



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC
RESEARCH
ECHAHID CHEIKH LARBI TEBESSI UNIVERSITY -
TEBESSA



Faculty of Exact Sciences and Natural Sciences
Department of Computer Science
Specialty: Networks and Computer Security

Graduation Project for obtaining Master Degree

A Deep Learning Approach for Detecting Distributed Denial of Service Attacks

Created by:
Bourougaa Khadidja

Role	Name	Grade	Institution
Supervisor	Souahi Mohamed Salah	MCB	Echahid cheikh larbi tebessi university
President	Tag Samir	MAA	Echahid cheikh larbi tebessi university
Examiner	Mzyani Ahlam	MCB	Echahid cheikh larbi tebessi university

Scholar Year: 2025 / 2026

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful

Praise be to Allah, who granted success and assistance, illuminated the path when darkness gathered, and completed this work by His grace alone, not by my own power or strength.

To those whose words cannot fully honor, nor ink ever give their due right, no matter how many pages stretch... To my dear parents, who taught me that a dream is only earned through long hardship and that a prayer in the depths of the night achieves what no other means can. You have always been the steadfast root that gives me the courage to reach toward the light. To my supervising professor, Souahi Mohamed Salah, who did not merely teach me but believed in me when I hesitated to believe in myself. And to my esteemed professors, who overwhelmed me with their vast knowledge and sound guidance and who planted in this soul what it will never forget.

To my brothers and sisters, companions of home and keepers of secrets, who know the details of this journey as no one else does.

To my loyal friends, whose names are mentioned only to make the heart smile before the lips.

And to every hand that extended help to me, and every kind word that touched my soul in a moment of weakness... You are all a part of this success, whether you feel it or not.

And finally... to that spirit that was patient when the horizon narrowed, persisted when exhaustion betrayed it, and believed when certainty abandoned it to Myself, this is not merely a graduation; this is a promise fulfilled.

Abstract

Distributed Denial of Service (DDoS) attacks remain one of the most significant threats to network security. By overwhelming a target with massive amounts of malicious traffic, these attacks disrupt essential online services and cause substantial financial losses. Traditional detection methods, such as signature-based and anomaly-based systems, struggle to detect new and evolving attack patterns. Moreover, most existing deep learning approaches focus on binary classification (normal vs attack), which limits their ability to distinguish between different types of DDoS attacks.

This study proposes a deep learning approach for multi-class DDoS attack detection using a one-dimensional Convolutional Neural Network (1D CNN). The proposed model is trained and evaluated on the CICDDoS2019 dataset, which contains various types of DDoS attacks. The dataset undergoes preprocessing steps including label mapping, removal of constant and identifying columns, duplicate removal, handling of missing values, normalization, and type conversion. The model is evaluated using standard metrics: accuracy, precision, recall, and F1-score.

The experimental results demonstrate that the 1D CNN model effectively distinguishes between different types of DDoS attacks. The main contributions of this work include a detailed analysis of the CICDDoS2019 dataset, a 1D CNN model for multi-class classification, and a comparison with existing approaches.

Keywords: DDoS attacks, deep learning, convolutional neural network, CICDDoS2019, multi-class classification.

Résumé

Les attaques par déni de service distribué (DDoS) constituent l'une des menaces les plus importantes pour la sécurité des réseaux. En submergeant une cible avec d'énormes quantités de trafic malveillant, ces attaques perturbent les services en ligne essentiels et causent des pertes financières considérables. Les méthodes de détection traditionnelles, telles que les systèmes basés sur les signatures et les anomalies, peinent à détecter les nouvelles attaques et les schémas évolutifs. De plus, la plupart des approches d'apprentissage profond existantes se concentrent sur la classification binaire (normal vs attaque), ce qui limite leur capacité à distinguer les différents types d'attaques DDoS.

Cette thèse propose une approche d'apprentissage profond pour la détection multi-classes des attaques DDoS utilisant un réseau de neurones convolutif unidimensionnel (1D CNN). Le modèle proposé est entraîné et évalué sur l'ensemble de données CICDDoS2019, qui contient différents types d'attaques DDoS. L'ensemble de données subit des étapes de prétraitement, y compris le mappage des étiquettes, la suppression des colonnes constantes et identifiantes, la suppression des doublons, le traitement des valeurs manquantes, la normalisation et la conversion de type. Le modèle est évalué à l'aide de métriques standard : la précision, le rappel et le F1-score.

Les résultats expérimentaux démontrent que le modèle 1D CNN distingue efficacement les différents types d'attaques DDoS. Les principales contributions de ce travail incluent une analyse détaillée de l'ensemble de données CICDDoS2019, un modèle 1D CNN pour la classification multi-classes, et une comparaison avec les approches existantes.

Mots-clés: Attaques DDoS, apprentissage profond, réseau de neurones convolutif, CICDDoS2019, classification multi-classes.

الملخص

تعتبر هجمات حجب الخدمة الموزعة (DDoS) واحدة من أهم التهديدات لأمن الشبكات. من خلال إغراق الهدف بكميات هائلة من حركة المرور الضارة، تعطل هذه الهجمات الخدمات الأساسية عبر الإنترنت وتسبب خسائر مالية كبيرة. تواجه طرق الكشف التقليدية، مثل الأنظمة القائمة على التوقيعات والسلوكيات الشاذة، صعوبة في اكتشاف الأنماط الجديدة والمتطورة للهجمات. علاوة على ذلك، تركز معظم أساليب التعلم العميق الموجودة على التصنيف الثنائي (طبيعي مقابل هجوم)، مما يحد من قدرتها على التمييز بين أنواع مختلفة من هجمات DDoS.

تقترح هذه الأطروحة نهجاً قائماً على التعلم العميق للكشف متعدد الفئات لهجمات DDoS باستخدام شبكة عصبية التفاضلية أحادية البعد (1D). (CNN) يتم تدريب النموذج المقترح وتقييمه على مجموعة بيانات CIC-DDoS2019 التي تحتوي على أنواع مختلفة من هجمات DDoS. تخضع مجموعة البيانات لخطوات معالجة مسبقة تشمل تعيين التصنيفات، وإزالة الأعمدة الثابتة والمعرفة، وإزالة التكرارات، ومعالجة القيم المفقودة، والتطبيع، وتحويل الأنواع. يتم تقييم النموذج باستخدام مقاييس قياسية: الدقة، والاستدعاء، ودرجة F1.

تظهر النتائج التجريبية أن نموذج CNN 1D يميز بشكل فعال بين الأنواع المختلفة لهجمات DDoS. تشمل المساهمات الرئيسية لهذا العمل تحليلاً مفصلاً لمجموعة بيانات CIC-DDoS2019 ونموذج CNN 1D للتصنيف متعدد الفئات، ومقارنة مع الأساليب الموجودة.

الكلمات المفتاحية:

هجمات DDoS، التعلم العميق، الشبكة العصبية التفاضلية، التصنيف متعدد الفئات، أمن الشبكات.

List of Abbreviations

AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
AWS	Amazon Web Services
BCCC	Behaviour-Centric Cybersecurity Center
BiLSTM	Bidirectional Long Short-Term Memory
C&C	Command and Control
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DDoS	Distributed Denial of Service
DL	Deep Learning
DNS	Domain Name System
DoS	Denial of Service
DCAE	Deep Contractive Autoencoder
EDAD	Enhanced Distributed DDoS Attack Detection
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FTP	File Transfer Protocol
GB	Gradient Boost
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
KNN	K-Nearest Neighbors
LDAP	Lightweight Directory Access Protocol
LSTM	Long Short-Term Memory

Table of Contents

Acknowledgment	i
Abstract	ii
Résumé	iii
Résumé (AR)	iv
List of Abbreviations	v
List of Figures	viii
List of Tables	ix
Introduction	x
Chapter 1 Background and Previous works	8
Introduction	9
1 Denial of Service Attacks (DoS)	9
1.1 Distributed Denial of Service Attacks (DDoS)	10
1.2 Taxonomy of DDoS Attacks	12
1.3 Limitations of Traditional DDoS Detection Methods	14
2 Artificial Intelligence in DDoS detection	14
2.1 Machine Learning	15
2.2 Deep Learning	16
2.3 Deep Learning Architectures	17
3 Previous work	20
3.1 Machine Learning for DDoS Detection	20
3.2 Deep Learning for DDoS Detection	20
3.3 Discussion and Research Gap	22
Conclusion	24
Chapter 2 Contribution and Future Works	25
Introduction	26
1 Datasets in DDoS Detection	26
1.1 Dataset Generation in DDoS Detection	26
1.2 Commonly Used Datasets in DDoS Research	27
2 CICDDoS2019 Dataset	28
2.1 Dataset Structure and Features	29
2.2 Attack Types in CICDDoS2019	30

2.3	Data Preprocessing	31
3	Contribution	33
3.1	Model Architecture	33
3.2	Development Environment	35
3.3	Evaluation Metrics	36
4	Experimental Results and Discussion	38
5	Future Perspectives	40
5.1	Dataset Conversion and Compatibility	40
5.2	Results, Limitations and Future Work	41
	Conclusion	43
	General Conclusion	44
	Bibliography	44

List of Figures

1.1	DoS attack diagram	9
1.2	Distributed Denial of Service (DDoS) Attack Architecture	11
1.3	Taxonomy of Distributed Denial of Service Attacks (DDoS)	12
1.4	Venn Diagram for AI, ML and DL	15
1.5	Convolutional Neural Network (CNN) Architecture	17
1.6	Long Short-Term Memory (LSTM) Cell Architecture	18
1.7	Basic structural layout of a fully connected Multi-Layer Perceptron (MLP). . .	19
1.8	Symmetrical data flow of a basic Autoencoder	19
2.1	Three-stage pipeline for DDoS dataset generation	27
2.2	Testbed architecture of CICDDoS2019 dataset. [1]	29
2.3	Classification of DDoS attacks into Reflection-based and Exploitation-based [1]	30
2.4	CNN model used for 13-class classification	33
2.5	Confusion matrix	38
2.6	Confusion matrix for 13-class classification	39

List of Tables

1.1	Comparison between DoS and DDoS attacks	10
1.2	Comparative Analysis of Related Work on DDoS Attack Detection	22
2.1	Commonly used datasets in DDoS research	29
2.2	Number of flow records per attack type in CICDDoS2019 dataset	31
2.3	Classification report for 13 classes	39
2.4	Statistics of the reprocessed CIC-DDoS2019 dataset	41

Introduction

The internet has become essential in our daily lives. We use it for banking, education, health-care, communication, and entertainment. However, this heavy reliance on online services has also attracted malicious actors. Among the many cyber threats, Distributed Denial of Service (DDoS) attacks are one of the most dangerous. In a DDoS attack, an attacker uses a network of compromised devices to send an overwhelming amount of traffic to a target server, making it unavailable to legitimate users. These attacks cause financial losses, damage reputations, and disrupt critical services.

Problem Statement

Traditional methods for detecting DDoS attacks, such as signature-based and anomaly-based systems, have significant limitations. Signature-based systems rely on predefined attack patterns and cannot detect new or unknown attacks. Anomaly-based systems monitor traffic for unusual behavior but often produce high false positive rates, flagging normal traffic as malicious. Both approaches struggle to keep up with the increasing volume and complexity of modern DDoS attacks.

In recent years, machine learning and deep learning have been proposed to improve detection. However, most existing work focuses on binary classification (attack vs. normal), which only tells us if there is an attack but not what type of attack it is. This limits the ability to respond appropriately to different DDoS attack categories.

Proposed Solution

To address these limitations, this thesis proposes a deep learning approach using a one-dimensional Convolutional Neural Network (1D CNN) for multi-class DDoS attack detection. The proposed model is trained and tested on the CIC-DDoS2019 dataset, which contains different types of DDoS attacks. The model is designed to not only detect whether an attack is happening but also to identify which specific type of DDoS attack it is (e.g., SYN flood, UDP flood, DNS amplification, etc.). This multi-class classification capability provides more useful information for network administrators to take appropriate countermeasures.

Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 1** provides background on DDoS attacks, artificial intelligence, machine learning, and deep learning. It also reviews related work and identifies research gaps.
- **Chapter 2** presents the methodology, including dataset description, preprocessing steps, model architecture, training process, evaluation metrics, experimental results, and discussion. Future Perspectives which discusses the limitations of the current work and suggests directions for future research.
- Finally, the **Conclusion** summarizes the main findings of the thesis.

Chapter 1

Background and Previous works

Introduction

Distributed Denial of Service (DDoS) attacks remain one of the most persistent threats to network security. Traditional detection methods struggle to keep up with new attack patterns. This chapter provides an explanation of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, including how they work, how they are classified, and the limitations of traditional detection methods. Then moves to artificial intelligence, introducing machine learning and deep learning concepts as well as the key neural network architectures commonly used in DDoS detection, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Multi-Layer Perceptrons (MLPs), and Autoencoders (AEs). Finally, the chapter reviews recent research in both machine learning and deep learning for DDoS detection, identifying gaps in the literature that motivate the work presented in this thesis.

1 Denial of Service Attacks (DoS)

A Denial of Service (DoS) attack is designed to prevent a computer system or network from delivering its services to its users. This attack is considered a true DoS attack if the accessibility of the computer services or the computer network is compromised in some way by the deliberate actions of another user. These attacks pose a serious threat to resource availability, rendering systems inaccessible to authorized users, even though they might not directly or permanently harm data.

The attacker sends a large number of packets to the target network or service, flooding the network/service with data, which makes it unable to respond to the legitimate users. The main objective of a DoS attack is to stop the target from functioning by restricting its access. [2]

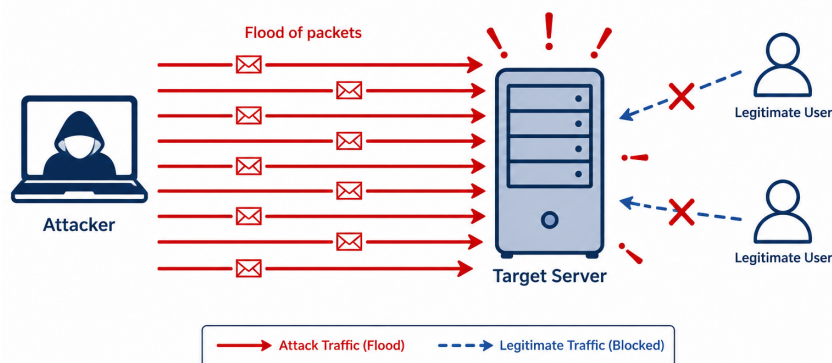


Figure 1.1: DoS attack diagram

1.1 Distributed Denial of Service Attacks (DDoS)

A Distributed Denial of Service (DDoS) attack is a cyberattack that aims to overwhelm a target and make its services unavailable to legitimate users by sending a large amount of traffic. The attacker remotely controls a group of compromised devices, known as a botnet, to generate this traffic. [3]

What is the difference between DoS and DDoS attacks?

Although both DoS and DDoS attacks aim to make a service unavailable to legitimate users, they differ in how they are initiated and in the number of sources that are involved. Table 1.1 summarizes the main differences between DoS and DDoS attacks.

Aspect	DoS (Denial of Service)	DDoS (Distributed Denial of Service)
Attack Source	Single machine or IP address	Multiple compromised devices forming a botnet
Traffic Volume	Lower traffic generated from one origin	Much higher traffic distributed across many sources
Traceability	Easier to trace and block the attacker	Harder to trace because traffic comes from many devices
Complexity	Simpler and often launched from a single system	More complex and coordinated using botnets
Speed and Impact	Slower buildup and usually less damaging	Faster attack with greater impact on the target

Table 1.1: Comparison between DoS and DDoS attacks

What is the Architecture of a DDoS Attack?

A distributed denial of service attack consists of four main components. The components include the attacker, Command and Control (C&C) servers, botnets (zombie machines), and the victim system. [4]

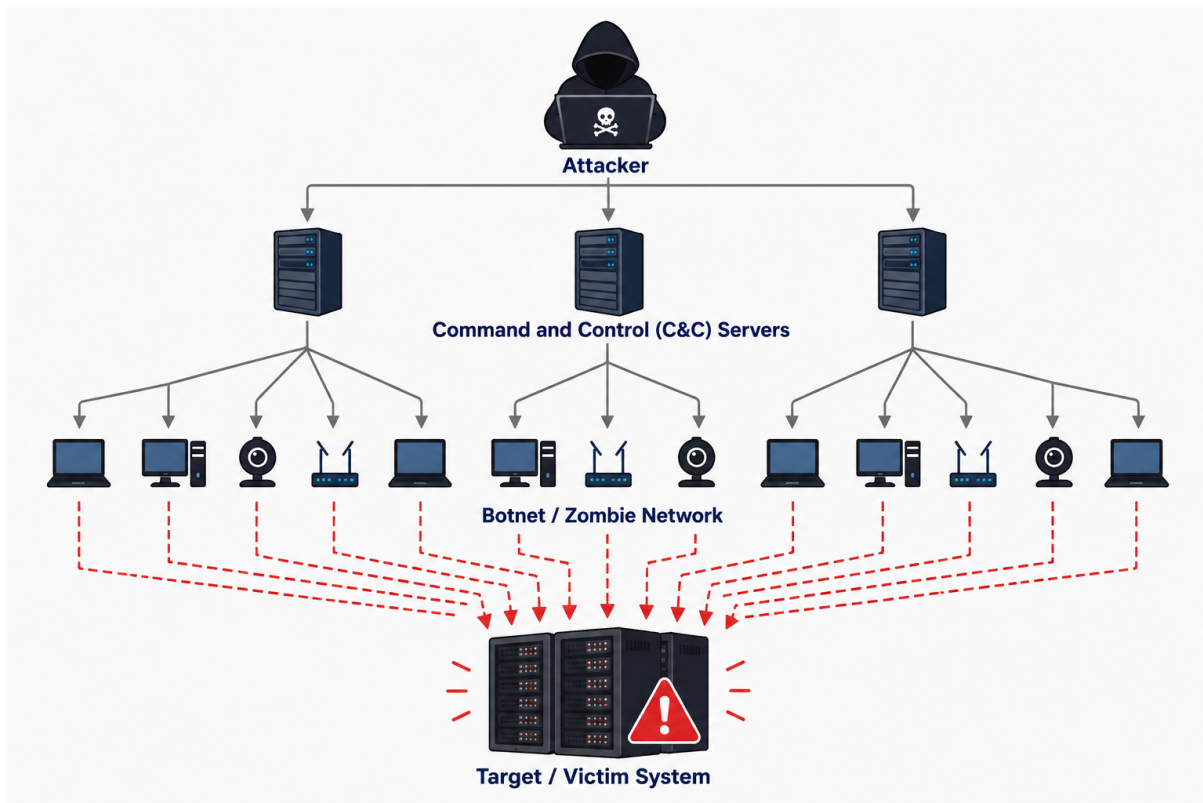


Figure 1.2: Distributed Denial of Service (DDoS) Attack Architecture

- **Attacker:** The person or group is responsible for initiating and coordinating the entire attack operation, as well as building and managing the network of compromised devices
- **Command and Control (C&C) Servers:** These are compromised machines that serve as intermediate command centers, receiving instructions from the attacker and relaying them to groups of infected devices.
- **Botnet:** Also known as a zombie, a botnet is a network of compromised devices, such as computers, laptops, or IoT devices, that send malicious traffic towards the victim.
- **Target/Victim:** The targeted system, server, or network receives malicious traffic, causing service disruption or complete unavailability.

These components work together in a hierarchical structure where compromised machines operate under a command chain. The attacker issues instructions to Command and Control (C&C) servers, which in turn control numerous zombie agents. These servers initiate dormant processes and instruct botnets to launch the assault upon receiving an attack order from the attacker. Then, until legitimate users are unable to access the service, these botnets bombard the target with enormous amounts of packets, overwhelming its processing capacity and using up all system resources. [5]

1.2 Taxonomy of DDoS Attacks

DDoS attacks can be categorized based on what they target and their method of execution. Understanding these categories as shown in 1.3 is essential for implementing effective defense mechanisms, as each type requires different mitigation strategies. [6]

The three main categories of DDoS attacks are *volume-based attacks*, *protocol attacks*, and *application layer attacks*:

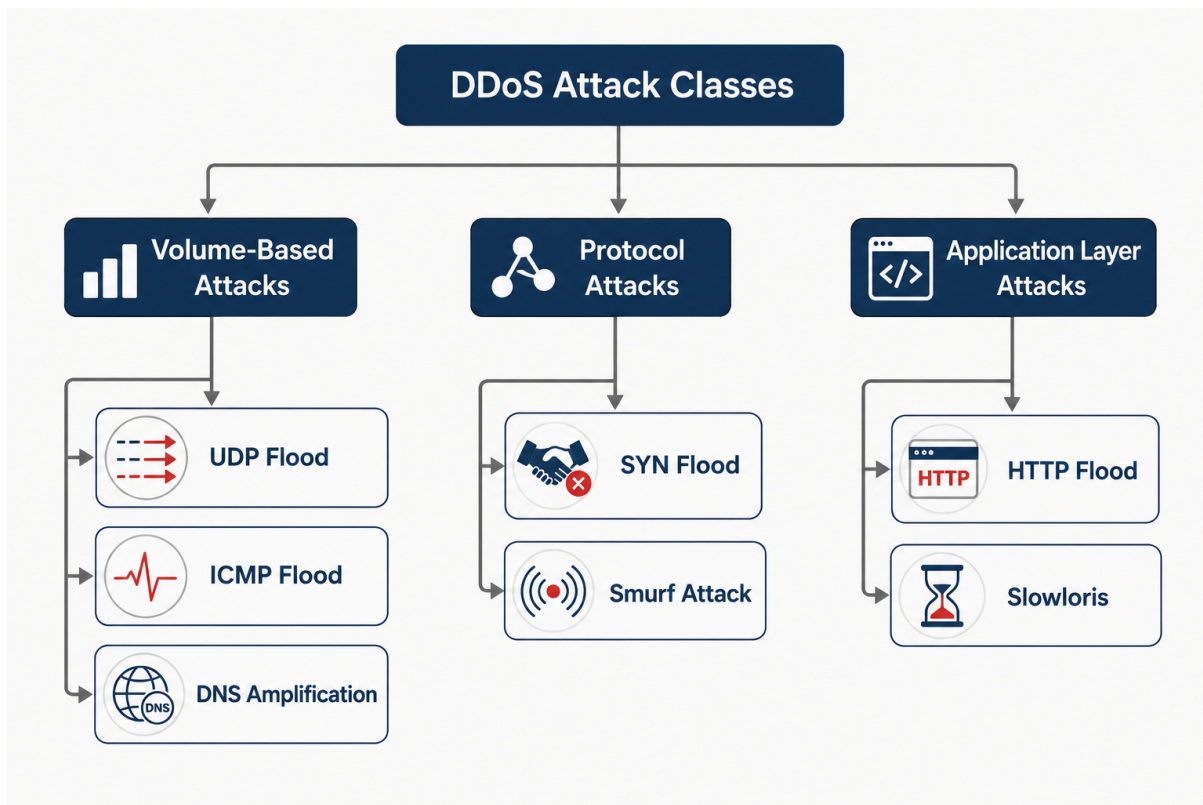


Figure 1.3: Taxonomy of Distributed Denial of Service Attacks (DDoS)

Volume-Based Attacks

Volume-based attacks are the most common type of DDoS attack, designed to saturate the target's bandwidth and make services unavailable. These attacks flood the victim with massive amounts of data, measured in bits per second (bps), targeting the network layer (Layer 3) and the transport layer (Layer 4) of the OSI model. [6]

- **UDP flood:** Attackers spoof source IP addresses and send a huge number of UDP packets to random ports of the targeted server. When the targeted server fails to identify the applications running on these ports, it responds with "destination unreachable" messages. If the number of packets is too large, the system becomes overloaded and cannot handle legitimate requests. [2]

- **ICMP flood:** Referred to as a "ping flood," it targets the overload of ICMP echo request packets by malicious actors. Although ping is typically used to confirm device connectivity, attackers use it as a weapon by sending so many requests that the target runs out of resources, attempting to reply to each one, which eventually disables the device. [2] [6]
- **DNS amplification:** The attacker uses several source IP addresses to send spoofed DNS request packets that look authentic. Because the victim's IP address is used as the source in these requests, DNS servers send the victim large response packets. The deluge of responses overwhelms the target, using up its bandwidth and depleting its resources. [7]

Protocol Attacks

Protocol attacks exploit vulnerabilities in protocols to consume server resources or disrupt essential services. These attacks send high volumes of packets per second, targeting specific weaknesses in network and transport layer protocols. They focus on reducing the capacity of connection tables, load balancers, and server processing capabilities. [6]

- **SYN flood:** The attacker sends multiple SYN packets with spoofed IP addresses. Each time a connection is partially opened, the target server allocates resources and waits for confirmation, but this never occurs. Eventually, the server's connection backlog becomes exhausted, making it impossible to establish new connections. [2][8]
- **Smurf attack:** The attacker spoofs the victim's IP address and sends ping requests to a network's broadcast address. As a result, all of the devices on that network respond to the victim at the same time, overwhelming it with traffic. [6]

Application Layer Attacks

Application layer attacks, also known as Layer 7 attacks, target the top layer of the OSI model in order to exhaust resources for specific applications. Measured in requests per second, these attacks are difficult to detect because they mimic legitimate user requests and traffic patterns. [6] [8]

- **HTTP Flood:** The attackers send high volumes of HTTP requests that appear completely normal. In order to respond to each request, the web server must allot resources like memory and processing power. Once the server receives an excessive number of requests, it overloads and fails to respond to legitimate users. [6] [8]
- **Slowloris:** Attackers attempt to maintain multiple connections open for as long as possible by sending incomplete HTTP requests on a regular basis. The server cannot accept new connections from authorized users once it has reached its connection limit. Because this attack uses little bandwidth, conventional detection methods are unable to identify it. [8]

1.3 Limitations of Traditional DDoS Detection Methods

Traditional DDoS detection techniques, which do not rely on artificial intelligence or machine learning, face a number of issues that need to be addressed as they continue to develop.

- **Signature-based techniques:** Are unable to identify anomalous attacks. Because they rely on predefined patterns, they are not effective against new attacks that lack signatures or zero-day attacks. [9]
- **Scalability issue:** A lot of the detection models currently used have scalability problems, especially in large-scale networks or high-traffic environments. They cannot handle the increasing volume and complexity of network traffic without compromising detection accuracy or performance. [10]
- **High False Positive Rates and False Negative Rates:** The trade-off between false positives and false negatives is another problem that needs to be addressed. High false-positive rates cause unnecessary interruptions of legitimate traffic, while high false-negative rates allow attackers to go undetected. [10]
- **Resource Restrictions:** A significant number of detection models, particularly those employing conventional methods, require significant memory and processing capabilities. This is a drawback for the use of such models in resource-constrained systems. [10]
- **Inability to Adapt to Evolving Threats:** DDoS attack techniques are constantly evolving, growing more complex and diverse. Traditional detection systems are not flexible enough to deal with novel and unexpected attack vectors. [10]

2 Artificial Intelligence in DDoS detection

Artificial intelligence is the creation of technological systems that replicate human mental processes such as understanding, logic, and reasoning to allow machines to function, learn, and make choices independently without direct human programming for every task [11]. The relationship between Artificial Intelligence, Machine Learning, and Deep Learning is illustrated in Figure 1.4.

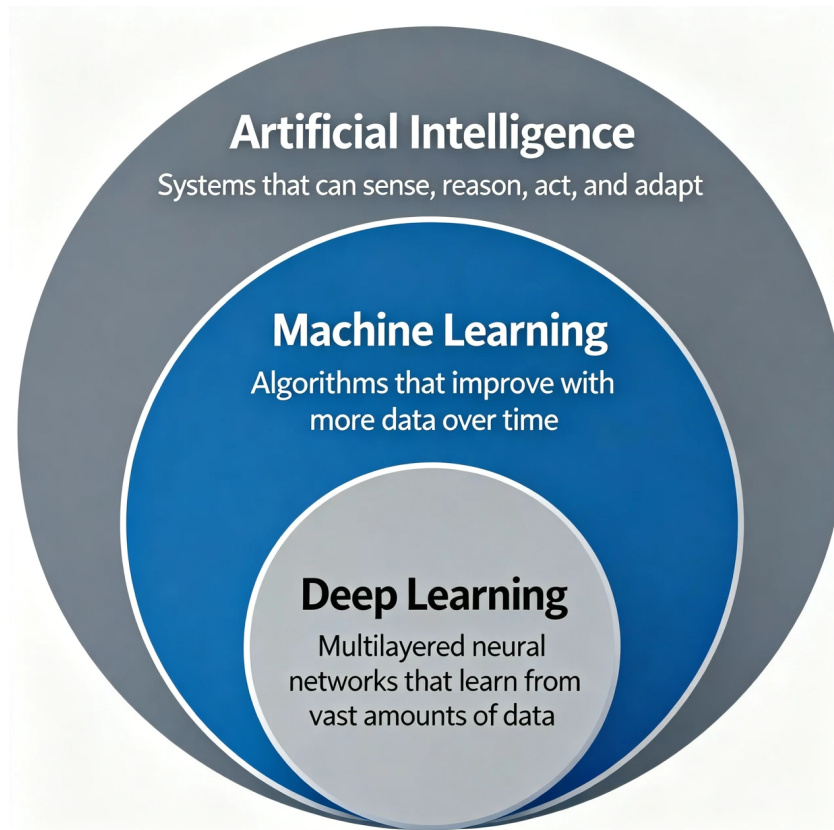


Figure 1.4: Venn Diagram for AI, ML and DL

2.1 Machine Learning

Machine learning can be described as the ability of computer systems to improve their performance over time based on experience. Rather than being manually programmed with fixed rules, these algorithms identify significant relationships and patterns within examples and observations. [12]

Machine learning uses an iterative optimization process (e.g. gradient descent) to fit a model to data to find associations by minimizing a loss or cost function. The goal is to learn a concept that generalizes to unseen data by finding a trade-off between underfitting and overfitting, generally by splitting datasets into training, validation and test subsets

Types of Machine Learning

Machine learning methods are typically categorized based on how the learning process is controlled by data. These categories describe whether the model learns from labeled examples or unlabeled data. The main types of machine learning are presented below.

- **Unsupervised Learning:** This field is concerned with systems that discover the hidden patterns or fundamentals in data without the need for pre-existing labels or categories. It

is frequently used for problems like clustering (subgroup identification), frequent pattern mining and dimensionality reduction

- **Supervised Learning:** In this category, the computer is trained on a data set where the correct answers (labels) are already known. The system learns a "model" or a set of rules to map inputs to outputs in order to predict results for new, unseen data. This is further divided into classification, which is used when the output of the forecast is a label or category. and regression when the output to be predicted is a specific number reduction.
- **Semi-supervised Learning:** This type of category learns on a small amount of labeled data and a large amount of unlabeled data. It guides the learning process using the labeled data and exploits the unlabeled data to obtain a better understanding of the underlying structure of the data distribution. This is especially useful when collecting labeled data is expensive/time-consuming but unlabeled data is plentiful.

2.2 Deep Learning

Deep learning uses multi-layered computational structures to automatically extract insights from data. It uses sequential layers of non-linear processing to provide a thorough understanding of information. This hierarchical approach allows machines to learn from mistakes and to master hard tasks by processing huge amounts of raw input without being specifically programmed for every situation. [13]

Components of Deep Learning Models

Deep learning models are based on four pillars that allow them to process data, recognize patterns, and improve their accuracy over time [13]

- **Structural Layers:** They are the skeleton of the model. There is an input layer that receives the raw data, hidden layers that extract complex features, and an output layer that produces the final result.
- **Activation Functions:** These are mathematical functions that impart non-linearity to the network. Without them a model could only perform simple linear transformations, but functions like ReLU, Sigmoid, and Softmax allow the system to learn complex non-linear relationships in the data.
- **Loss Functions:** Also known as objective functions or cost functions, these measure the mathematical distance between the models predictions and the actual target values. They are a measure of performance.
- **Optimization Algorithms:** These are the mechanisms that actually train the model by iteratively tuning internal parameters to minimize the error reported by the loss function.

Modern techniques like Adam optimizer or Stochastic Gradient Descent (SGD) ensure the model converges towards an accurate solution efficiently.

2.3 Deep Learning Architectures

Deep learning architectures are neural networks with multiple layers that automatically learn hierarchical representations from raw data. Several architectures have been developed for different types of tasks, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Multi-Layer Perceptrons (MLPs), and Autoencoders (AEs).

- **Convolutional Neural Networks (CNN)** is a model designed for visual data, like images, to identify and classify objects. These layered systems pass a raw input through a series of digital filters that automatically detect patterns and features. The network eventually builds a complex knowledge base of the data, enabling tasks such as facial recognition, medical diagnosis, and object tracking. [14]

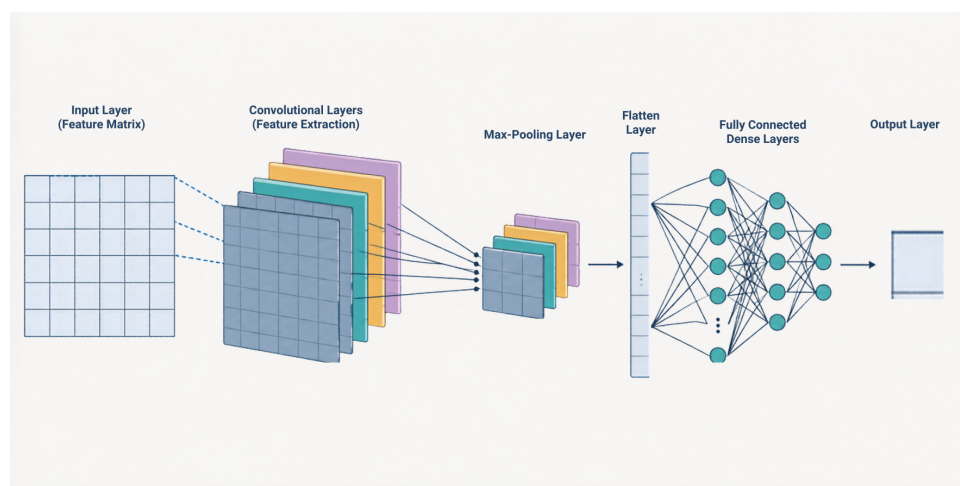


Figure 1.5: Convolutional Neural Network (CNN) Architecture

- **Long Short-Term Memory (LSTM)** is a refined version of recurrent neural networks built to overcome the inability to learn from long sequences. These networks can selectively decide which information is important to keep and which should be discarded, making them highly reliable for understanding context in long-form text, speech, or complex time-based patterns. [15]

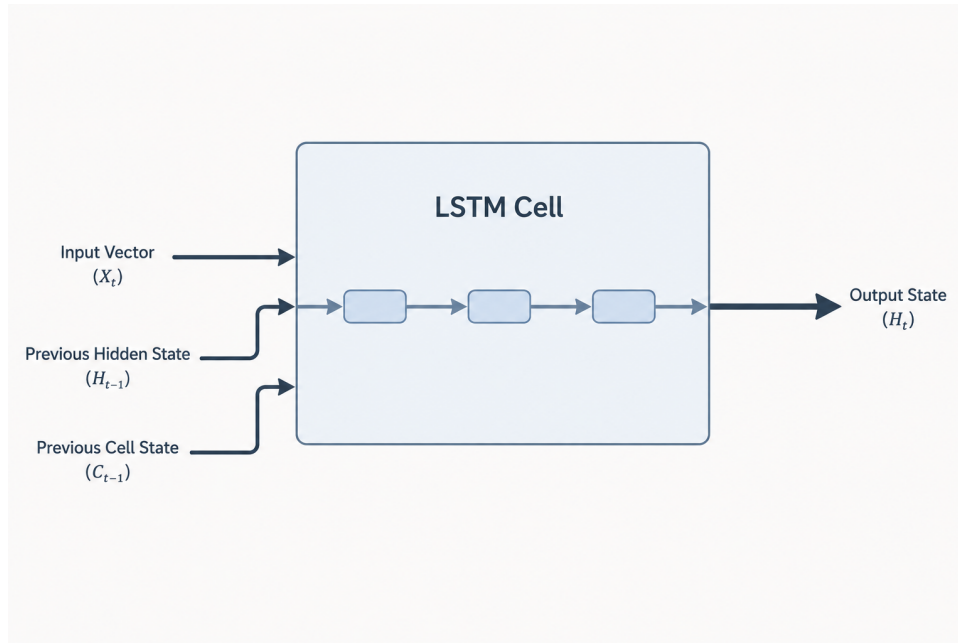


Figure 1.6: Long Short-Term Memory (LSTM) Cell Architecture

- **Multi-Layer Perceptron (MLP)** is a straightforward technical architecture with two layers of dense connections, using a mathematical step to apply a non-linear change to data. It does typical multiplication operations on data tables and transforms individual values to perform information processing between different locations or channels. [16]
- **Autoencoder (AE)** is a type of neural network designed to copy its incoming data. It does this by translating the input into a compressed internal code and then decoding it to produce a version of the data that is nearly the same as the original. It has three layers: input, hidden, and output layers. The system can automatically identify and extract complex features from the data without the need for human-labeled samples. [17]

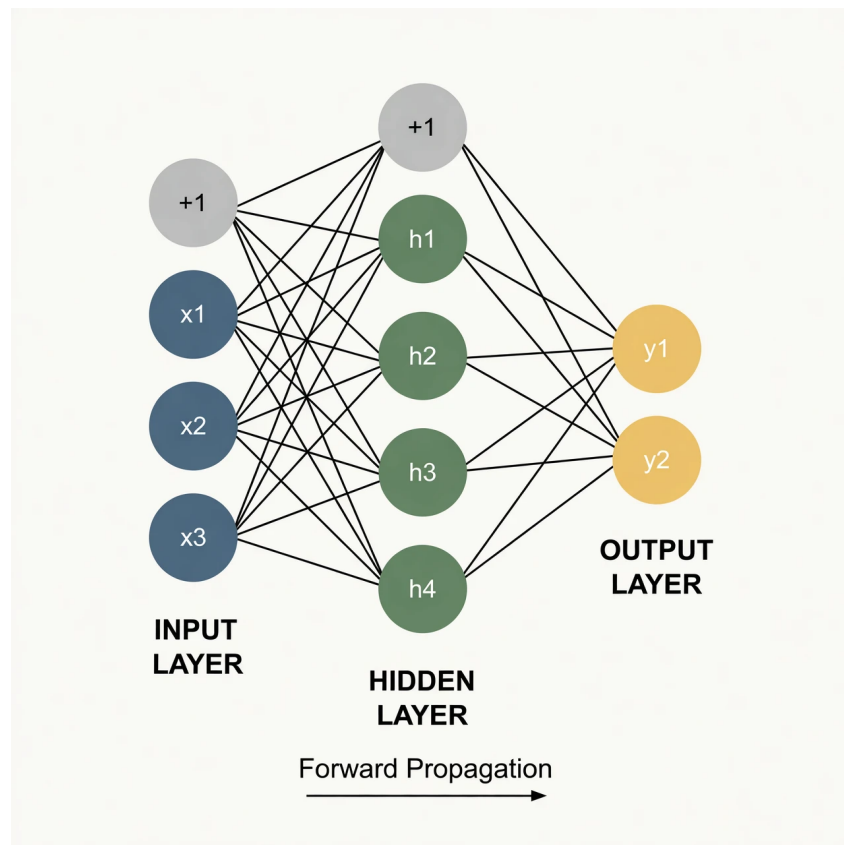


Figure 1.7: Basic structural layout of a fully connected Multi-Layer Perceptron (MLP).

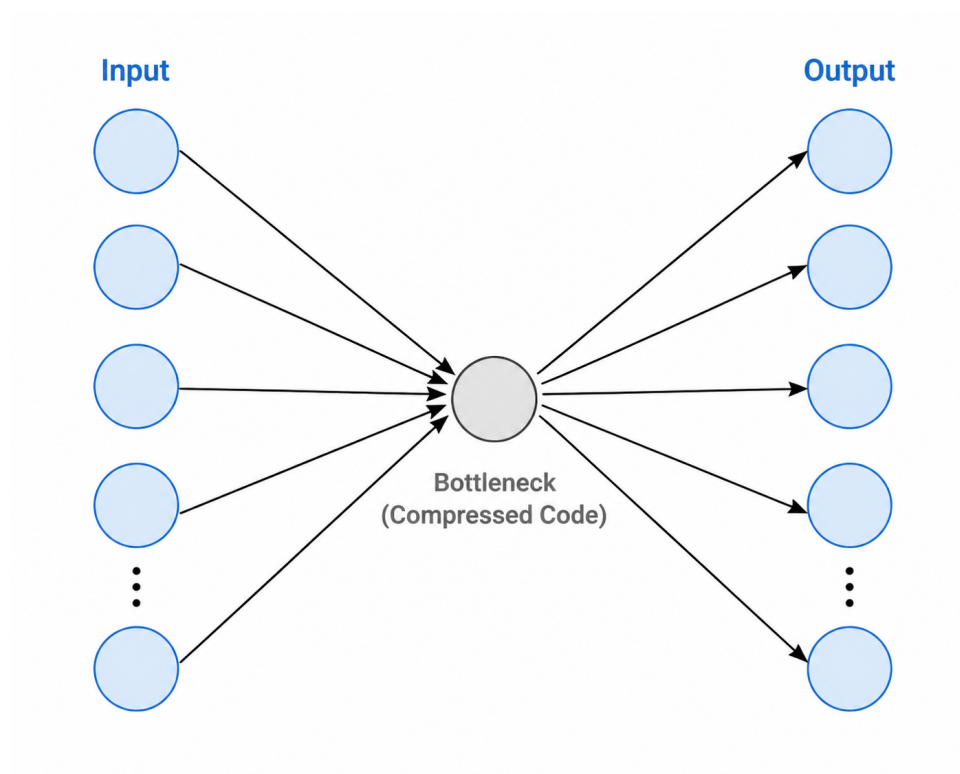


Figure 1.8: Symmetrical data flow of a basic Autoencoder

3 Previous work

3.1 Machine Learning for DDoS Detection

Supervised machine learning (ML) frameworks have proved to be very effective in classifying malicious traffic from legitimate network activity using different classification algorithms and feature selection techniques.

Abiramasundari and Ramaswamy [18] proposed an Enhanced Distributed DDoS Attack Detection (EDAD) framework, which uses *Principal Component Analysis (PCA)* to reduce the dimension of network traffic and refines the dataset from 79 to 61 essential features. In their evaluation of the *CICIDS2017*, *CICIDS2018*, and *CICDDoS2019* datasets, *Random Forest (RF)* achieved an accuracy of **98.9%** on the 2017 data, while *Support Vector Machine (SVM)* and *K-Nearest Neighbors (KNN)* achieved the best accuracies of **98.7%** and **98.6%** on the 2018 dataset, respectively.

Similarly, Dasari and Devarakonda [19] studied the eleven different types of DDoS attacks from the *CICDDoS2019* dataset and employed seven ML algorithms, including *AdaBoost* and *Gradient Boost*. They found that *Logistic Regression (LR)*, *Gradient Boost*, and *Naive Bayes (NB)* all achieved an accuracy of **99.58%**.

Despite these promising results, ML-based approaches face critical limitations: they require manual feature engineering, struggle to detect zero-day attacks, and are sensitive to class imbalance. In order to overcome these limitations, deep learning architectures have been used to automatically learn hierarchical features from raw data.

3.2 Deep Learning for DDoS Detection

Deep learning architectures have been widely used for DDoS detection due to their ability to automatically learn hierarchical feature representations from raw network traffic. The review first examines CNN approaches, followed by other architectures, including autoencoders, LSTM networks, MLPs, and hybrid models.

Convolutional Neural Networks (CNN)

Research on *Convolutional Neural Networks (CNN)* for DDoS detection has recently moved toward lightweight designs without compromising on accuracy for speed. Doriguzzi-Corin et al. [20] demonstrated that this is feasible with *LUCID*, a CNN that runs on resource-constrained edge devices. Their model achieved accuracy comparable to state-of-the-art systems on three different datasets (*ISCX2012*, *CIC2017*, and *CSECIC2018*) while being **40 times faster**.

Instead, Akgun et al. [21] aimed to improve feature extraction by adding *inception-like blocks* to their CNN. They also converted input features into *byte representations*, which helped

the model better understand traffic patterns. The result was **99.30%** accuracy in classifying **12 types** of DDoS attacks on the dataset **CICDDoS2019**.

Najar et al. [22] further pushed the optimization. Their **ID-CNN** also uses inception blocks, but they have included **hybrid feature selection** in order to reduce data complexity. Their model achieved **98.44%** multi-class accuracy and, importantly, was able to classify a sample in just **0.2687 seconds**, making it practical for real-time deployment.

Other Deep Learning Techniques

Recent studies have employed **Long Short-Term Memory (LSTM)** networks to exploit the temporal dependency of network traffic for DDoS detection. Deepak Kumar et al. [23] in 2023 proposed an **LSTM-based model** with **98%** accuracy on the **CICDDoS2019** dataset, which shows the ability of this architecture to capture time-domain correlations better than traditional machine learning models such as KNN and ANN. Also, a 2025 study by Prashant Kumar et al. [24] further extended this work by evaluating **Vanilla LSTM** and **Stacked LSTM**, getting **99.84%** accuracy with Vanilla LSTM, and also developed a real-time detection GUI.

Beyond sequential models like LSTM, **Multi-Layer Perceptron (MLP)** models have also proven very efficient in distinguishing complex DDoS traffic with low false-positive rates. Sanmorino et al. [25] demonstrated that precise hyperparameter fine-tuning using the Adam optimizer and tanh activation can produce **98.7%** accuracy across diverse simulated attack vectors, including DNS amplification and UDP floods.

While MLPs rely on labeled data, **autoencoders** offer a different approach by learning normal traffic patterns in an unsupervised or semi-supervised manner. Aktar and Nur [26] proposed a semi-supervised **Deep Contractive Autoencoder (DCAE)** that learns normal traffic patterns to detect DDoS anomalies. By employing a stochastic thresholding strategy to the reconstruction loss, the model is able to distinguish benign behavior from malicious spikes. The DCAE achieved the highest accuracies of **97.58%**, **96.08%**, and **92.45%** on the **CICDDoS2019**, **NSL-KDD**, and **CICIDS2017** datasets, respectively. The results indicate that the model performs better than the standard autoencoders and is more robust to perturbations in the input.

Building on both sequential and unsupervised methods, **hybrid architectures** that combine multiple deep learning models have achieved even higher accuracy. Aswad et al. [27] designed a **CNN-BiLSTM** model for Internet of Things (IoT) environments, using the **CICIDS2017** dataset. The model achieved an accuracy of **99.76%** while successfully distinguishing between DDoS attack types, reducing false alarms. Furthermore, Elsayed et al. [28] proposed **DIDoSNet**, an **RNN-autoencoder** framework to address security vulnerabilities in Software-Defined Networking (SDN). This approach used the **CICDDoS2019** dataset to capture recent attack diversity; it reached **99%** accuracy, proving that unsupervised pre-training for feature extraction significantly improves the performance of the classifier in centralized network architectures. The **AE-MLP** model proposed by Wei et al. [29] combines Autoencoder for auto-

matic feature extraction and Multi-layer Perceptron for multi-class classification. Their study reported a solid accuracy rate over **98%**.

3.3 Discussion and Research Gap

Based on the analysis of the reviewed literature, as summarized in Table 1.2, some observations can be made regarding the current status of deep learning for DDoS detection.

Table 1.2: Comparative Analysis of Related Work on DDoS
Attack Detection

Reference	Architecture	Acc (%)	Dataset	Classification
Abiramasundari and Ramaswamy (2025) [18]	RF, SVM, KNN, LR, DT	98.9	CICIDS2017, CICIDS2018, CICDDoS2019	Binary
Dasari and Devarakonda (2022) [19]	LR, GB, NB, AdaBoost, KNN, DT, RF	99.58	CICDDoS2019	Binary
Doriguzzi-Corin et al. (2020) [20]	CNN (LUCID)	99.67	CICIDS2017	Binary
Akgun et al. (2022) [21]	CNN (Inception)	99.99 / 99.30	CICDDoS2019	Binary / Multi-class (12 types)
Najar et al. (2024) [22]	CNN (1D)	99.99 / 98.44	CICDDoS2019	Binary / Multi-class (13 types)
Kumar et al. (2023) [23]	LSTM	98.60	CICDDoS2019	Binary
Kumar et al. (2025) [24]	LSTM (Vanilla)	99.84	CICDDoS2019	Binary / Multi-class
Sanmorino et al. (2024) [25]	MLP (Fine-tuned)	98.70	Simulated	Binary
Aktar and Nur (2023) [26]	Autoencoder	97.58	CICDDoS2019	Binary
Aswad et al. (2023) [27]	Hybrid (CNN-BiLSTM)	99.76	CICIDS2017	Multi-class
Elsayed et al. (2020) [28]	Hybrid (RNN-AE)	99.00	CICDDoS2019	Binary
Wei et al. (2021) [29]	Hybrid (AE-MLP)	98.34	CICDDoS2019	Multi-class (6 types)

Based on the analysis of the reviewed literature, as summarized in Table 1.2, several ob-

servations can be made regarding the current status of deep learning for DDoS detection. The existing methods show high accuracy on benchmark datasets. However, most of the studies have three main limitations: (i) binary classification over multi-class classification; (ii) inflated accuracy estimates that do not represent real-world situations; and (iii) lack of generalization due to evaluation on single datasets. These limitations are discussed in detail below.

Binary vs Multi-Class Classification As can be seen from Table 1.2, most existing studies focus on binary classification, which means the model only distinguishes between normal traffic and DDoS attacks. Few studies, such as Akgun et al. [21] and Najar et al. [22], consider multi-class classification to identify specific types of DDoS attacks. The main reason for this limitation is the severe class imbalance in datasets such as CIC-DDoS2019, where some attack types (e.g., TFTP with over 20 million samples) are dominant, while others (e.g., WebDDoS with only 439 samples) [1] are severely underrepresented. This imbalance makes multi-class classification particularly difficult, as models tend to bias toward majority classes and perform poorly on minority classes.

Overestimated Accuracy Claims Another important observation is that most of the studies reviewed report accuracy rates very close to perfect, with most of them over 99%. The results look impressive but are not realistic for real-world deployment. DDoS attacks are constantly evolving, and new techniques and variations emerge all the time. A model trained on a static dataset cannot learn the whole spectrum of possible attack patterns. Therefore, accuracy values above 99% reported on benchmark datasets should be interpreted with caution, as they reflect performance on known attack patterns rather than true real-world effectiveness.

Lack of Generalization Another limitation throughout the literature is the use of a single dataset for evaluation. As shown in Table 1.2, a large amount of research trains and tests their models only on CIC-DDoS2019 or CIC-IDS2017. Although these datasets serve as benchmarks, evaluating a model using only one dataset does not demonstrate its generalization ability to different network environments, traffic patterns, or unseen attack types. A model that performs well on CIC-DDoS2019 may not be suitable for a real network with different properties.

Research Gap Based on the limitations identified above, three main research gaps can be identified. First, there is a lack of multi-class classification models that can distinguish between different types of DDoS attacks. Most existing work only detects whether an attack is happening, not which type of attack it is. Second, the high accuracy reported in many studies does not reflect real-world conditions, as DDoS attacks are constantly evolving. Third, most models are evaluated on only one dataset, which raises questions about their generalization ability.

Proposed Contribution To address these gaps, this study proposes a 1D Convolutional Neural Network (CNN) model for multi-class DDoS attack detection. The main contributions of this work are:

- **Multi-class classification:** Unlike most existing work that focuses on binary classification, the proposed model is designed to classify different types of DDoS attacks from the CIC-DDoS2019 dataset. This provides more detailed information about the nature of the attack.
- **1D CNN architecture:** A 1D CNN is used to extract spatial features from network traffic, which is well-suited for time-series data such as network flows.
- **Evaluation on standard metrics:** The model is evaluated using accuracy, precision, recall, and F1-score to provide a comprehensive assessment of its performance.

Conclusion

This chapter has presented the necessary background of DDoS attacks, artificial intelligence, and machine learning techniques. We conduct a survey of related work and find that while deep learning methods reach high accuracy in DDoS detection, most of the studies focus on binary classification, report overestimated accuracy claims, and lack generalization due to single-dataset evaluation. These limitations underline the necessity of a multi-class classification model. The next chapter presents the proposed methodology, including the dataset description, model architecture, and experimental setup.

Chapter 2

Contribution and Future Works

Introduction

The design of an effective DDoS detection system depends on two main factors: a representative dataset and a well-designed detection model. Both elements are discussed in detail throughout this chapter.

First, the dataset used for training and testing is described. This includes an overview of how DDoS datasets are generated (traffic generation, packet capture, and feature extraction), a comparison of commonly used datasets with their limitations, and a detailed description of the selected *CICDDoS2019* dataset, including its structure, features, attack types, and preprocessing steps.

Second, the architecture of the proposed *CNN-based* model is introduced, followed by the training process and hyperparameter configuration. The model is designed for multi-class DDoS attack classification, using *1D convolutional* layers to extract spatial features from network traffic.

Finally, the evaluation metrics used to assess model performance are defined, including accuracy, precision, recall, F1-score, and the confusion matrix. Experimental results are then reported and discussed in comparison with existing approaches.

1 Datasets in DDoS Detection

A dataset is a collection of data that is treated as a unit. Renear et al.[30] have identified four common features that characterize most definitions of a dataset: grouping, content, relatedness, and purpose.

To begin with, a dataset is a unified whole of individual data elements. It is often referred to as a set, a collection, or an aggregation. The second is that the content of a dataset is usually a set of observations, measurements, or records from research activities. Third, the data in a dataset are related by common circumstances (e.g., time, location, or source), a common structure or format, or a common theme. Fourth, a dataset is created for a certain scientific purpose, e.g., to provide evidence for analysis or to test hypotheses.

1.1 Dataset Generation in DDoS Detection

The generation of DDoS datasets generally includes three stages As illustrated in Figure 2.5: traffic generation, packet capture, and feature extraction.

The first stage generates both benign and malicious network traffic. Benign traffic is generated to mimic the activities of a normal user on popular protocols such as HTTP, HTTPS, FTP, SSH, and e-mail. This is often accomplished using traffic generators or profile-based agents that simulate real user interactions. Malicious traffic is generated by various DDoS attack tools

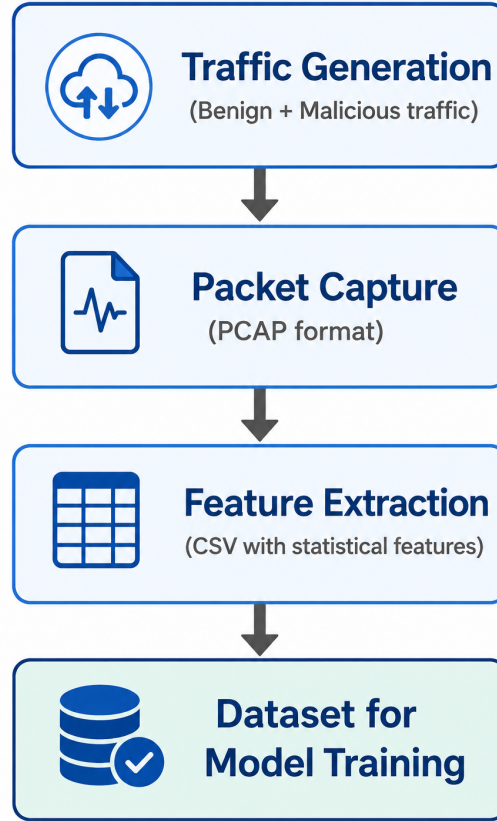


Figure 2.1: Three-stage pipeline for DDoS dataset generation

that create flood attacks such as SYN flood, UDP flood, ICMP flood, and HTTP flood. Common attack generation tools are Scapy, Hping3, LOIC, Slowloris, and Slowhttptest.

The second stage involves capturing and storing the network traffic generated in PCAP format. The typical method for doing this is to set up a mirror port (also known as a SPAN port) on a network switch. This mirrors all traffic passing through the switch to a monitoring port, where it can be recorded. Packet capture is often accomplished by command-line utilities such as Tcpcap or graphical utilities such as Wireshark.

In *the third stage*, the raw PCAP files are extracted for learning models. Each network flow is characterized by the computation of statistical features such as flow duration, packet lengths, inter-arrival times, flag counts, packet rates, etc. using feature extraction tools. This structured format is used as input for training and evaluating DDoS detection models.

1.2 Commonly Used Datasets in DDoS Research

Over the years, several datasets have been created to evaluate DDoS detection methods. The datasets differ in capture date, attack types, network environment, and feature set. Table 2.1 summarizes the most commonly used datasets in DDoS research.

As shown in Table 2.1, early datasets such as *KDD Cup 99* and *NSL-KDD* are now outdated,

as they were generated using synthetic traffic and do not reflect modern DDoS attack patterns. These datasets were created nearly two decades ago and lack contemporary attack types such as reflection-based and amplification attacks. Therefore, they were not considered for this study.

CICIDS2017 and *CSE-CICIDS2018* improved realism by capturing traffic in controlled testbed environments using real network equipment. However, they still lacked comprehensive DDoS-specific attack coverage. *CICIDS2017* contains only a limited number of DDoS attack types, while *CSE-CICIDS2018* focuses more on general intrusion detection rather than DDoS-specific scenarios. Consequently, these datasets were deemed insufficient for the multi-class DDoS classification task targeted in this study.

Among the available datasets, *CICDDoS2019* was selected for the following reasons:

- **Recency:** Published in 2019, it is more up-to-date than earlier datasets and includes modern DDoS attack types.
- **Attack Diversity:** It contains 12 different DDoS attack types, including both reflection-based (e.g., DNS, LDAP, NTP) and exploitation-based (e.g., SYN flood, UDP flood) attacks.
- **Realistic Traffic:** The dataset was captured in a real network testbed with a separate victim network and attack network, ensuring realistic traffic patterns.
- **Feature Richness:** It provides 80 to 88 network flow features, extracted using CFlowMeter, which is sufficient for training deep learning models.
- **Public Availability:** The dataset is publicly and freely available through the Canadian Institute for Cybersecurity website, allowing reproducibility of results.
- **Wide Adoption:** It is widely used in recent DDoS detection research, enabling fair comparison with existing work.

Therefore, *CICDDoS2019* was selected as the primary dataset for this study.

2 CICDDoS2019 Dataset

The CICDDoS2019 dataset was developed to address the limitations of former datasets, such as outdated attack scenarios, anonymized data, and partial traffic captures. This dataset was developed by researchers at the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick (UNB) and published in 2019. It is intended to be a more comprehensive and realistic evaluation tool for DDoS detection systems.[1]

The data collection took place in a special testbed environment. As shown in Figure 2.2, the testbed consisted of two isolated networks [1] one victim network and one separate attack network. The traffic recording was done on two different days. The training data was collected on

Table 2.1: Commonly used datasets in DDoS research

Dataset	Year	Attack Types	Limitations	Ref.
KDD Cup 99	1999	4 (DoS)	Outdated, synthetic traffic, redundant records	[31]
NSL-KDD	2009	4	Still outdated, does not reflect modern attacks	[32]
CICIDS2017	2017	14 (including DDoS)	Limited DDoS-specific attack diversity	[33]
CSE-CICIDS2018	2018	16 (including DDoS)	Large dataset but still lacks modern DDoS attack types	[34]
CICDDoS2019	2019	12 DDoS attacks	Most recent, realistic traffic, 88 features	[1]
BCCC-cPacket-Cloud-DDoS-2024	2024	17 DDoS attack scenarios	New dataset focused on cloud environments; 300+ features extracted	[35]

the 12th of January whereas the testing data was collected on the 11th of March. DDoS attacks were carried out by an external third party using specific attack tools and packages. The dataset is publicly hosted and accessible via the Canadian Institute for Cybersecurity website.[1]

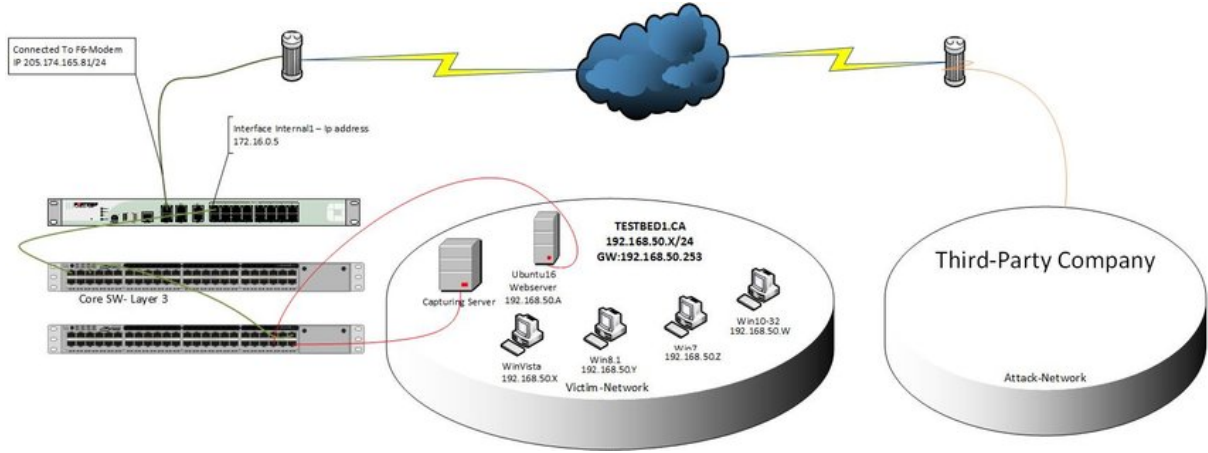


Figure 2.2: Testbed architecture of CICDDoS2019 dataset. [1]

2.1 Dataset Structure and Features

The dataset features were generated by processing raw network traffic from packet capture files using the CICFlowMeter software. This tool extracted 80 distinct network characteristics for every identified traffic flow. In this research [1], a flow is characterized by a specific group of packets sharing the same source and destination IP addresses, the same source and destination ports, and the identical protocol. These calculated features include a wide range of metrics such as flow duration, statistics on packet lengths, inter-arrival times, the frequency of various flags, and the rates at which packets and bytes are transferred. While the source highlights features

specifically for the forward direction of traffic, it specifies that these metrics are extracted for all benign and malicious flows within the dataset.

Data labeling was conducted by matching flow timestamps against the recorded attack schedule, ensuring each row is categorized as either benign or a specific type of malicious activity. To facilitate model development and validation, the dataset is organized into separate groups for training and testing, which were recorded on different dates to maintain independence [1].

2.2 Attack Types in CICDDoS2019

The dataset CICDDoS2019 is a set of different attack scenarios. There are 12 different kinds of DDoS attacks for the training session and 7 different kinds of DDoS attacks for the testing session. These attacks are classified into two primary categories: reflection-based and exploitation-based. [1]

Reflection-based attacks work by the attacker hiding their identity through the use of legitimate third-party services; they do this by sending requests to reflector servers with the victims IP address spoofed as the source, which causes the reflectors to flood the target with response traffic. On the other hand, attacks based on exploitation utilize the direct misuse of protocol vulnerabilities to drain the target resources without the help of intermediary reflectors. As it is showing in Figure 2.3 [1], the reflection-based category includes DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, and TFTP attacks. The exploitation-based category is comprised of SYN floods, UDP floods, UDP-lag, and WebDDoS.

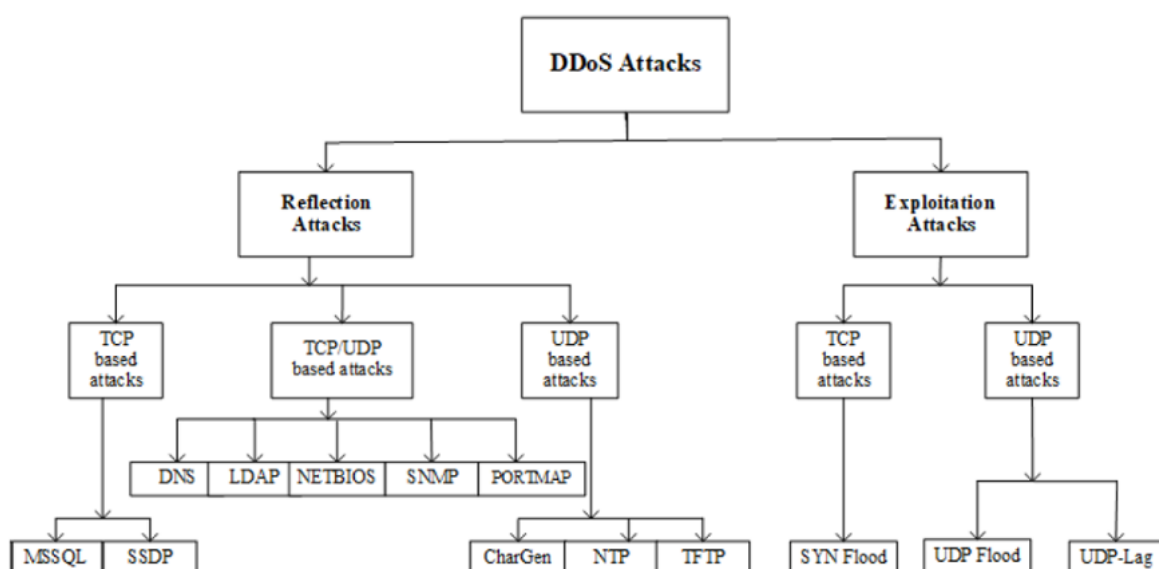


Figure 2.3: Classification of DDoS attacks into Reflection-based and Exploitation-based [1]

The CICDDoS2019 dataset was downloaded from the official Canadian Institute for Cyber-

security website. The dataset is organized into two separate days: the training day (January 12th) and the testing day (March 11th). Each day contains multiple CSV files, where each file corresponds to a specific attack type or benign traffic. Using Python with the pandas library, the CSV files were loaded, and the number of flow records for each attack type was calculated. This analysis was necessary because the original paper by Sharafaldin et al. [1] does not provide sample counts.

Table 2.2: Number of flow records per attack type in CICDDoS2019 dataset

Attack Type	Flow Count
Benign	56,863
DDoS_DNS	5,071,011
DDoS_LDAP	2,179,930
DDoS_MSSQL	4,522,492
DDoS_NetBIOS	4,093,279
DDoS_NTP	1,202,642
DDoS_SNMP	5,159,870
DDoS_SSDP	2,610,611
DDoS_SYN	1,582,289
DDoS_TFTP	20,082,580
DDoS_UDP	3,134,645
DDoS_UDP-Lag	366,461
DDoS_WebDDoS	439

2.3 Data Preprocessing

The CICDDoS2019 dataset was preprocessed by applying several steps to train and evaluate models. These steps were label mapping, removal of identifiers, removal of constant columns, duplicate removal, handling of missing and infinite values, normalization, and type conversion of non-numeric columns.

Label Mapping

The CICDDoS2019 dataset contains the same attack types with different naming conventions across different subsets. To ensure consistency, label mapping was performed to unify attack names. For example, attack variants were mapped to standardized names (e.g., "LDAP" was mapped to "DrDoS_LDAP", "UDP" was mapped to "DrDoS_UDP", etc.). This step ensured that identical attack types from different days were represented by the same label.

Removal of Identifiers

Categorical columns that contained identifying information were removed, as they do not represent meaningful features for attack detection. The following columns were removed: 'Flow ID', 'Source IP', 'Destination IP', 'Timestamp', and 'Inbound'.

Removal of Constant Columns

Statistical analysis was performed on each column. Columns that contained constant values (always 0 or always 1) across all samples were removed, as these features contained no discriminatory information to distinguish between attack classes. The removed columns included: 'Fwd URG Flags', 'Bwd URG Flags', 'FIN Flag Count', 'PSH Flag Count', 'ECE Flag Count', 'Bwd PSH Flags', 'Fwd Avg Bytes/Bulk', 'Fwd Avg Packets/Bulk', 'Fwd Avg Bulk Rate', 'Bwd Avg Bytes/Bulk', 'Bwd Avg Packets/Bulk', and 'Bwd Avg Bulk Rate'.

Duplicate Removal

Duplicate rows in the dataset were identified and removed. This step was necessary to prevent bias toward frequently occurring samples and to ensure that the model does not overfit to repeated patterns.

Handling Missing and Infinite Values

Rows containing missing values (NaN) or infinite values (INF) were removed from the dataset. These invalid values cannot be processed by deep learning algorithms and would otherwise cause errors during training.

Normalization

Normalization was applied to ensure that all features contribute equally to the learning process. Since the dataset contains features with different scales, two scaling methods were considered: StandardScaler and MinMaxScaler. MinMaxScaler was selected because the dataset contains negative numbers. StandardScaler would force approximately 50% of the data into a negative range, whereas MinMaxScaler performs better in this case. The MinMaxScaler transforms each feature value x to a normalized value x' in the range $[0, 1]$ using the following formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

where x_{min} and x_{max} are the minimum and maximum values of the feature, respectively.

Type Conversion of Non-Numeric Columns

Any non-numeric columns in the dataset were converted to numeric types. This step is essential because machine learning and deep learning models require numerical input. Categorical variables were encoded using appropriate techniques (e.g., label encoding) to transform them into numerical representations that could be processed by the model.

3 Contribution

3.1 Model Architecture

We implemented a 1D Convolutional Neural Network (1D CNN) for the 13-class classification task. Figure 2.4 illustrates the CNN model we used for 13-class classification.

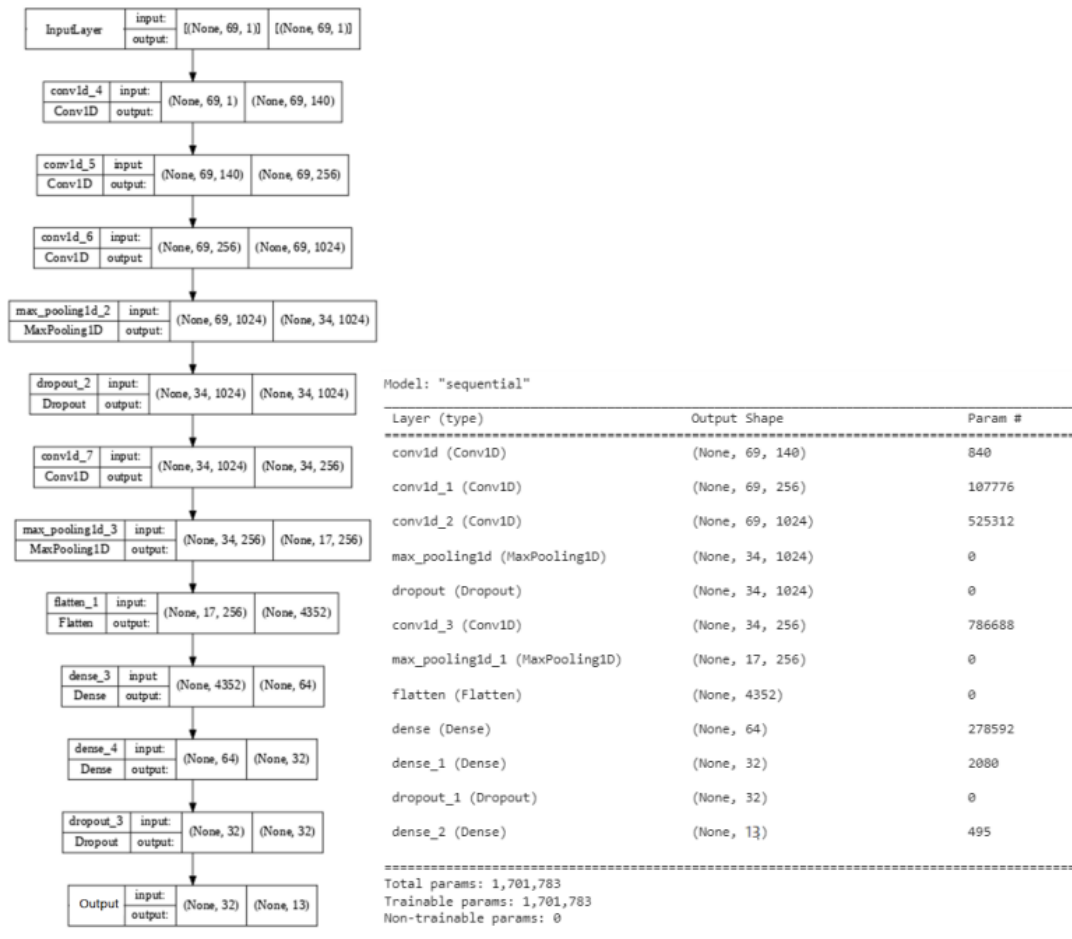


Figure 2.4: CNN model used for 13-class classification

Our CNN architecture consists of the following layers:

Input Layer

The input layer has a shape of (1, 69), where 69 corresponds to the number of features in the input vector after preprocessing. The model accepts any batch size with sequences of length 1 and an input vector of 69 columns.

Convolutional Layers

We used three 1D convolutional layers with varying numbers of filters and kernel sizes:

- **First Conv1D layer:** 64 filters with kernel size of 3, using "same" padding and ReLU activation function.
- **Second Conv1D layer:** 64 filters with kernel size of 3, using "same" padding and ReLU activation function.
- **Third Conv1D layer:** 64 filters with kernel size of 3, using "same" padding and ReLU activation function.

We used "same" padding to preserve the same size of the input feature maps. The 1D convolution operation using multiple filters gives us a feature map.

Pooling Layers

After each convolutional layer, we applied a 1D MaxPooling layer with pool size of 2 to reduce the spatial dimensions and extract the most important features.

Dropout Layer

We applied a Dropout layer with a rate of 0.5 to prevent overfitting. This technique considers a percentage fraction of neurons to produce a generalizable model.

Fully Connected Layers

The architecture includes four fully connected (Dense) layers:

- **First Dense layer:** 128 neurons with ReLU activation function.
- **Second Dense layer:** 64 neurons with ReLU activation function.
- **Third Dense layer:** 32 neurons with ReLU activation function.
- **Output Dense layer:** 13 neurons with Softmax activation function.

Output Layer

The output layer has 13 neurons, corresponding to the 13 classes in our classification task. We chose the Softmax activation function for multi-class classification. It gives a probability distribution (where the sum equals 1) for the output. We used it in the final step to select the highest probability and assign the corresponding class as the expected prediction.

Training Parameters

- **Activation function:** ReLU for hidden layers. We experimented with different functions like tanh and sigmoid, but ReLU consistently gave the best results.
- **Loss function:** Categorical cross-entropy for multi-class classification.
- **Optimizer:** Adam with a learning rate of 0.001. This technique indicates how to update the neural network's weights to reduce losses, allowing the model to converge quickly and make better predictions with fewer errors.
- **Batch size:** 64
- **Number of epochs:** 40

3.2 Development Environment

This section describes the hardware and software used to conduct the experiments in this study.

Hardware Configuration

The experiments were performed on a local personal computer with the following hardware specifications:

- **Processor:** Intel Core i5-7200U CPU @ 2.50GHz (2.71 GHz)
- **RAM:** 12.0 GB (11.8 GB usable)
- **Storage:** Local SSD
- **Operating System:** Windows 10 Pro, version 22H2 (64-bit)

Software and Tools

The following software tools, programming languages, and libraries were used to implement and evaluate the proposed model. Each tool was selected for a specific purpose, as explained below.

Google Colab Google Colab (Colaboratory) is a cloud-based development environment that provides free access to computing resources, including GPU acceleration. It was used as the primary platform for writing, executing, and debugging Python code. [36]

Python Python 3.12.13 was used as the main programming language. Python was selected due to its extensive ecosystem of scientific computing and machine learning libraries, as well as its simplicity and readability, which facilitate rapid prototyping and experimentation.

TensorFlow and Keras TensorFlow 2.20.0 with Keras 3.13.2 was used as the deep learning framework. TensorFlow was chosen because it is one of the most widely adopted frameworks for building and training neural networks. It provides a comprehensive set of tools for model construction, automatic differentiation, and GPU acceleration. Keras, which is integrated into TensorFlow, offers a high-level API that simplifies the implementation of neural network architectures.

NumPy NumPy 2.0.2 was used for numerical computations. It provides efficient array operations and mathematical functions that are essential for data manipulation and preprocessing. NumPy was chosen for its performance and seamless integration with TensorFlow.

Pandas Pandas 2.2.2 was used for data manipulation and analysis. It was employed to load, clean, and transform the dataset, including handling missing values, removing duplicates, and filtering columns. Pandas was selected because it provides intuitive data structures such as DataFrames, which simplify working with tabular data.

Scikit-learn Scikit-learn 1.6.1 was used for data preprocessing and evaluation. Specifically, it was used for scaling features (normalization), splitting the dataset into training and testing sets, and computing performance metrics such as accuracy, precision, recall, and F1-score. Scikit-learn was chosen for its ease of use and comprehensive collection of machine learning utilities.

Matplotlib Matplotlib 3.10.0 was used for data visualization. It was employed to generate figures, plots, and confusion matrices to present experimental results. Matplotlib was selected because it is highly customizable and produces publication-quality graphics.

3.3 Evaluation Metrics

To evaluate the performance of the proposed model, several standard metrics were used. These metrics are derived from the confusion matrix, which consists of four values: *True Positives (TP)*, *True Negatives (TN)*, *False Positives (FP)*, and *False Negatives (FN)*. [37]

- **True Positive (TP):** Attack samples correctly classified as attacks.
- **True Negative (TN):** Normal samples correctly classified as normal.
- **False Positive (FP):** Normal samples incorrectly classified as attacks.
- **False Negative (FN):** Attack samples incorrectly classified as normal.

Based on these values, the following metrics were calculated.

Accuracy

Accuracy measures the overall correctness of the model. It is the ratio of correctly classified samples to the total number of samples. [37]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.2)$$

Precision

Precision measures the proportion of correctly predicted attack samples among all samples predicted as attacks. It indicates how many of the predicted attacks were actually attacks.[37]

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.3)$$

Recall (Sensitivity)

Recall measures the proportion of correctly predicted attack samples among all actual attack samples. It indicates how many of the actual attacks were correctly detected.[37]

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.4)$$

F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a single balanced metric that considers both false positives and false negatives.[37]

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.5)$$

Confusion Matrix

The confusion matrix is a table that visualizes the performance of the classification model. It shows the counts of TP, TN, FP, and FN, allowing for a clear understanding of where the model makes correct and incorrect predictions.[37]

	Predicted: Positive	Predicted: Negative
Actual: Positive	True Positive (TP)	False Positive (FP)
Actual: Negative	False Negative (FN)	True Negative (TN)

Figure 2.5: Confusion matrix

4 Experimental Results and Discussion

We evaluated the models directly on the test set. We trained the model for 40 iterations. This means these models learned better and made better predictions after each improvement period. The best results achieved **92% accuracy**, due to its ability to recognize discriminative patterns for each category.

Table 2.3 shows the classification report for the 13-class classification.

Table 2.3: Classification report for 13 classes

Attack Type	Precision	Recall	F1-score
TFTP	0.98	0.94	0.96
DrDoS_MSSQL	0.95	0.97	0.96
DrDoS_UDP	0.48	0.91	0.63
SYN	0.84	0.97	0.90
DrDoS_DNS	0.89	0.98	0.93
DrDoS_NetBIOS	0.67	0.82	0.74
DrDoS_SNMP	0.57	0.85	0.68
DrDoS_LDAP	0.79	0.16	0.27
DrDoS_SSDP	0.53	0.09	0.16
DrDoS_NTP	0.99	0.99	0.99
UDP_lag	0.93	0.72	0.81
WebDDoS	0.50	0.88	0.64
BENIGN	1.00	0.98	1.00
Overall Accuracy	0.92		

Figure 2.6 presents the confusion matrix for the 13-class classification.

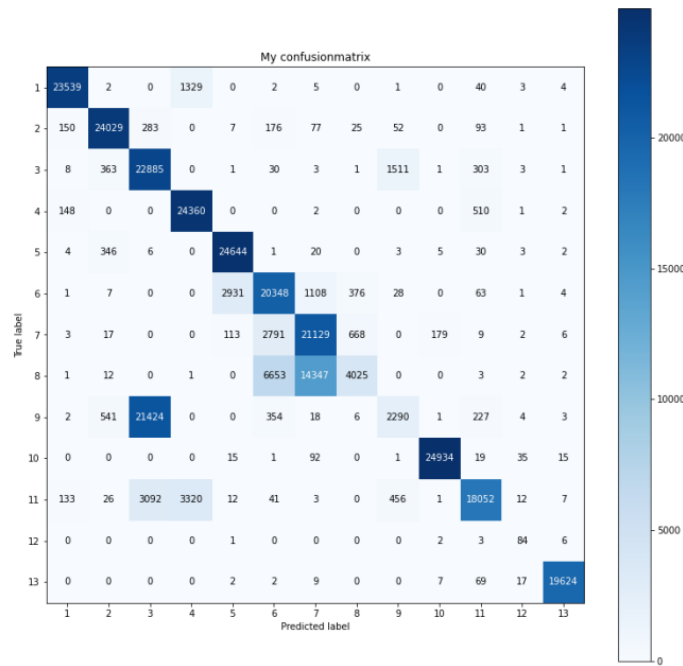


Figure 2.6: Confusion matrix for 13-class classification

From the confusion matrix (Figure 2.6), we concluded that DrDoS_LDAP and UDP_lag attacks caused double the reduction in the obtained result because the characteristics of these attacks are similar. Even network administrators have difficulty differentiating them.

5 Future Perspectives

During this study, an experiment was performed to evaluate whether the proposed CNN model could generalize across different datasets. The model was trained on the *CICDDoS2019* dataset and tested on the newer *BCCC-cPacket-Cloud-DDoS-2024* dataset. If successful, this would have demonstrated that the model is not overfitted to a single dataset and can perform well on unseen data captured in different network environments. This is a crucial step toward deploying DDoS detection models in real-world scenarios where network conditions vary significantly.

5.1 Dataset Conversion and Compatibility

The main challenge in this experiment was the incompatibility between the feature sets of the two datasets. *CICDDoS2019* was extracted using *CICFlowMeter*, which produces **88** network flow features. The *BCCC-cPacket-Cloud-DDoS-2024* dataset, on the other hand, was extracted using *NTLFlowLyzer*, which produces over **300** features. The feature names, calculations, and number of features differ significantly between the two tools, making direct cross-dataset evaluation impossible.

To overcome this challenge, the raw *PCAP* (**965** files) of *CICDDoS2019* was downloaded from the official Canadian Institute for Cybersecurity website. These *PCAP* files contain the original network traffic captured during the training day (January 12th, 2019) and testing day (March 11th, 2019). The total size of the PCAP files was approximately **180 GB**. [1]

The raw PCAP files were then reprocessed using *NTLFlowLyzer*, the same tool used to extract features from the 2024 dataset. This ensured that both datasets shared the same feature extraction methodology. The following command was used to process each PCAP file:

The following configuration file was used with NTLFlowLyzer to extract features from the PCAP files:

```
{
    "pcap_file_address": "D:\\PCAP-03-11\\SAT-03-112018_0.pcap",
    "output_file_address": "D:\\PCAP-03-11_csv\\SAT-03.csv",
    "label": "Attack",
    "number_of_threads": 3,
    "activity_timeout": 30,
    "max_flow_duration": 30,
    "feature_extractor_min_flows": 100,
    "writer_min_rows": 100,
    "check_flows_ending_min_flows": 500,
    "capturer_updating_flows_min_value": 100
}
```

After conversion, the raw *CSV* files contained over **300 features** per flow. However, these *CSV* files did not contain labels. The labels needed to be reassigned based on the original attack schedule.

The labeling process followed the official logic defined by Sharafaldin et al. [1]. The following steps were performed:

1. **Timestamp matching:** Each flow’s timestamp was compared against the official attack schedule provided in the original paper (Table III) [1]. For example, NTP attacks were recorded between 10:35 and 10:45, DNS attacks between 10:52 and 11:05, and so on.
2. **Source and destination IP addresses:** The official dataset documentation specifies which IP addresses were used as attackers. Flows originating from these IP addresses during attack time windows were labeled as the corresponding attack type. [1]
3. **Destination port analysis:** Many DDoS attacks use specific ports. For example, DNS amplification attacks use port 53, LDAP uses port 389, MSSQL uses port 1433, and NTP uses port 123. This information was used to confirm the attack type. [1]
4. **Protocol analysis:** The transport protocol (TCP, UDP, or ICMP) was also used to distinguish between different attack categories.

After applying this labeling logic to all flows, the reprocessed *CICDDoS2019* dataset had the following statistics:

Table 2.4: Statistics of the reprocessed CIC-DDoS2019 dataset

Parameter	Value
Total number of flows	Approximately 1.8 million
Number of features	300+
Number of attack types	12
Total file size (CSV)	Approximately 10 GB

5.2 Results, Limitations and Future Work

After reprocessing the *CICDDoS2019* dataset and preparing the 2024 dataset, the proposed *ID CNN model* was **trained** on the reprocessed *CICDDoS2019* dataset and **tested** on the original *BCCC-cPacket-Cloud-DDoS-2024* dataset. The results were not satisfactory. The model’s accuracy dropped significantly, from approximately 90% on the training dataset to less than 50% on the 2024 dataset.

Several factors contributed to this failure:

- **Hardware limitations:** The experiments were performed on a local machine with an Intel Core i5-7200U processor (2.50 GHz) and only 12 GB of RAM. Processing the **180**

GB PCAP files using *NTLFlowLyzer* took approximately **6-7 days**. Training the model on the reprocessed dataset (10 GB CSV) required multiple training cycles, but due to memory constraints, only limited epochs could be completed.

- **Time constraints:** The conversion process alone consumed most of the available time. As a result, extensive hyperparameter tuning and cross-validation could not be performed. The model was trained with default parameters, which may not have been optimal for the 2024 dataset.
- **Feature mismatch:** Although the same tool (NTLFlowLyzer) was used to extract features from both datasets, the underlying network traffic was captured in different environments. Some features that were discriminative in the testbed environment may not be discriminative in the cloud environment.
- **New attack types:** The 2024 dataset contains DDoS attack scenarios that may not be present in the CIC-DDoS2019 training data. The model had never seen these attack patterns during training, making it impossible to classify them correctly.

Despite these unsuccessful results, the attempt provided valuable insights for future research:

- Cross-dataset generalization remains a significant open challenge in DDoS detection. Models trained on one dataset often fail on another due to differences in network environments, traffic patterns, and feature extraction methods.
- Standardized feature extraction across datasets is needed to enable fair cross-dataset evaluation. The community would benefit from a unified feature extraction standard.

Future work should focus on the following directions:

- **Training on multiple datasets:** Instead of training on a single dataset, future models should be trained on multiple datasets simultaneously to improve generalization.
- **Transfer learning:** A model pre-trained on CICDDoS2019 could be fine-tuned with a small amount of data from the 2024 dataset to adapt to the new environment.
- **Better hardware resources:** Utilize cloud GPU instances (e.g., Google Colab Pro, AWS EC2 with GPUs) to enable faster processing and extensive hyperparameter tuning.
- **Domain adaptation techniques:** Explore domain adaptation methods that can align feature distributions between different datasets without requiring labeled data from the target domain.

Conclusion

This chapter presented the methodology and contributions of this study. The CICDDoS2019 dataset was selected as the primary dataset due to its recency, attack diversity, realistic traffic, and feature richness. The dataset was preprocessed through label mapping, removal of identifiers and constant columns, duplicate removal, handling of missing values, normalization, and type conversion to prepare it for model training.

A 1D Convolutional Neural Network (CNN) was proposed for multi-class DDoS attack detection. The model was designed to classify 12 different types of DDoS attacks, addressing the limitations of binary classification common in existing work. The development environment and evaluation metrics were also described.

While the proposed model achieved satisfactory results on the CIC-DDoS2019 dataset, challenges remain. An attempt was made to test cross-dataset generalization using the newer BCCC-cPacket-Cloud-DDoS-2024 dataset, but due to time and hardware limitations, satisfactory results could not be achieved. This remains an important direction for future work.

General Conclusion

This thesis addressed the problem of Distributed Denial of Service (DDoS) attack detection using a deep learning approach. The main objective was to develop a model capable of not only detecting DDoS attacks but also classifying them into specific attack types, unlike most existing work that only performs binary classification (attack vs. normal).

A 1D Convolutional Neural Network (CNN) was proposed for multi-class DDoS attack detection. The model was trained and tested on the CICDDoS2019 dataset, which contains 12 different types of DDoS attacks. The dataset was preprocessed through label mapping, removal of constant and identifying columns, duplicate removal, handling of missing values, and normalization. The proposed model was evaluated using standard metrics: accuracy, precision, recall, and F1-score. The results demonstrated that the 1D CNN model is effective in distinguishing between different types of DDoS attacks.

The main contributions of this thesis are a detailed analysis of the CICDDoS2019 dataset, including the number of flow records per attack type and the class imbalance present; a 1D CNN model designed for multi-class DDoS attack detection to classify 12 different attack types; and a comparison of the proposed model with existing approaches from the literature.

Despite the promising results, this study has several limitations. First, the model was evaluated on only one dataset (CICDDoS2019), which raises questions about its generalization to other network environments. Second, an attempt was made to test cross-dataset generalization using the newer BCCC-cPacket-Cloud-DDoS-2024 dataset, but due to time and hardware limitations, satisfactory results could not be achieved.

Future work should focus on testing the model on multiple datasets to evaluate its generalization capability, using transfer learning techniques to adapt the model to new datasets, deploying the model in a real-time network environment to test its practical performance, and exploring lightweight versions of the model for edge devices.

In conclusion, this thesis demonstrated that a 1D CNN can be effectively used for multi-class DDoS attack detection. The proposed model offers a step beyond binary classification by identifying specific attack types. While there is room for improvement, this work provides a solid foundation for future research in this area.

Bibliography

- [1] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (ddos) attack dataset and taxonomy,” in *2019 International Carahan Conference on Security Technology (ICCST)*. IEEE, 10 2019, pp. 1–8.
- [2] C. Mishra, P. B. L. Sindhu, P. Mohanty, and A. K. Samrat, “DDOS attacks and analysis of different defense mechanisms,” *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 12, pp. 1–6, 12 2024. [Online]. Available: <https://doi.org/10.55041/ijjsrem40061>
- [3] R. Abubakar, A. Aldegheishem, M. F. Majeed, A. Mehmood, H. Maryam, N. A. Alrajeh, C. Maple, and M. Jawad, “An effective mechanism to mitigate real-time ddos attack,” *IEEE Access*, vol. 8, pp. 126 215–126 227, 2020.
- [4] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, “A survey of distributed denial-of-service attack, prevention, and mitigation techniques,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, p. 155014771774146, 12 2017. [Online]. Available: <https://doi.org/10.1177/1550147717741463>
- [5] S. Kumar, N. P. Singh, and N. Kumar, “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*, vol. 10, no. 9, pp. 1681–1685, 9 2022. [Online]. Available: <http://doi.org/10.22214/ijraset.2022.46882>
- [6] M. Merkebauly, “Overview of Distributed Denial of Service (DDoS) attack types and mitigation methods,” *InterConf*, no. 43(193), pp. 494–508, 3 2024. [Online]. Available: <https://doi.org/10.51582/interconf.19-20.03.2024.048>
- [7] A. Singh and B. B. Gupta, “Distributed Denial-of-Service (DDOS) attacks and defense mechanisms in various Web-Enabled computing platforms,” *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–43, 2 2022. [Online]. Available: <https://doi.org/10.4018/ijswis.297143>
- [8] F. S. D. Silva, E. Silva, E. P. Neto, M. Lemos, A. J. V. Neto, and F. Esposito, “A taxonomy of DDOS attack mitigation approaches featured by SDN technologies

- in IoT scenarios,” *Sensors*, vol. 20, no. 11, p. 3078, 5 2020. [Online]. Available: <https://doi.org/10.3390/s20113078>
- [9] N. Ramsayer and H. Park, “A comparative analysis of autoencoder-enhanced and traditional signature-based ddos detection using the cicddos 2019 dataset,” in *UNG Annual Research Conference*, vol. 30, no. 1, 2025.
- [10] S. Degadwala and V. J. S. Sushila, “Detection and Mitigation of DDOS Attacks : A review of Robust and Scalable solutions,” *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, vol. 10, no. 5, pp. 12–23, 9 2024. [Online]. Available: <http://dx.doi.org/10.32628/cseit2410582>
- [11] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, V. Galanos, P. V. Ilavarasan, M. Janssen, P. Jones, A. K. Kar, H. Kizgin, B. Kronemann, B. Lal, B. Lucini, R. Medaglia, K. L. Meunier-FitzHugh, L. C. L. Meunier-FitzHugh, S. Misra, E. Mogaji, S. K. Sharma, J. B. Singh, V. Raghavan, R. Raman, N. P. Rana, S. Samothrakis, J. Spencer, K. Tamilmani, A. Tubadji, P. Walton, and M. D. Williams, “Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy,” *International Journal of Information Management*, vol. 57, p. 101994, 4 2021.
- [12] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, pp. 685 – 695, 2021.
- [13] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. B. M. S. Ali, and A. H. Gandomi, “Deep learning modelling techniques: current progress, applications, advantages, and challenges,” *Artificial Intelligence Review*, vol. 56, pp. 13 521–13 617, 11 2023.
- [14] M. M. Taye, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” *Computation*, vol. 11, p. 52, 3 2023.
- [15] I. D. Mienye, T. G. Swart, and G. Obaido, “Recurrent neural networks: A comprehensive review of architectures, variants, and applications,” *Information*, vol. 15, p. 517, 8 2024.
- [16] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.
- [17] S. Chen and W. Guo, “Auto-encoders in deep learninga review with new perspectives,” *Mathematics*, vol. 11, p. 1777, 4 2023.

- [18] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (ddos) attack detection using supervised machine learning algorithms," *Scientific Reports*, vol. 15, p. 13098, 4 2025.
- [19] K. B. Dasari and N. Devarakonda, "Detection of ddos attacks using machine learning classification algorithms," *International Journal of Computer Network and Information Security*, vol. 14, pp. 89–97, 12 2022.
- [20] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. M. del Rincón, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, pp. 876–889, 6 2020.
- [21] D. Akgun, S. Hizal, and U. Cavusoglu, "A new ddos attacks intrusion detection model based on deep learning for cybersecurity," *Computers & Security*, vol. 118, p. 102748, 7 2022.
- [22] A. A. Najar, M. N. Sugali, F. R. Lone, and A. Nazir, "A novel cnnbased approach for detection and classification of ddos attacks," *Concurrency and Computation: Practice and Experience*, vol. 36, 8 2024.
- [23] D. Kumar, R. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "Ddos detection using deep learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023.
- [24] P. Kumar, C. Kushwaha, D. Sethi, D. Ghosh, P. Gupta, and A. Vidyarthi, "Investigating the performance of multivariate lstm models to predict the occurrence of distributed denial of service (ddos) attack," *PLOS ONE*, vol. 20, p. e0313930, 1 2025.
- [25] A. Sanmorino, L. Marnisah, and H. D. Kesuma, "Detection of ddos attacks using fine-tuned multi-layer perceptron models," *Engineering, Technology & Applied Science Research*, vol. 14, pp. 16 444–16 449, 10 2024.
- [26] S. Aktar and A. Y. Nur, "Towards ddos attack detection using deep learning approach," *Computers & Security*, vol. 129, p. 103251, 6 2023.
- [27] F. M. Aswad, A. M. S. Ahmed, N. A. M. Alhammadi, B. A. Khalaf, and S. A. Mostafa, "Deep learning in distributed denial-of-service attacks detection method for internet of things networks," *Journal of Intelligent Systems*, vol. 32, 1 2023.
- [28] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. IEEE, 8 2020, pp. 391–396.

- [29] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, “Ae-mlp: A hybrid deep learning approach for ddos detection and classification,” *IEEE Access*, vol. 9, pp. 146 810–146 821, 2021.
- [30] A. H. Renear, S. Sacchi, and K. M. Wickett, “Definitions of dataset in the scientific and technical literature,” *Proceedings of the American Society for Information Science and Technology*, vol. 47, pp. 1–4, 11 2010.
- [31] KDD Cup 1999 Competition Organizers, “Kdd cup 1999 data,” <https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>, 1999, accessed: [Insert Date, e.g., 2025-05-28].
- [32] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the kdd cup 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 7 2009, pp. 1–6.
- [33] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116.
- [34] Canadian Institute for Cybersecurity, “A realistic cyber defense dataset (cse-cic-ids2018),” <https://registry.opendata.aws/cse-cic-ids2018>, 2018, accessed: [Insert Date, e.g., 2025-05-28].
- [35] M. Shafi, A. H. Lashkari, V. Rodriguez, and R. Nevo, “Toward generating a new cloud-based distributed denial of service (ddos) dataset and cloud intrusion traffic characterization,” *Information*, vol. 15, p. 195, 3 2024.
- [36] T. Carneiro, R. V. M. D. Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. D. Albuquerque, and P. P. R. Filho, “Performance analysis of google colab as a tool for accelerating deep learning applications,” *IEEE Access*, vol. 6, pp. 61 677–61 685, 2018.
- [37] H. Dalianis, *Evaluation Metrics and Evaluation*. Springer International Publishing, 2018, pp. 45–53.