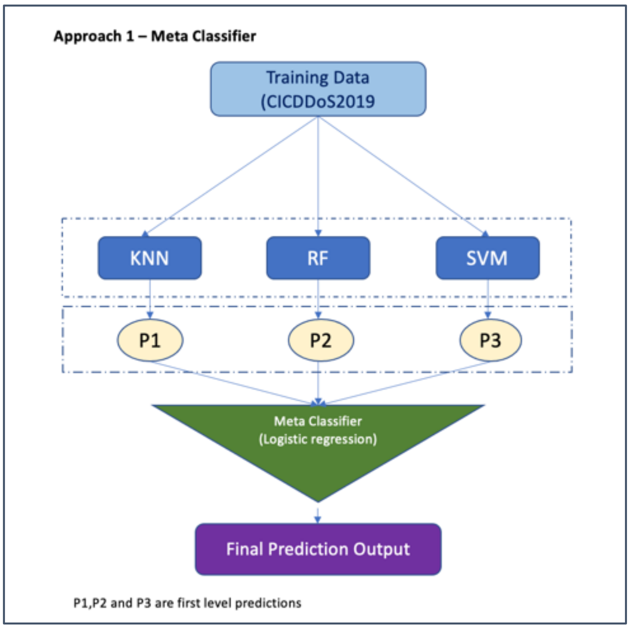

Approach 1 – Meta Classifier
Training Data (CICDDoS2019
KNN
RF
SVM
P1
P2
P3
Meta Classifier (Logistic regression)
Final Prediction Output
P1,P2 and P3 are first level predictions

Figure 1. Approach 1 Meta-Classifier

- Transfer learning is the process of applying a learned model to a new issue. It's an efficient approach when Large datasets that were used to train the model would be expensive and time-consuming to computely replicate. (Gurjar and Voditel, 2022). As illustrated in Figure 2 and Figure 3, This method enables leveraging knowledge acquired from one task to enhance output on an associated task, thereby reducing the requirement for abundant data in the new domain and speeding up model convergence(Islam, 2024).

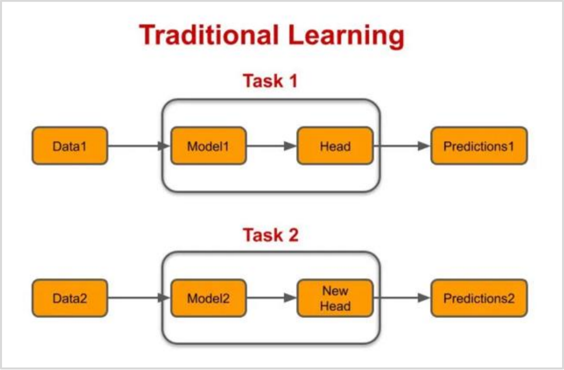


Figure 3. Traditional Learning

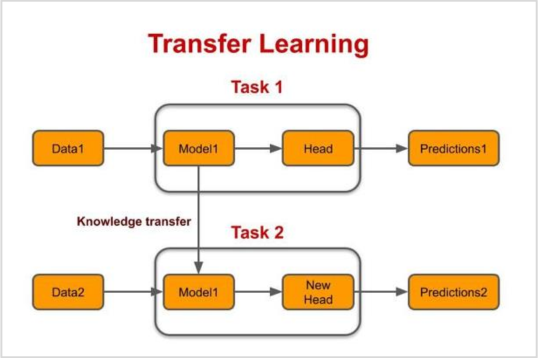


Figure 2. Transfer Learning

(Source: https://medium.com/modern-nlp/transfer-learning-in-nlp-f5035cc3f62f)

3.2. Data Collection and Preprocessing

3.2.1.Selection of Datasets

For the purpose of this study, two well-known datasets in the field of cybersecurity were selected: the CICDDoS2019 and the CICIDS2017 datasets. Both datasets were generated within the Canadian Institute of Cybersecurity.

The CICDDoS2019 dataset includes both harmless and recent typical DDoS attacks that closely mirror actual data (PCAPs). It includes the outcomes of analyze the traffic on the network using CICFlowMeter-V3, with flows labeled according to timestamp, source and destination IPs, source and destination ports, protocols, and the type of attack (Sharafaldin et al., 2019)

The CICDDoS2019 dataset is specifically designed to provide comprehensive data on Distributed Denial of Service (DDoS) attacks. It includes traffic logs from a wide variety of distributed denial of service attacks, capturing a diverse array of methods of assault patterns and behaviors. This dataset is valuable for understanding the dynamics and characteristics of DDoS attacks, rendering it a critical resource for the development of effective detection and mitigation strategies.

The CICIDS2017 dataset is extensively utilized for research on intrusion detection systems (IDS). For this study, a specific portion of the CICIDS2017 dataset was utilized, namely the “Friday-Working-Hours-Afternoon-DDos.pcap_ISCX.csv” file. This subset focuses exclusively on DDoS attack data captured during Friday working hours. It provides detailed records of network traffic during the DDoS attack, making it suitable for analyzing the specific characteristics and patterns of such attacks.

Table 2 provides a detailed summary of the information described in the datasets, including the types of attacks, the duration of data collection, the number of records, and the features available in each dataset.

Table 2. The detail of CICIDS2017 and CICDDoS2019 dataset

Dataset Name	**CICIDS2017**	**CICDDoS2019**
Total Number of Samples	225745	431371
Number of Attributes	79	88
Number Of Classes	2 class (Benign & DDoS) DDoS: 128027 Benign: 97718	2 class (Benign & DDoS) DDoS: 330540 Benign: 97831
URL	https://kaggle/input/cicids2017/DDoS-Friday-no-metadata.parquet	https://kaggle.com/code/dhoogla/cic-ddos2019-00-cleaning/input

The features are listed in Table 3.

Table 3. All Fetures of Dataset

Column Name	Description
Dst Port	Destination port number
Protocol	Protocol used in the communication
Timestamp	Timestamp of the flow
Flow Duration	Duration of the flow in seconds
Tot Fwd Pkts	Total number of packets in the forward direction
Tot Bwd Pkts	Total number of packets in the backward direction
TotLen Fwd Pkts	Total size of the forward packets in bytes
TotLen Bwd Pkts	Total size of the backward packets in bytes
Fwd Pkt Len Max	Maximum length of the forward packets
Fwd Pkt Len Min	Minimum length of the forward packets
Fwd Pkt Len Mean	Mean length of the forward packets
Fwd Pkt Len Std	Standard deviation of the length of the forward packets
Bwd Pkt Len Max	Maximum length of the backward packets
Bwd Pkt Len Min	Minimum length of the backward packets
Bwd Pkt Len Mean	Mean length of the backward packets
Bwd Pkt Len Std	Standard deviation of the length of the backward packets
Flow Byts/s	Flow bytes per second
Flow Pkts/s	Flow packets per second
Flow IAT Mean	Mean inter-arrival time of the flow
Flow IAT Std	Standard deviation of the inter-arrival time of the flow
Flow IAT Max	Maximum inter-arrival time of the flow
Flow IAT Min	Minimum inter-arrival time of the flow
Fwd IAT Tot	Total inter-arrival time of the forward packets
Fwd IAT Mean	Mean inter-arrival time of the forward packets
Fwd IAT Std	Standard deviation of the inter-arrival time of the forward packets
Fwd IAT Max	Maximum inter-arrival time of the forward packets
Fwd IAT Min	Minimum inter-arrival time of the forward packets
Bwd IAT Tot	Total inter-arrival time of the backward packets
Bwd IAT Mean	Mean inter-arrival time of the backward packets

Bwd IAT Std	Standard deviation of the inter-arrival time of the backward packets
Bwd IAT Max	Maximum inter-arrival time of the backward packets
Bwd IAT Min	Minimum inter-arrival time of the backward packets
Fwd PSH Flags	Number of times the PSH flag is set in the forward packets
Bwd PSH Flags	Number of times the PSH flag is set in the backward packets
Fwd URG Flags	Number of times the URG flag is set in the forward packets
Bwd URG Flags	Number of times the URG flag is set in the backward packets
Fwd Header Len	Total header length in the forward direction
Bwd Header Len	Total header length in the backward direction
Fwd Pkts/s	Forward packets per second
Bwd Pkts/s	Backward packets per second
Pkt Len Min	Minimum length of the packets
Pkt Len Max	Maximum length of the packets
Pkt Len Mean	Mean length of the packets

Table 3. [Countinue]

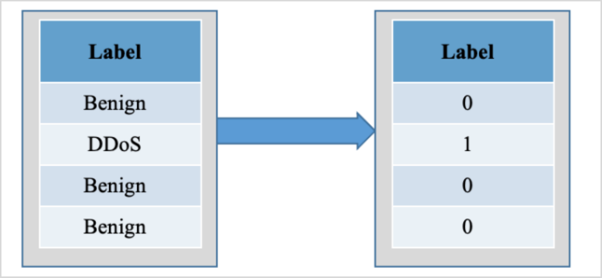
Column Name	**Description**
Pkt Len Std	Standard deviation of the length of the packets
Pkt Len Var	Variance of the length of the packets
FIN Flag Cnt	Number of times the FIN flag is set
SYN Flag Cnt	Number of times the SYN flag is set
RST Flag Cnt	Number of times the RST flag is set
PSH Flag Cnt	Number of times the PSH flag is set
ACK Flag Cnt	Number of times the ACK flag is set
URG Flag Cnt	Number of times the URG flag is set
CWE Flag Count	Number of times the CWE flag is set
ECE Flag Cnt	Number of times the ECE flag is set
Down/Up Ratio	Downstream to upstream ratio
Pkt Size Avg	Average packet size
Fwd Seg Size Avg	Average segment size in the forward direction
Bwd Seg Size Avg	Average segment size in the backward direction
Fwd Byts/b Avg	Average number of bytes per forward packet
Fwd Pkts/b Avg	Average number of packets per forward packet
Fwd Blk Rate Avg	Average block rate in the forward direction
Bwd Byts/b Avg	Average number of bytes per backward packet
Bwd Pkts/b Avg	Average number of packets per backward packet
Bwd Blk Rate Avg	Average block rate in the backward direction
Subflow Fwd Pkts	Number of packets in the forward subflow
Subflow Fwd Byts	Number of bytes in the forward subflow
Subflow Bwd Pkts	Number of packets in the backward subflow
Subflow Bwd Byts	Number of bytes in the backward subflow
Init Fwd Win Byts	Initial forward window size
Init Bwd Win Byts	Initial backward window size
Fwd Act Data Pkts	Number of forward packets with payload
Fwd Seg Size Min	Minimum segment size in the forward direction
Active Mean	Mean time of active connections
Active Std	Standard deviation of the time of active connections

Active Max	Maximum time of active connections
Active Min	Minimum time of active connections
Idle Mean	Mean time of idle connections
Idle Std	Standard deviation of the time of idle connections
Idle Max	Maximum time of idle connections
Idle Min	Minimum time of idle connections
Label	Label indicating the class or category of the network flow

- **Data Cleaning and Transformation**

The preprocessing of data is a critical preceding phase in guaranteeing the accuracy and dependability of information for future analysis and modeling operations. In this section, we outline the steps taken to clean and transform the datasets ***CICIDS2017*** and ***CICDDoS2019***, focusing on removing unnecessary columns and addressing missing or infinite values.

- **Data Cleaning**
  - **Unnecessary Feature Removal:** Removed non-contributory columns such as "Destination Port" in CICIDS2017 and detailed columns like ‘Flow ID’, ‘Source’ and ‘Destination’, Port and IP Addresses in CICDDoS2019 to streamline the datasets.
  - **Handling Missing Values:** Applied imputation to fill missing values and removed instances with high missing values to ensure data quality.
  - **Removing Duplicates:** Eliminated redundant records based on attributes like source and destination IP addresses, timestamps, and attack types to maintain data integrity.
  - **Dealing with Infinite Values:** Addressed instances with infinite values by replacing or removing them to prevent computational issues.
- **Data Transformation**
  - **One-Hot Encoding:** Converted categorical variables, such as "Protocol," into numerical format using one-hot encoding for better interpretation by machine learning algorithms.
  - **Label Encoding for Target Variable:** Used label encoding for binary classification between "Benign" and "DDoS" traffic, assigning numerical labels (e.g., 0 for "Benign" and 1 for "DDoS") for better prediction by machine learning models as shown in Figure 4.



Label
Benign
DDoS
Benign
Benign

Label
0
1
0
0

Figure 4. Label Encoding

3.2.2.Feature Normalization

Feature normalization and scaling are essential preprocessing steps for network traffic data, ensuring that features with varying scales do not bias the analysis. This improves the performance and stability of machine learning models. Several methods are used for normalization and scaling:

- **Z-score Normalization (StandardScaler):** This technique standardizes feature values to have a mean of 0 and a standard deviation of 1. It ensures that all features contribute uniformly to model training, enhancing the stability and efficiency of optimization algorithms.
- **Min-Max Scaling (Normalization):** This method scales features to a range of 0 to 1 by subtracting the minimum value and dividing by the range. It preserves the relative relationships between feature values.
- **Maximum Absolute Value Normalization:** This technique scales features to a range of [-1, 1] by dividing each feature value by the maximum absolute value in the dataset, useful when feature distributions are unknown or contain outliers.

In this study, the StandardScaler method is used for feature normalization and scaling due to its effectiveness with machine learning algorithms sensitive to feature scale variations.

3.2.3.Data Balancing

To address class imbalance in the CICIDS2017 and CICDDoS2019 datasets, different techniques were employed for each approach:

- **Approach 1 (Metaclassifier):** Class imbalance was managed by adjusting the weight parameter in KNN, SVM, and RF models. The CICDDoS2019 dataset, with 97,831 benign instances and 333,540 DDoS instances, was balanced by prioritizing the minority class (benign) through increased class weights. This method ensured models paid more attention to the minority class, enhancing sensitivity to benign traffic while maintaining computational efficiency.
- **Approach 2 (Transfer Learning):** The class_weight parameter in Keras was used to balance class distribution in both datasets. This parameter assigns a weight inversely proportional to class frequency, balancing the impact of each class during training. This approach ensured the model remained unbiased towards the majority class (DDoS) while retaining sensitivity to the minority class (benign), leading to an effective and fair DDoS detection system.

3.2.4.Feature Selection and Engineering

Feature selection is a crucial step in machine learning, involving the selection of relevant characteristics from the dataset to enhance model performance, reduce overfitting, and improve generalization. In this study, we used correlation analysis for feature reduction and XGBoost for feature selection.

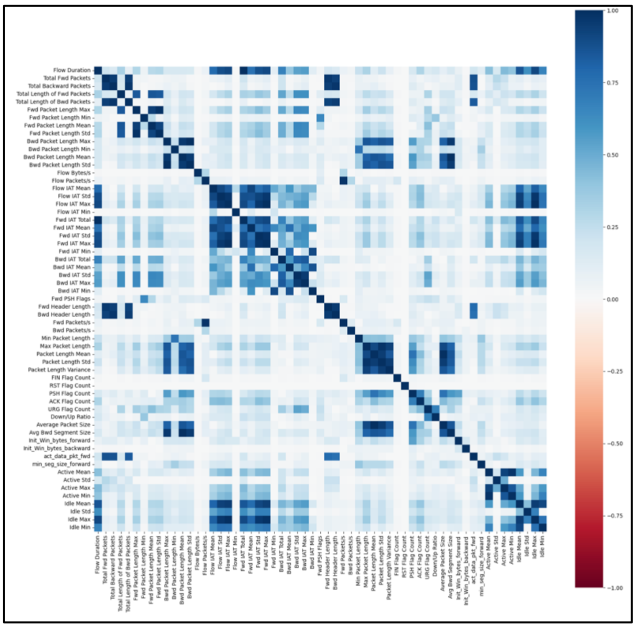


Figure 5. Correlation Matrix before reduction

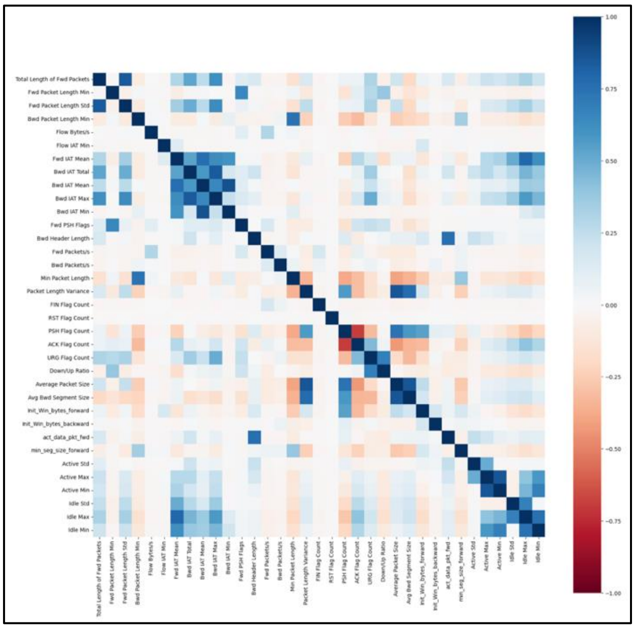


Figure 6. Correlation Matrix after reduction

- **Correlation-based methods:** These methods evaluate relationships between features. Highly correlated features are identified and eliminated to reduce redundancy. In this study, a threshold of 0.90 was used to identify and remove highly correlated features, reducing the dataset from 79 to 36 features as shown in Figure 5 and Figure 5. Correlation Matrix before reduction Figure 6.
- **Feature Selection with XGBoost:** XGBoost, a robust ensemble learning method based on decision trees, was used for automatic feature selection during training. It evaluates each feature's influence on model accuracy using feature significance scores.
- **Feature Importance Calculation:** XGBoost calculates feature importance scores based on their impact on reducing impurity in decision trees. Higher scores indicate greater relevance. After calculating feature importance scores, XGBoost selects the top features contributing most to predictive performance. This study refined the feature set from 36 to the top 15 most informative features as shown in Figure 7.

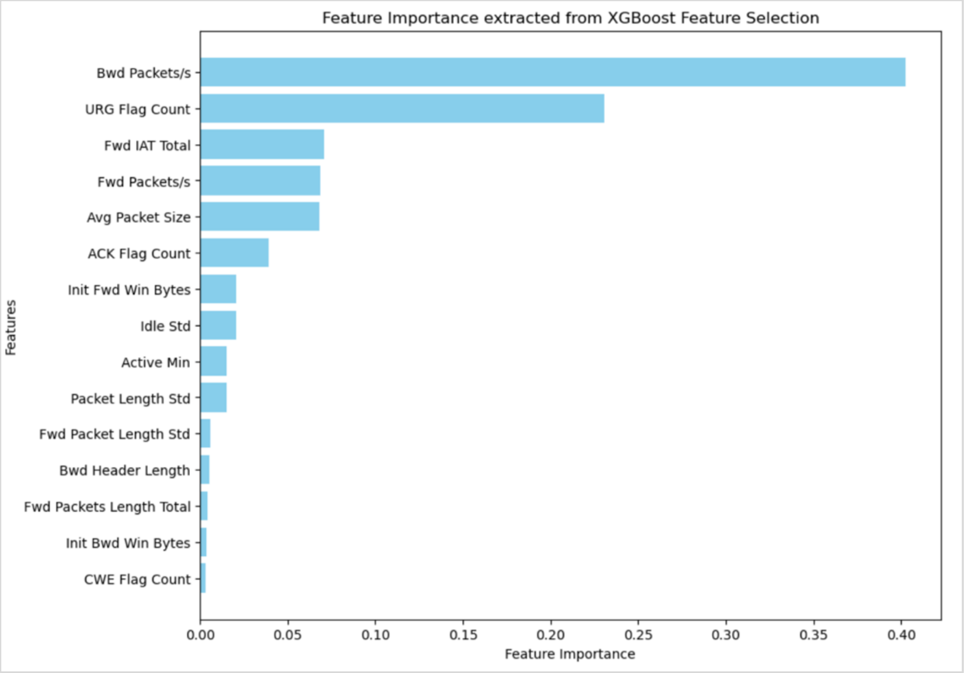


Figure 7. The 15 important feature selections

3.3. Model development

This study utilizes two different approaches, **Metaclassifier and Transfer Learning Approach:**

3.3.1.Metaclassifier Approach:

In this approach, as shown in Figure 8, three methods, including Random Forest, Support Vector Machine, and K-Nearest Neighbors, were trained on the CICDDoS2019 dataset. Subsequently, these algorithms were combined using a meta-classifier based on logistic regression. For each algorithm, the optimal hyperparameters were tuned using the Random Search method.

- **Base Algorithms:**
  - **KNN:** Classifies data points based on the predominant class among its K closest neighbors.
  - **SVM:** Determines the ideal hyperplane to classify data into distinct groups. It maximizes the margin between support vectors.
  - **RF:** Uses an ensemble of decision trees for classification tasks. It assesses branching decisions using the Gini index or entropy.

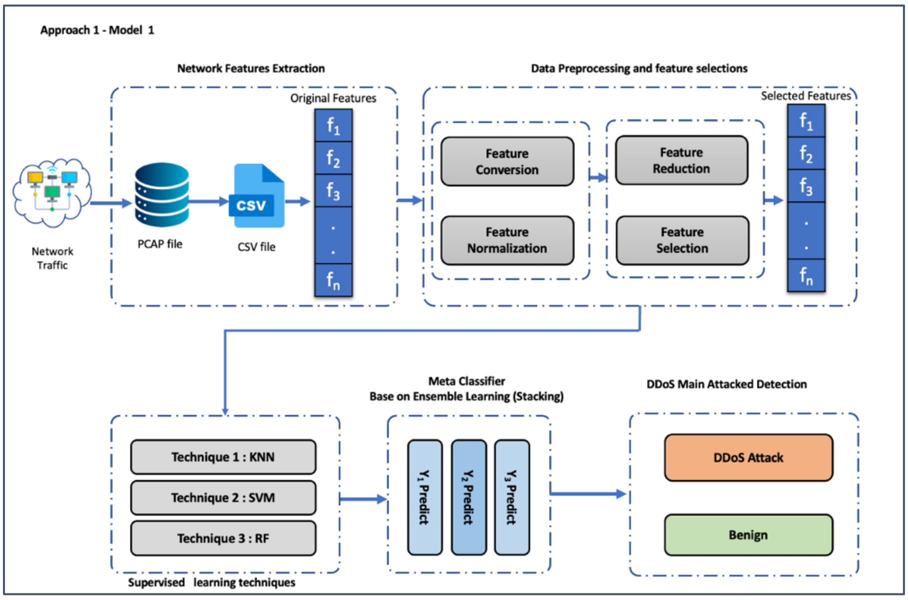


Figure 8. Diagram of approach 1

- **Training the Algorithms:**

  The dataset was divided in the following manner: 75% was used as the training set for fitting the model parameters. 10% was designated as the validation set, which was used for hyperparameter optimization using the Random Search method. The remaining 15% was set aside as the testing set to evaluate the model's performance and its ability to generalize to unseen data.

- **Hyperparameters**

  Machine learning algorithms possess adjustable hyperparameters that provide control over their behavior and performance. Our research focuses on identifying and utilizing these hyperparameters are the main areas of focus. In this study, some hyperparameters were set to their default values, while others were selected using random search method. Additionally, class_weight was set to 'balanced' to address the imbalance in the dataset. Random search is used to explore a broad range of possible values, allowing the model to find optimal configurations that might not be immediately obvious.

  The best hyperparameters were selected based on performance metrics, as detailed Table 4 this resource offers a thorough examination of the selected hyperparameters and how they influence the performance of the model.

Table 4. Selected Best Hyperparameters

Algorithm	Hyperparameter	Possible Values	Selected Hyperparameter	Selection Method
KNN	**n_neighbors**	[1, 3, 5, 7, 9, 11, 13, 15]	7	**Random Search**
	Class_weight	['uniform', 'distance']	**'distance'**	
	algorithm	['auto', 'ball_tree', 'kd_tree', 'brute']	'auto'	
	leaf_size	[10, 20, 30, 40, 50]	30	**Random Search**
	p	[1, 2]	2	
SVM	**C**	[0.1, 1, 10, 100, 1000]	10	**Random Search**
	kernel	['linear', 'poly', 'rbf', 'sigmoid']	'rbf'	
	degree	[2, 3, 4, 5] (only for 'poly' kernel)	3	
	gamma	['scale', 'auto', 0.001, 0.01, 0.1, 1]	'scale'	
	Class_weight	[None, 'balanced']	**'balanced'**	
	coef0	[0.0, 0.1, 0.5, 1.0]	0.0	
RF	**n_estimators**	[100, 200, 300, 400, 500]	300	**Random Search**
	max_features	['auto', 'sqrt', 'log2']	'sqrt'	
	max_depth	[None, 10, 20, 30, 40, 50]	30	**Random Search**
	min_samples_split	[2, 5, 10]	2	
	min_samples_leaf	[1, 2, 4]	1	
	Bootstrap	[True, False]	True	
	Class_weights	[None, 'balanced','balanced_subsample']	**'balanced'**	

- **Combining Algorithms with a Meta-Classifier**

A metaclassifier enhances classification accuracy by combining the outputs of multiple base models to make a final decision. In this study, a metaclassifier architecture is used, comprising KNN, SVM, and RF as base models, with Logistic Regression as the metaclassifier. As Figure 9 shown the methodology involves several steps: first, KNN, SVM, and RF are independently trained on the training dataset to learn patterns and generate predictions. The trained base algorithms then make predictions on the same dataset, providing probability scores or class labels. These predictions from the base algorithms are used as input features for the meta-classifier. for example If the original dataset has *n* features and three base algorithms, the input to the meta-classifier will consist of three new features. Logistic Regression is used as the meta-classifier and is trained on the new feature set (predictions from KNN, SVM, and RF) to learn the optimal way to weigh these outputs for improved decision-making. For new data points, the base algorithms generate predictions, which are then processed by the Logistic Regression meta-classifier to produce the final prediction, effectively combining the strengths of each base algorithm (JagadeeswaraRao and Sivaprasad, 2024)(Ali et al., 2022).

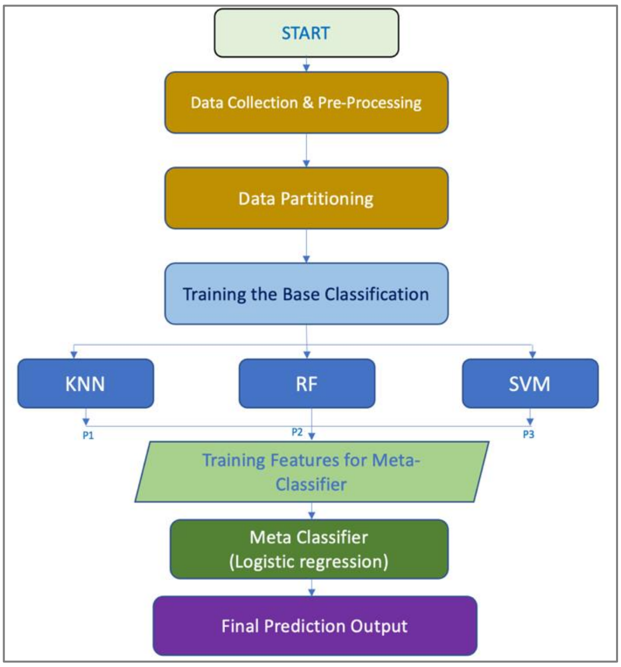

START
Data Collection & Pre-Processing
Data Partitioning
Training the Base Classification
KNN
RF
SVM
P1
P2
P3
Training Features for Meta-Classifier
Meta Classifier
(Logistic regression)
Final Prediction Output

Figure 9. The proposed stacking model

3.3.2.Transfer Learning Approach (Second Approach)

The objective of this technique is to enhance the effectiveness of the DDoS attack detection model by utilizing a pre-trained model on the CICIDS2017 dataset, leveraging fine-tuning and transfer learning techniques. This methodology addresses imbalanced class distribution and improves the model's capacity to provide accurate predictions in various network settings. The process involves network feature extraction from raw traffic data, saved in CSV format, followed by preprocessing and feature selection to refine the dataset for optimal input.(Gurjar and Voditel, 2022) (Islam, 2024).

- **Transfer Learning Model 1:** As shown in Figure 10 this model employs a pre-trained network initially trained on a large dataset, optimizing specific layers to adapt to DDoS attack identification. The process includes data preprocessing, feature extraction, and classifier training to distinguish between DDoS attacks and benign traffic.
- **Transfer Learning Model 2:** Figure 11 provides a comprehensive overview of Transfer Learning Model 2. This model extracts network properties from raw traffic data saved in PCAP files and converts them into a CSV format. The data undergoes preprocessing, including feature conversion and normalization. Features selected and reduced in Model 1 are reused directly. The pre-trained model acts as a feature extractor, with the encoder layers providing essential features passed to the head layers for DDoS attack detection, classifying traffic as DDoS or benign.

Steps of the Approach:

- **Selection of the Base Model:**

An Artificial Neural Network (ANN) model trained on the CICIDS2017 dataset was employed.

- **Training the Base Model:**

The ANN model was trained using the CICIDS2017 dataset with an input layer of 15 nodes, 7 hidden layers with nodes: 128, 64, 32, 16, 8, 4, and 2, and an output layer with 1 node. Activation functions used were 'tanh', 'selu', and 'sigmoid' for the output layer. The Adam optimizer was used, and the model had 13,097 trainable parameters.

- **Transferring Weights and Fine-Tuning the Model:**

  Weights from the pre-trained ANN model on the CICIDS2017 dataset as outlined in Table 5 were transferred to a new model with the same architecture. The new model was trained using the Adam optimizer and binary cross-entropy loss function, with evaluation metrics of accuracy, precision, and recall. Fine-tuning was done using the CICDDoS2019 dataset, employing techniques like Early Stopping and Learning Rate Reduction to optimize training and prevent overfitting. Training stopped if there was no improvement in validation loss for 30 consecutive epochs, restoring the best validation performance weights.

Table 5. Hyperparameters for Early Stopping and Learning Rate Reduction

Hyperparameter	Value	Description
EarlyStopping		
monitor	val_loss	Metric to monitor for early stopping.
patience	30	Number of epochs with no improvement after which training will be stopped.
verbose	1	Verbosity mode; 1 = verbose output when early stopping is triggered.
restore_best_weights	True	Whether to restore model weights from the epoch with the best value of the monitored metric.
ReduceLROnPlateau		
monitor	val_loss	Metric to monitor for learning rate reduction.
patience	30	Number of epochs with no improvement after which learning rate will be reduced.
min_lr	1e-07	Minimum learning rate.
verbose	1	Verbosity mode; 1 = update messages when learning rate is reduced.
factor	0.1	Factor by which the learning rate will be reduced. (new_lr = lr * factor)

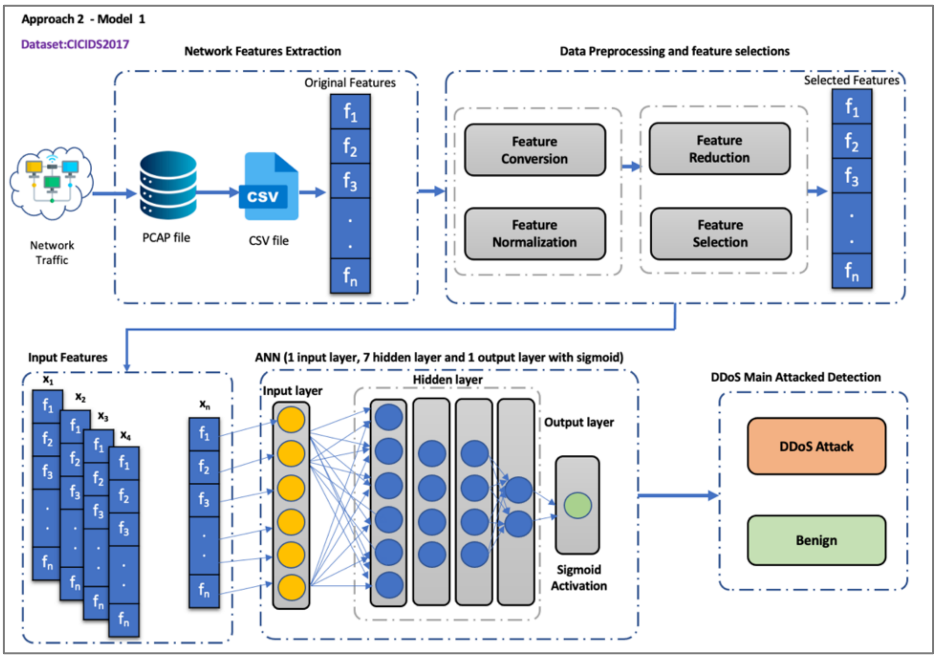


Figure 10. Transfer Learning Model 1

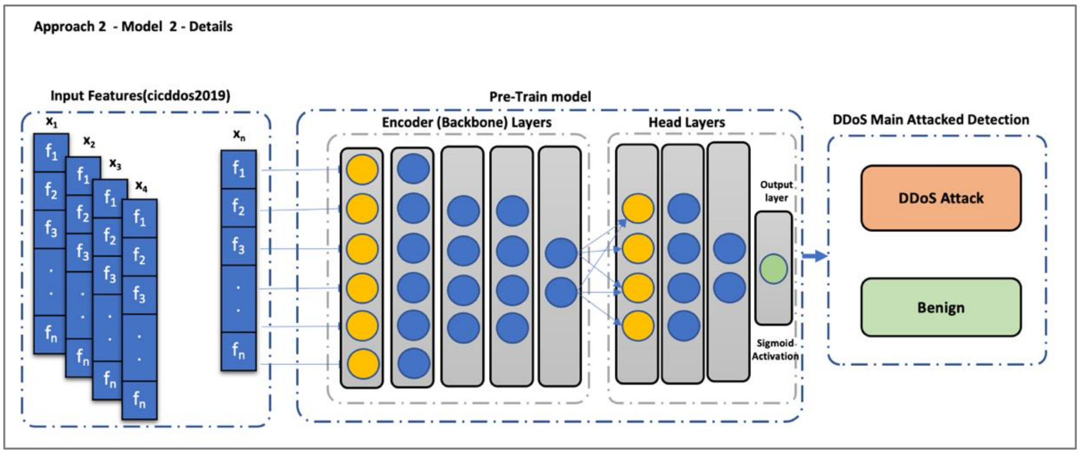

Approach 2 - Model 2 - Details
Input Features(cicddos2019)
Pre-Train model
Encoder (Backbone) Layers
Head Layers
Output layer
Sigmoid Activation
DDoS Main Attacked Detection
DDoS Attack
Benign

Figure 11.Transfer learning model 2

- **Baseline Model:**

To assess the efficacy of the transfer learning method, a baseline model (Model 3) was implemented. Model 3, an ANN, was trained from scratch solely on the CICDDoS2019 dataset without any pre-trained knowledge. This model serves as a comparative benchmark to highlight the advantages of transfer learning as illustrated in .

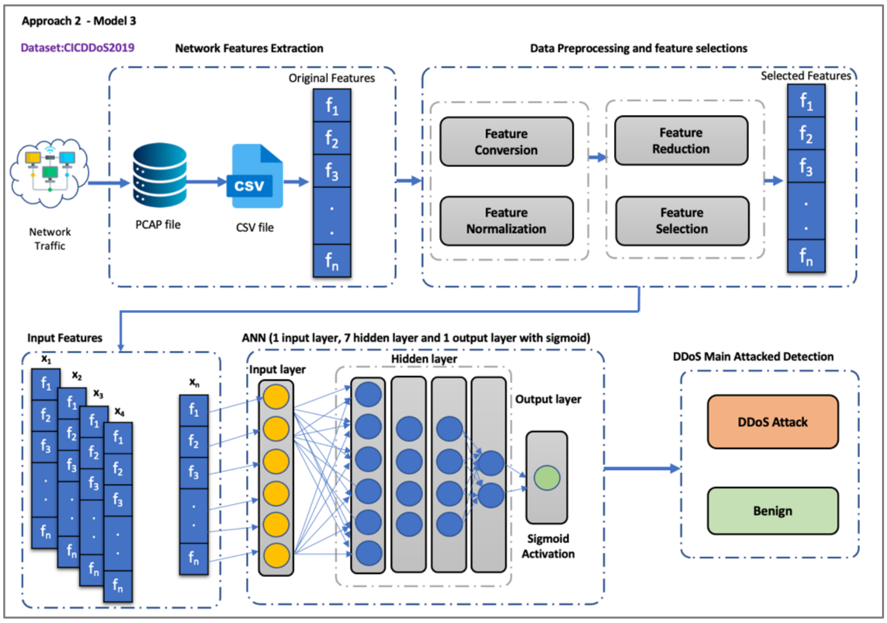


Figure 12 . Architecture of Model 3

- **Statistical Analysis:**

Statistical analysis enhances DDoS detection systems by extracting insights from network traffic data. The Confusion Matrix, a table format used in classification tasks, represents the predicted and true classes, allowing for a visual assessment of the model's performance.

- **Basic Measures**
  - **True Positive (TP):**

    The count of instances that are accurately categorized as belonging to a specific class. True Positives indicate the accurate predictions made by the model for each positive category.

  - **False Positive (FP):**

    False negatives refer to the cases that are mistakenly categorized as belonging to a particular class, whereas in reality they belong to a different class. The calculation involves adding up the values in the column, but excluding the True Positives.

  - **False Negative (FN):**

    Misclassified instances refer to the count of occurrences that are categorized under a specific class, but are actually supposed to be assigned to a different class. The calculation involves summing the values in the same row, but eliminating the True Positives.

  - **True Negative (TN**):

    The amount of occurrences that are accurately categorized as not belonging to a specific class. True Negatives represent the model's correct predictions for the negative classes.

- **Performance Metrics:**
  - **Recall:** Proportion of actual positives correctly identified as shown in equation (1).

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

  - **Precision:** Proportion of positive class predictions that are correct as shown in equation (2).

$$Precision\ = \frac{TP}{TP+FP} \quad (2)$$

- **Accuracy:** Measures overall proficiency in identifying both positive and negative classes. It is calculated using the equation (3):

$$Accuracy\ = \frac{TP+TN}{TP+TN+\ FP+FN} \quad (3)$$

- **F1-Score:** Combines recall and precision, it is calculated using the equation (4):

$$F1 - Score\ = \frac{2*Recall*Precisson}{Recall+Precision} \quad (4)$$

4. RESULTS

This section showcases the outcomes derived from two separate methodologies employed in identifying DDoS assaults. The first approach involves combining multiple algorithms with a meta-classifier, while the second approach leverages transfer learning using a pre-trained model.

4.1. Approach 1: Combining Algorithms with a Meta-Classifier

Multiple tests were conducted on four models, utilizing diverse criteria for example, metrics such as accuracy, precision, and F1-score. Below includes a collection of the models used in the simulation: The algorithms employed in this investigation include KNN, SVM, RF, and the Proposed Stacking Classifier.

4.1.1. Confusion Matrix

A confusion matrix is a systematic table employed to evaluate the efficacy of a classification model. It offers a comprehensive analysis of Evaluate the model's predictions by comparing the expected categories with the observed categories. The matrix is organized in such a way that each row represents cases belonging to a specific actual class, and each column represents instances belonging to a predicted class.

I. K-Nearest Neighbors

The confusion matrix for the KNN model used in DDoS detection is depicted Figure 13. The matrix offers a graphical depiction of the model's effectiveness by contrasting the real classes with the anticipated classes. The number of true positives (TP) is 50,002, the number of true negatives (TN) is 14646, the number of false positives (FP) is 29, and the number of false negatives (FN) is similarly 29.

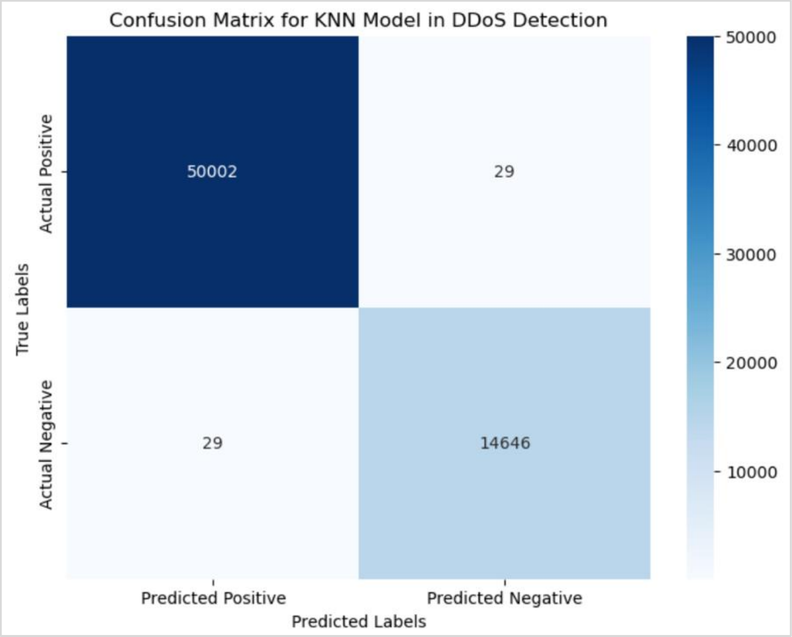


Figure 13. KNN model confusion matrix

II. Support Vector Machine

The confusion matrix of the Support Vector Machine model that is utilized in DDoS detection is displayed Figure 14. The matrix offers a graphical depiction of the model's efficacy by contrasting the observed classes with the anticipated classes. The total number of true positives (TP) is 49,900, the total number of true negatives (TN) is 14,665, the total number of false positives (FP) is 131, and the total number of false negatives (FN) is also 10.

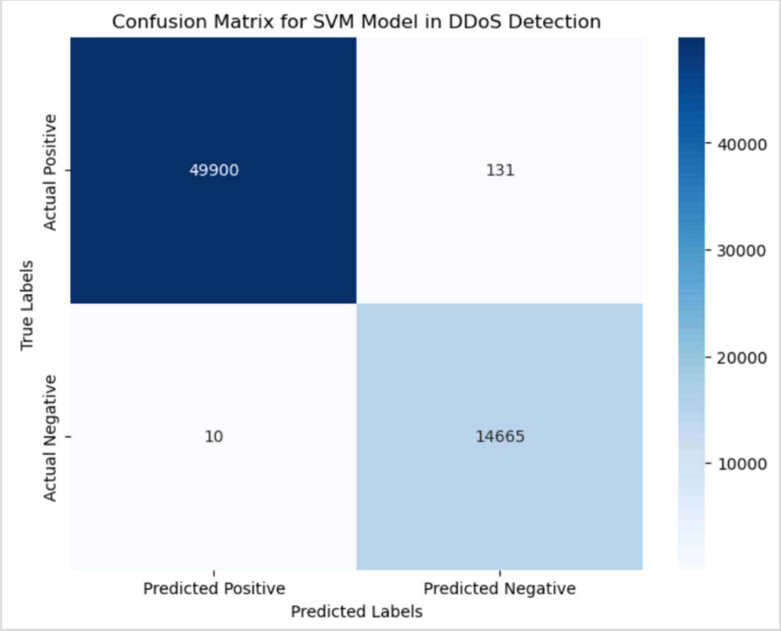


Figure 14. SVM model confusion matrix

III. Random Forest

Figure 15 displays the confusion matrix of the Random Forest algorithm employing recognizing DDoS assualts. The matrix offers a graphical depiction of the model's efficacy by contrasting actual classes with the predicted classes. The number of true positives (TP) is 50,005, the number of true negatives (TN) is 14,665, the number of false positives (FP) is 26, and the number of false negatives (FN) is 11.

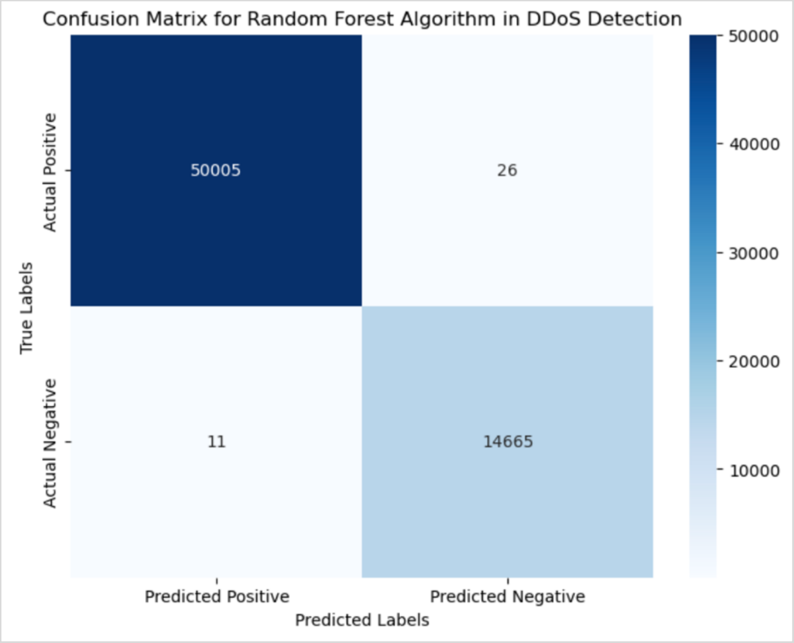


Figure 15. Random Farest confusion matrix

IV. Stacked Model

The confusion matrix associated with the Staked model used in DDoS detection is seen in Figure 16. The matrix provides a visual representation of the effectiveness of the model. By comparing the observed classes with the expected classes. The number of true positives (TP) is 50,012, the number of true negatives (TN) is 14,656, the number of false positives (FP) is 19, and the number of false negatives (FN) is similarly 19.

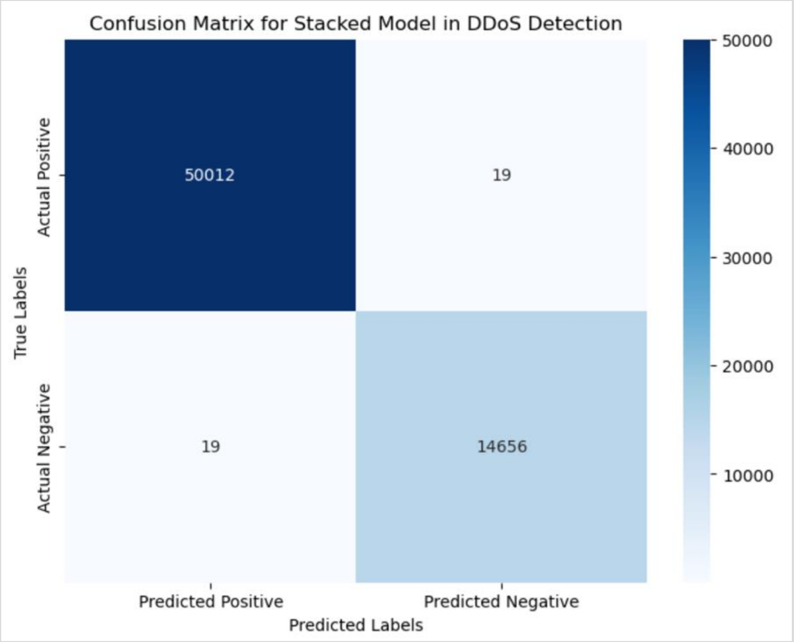


Figure 16. Confusion matrix for the stacked model

V. Performance Metrics of Classifiers

Table 6 presents a comprehensive analysis of the performance measures for four classifiers: SVM, KNN, and RF. The assessment encompasses crucial parameters, including accuracy, precision, and F1-score, which are vital for analyzing the classification capabilities of each algorithm.

Table 6. Comparison of Stacking Classifier

Metric	Model	Training Set	Test Set
Accuracy	KNN	0.999842	0.999104
	SVM	0.998000	0.997821
	Random Forest	0.999774	0.999428
	Stacked Model	0.999716	0.999413
Precision	KNN	0.999904	0.999420
	SVM	0.999944	0.999800
	Random Forest	0.999972	0.999780
	Stacked Model	0.999824	0.999620
F1-score	KNN	0.999842	0.999104
	SVM	0.998003	0.997824
	Random Forest	0.999774	0.999428
	Stacked Model	0.999716	0.999413
Recall-score	KNN	0.999892	0.999420
	SVM	0.997470	0.997382
	Random Forest	0.999736	0.999480
	Stacked Model	0.999413	0.999413

The performance metrics differences of KNN, SVM, RF, and the stacking classifier are shown in the Figure 17 to Figure 20.

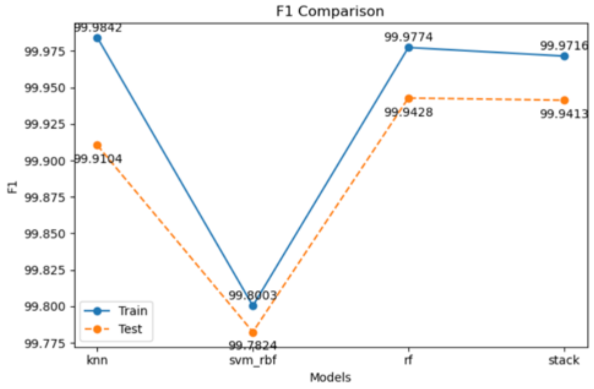


Figure 17. F1 score comparison

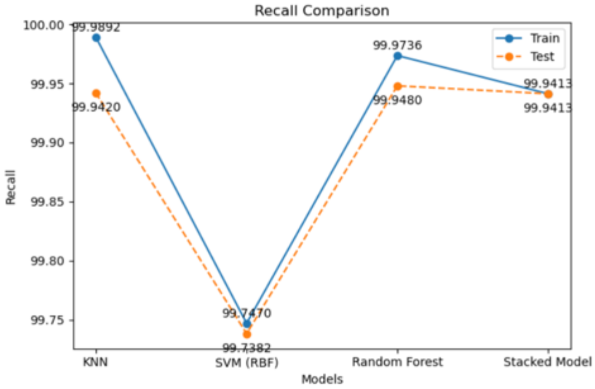


Figure 18. Recal compoarison

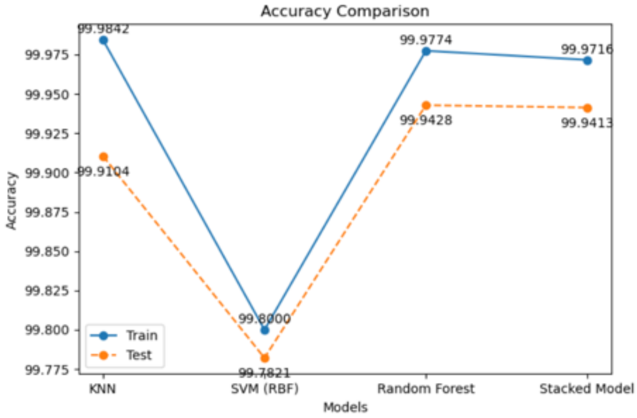


Figure 20. Accuracy comparison

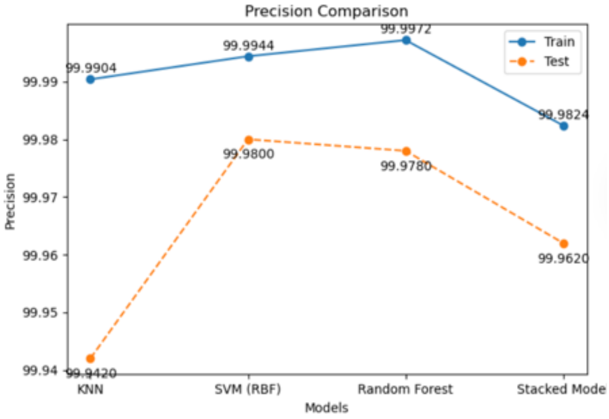


Figure 19. Precison Comparison

While the SVM algorithm achieves 99.78% accuracy, the KNN algorithm obtains 99.91% accuracy. Random Forest did exceptionally well, with an accuracy of 99.94%. In addition, The proposed stacking classifier has an accuracy value of 99.94%. According to this study, integrated classifiers provide superior performance compared to separate classifiers. The proposed stacking model, including a combination of the three previously described classifiers, has superior accuracy compared to the individual classifiers.

4.2. Approach 2: Transfer Learning Approach

In this approach, a pre-trained model developed using the ANN algorithm on the CICIDS2017 dataset was utilized (model 1). The pre-trained weights were applied to a new model (model 2) to analyze the CICDDoS2019 dataset.

Model 3, trained without pre-training, achieved commendable performance with a test accuracy reaching 97.75%, 98.23% for test precision , test recall of 98.88%, and a test F1 score reaching 98.55%. During training, it demonstrated a train accuracy of 97.84%, train precision of 98.23%, train recall of 98.98%, and a train F1 score of 98.61%.

Model 2, benefiting from pre-training, displayed slightly superior performance metrics. It was able to get a score of 99.87% on the test F1 and a score of 99.81% on the test accuracy, 99.97% on the test precision, and 99.78% on the test recall. It demonstrated a train accuracy of 99.81%, a train precision of 99.99%, a train recall of 99.77%, and a train F1 score of 99.88% when it was being trained. Table 7 presents a comparison of the performance metrics between Model 3 (without pre-training) and Model 2 (utilizing pre-training). This comparison highlights the efficacy of pre-training in enhancing the performance of the models in detecting DDoS assaults.

Table 7. Summarized of Transfer learning evaluation metrics

Model	*Phase*	*Loss*	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*Latency(ms)*	*Best Epoch*
Model 2	Train	0.0056	99.81%	99.99%	99.77%	99.88%	0.035	50
	Test	0.0061	99.81%	99.97%	99.78%	99.87%		
Model 3	Train	0.0939	97.84%	98.23%	98.98%	98.61%	0.036	120
	Test	0.0957	97.75%	98.23%	98.88%	98.55%		

The figures that follow depict the loss and accuracy trends for the two models as the epochs progress.Figure 21 and Figure 22 show the training and validation loss and accuracy for Model 3, respectively. Model 3 takes significantly more epochs to stabilize, with the loss value gradually decreasing and accuracy gradually increasing over **120** epochs. In contrast, Figure 23 and Figure 24 depict the training and validation loss and accuracy of Model 2, respectively. Model 2 converges much faster, achieving a lower loss value and higher accuracy within the first 50 epochs.

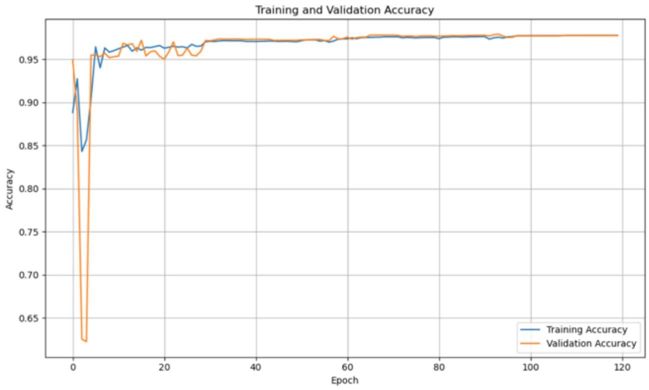


Figure 21. Training and validation accuracy model 3

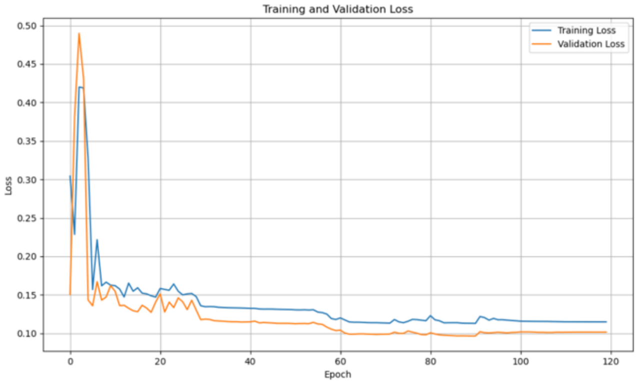


Figure 22. Training and validation loss model 3

This rapid convergence highlights the effectiveness of transfer learning in achieving better performance quickly. Overall, the figures demonstrate that Model 2 reaches high performance levels more efficiently than Model 3.

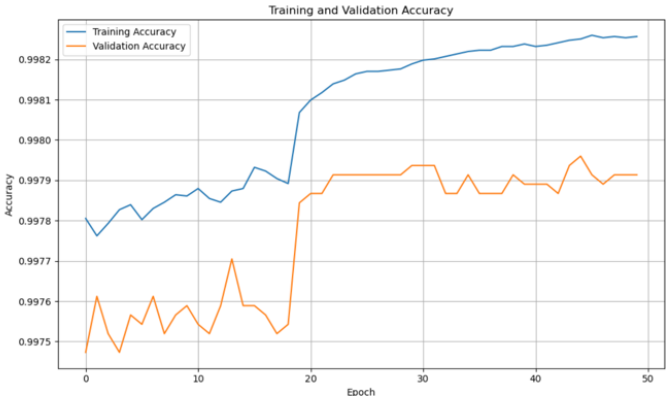


Figure 24. Training and validation accuracy model 2

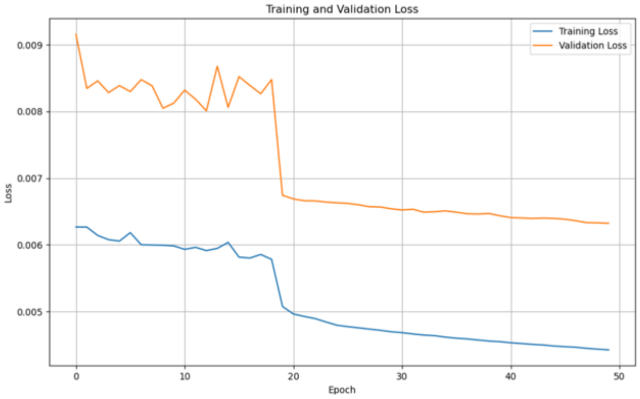


Figure 23. Training and validation accuracy model 2

Additionally, Figure 25 *and* Figure 26 provide a direct comparison between Model 2 and Model 3 regarding the aspects of loss and accuracy, respectively. These figures clearly demonstrate that Model 2 achieves higher performance levels more efficiently than Model 3. Model 2 shows a sharper decline in loss and a quicker rise in accuracy compared to Model 3, further emphasizing the benefits of transfer learning in enhancing model performance and training efficiency.

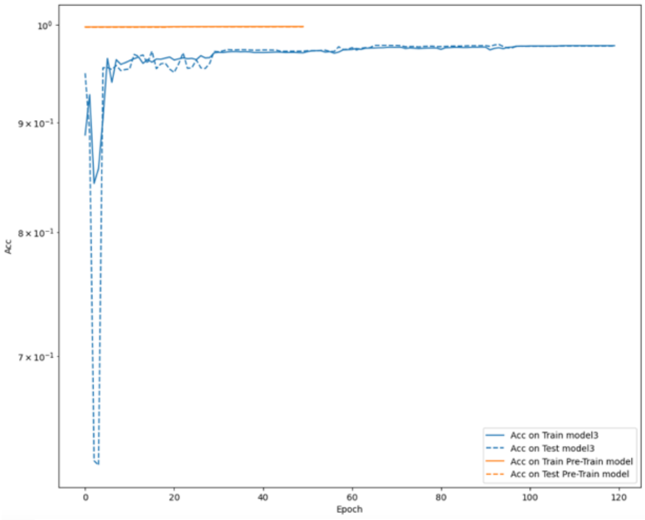


Figure 25. Accuracy comparison of Model 2 and Model 3

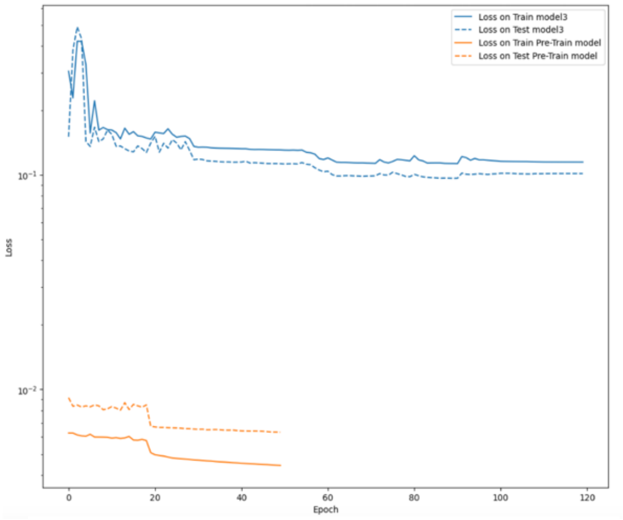


Figure 26. Loss comparison of Model 2 and Model 3

5. DISCUSSION

The analysis and findings of this study provide a substantial contribution to the current research on detecting Machine learning techniques used to carry out DDoS assaults. This section discusses the implications of our results, compares them with previous studies, and Identifies key areas that require more investigation, emphasizing the innovative approach of using a meta-classifier stacking model and advanced feature extraction and selection methods.

5.1. Model Performance

In the study, two distinct approaches to DDoS detection were explored, each yielding remarkable results.

5.1.1.Analysis of Performance Metrics for Approach 1

Firstly, Approach 1 involved the utilization of a meta-classifier stacking model, which combined KNN, SVM, and RF algorithms. This hybrid approach significantly surpassed the performance of individual algorithms, showcasing a notable improvement in detection accuracy. Prior research, such as the experiments undertaken by Seifousadati et al. (2021) and Gopinaath et al. (2022), primarily evaluated The effectiveness of individual algorithms. In contrast, our approach underscored the potential advantages of integrating multiple models to leverage their complementary strengths (Gopinaath et al., 2022; Seifousadati et al., 2021).

The stacking model in this study exhibited outstanding performance metrics, with accuracies consistently approaching 99.94% and above. This outcome aligns with the high accuracy rates reported for individual algorithms in existing literature, thereby affirming the efficacy of our ensemble method. Furthermore, this hybrid approach offers an enhanced and reliable detecting technique, highlighting the considerable potential of ensemble methods in enhancing cybersecurity measures.

- **Accuracy**

Table 8 compares the training and test accuracy of different models. It shows that Random Forest and the Stacked Model have the highest accuracy and generalize well to new data.

Table 8. Accuracy analysis

Model	Analysis
KNN	KNN shows very high accuracy on both training and test sets, indicating good generalization.
SVM	SVM also demonstrates high accuracy but slightly lower than KNN and Random Forest.
Random Forest	Random Forest performs excellently on both sets, with very close results between training and test sets, indicating robust generalization.
Stacked Model	The Stacked Model performs similarly to Random Forest, with excellent generalization from training to test sets.

- **Precision**

Table 9 displays the precision of different models on both the training and test sets. The results indicate SVM and Random Forest exhibit the best accuracy, suggesting their efficacy in reducing false positives.

Table 9. Precision analysis

Model	Analysis
KNN	KNN maintains high precision, ensuring that the majority of predicted positive cases are true positives.
SVM	SVM shows the highest precision among the models, indicating very few false positives.
Random Forest	Random Forest shows very high precision, slightly lower than SVM on the test set.
Stacked Model	The Stacked Model has high precision but is slightly lower than SVM and Random Forest.

- **F1-score**

Table 10 illustrates the F1-scores for each model on the training and test sets. The Random Forest and Stacked Model demonstrate the highest F1-scores, reflecting their balanced performance between precision and recall.

Table 10. F1-Score analysis

Model	Analysis
KNN	KNN shows a balanced performance between precision and recall, maintaining a high F1-score.
SVM	SVM has a lower F1-score compared to KNN and Random Forest, reflecting its slightly lower recall.
Random Forest	Random Forest achieves a high F1-score, reflecting its strong balance between precision and recall.
Stacked Model	The Stacked Model also maintains a high F1-score, indicating balanced performance.

- **Recall**

Table 11 summarizes the recall metrics for the models. It shows that the Random Forest model has the highest recall, ensuring most positive cases are correctly identified, closely followed by the Stacked Model.

Table 11. Recall analysis

Model	Analysis
KNN	KNN shows high recall, ensuring most positive cases are correctly identified.
SVM	SVM has the lowest recall among the models, indicating it misses more positive cases compared to others.
Random Forest	Random Forest demonstrates strong recall, slightly better than KNN.
Stacked Model	The Stacked Model maintains balanced recall performance across both sets.

- **Conclusion**

  - **Random Forest** generally shows the best overall performance with high scores across all metrics and minimal overfitting.
  - **KNN** and the Stacked Model also perform excellently, with slight differences in precision and recall.
  - **SVM** shows slightly lower performance in terms of recall, which affects its F1-score, but it has the highest precision.
  - **The Stacked Model** leverages the strengths of individual models, resulting in balanced and robust performance metrics.

Overall, the analysis indicates that while all models perform well, the Random Forest and Stacked Model provide the best balance between precision, recall, and overall accuracy.

5.1.2.Approach 2: Transfer Learning Approach Analysis

Secondly, Approach 2 employed transfer learning by leveraging a pre-trained model on the CICIDS2017 dataset, utilizing ANN algorithm, and applying it to the CICDDoS2019 dataset. This innovative approach also yielded superior performance metrics, achieving an accuracy rate of 99.78%.

- **Training Method**
  - Model 3 was trained without transfer learning, starting from scratch and training for 120 epochs.

- Model 2 utilized transfer learning, starting with pre-trained weights and training for only 50 epochs. This indicates that transfer learning helped Model 2 converge faster and achieve better performance in fewer epochs.

- **Performance**
  - Model 2 outperformed Model 3 in both training and testing phases across all performance metrics (loss, accuracy, precision, recall, and F1 score).
  - The superior performance of Model 2, as indicated by its lower loss values and greater accuracy, precision, recall, and F1 score, suggests that it is more proficient in both accurately representing the training data and making accurate predictions on unseen test data.
- **Latency**
  - Model 2 has a slightly lower latency (0.035 ms) compared to Model 3 (0.036ms), which indicates that Model 2 is marginally faster in terms of inference time.
- **Efficiency**
  - Model 2 achieved high performance metrics in significantly fewer epochs (50 vs. 120), demonstrating the efficiency of transfer learning. This efficiency not only decreases the amount of time required for training, but also minimizes the usage of computing resources.
- **Conclusion**

  Model 2, utilizing transfer learning, exhibits superior accuracy, recall, precision, and F1 score in comparing with Model 3, which underwent training from scratch. The use of pre-trained weights allowed Model 2 to converge faster and achieve better generalization with fewer epochs. Additionally, Model 2 shows a slight improvement in inference latency, making it both an effective and efficient choice for the task. The figures further validate these findings by showing faster convergence and higher performance for Model 2.

5.1.3. Comparison of Performance Metrics for Both Approaches

Table 12 compares the performance metrics of two approaches: Approach 1 (Stacked Model) and Approach 2 (Transfer Learning with a Pre-trained ANN).

Table 12. Both Approach Performance Metrics Comparison

Metric	Stacked Model	Transfer Learning	Analysis
Training Accuracy	99.9716%	99.81%	Both models have high training accuracy, but the Stacked Model is slightly higher.
Test Accuracy	99.9413%	99.81%	Test accuracy is slightly lower for the Stacked Model compared to its training accuracy.
Training Precision	99.9824%	99.99%	Both models have very high precision, with Transfer Learning being slightly higher.
Test Precision	99.9620%	99.97%	Precision remains very high for both models on the test set.
Training Recall	99.9413%	99.77%	The Stacked Model has higher training recall compared to Transfer Learning.
Test Recall	99.9413%	99.78%	Recall is slightly lower for Transfer Learning compared to its training recall.

Table 12 Countinue

Metric	Stacked Model	Transfer Learning	Analysis
Training F1-score	99.9716%	99.88%	Both models show high F1-scores, with the Stacked Model being slightly higher.
Test F1-score	99.9413%	99.87%	F1-score remains very high for both models on the test set.
Difference in Accuracy (Train vs Test)	0.03%	0%	Transfer Learning shows no difference, indicating better generalization and less overfitting.
Difference in Precision (Train vs Test)	0.0204%	0.02%	Both models have minimal differences, but Transfer Learning's is sli continue
Difference in Recall (Train vs Test)	0%	0.01%	Transfer Learning shows a minimal difference, indicating good generalization.
Difference in F1-score (Train vs Test)	0.0303%	0.01%	Transfer Learning has a smaller difference, indicating better handling of overfitting.
Training Time	4450s	125s	Transfer Learning significantly reduces training time compared to the Stacked Model.
Inference Latency	0.3 ms	0.036 ms	Transfer Learning offers much lower inference latency, making it more suitable for real-time applications.

- **Analysis**

  - **Stacked Model**
    - The difference between training and test accuracy is very small (0.03%).
    - The precision, recall, and F1-scores for training and test sets are also very close.
    - This indicates that the Stacked Model generalizes well and does not show significant signs of overfitting.

  - **Transfer Learning**
    - The difference between training and test accuracy is negligible (0%).
    - Precision, recall, and F1-scores are very similar between training and test sets.
    - This also indicates that the Transfer Learning model generalizes very well without overfitting.

  - **Conclusion**

  Both models show minimal signs of overfitting based on the provided metrics. However, if we need to choose the one that handles overfitting slightly better, we can consider the following:

    - **Transfer Learning**: The training and test metrics are almost identical, indicating that the model has learned to **generalize** very well to unseen data. Additionally, the **lower training time** and **inference latency** make it a more efficient choice.
    - **Stacked Model:** Although it shows excellent performance and minimal overfitting, the slight difference in training and test metrics compared to Transfer Learning indicates a marginally higher risk of overfitting.

5.2. Feature Extraction and Selection

Unnecessary features were removed to reduce dataset dimensionality. Correlation analysis and XGBoost were used to refine the feature set from 79 to 15, enhancing model performance and efficiency.

5.3. Dataset Utilization

The CICDDoS2019 dataset, providing diverse DDoS attack types, improved model robustness and generalizability, crucial for real-world applications.

5.4. Optimization Strategies

Hyperparameter optimization and class weighting techniques, such as Random Search and SMOTE, improved model accuracy, precision, recall, and F1-score.

5.5. Generalization Capability

The combined approach showed better generalization on test data, indicating lower susceptibility to overfitting compared to individual models.

5.6. Comparison with Literature

Compared to the other studies shown in Table 13, which all use the CICDDoS2019 dataset, our research uniquely employs ensemble methods and transfer learning. This approach has proven effective in developing precise DDoS detection systems and enhancing cybersecurity.

Table 13. Compare with other studies

Title	Authors, Year	Algorithms	Features	Accuracy
A Machine Learning Approach for DDoS Detection on IoT Devices	Seifousadati, Ghasemshirazi, Fathian, 2021	Naïve Bayes, SVM, AdaBoost, XGBoost, KNN, Random Forest	Top 10 important features	**100%**
Detection and Characterization of DDoS Attacks Using Time-Based Features	Halladay, Cullen, 2022	LightGBM, XGBoost, Adaptive Boosting, DNN	Time-based features(25 features)	**98-99%**
DDoSNet: A Deep-Learning Model for Detecting Network Attacks	Elsayed, Le-Khac, Dev, Jurcut, 2022	RNN, Deep Learning techniques	77 features	**99%**
DDoS Detection using Machine Learning Techniques	Gopinaath, Amrish, Kumar, Bavapriyan, 2022	KNN, Decision Tree, Random Forest, ANN	Top 15 features by ExtraTrees	**99.95%**
Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices	Kumar Ranjeesh, Gaur Vimal ,2022	KNN, Decision Tree, Random Forest, ANN	Top 15 features from	**99.95%**
Enhanced Ensemble-Based DDoS Attack Detection	Md. Alamgir Hossain, 2023	Ensemble Method (e.g., Random Forest)	--	**100%**
This study		**Stacking Model**	**Correlation and XGBoost (Top 15 features)**	**99.94%**
		Transfer-learning		**99.81%**

5.7. Complexity and Implementation:

The ensemble approach requires intensive computation due to multiple model training and combination. Transfer learning, though needing initial pre-training, is easier to implement and fine-tune.

5.8. Scalability:

Transfer learning is more scalable, quickly adapting pre-trained models to new datasets without extensive retraining.

5.9. Implications for Cybersecurity:

This research enhances cybersecurity by demonstrating effective DDoS attack detection, particularly valuable for protecting vulnerable IoT devices.

5.10. Future Directions:

- **Exploration of Additional Feature Groups:** Investigate more feature sets from datasets to improve DDoS detection performance.
- **Optimal Flow Intervals:** Experiment with different flow intervals to refine detection mechanisms.
- **Extended Hyperparameter Optimization:** Apply advanced optimization techniques across more classifiers and neural networks.
- **Comparative Analysis of Class Balancing Techniques:** Compare oversampling, undersampling, and ensemble methods to evaluate their impact on model performance and efficiency.
- **Real-Time DDoS Detection Systems:** Develop and test real-time systems integrating optimized models for live network environments.
- **Cross-Dataset Generalization:** Test model robustness on various datasets and real-world traffic to ensure adaptability to different attack patterns and network conditions.

6. CONCLUSION

Distributed Denial-of-Service (DDoS) assaults pose a significant threat to cybersecurity, necessitating the development of robust detection techniques. This thesis explored the application of machine learning techniques to enhance DDoS detection using the CICDDoS2019 and CICIDS2017 datasets. Two distinct approaches were examined: combining multiple algorithms using a meta-classifier and utilizing a pre-trained model through transfer learning. These approaches provide valuable insights for developing efficient and effective DDoS detection systems.

The first approach involved training K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF) on the CICDDoS2019 dataset, and then combining them using a logistic regression meta-classifier. This ensemble method capitalized on the strengths of each algorithm, resulting in a stacked model with a high accuracy of 99.94%. Despite the superior performance, this approach was computationally intensive, requiring significant training time and resources.

The second approach utilized transfer learning, where a pre-trained Artificial Neural Network (ANN) model on the CICIDS2017 dataset was fine-tuned using the CICDDoS2019 dataset. This method achieved an accuracy of 99.78% with a significantly reduced training time of 2.75 minutes, compared to the 3 hours required for the stacked model. The transfer learning approach also demonstrated lower inference latency, making it a more efficient solution for scenarios requiring rapid model deployment.

In comparing these methods, the meta-classifier approach offers the highest detection accuracy, making it ideal for scenarios where computational resources are not a limiting factor. In contrast, the transfer learning approach, while slightly lower in performance metrics, provides a more practical balance between effectiveness and efficiency, particularly suitable for real-time applications.

The findings of this study have significant implications for cybersecurity, demonstrating the efficacy of combining multiple algorithms and leveraging transfer learning techniques. These methods offer a foundation for developing advanced DDoS detection systems that are resilient to evolving attack patterns and adaptable to new data. The scalability and efficiency of the transfer learning approach, in particular, present a promising avenue for future research.

Future research could explore the integration of the two approaches examined in this thesis, potentially yielding further improvements in detection performance and robustness. Additionally, investigating the application of these methods across different datasets and exploring the use of unsupervised learning techniques and anomaly detection methods could further enhance the ability to detect novel and sophisticated attack vectors.

In conclusion, this study demonstrates the potential of advanced machine learning techniques in enhancing the detection of Distributed Denial-of-Service (DDoS) attacks. The combination of multiple algorithms through ensemble methods and the utilization of pre-trained models via transfer learning offer effective strategies for improving detection accuracy and efficiency. These findings contribute to ongoing efforts to develop robust, adaptive, and scalable cybersecurity solutions capable of safeguarding against the persistent threat of DDoS attacks.

REFERENCES

Bhushan, B., Chaganti, R., & Ravi, V. (2022). A survey on Blockchain solutions in DDoS attacks mitigation: Techniques, open challenges and future directions. *Computer Communications*, *197*. https://doi.org/10.1016/j.comcom.2022.10.026

Chong, Y.-W., Ali, T. E., & Manickam, S. (2023). Comparison of ML/DL Approaches for Detecting DDoS Attacks in SDN. *Applied Sciences*, *13*. https://doi.org/10.3390/app13053033

Elsayed, M. S., Le-Khac, N.-A., Dev, S., & Jurcut, A. D. (2020). *DDoSNet: A Deep-Learning Model for Detecting Network Attacks* (arXiv:2006.13981). arXiv. http://arxiv.org/abs/2006.13981

Gaur, V., & Kumar, R. (2022). Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. *Arabian Journal for Science and Engineering*, *47*(2), 1353–1374. https://doi.org/10.1007/s13369-021-05947-3

Gopinaath, V., Amrish, R., Kumar, C. V., Jawahar, A., & Bavapriyan, K. (2022). DDoS Detection using Machine Learning Techniques. *Journal of ISMAC*, *4*. https://doi.org/10.36548/jismac.2022.1.003

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. A. (2023). Enhanced Ensemble-Based Distributed Denial-of-Service (DDoS) Attack Detection with Novel Feature Selection: A Robust Cybersecurity Approach. *Artificial Intelligence Evolution*. https://doi.org/10.37256/aie.4220233337

Kasture, P. (2023). DDoS Attack Detection using ML. *International Journal for Research in Applied Science and Engineering Technology*, *11*. https://doi.org/10.22214/ijraset.2023.53133

Khan, M. A. (2021). HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. *Processes*, *9*(5), Article 5. https://doi.org/10.3390/pr9050834

Li, S., & Wang, D. (2022, December 2). *Automated DDoS Attack Mitigation for Software Defined Network*. https://doi.org/10.1109/asid56930.2022.9996013

Ojha, S. P., Qureshi, Z., Kumar, S. P., & Sadhu, A. (2023, April 19). *Detection and Prevention of Distributed Denial of Service in Mobile ADHOC Network*. https://doi.org/10.1109/raeeucci57140.2023.10134098

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

Shakya, H. K., & Karnani, S. (2022). Mitigation strategies for distributed denial of service (DDoS) in SDN: A survey and taxonomy. *Information Security Journal: A Global Perspective*, *32*. https://doi.org/10.1080/19393555.2022.2111004

Singh, D. N. P., Kumar, D. N., & Kumar, S. (2022). Literature Review of Distributed Denial of Service (DDoS) Attacks, its Detection Techniques and Prevention Mechanisms. *International Journal for Research in Applied Science and Engineering Technology*, *10*. https://doi.org/10.22214/ijraset.2022.46882

Suhag, A., & Daniel, A. (2022). Study of statistical techniques and artificial intelligence methods in distributed denial of service (DDOS) assault and defense. *Journal of Cyber Security Technology*, *7*. https://doi.org/10.1080/23742917.2022.2135856

T, R., E, A., U, D., Sumathi, A. C., Yuvaraj, N., & Ghazali, N. H. (2023). Improved Intrusion Detection System That Uses Machine Learning Techniques to Proactively Defend DDoS Attack. *ITM Web of Conferences*, *56*. https://doi.org/10.1051/itmconf/20235605011

Umamaheswari, K., Subramanian, N., & Subramaniyan, M. (2023). Distributed Denial of Service Attack Detection Using Hyper Calls Analysis in Cloud. *International Journal of Computer Network and Information Security*, *15*. https://doi.org/10.5815/ijcnis.2023.04.06

Wang, C., Zheng, J., & Li, X. (2017). Research on DDoS Attacks Detection Based on RDF-SVM. *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 161–165. https://doi.org/10.1109/ICICTA.2017.43

Jia, Y., Zhong, F., Alrawais, A., Gong, B., & Cheng, X. (2020). FlowGuard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet of Things Journal, 7(10), 9552–9562.

Cheema, A., Khan, M.M., Anwar, M., Tariq, M., Ahmad, F., & Hafiz, A. (2022). Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review. Security and Communication Networks, 2022. https://doi.org/10.1155/2022/8379532

Rajendran, N.A., & Vincent, D.R. (2021). Heart Disease Prediction System using Ensemble of Machine Learning Algorithms. Recent Patents on Engineering, 15. https://doi.org/10.2174/1872212113666190328220514

Sultana, N., & Islam, M.M. (2019). Meta Classifier-Based Ensemble Learning For Sentiment Classification. https://doi.org/10.1007/978-981-13-7564-4_7

Islam, M.M. (2024). The Impact of Transfer Learning on AI Performance Across Domains. Journal of Artificial Intelligence General Science (JAIGS), 1. https://doi.org/10.60087/jaigs.v1i1.37

Sharafaldin, I., Lashkari, A.H., Hakak, S., & Ghorbani, A.A. (2019). CICDDoS2019 Dataset. Canadian Institute for Cybersecurity, University of New Brunswick. Available at: http://www.unb.ca/cic/datasets/CICDDoS2019

Canadian Institute for Cybersecurity. (2017). CICIDS2017 Dataset. University of New Brunswick. Available at: https://www.unb.ca/cic/datasets/ids-2017.html

Gurjar, A., Voditel, P., 2022. Transfer Learning: A Paradigm for Machine Assisted Knowledge Transfer. ECS Transactions 107. https://doi.org/10.1149/10701.7179ecst

ACKNOWLEDGEMENT and DECLARATIONS

The authors contributed equally to the study. There are no potential conflicts of interest in this study.
Research and publication ethics were complied with in the study.

Note: This article was prepared using the preliminary studies of the master's thesis titled "Enhanced DDOS Attack Detection Through Hybrid Machine Learning Techniques", which will be carried out by Feraidoon Farahmandnia, under the supervision of Prof. Dr. Serhat Özekes, at Üsküdar University Institute of Science, Cybersecurity Thesis Master's Program.