Journal of
Sensor and Actuator Networks

MDPI

article

Detecting DDoS Attacks Using Machine Learning Techniques

Dhiaa A. Musleh, Nasser Alhajri, Rayyan AlNahwi, Aseel Alrudayni, Ibrahim Alhussaini, Noor Hadari

[1] Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[2] SAUDI ARAMCO Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
* Correspondence: damusleh@iau.edu.sa

Abstract: All operations in the contemporary digitized world rely on services that exist online. The availability of online systems faces serious challenges from Distributed Denial of Service (DDoS) attacks each year.. This paper aims to improve the latest techniques used in DDoS using Artificial Intelligence (AI) and Machine Learning (ML). Conventional methods of DDoS detection have thus been proved to be inadequate given the complexity of the attacks. Consequently, the paper looks at the different ML techniques to improve the functionality of DDoS detection systems. Furthermore, it could later assess the use of these techniques in cloud and Internet of Things (IoT) computing paradigms to overcome different issues arising from these frameworks. The idea is to build upon current lacks detection and to enhance model interpretability such that cybersecurity experts can address threats. This study will help to build effective, flexible and optimal anti-DDoS approaches that will help to strengthen the digital systems against new forms of cyber threats. For our paper we used the CICDDoS2019 dataset to build and test our model in this paper.

Keywords: Artificial intelligence (AI); Distributed Denial of Service (DDoS); Machine learning (ML); Support Vector Machine (SVM); k-NN; Random Forest (RF); Decision Tree (DT); Naive Bayes (NB); Intrusion Prevention System

1. Introduction

With the expansion of the Internet and increased reliance on it, cyber threats and their complexity have increased. One of these threats is DDoS which aims to flood available software with a large amount of traffic. We will focus on using AI to ease the process of detecting DDoS. In this paper, we will start talking about the purpose and scope, then the methodology for our work, then the implementation and result with findings. Imagine working in a reputable company such as Google, then suddenly, your computer freezes making you unable to continue the work intended. While the employee struggles with the computer, it turns out that many employees are facing the same issue. Symptoms such as a frozen computer or a sluggish responding computer are related to a widely known attack called a DDOS attack, this attack has caused devastating incidents which altered the way people handle the organization's security. Moreover, as we progress in this evolution in techno-related sub-jects, we need to provide better security for our organizations to ensure that many of the attacks such as the DDOS attacks can be minimized and captured before any major issues arise. AI and ML are making a noticeable evolution in

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

modern days [1], so why don't we use AI and ML to our advantage? How can we achieve this goal.

2. Purpose, Scope and Objective

The goal of our paper is to detect DDoS attacks in datasets of network traffic using simple ML algorithms. The objective is to quantify, to what extent ML can identify DDoS patterns from past network information. While the goal of our work is to be able to detect adversarial attacks in various scenarios, it is only applicable to predict network traffic datasets that are obtained from the public domain within a research laboratory environment. In the case of our work, a simple set of classification models including the Decision Tree or Random Forest algorithms, etc., will be employed to identify moving traffic as either normal or potentially carrying DDoS. In the next sections, we will concentrate on the models' training and their evaluation with small dataset sizes to compare their DDoS detection performance. The outcome is a report of the results gotten from the models and how accuracy and performance was done hence advancing the knowledge of DDoS using ML. So the main goal of our paper is to develop both the understanding of DDoS attack dynamics and the execution mechanisms of these attacks to achieve effective detection and mitigation through machine learning techniques. Our initial research will later expand into examining behavioral aspects of DDoS attacks together with their pattern varieties so we can make knowledgeable technical choices. The data collection along with preprocessing steps represents a crucial aspect of this venture because it requires obtaining high quality datasets from both actual network traffic or simulated networks that demonstrate typical and harmful behavior patterns. Data cleaning combined with normalization processes and feature extraction together with labeling methods work to maintain dataset integrity for training applications. After acquiring a structured and cleansed dataset we start building our model through the chosen machine learning techniques which includes decision trees and support vector machines as well as neural networks and many more algorithms. The data characteristics along with the desired features of DDoS traffic determine which algorithm should be selected for the analysis. The examination of system effectiveness begins after model training phase when performance metrics consisting of accuracy, precision, recall and F1-score are extracted for validating correct identification between legitimate traffic and DDoS at-tacks and reducing false positive. Evaluating the model enables us to test its performance in real conditions for robust deployment as an operational component of active network defense platforms. Our paper works toward developing both theoretical and practical elements which result in an effective data-based solution to strengthen DDoS protection capabilities for cybersecurity systems.

3. Background and Review of Literature

As we progress in today's vast evolving digital development, as is the case for any cyber-attacks, hence the need for a powerful tool to detect such attacks. One of the most devastating attacks is the DDoS attack, which targets a server by flooding it with a huge amount of traffic through the network. This caused many challenges to all private and public sectors across the world, even global companies such as Microsoft, Google and others faced these challenges at some point. Therefore, ML is one of the most promising solutions to detect all sophisticated DDoS attacks, by using many ML algorithms such as: Random Forest, DT and many others will aid many companies to be safe and cautious from DDoS attacks. Additionally, even with the evolution of DDoS attacks. Hackers are advancing with their improvements; therefore, we should be prepared with solutions to mitigate these attacks in any shape or form.[2]. A DDoS attack is an intentional attempt to orchestrate the stoppage of a specified server, service, or network through floods of

External Traffic. While a simple DDoS comes from a single location, DDoS consists of many contaminated machines, usually affiliated with a botnet, that overwhelm the intentioned site. These compromised devices can be situated in various parts of the world; thus, the attack is of a distributed type. The mechanism of a DDoS attack is aimed at completely consuming the bandwidth or resources of the targeted system so that genuine clients cannot access the Service. Traditional large-scale attack types such as DDoS are problematic to manage as they bring in a lot of traffic and it is hard to determine which traffic is bad.[3]

3.1 Types of DDoS attacks

DDoS attacks can be broadly classified into three main types based on the method of attack [1][4]:

- Volumetric Attacks: These are some of the most familiar kinds of DDoS attacks and the goal of those kinds of attacks is to use up the bandwidth of the target. Currently, the attackers congest the network with a hard and overwhelming bandwidth utilization. Some of the examples are User Datagram Protocol (UDP) flood or Internet Control Message Protocol (ICMP) flood.

- Protocol Attacks: These attacks take advantage of unrepaired weak points in network protocols to flood the network devices, including firewalls or load balancers, with excessive traffic. One classic example is the Synchronize (SYN) flood; this involves a suspect that sends SYN requests to begin a connection but does not proceed to the third stage of connection.

- Application Layer Attacks: These attacks flood the application with apparently valid requests improving its capability to pass through the system defenses. They are aimed at flooding the application layer (Layer 7 on the OSI model) that has to do with Web traffic. An example with the same intent is the Hypertext Transfer Protocol (HTTP) flood in which the attacker floods the web server with multiple requests with an intention of overloading

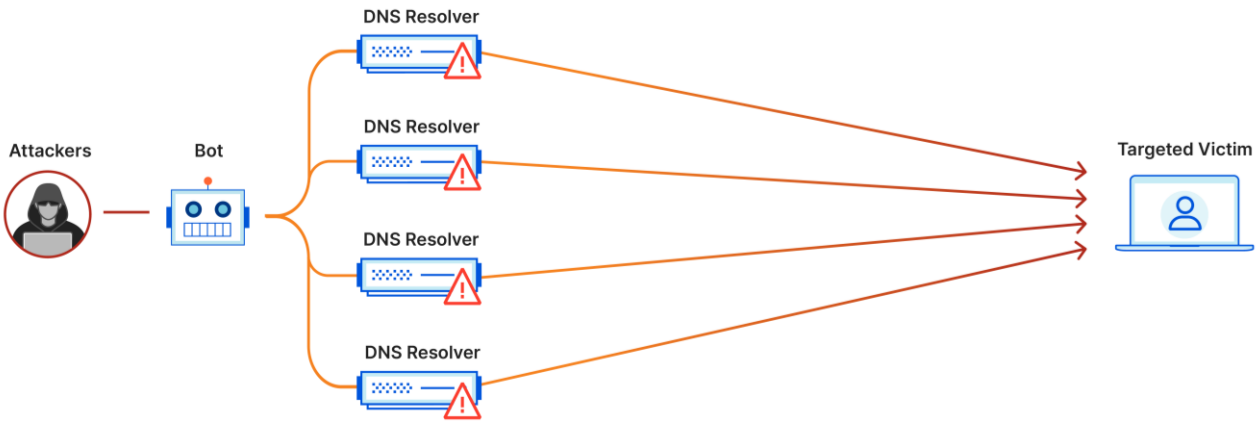

Figure 1: DDoS attack example [5]

3.2 Literature review

The combination of centralized computing systems and the Internet of Things has driven digital service expansion, yet it has created new vulnerabilities because attackers primarily target networks with DDoS attacks through traffic flooding. This part reviews current research about the utilization of ML and DL methods to detect DDoS traffic while considering changing attack methods and intelligent adversaries. [6]

3.2.1 Evolution of DDoS Attacks and Traditional Detection Methods

Volumetric attacks used to be the main type in DDoS operations until their evolution created complex application-layer and reflection-based forms of attack. Rate limiting together with deep packet inspection fails to detect the low-volume attacks which employ stealthy methods. Through ML based techniques analysts gain higher detection ability through their ability to learn from existing traffic data trends. The authors in [7] suggested using Decision Trees and Random Forest algorithms to detect DDoS traffic in order to establish proactive defense systems.

3.2.2 Software-Defined Networking (SDN) Frameworks and Smart City DDoS Detection

Smart city networks face security risks because they consist of complex structures linked by many connections with the potential to suffer from DDoS attacks. The authors introduced an SDN architecture design that merges AI solutions for detecting anomalies throughout smart city infrastructure networks. [8] The centralized control from SDN allows dynamic traffic management operations. An extension of this concept added ML-based defenses to cloud-based SDN systems for transport-layer and application-layer real-time attack identification. [9]

3.2.3 Anomaly Detection Techniques for DDoS Defense

This method detects unknown attacks through its ability to detect abnormal behavior patterns. This research team applied their methodology toward DNS authoritative servers to differentiate between regular and harmful traffic variations. [10] Anomaly detection received further research attention for IoT specifically with an emphasis on developing suitable lightweight ML techniques for low-resource environments. Early detection through network function features demonstrated its capabilities according to both [11] and [12].

3.2.4 IoT Vulnerabilities and ML Solutions

IoT devices remain highly susceptible to DDoS botnets owing to their poor security setup and minimal resources capabilities. The authors proposed an XGBoost-based method which achieved 99.993% detection accuracy for feature selection in the IoT environment according to their paper. [13] Chakraborty et al. [14] evaluated various ML models with LR, DT and RF among them but the study revealed that LR delivered optimal performance at a lower computational cost that fit constrained IoT networks.

3.2.5 DL for Complex DDoS Detection

The complex nature of stealthy attacks which imitate normal systems behavior makes DL models a preferable detection solution. Researchers Kumar et al. applied combination CNN and RNN models for detecting application-level attacks [15]. A research study built up-on DL technology inside SDN networks to show how the combination of ML with DL generates better attack recognition through lower instances of false identification [16]. The authors of [17] stressed the necessity of using ensemble models with DL for real-time processing of IoT attack detection data.

The research by Hameed et al. [18] analyzed the optimization of attack impact reduction through accelerated detection and precise analysis by integrating anomaly detection systems with ML into SDN and IoT environments.

3.2.6 Real-Time Detection and Application-Layer Attacks

A real-time detection of Slowloris attacks through Gradient Boosting and Random Forest ML models showed successful identification of legitimate and malicious traffic according to Wibowo et al. [19]. The researchers from Anley et al. [20] developed transfer learning as a solution to address the difficulties associated with real-time training and adaptable models which enabled threat detection speedup without requiring complete retraining.

3.2.7 Collaborative Detection and Cross-Network Defense

The research by Wagner et al. [21] developed a collaborative detection system based on Internet Exchange Points (IXPs) that serve as sensors to distribute traffic data between organizations. The detection using this method identifies amplification attacks and prevents large-scale traffic anomalies which individual networks would fail to detect.

3.2.8 AI and Ensemble Learning for DDoS Detection

Jaszcz proposed a cooperative learning system that integrates Decision Tree with SVM and k-NN to reach higher detection accuracy and minimize wrong alarm flags. The authors in [22] established a signature-anomaly hybrid detection system for protection of cloud environments from DDoS and brute-force attacks.

3.2.9 Efficient Detection in IoT with ML

A DL-based approach from Najafimehr [23] selects features through hybrid methods to operate with strong accuracy levels while needing low computational resources suitable for limited IoT settings. Kalutharage applied Decision Trees and ensemble models to IoT networks for reducing false positives while maintaining detection accuracy according to his research in [24].

3.2.10 Transfer Learning and Cross-Context Adaptation

The research conducted by Alshahrani [25] demonstrated that pre-trained models achieve effective new attack context adaptation through minimal retraining steps to detect un-known DDoS patterns.

3.2.11 The Future of ML in DDoS Detection

The authors Maksimović et al. [26] recommended integrating Triple Modular Redundancy (TMR) with ML-based detection methods to reduce false positive outcomes through the differentiation of actual traffic anomalies. Future innovative work in ML/DL must target three elements that enhance their abilities to provide real-time detection across diverse network settings.

3.3 Knowledge Gap.

That being the case, fundamental gaps persist when it comes to applying ML in DDoS detection. It should be recognized that the majority of works concern current, labelled datasets, while real traffic is much richer and more diverse. Little has been done to investigate how predictive models continue to learn usable patterns from emergent, unorganized, or noisy data types in real-time scenarios [1][2].

Furthermore, we will be testing multiple ML algorithms in order to search for the best algorithm which will aid in the detection of DDoS attacks. Additionally, there will be a criterion for the algorithms which are, firstly, accuracy of detecting DDoS attacks. Secondly, the precision of the detection of the algorithms. Thirdly, the speed at which the algorithm can identify a DDoS attack from other normal traffic.

Our work will make the algorithm learn from datasets to differentiate between normal traffic spikes and false positive from DDoS attacks. This will change the perception of choosing criteria of the accuracy, which will consider other factors such as: identifying

DDoS attacks from other normal traffic, additionally, this will benign traffic that could potentially be considered as a false positive in a different system.

Ref	Authors	Aim	Classifier	Accuracy	Strength	Weakness
[26]	Maksimović et al. (2024)	Improve thresholding in DDoS IDS	TMR, KNN	TMR 99.6%	Avoided over-voting; high accuracy	Scalability, execution time
[12]	Wibowo (2024)	Real-time Slowloris detection	GB, RF, SVM, NN	High (GB best)	High precision, scalable	Limited to Slowloris
[19]	Bitew et al. (2024)	Robust network defense	CNN, VGG19, DNN	VGG19 99.99%	Multiple datasets tested	Coordination complexity
[25]	Alshahrani (2023)	SDN framework for smart cities	XGBoost	99.99%	High accuracy	Implementation complexity
[16]	Yungaicela-Naula et al. (2023)	IoT DDoS detection	KNN, SVM	KNN 99.97%	Explainable model	Limited generalization
[7]	Islam et al. (2023)	DDoS detection using AI	ANN, DL	>96%	High detection accuracy	Data-specific applicability
[13]	Modi (2023)	IoT DDoS detection using XGBoost	XGBoost	99.99%	High accuracy, explainable	Dataset dependency
[17]	Al-Shareeda et al. (2023)	Compare ML & DL in IDS	SVM, DT	DT 99.93%	Effective anomaly-based IDS	No specific dataset
[19]	Kumar et al. (2023)	LSTM for DDoS detection	LSTM	NDAE 99.6%	High performance, automated features	Dataset quality dependent
[24]	Kalutharage et al. (2023)	IoT DDoS detection	XAI, RF	RF 98%	Explainable, accurate	Limited to specific attacks
[21]	Al-Shareeda et al. (2023)	Compare ML and DL in IDS	ML, DL	DT 99.93%	Detailed comparison, IDS effectiveness	No dataset provided
[15]	Kumar et al. (2023)	LSTM-based DDoS detection	LSTM	NDAE 99.6%	Automated feature learning	Relies on dataset quality
[28]	Kalutharage et al. (2023)	IoT DDoS identification	XAI, RF	RF 98%	High detection, explainability	Specific attack types

[23]	Najafimehr et al. (2023)	IoT/cloud DDoS detection	RF, SVM, CNN, LSTM	RF 99.9%	Effective, high accuracy	Limited deployment
[10]	Bushart & Rossow (2022)	DNS resilience to DDoS	Anomaly-based	FPR 1.2-5.7%	Effective filtering	Concept drift risk
[9]	Nadeem et al. (2022)	SDN protection from DDoS	Random Forest	99.97%	Highest accuracy among ML methods	Single model focus
[14]	Mondal et al. (2022)	IoT DDoS removal	LR, NB, SVM, DT, RF	RF 98%	High accuracy	System limitations
[22]	Jaszcz & Połap (2022)	AI-based hybrid DDoS detection	NN, k-NN	99.5%	Flexible, small dataset ready	Imbalance sensitivity
[18]	Ismail et al. (2022)	ML framework for DDoS classification	RF, XGBoost	XGBoost 90%	Modern datasets, good results	Real-world validation needed
[8]	Lee et al. (2022)	Autonomous IoT defense	CNN	99.5%	Effective DDoS distinction	Computational load
[11]	Alzahrani & Alzahrani (2021)	Evaluate ML algorithms via WEKA	DT, RF, etc.	DT, RF = 99%	Speed comparison	Dataset generalization
[20]	Wagner et al. (2021)	Collaborative DDoS mitigation via IXPs	N/A	Detected 80% more attacks	Distributed mitigation	Data quality reliance
[21]	Nadeem et al. (2021)	Cloud IDPS framework	RF, CatBoost	99.99%	Robust IDS framework	Lacks dataset details

*Table 1: summarize of **literature review***

4. Methodology

Several proposed ML techniques are used in our paper to detect DDoS attacks successfully. The algorithms considered are Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (K-NN), Artificial Neural Network (ANN). A quantitative comparison of each technique is made and based on the results of the technique with respect to accuracy, precision, recall etc. The objective is to find the model with the best detection performance, the highest level of accuracy and reduce the false positive alarms during the detection duration.

For this work, the dataset used is network traffic which is collected from public datasets (CICDDoS2019) and other emulated environments to achieve all-weather Benign and Attack traffic. Most of the time, data preprocessing is done very orderly in a way to enhance the quality of the data fed into the models. It entails data cleaning which involves deletion of noises and irrelevant attributes, numerical attributes scaling and normalization, transformation if necessary. The obtained data is then randomly divided into training and testing datasets in a 4:1 ratio as a simplified application of the k-fold cross-validation technique to avoid the problem of overtraining.

After such an impressive campaign has been developed and optimized, it is tested for its efficiency in a validation process. The model is then used in real-time traffic analysis in our work architecture as mentioned above. New data is used to make predictions, and Dynamic Analysis is constantly in feedback and adaptive mode checking on its performance against new interfering signals and or new attack patterns etc. The results of different models are compared with respect to these pre-defined objective criteria to determine which model should be incorporated henceforth. The last one is predetermined by its capacity to retain a high rate of detections while still sliding low through the detection of false positives, thus making a reliable DDoS detection and proper solutions to it.

4.1. Data Collection

The research obtained network traffic data from CICDDoS2019 public repositories. The traffic data was classified into two alternative categories which included benign or DDoS attack activity.

4.2 Feature Extraction and Preprocessing, Key phases [30]:

- Preprocessing: Removal of noise, packet anomalies, and normalization.
- The process of noise removal eliminates both duplicate packets and those with missing information.
- The scale standardization process uses NumPy or Pandas to transform features.
- Irrelevant Feature Removal: Dropping unhelpful attributes.
- Dimensionality reduction through PCA constitutes the reduction step for features in the methodology.
- The clustering method enables detection of patterns between attributes by grouping similar features and it removes redundancy between corresponding features.
- The evaluation process of ranking important features happens through Feature Importance Analysis which determines the significance of attributes by assessing their appearance frequency and scarcity.

4.3 Classification Techniques

A variety of classifiers namely NB, SVM, RF, DT, K-NN and Artificial Neural Networks (ANN) were applied after the preprocessing stage.

4.4 Model Evaluation

The metrics for model evaluation included accuracy, precision, recall, F1-score and confusion matrix. The true/false positive/negative evaluation rates help identify malicious traffic among benign traffic.

4.5 System Design

4.5.1 Architecture: The system uses a modular design which enables different ML models to undergo evaluation tests. Components include:

- The Data Collection Module performs data processing duties for raw data normalization.
- The Model Evaluation Engine performs both training operations and model evaluation procedures.
- Through the Visualization and Reporting Module the system produces both performance reports and visualizations.

4.6 Workflow [27]

- Data Processing: Cleaning, scaling, encoding.
- The testing process starts with dividing the dataset into three parts for training together with validation and testing segments.
- The system performs comparison and analysis through its plotting and analytical functions.
- Documentation: Recording findings, stakeholder reporting.

4.7 Hardware and Software Requirements

- Hardware: 8GB RAM, i5 processor, SSD storage. Optional cloud computing.
- The software selection includes Python version Pycharm data analysis through Pandas with NumPy alongside Scikit-learn, TensorFlow, Matplotlib and Seaborn.

5. Implementation, Result and Key Findings

This section details step-by-step how the proposed machine learning-based DDoS detection model should be implemented. The methodology section selects ML algorithms to apply them to a specific dataset which was chosen by careful research and was done by literature review. The implementation involves traditional machine learning processing by acquiring data while performing preprocessing actions before choosing models for optimization and evaluation

5.1 Testing Plan

In the testing plan stage, we test our model in order to see how well our model is and is there room for improvement. Performance metrics such as Accuracy, Recall, Precision and F1-Score as well as the reduction of false positives are the core data that will indicate how well the model is, furthermore this will aid in the optimization step and improvement of the model which will indicate the room for improvements. The targeted

Accuracy would be more than 94% to be qualified as good accuracy, any percentage under the 94% margin will show that the model is insufficient and lacks necessary adjustments. For the Recall and Precision and F1-Score are also as same as the Accuracy to be precise. The model should be able to differentiate between false positives and real DDoS attacks, hence the necessity for the false positive percentage being very close or 0% to be credible and authentic.

5.2 Implementation Progress

In this section, the process of implementing this paper is discussed. The implementation process can be represented in three parts: Data collection, Data Preprocessing, Feature engineering, and Generating Models.

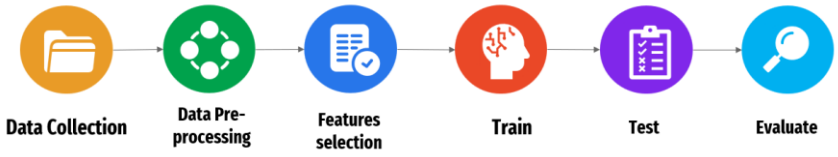

Figure 2: Machine Learning Pipeline [27]

5.2.1 Data collection

A comprehensive data collection process forms the foundation of our work since it determines how well our machine learning models will function based on the quality of the dataset selected. Our main task focused on obtaining research data representing genuine DDoS attacks which also holds a validated status and enjoys widespread research usage.

Our team conducted a comprehensive review through Google Scholar for research papers which attained high citation stats among respected members of cybersecurity and machine learning fields. The selection process focused on studies that made available datasets which numerous research works have utilized along with providing references for ensuring reliable and consistent analysis. Our selection of the CICDDoS2019 dataset followed a comprehensive review process because researchers commonly use it for DDoS detection research.

5.2.1.1 Data description

CICDDoS2019 includes both benign and contemporary standard DDoS attacks that closely represent actual world PCAPs. The analysis leads include time stamps along with source IPs and destination IPs and source and destination ports and protocols and attack types supplied using CICFlowMeter-V3 and CSV files. [31]

As their main focus they dedicated efforts toward creating background traffic that resembled realistic scenarios. The B-Profile system developed by Sharafaldin et al. (2016) has generated naturalistic benign background traffic within the proposed testbed (Figure 2) by profiling abstract human behavior through its method (Sharafaldin, et al. 2016). The abstract user behavior of 25 individuals was constructed by utilizing HTTP, HTTPS, FTP, SSH and email protocols. [31]

5.2.1.1.1 Number of Attributes & Feature Description

The dataset consists of more than 80 various attributes which originated from network traffic log analysis. The features in the dataset fall into three categories which

comprise packet-based characteristics together with flow-based indicators and statistical properties for detecting normal and malicious activities.

Key Feature attributes:

1. Basic Flow Features.
2. Time-Based Features.
3. Packet Size Features.
4. Flow Behavior & Statistical Features.
5. Header & Flag-Based Features.
6. Label Attribute.

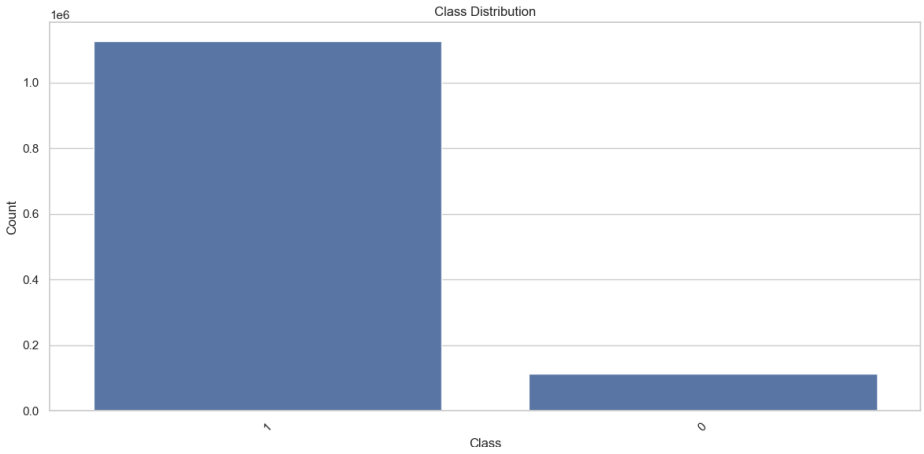

Figure 3: Bar chartr for Label class before balnce the data

The distribution of classes in the dataset appears in the bar chart which separates DDoS attack traffic into group 0 while normal traffic falls under group 1. The data set displays an extreme imbalance through the bar chart because normal traffic samples significantly outnumber DDoS attack samples. The data consists of 1.1 million normal traffic records which represent the majority class but only a small number of DDoS attack records make up the minority class. Model training operations will face difficulties because of data imbalance in the dataset which produces machine learning algorithms to favor predictions for the prevalent class thus decreasing detection performance for actual attack conditions. The successful development of an effective DDoS detection model requires solving the existing class imbalance problem

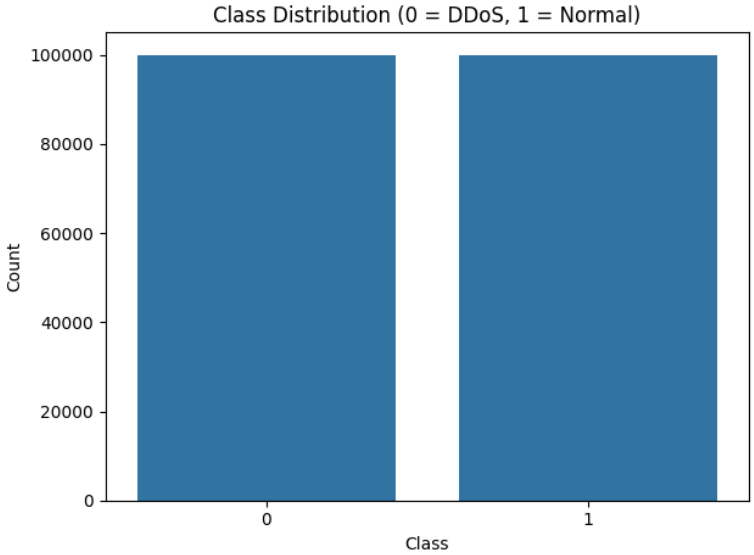

Figure 4: Balnces label each class have 100,000 rows

5.2.2 Pre-processing:

Data pre-processing is a critical step that must be applied in machine learning before training the models on the dataset to get the best results, and it contains four stages which are data Integration, data cleaning, data reduction, data transformation [6]

5.2.2.1 Data Integration

We discovered that the CICDDoS2019 dataset contained nineteen Excel files when we downloaded it which held network traffic records. We integrated all files into one dataset through data processing to optimize model training operations.

The merged information expanded to reach a 50GB size with more than 50 million records of network traffic data. The data integration combines all critical attack and normal traffic records to establish a unified dataset for extensive training process analysis.

5.2.2.2 Data cleaning

The following step involved cleaning the CICDDoS2019 dataset which integrated into a single file for improving its quality and consistency. We applied the following cleaning approaches on the dataset since it contained inconsistent information

- Removing Duplicates: We deleted duplicate records to control model bias through removal of redundant information.
- Handling Missing Values: We eliminated both missing and incomplete data from rows to keep the dataset free from inaccuracies and unreliable data points.
- Removing Empty Rows: We eliminated all completely empty rows through this process which helped minimize unnecessary computing costs and maintain data validity.

5.3.2.3 Data reduction

We performed data reduction through random sampling of 1 million records from the 50 million-row dataset and maintained class balance for the data. Data sampling of one million records enabled us to handle a reduced dataset while retaining important information.

5.2.2.4 Data transformation

Our procedure converted the two-class nominal targets into numeric format with 0 and 1 values to enable model training in machine learning.

Additionally, Excel performed an automatic conversion of big numbers to scientific notation while maintaining the unchanged actual value quantities. Data transformation enables more effective handling of substantial numeric data sets

5.2.3 Feature engineering

Feature Engineering is the process of improving ML model performance that involves selecting and transforming existing data features along with generating new features from the data. The practice involves analyzing the domain together with data and the specific problem statement to build features which help models identify critical data relationships. This process includes two fundamental methods: first selecting specific features while the second method extracts new features from existing data. [5] [6]

5.2.3.1 Feature selection:

We performed feature selection by eliminating Source IP, Destination IP as well as Similarity HTTPS from our dataset. The model excluded these features since they either provided irrelevant information or they were redundant or introduced unnecessary bias.

5.2.3.2 Feature extraction:

No feature extraction techniques were applied in our approach, as we did not create new features or transform existing ones into different representations.

5.3 Building Machine Learning models:

After applying the pre-processing phase, building the models phase started. To gain the best results and performance, several models were bult using different Machine Learning algorithms. The results of each model recorded first before the pre-processing phase. After that, they trained with the processed data. The performance of each model measured with Accuracy, Precision, Recall, F1 score, and False positive rate.

We divided the extensive dataset into various chunks while each portion contained 200,000 rows. We adopted this data partitioning method for maximum computational efficiency on our available hardware platform to achieve better execution speed. Data was partitioned so 70% served for training with the remaining 30% reserved for testing purposes.

5.3.1 Naïve Bayes

The Naive Bayesian (NB) classifier implements Bayes' theorem for object classification while NB identifies available knowledge to determine specific outcome probabilities because its Independence Assumptions lead the algorithm to view all features and attributes apart from each other. NB presents an easy-building model which performs better with big datasets compared to various sophisticated algorithms that Weka offers a Naïve Bayesin tool for implementing NB. [28]

5.3.2 Random Forest

The automated Random Forest system functions as a very strong technique. The model reaches high levels of success through the utilization of a small dataset. NB can solve both regression and classification issues as part of its operation. The classification problems will perform feature selection through information gain or gain ratio indexes or Gini index methods. The algorithm selects the major voting class among its options. [32] [33]

5.3.3 Decision Tree

As a supervised learning method the Decision Tree (DT) algorithm operates through tree-like structure nodes where each decision depends on feature values. The algorithm divides data incrementally into subsegments through Gini Index or Information Gain or Gain Ratio evaluation which stops when it obtains the final classification result. At the top of the tree structure the root functions as the most important variable whereas the final endings or leaves display the predicted output classes. Due to their interpreter-friendly arrangement DTs provide usable visual displays which help both feature selection and classification efforts. The algorithm has difficulties with overfitting when excessive dimensions make up the tree yet lack appropriate pruning techniques. The algorithm finds its main applications in medicine to diagnose patients as well as credit assessment and the detection of intrusions within systems. [32][33]

5.3.4 KNN

The k-Nearest Neighbors (KNN) algorithm functions as a basic instance-based learning approach which performs classification tasks as well as regression activities. The algorithm selects k nearest data points from an input while assigning the majority class label from this selection. KNN demonstrates non-parametric characteristics which enable it to handle diverse data distributions though its operation becomes slower when processing big data collections. [29] [33]

5.3.5 ANN

The computational design of Artificial Neural Networks (ANNs) derives from the structure of human brain neurons. Each Artificial Neural Network (ANN) uses connected nodes (neurons) to process data within weighted connection paths between nodes. ANNs serve as standard tools for solving difficult pattern recognition problems including image processing and speech recognition together with anomaly detection. The implementation of ANN-based classification through Multilayer Perceptron (MLP) which exists in Weka forms the main approach of our study. [28] [23]

5.3.6 SVM

Support Vector Machine provides advanced supervised learning capabilities that enable classification as well as regression operations. SVM uses an algorithm that identifies the best hyperplane which creates the largest margin between dataset classes. SVM succeeds particularly well in dimensions with many variables and operates effectively on data points which cannot be separated by simple lines through kernel functions including polynomial and radial basis function (RBF). The SVM algorithm finds uses in three domains: bioinformatics research, text classification analysis and fraud detection. [28] [29]

5.4 Result and Finding

After building our model, we trained and tested the model with various ML techniques to extract the performance metrics of the model which consists of: Accuracy, Precision, Recall, F1 - Score and False Positive rate. Data preprocessing and feature engineering were made on the CICDDoS2019 dataset:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**FPR**
NB	0.951	0.921	0.992	0.955	0.854
DT	0.991	0.987	0.997	0.995	0.141
RF	0.984	0.986	0.996	0.991	0.145
KNN	0.891	0.895	0.939	0.896	0.157
ANN	0.941	0.946	0.934	0.940	0.053
SVM	0.988	0.992	0.995	0.994	0.081

Table 2: the result after run our models

5.5 Discussion

The implementation of different machine learning algorithms helped achieve enhanced detection performance for DDoS attacks after applying strong preprocessor methods and creating engineered features.

The study confirms that proper data handling techniques result in better performance of ML models for identifying benign and malicious traffic. When the model was created, the main objective of it is to detect DDoS attacks with high accuracy, precision, recall, F1 – score and finally reduced false positive rate. Many ML techniques can perform either poorly or greatly depending on the labels in the dataset. With a little improvement to the labels in a dataset the performance could enhance greatly, which will be shown in this section.

After data preprocessing and feature engineering and the optimization techniques, the ML techniques with the highest performance metrics were Decision Tree, Support Machine Vector and Random Forest. Decision Tree had a very low accuracy for starters due to an unbalanced dataset with many labels that could possibly distract the ML techniques to perform in its best possible way, the accuracy of Decision Tree became 99% which increased significantly after the optimization process along with the other two ML techniques which also had a 98.8% for the SVM and 98.8% for the Random Forest. The optimization also increased the other performance metrics significantly.

We compared our model to an existing paper that showed that the KNN got an accuracy of 89% and the ANN got an accuracy of 93.6%, while our model enhanced the accuracy figure and provided improved results. KNN accuracy was 89.1% and the ANN improved to 94.1% with just optimization for the dataset.

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.89	0.89	0.91	0.92
ANN	0.93	0.94	0.93	0.94

Table 3: Result of the research paper that we are compared with [15]

6. Conclusion

We performed a complete examination of machine learning algorithms to determine their ability for detecting Distributed Denial of Service (DDoS) attacks within this research. The necessity for intelligent adaptive detection methods in cybersecurity has surged because DDoS attacks keep becoming more complex in traditional and emerging digital infrastructures. The research constructed the framework consisting of various supervised ML models which achieve accurate DDoS pattern detection with minimal wrong alert activations from network traffic analysis.

The research paper had a single defined purpose which was to measure machine learning algorithms' capacity for detecting DDoS attacks from labeled network traffic datasets. The CICDDoS2019 dataset served our needs because it delivered exceptional realism and academic validity as well as detailed dataset features. The dataset included equal numbers of records from benign traffic and attack sources which made it an ideal choice for detecting security threats under real-world simulation.

Multiple crucial phases composed the research process beginning with information acquisition followed by information merging before handling data. Our model training became efficient after we processed raw traffic logs through thorough data cleaning and noise reduction followed by transformation procedures. Model generalizability increased significantly after feature selection executed necessary removal of attributes that were

unneeded or duplicative and irrelevant. Clean data processing methods transformed 50 million raw records into one million balanced data rows which became optimal for performing thorough experiments.

Our experiments utilized six machine learning classifiers namely Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) as well as Artificial Neural Networks (ANN). The trained models received performance evaluation through accuracy and precision metrics as well as recall and F1-score and false positive rate (FPR).

The first assessments of raw data produced mediocre results mainly due to elevated FPR and limited accuracy in differentiating hostile from reputable traffic. The ML pipeline's complete implementation starting from preprocessing and feature engineering yielded substantial enhancement of model performance. The Decision Tree model outperformed other models by attaining a 99% accuracy level and a 99.5% F1-score while Random Forest and SVM achieved close to identical results that reduced incorrect detection rates.

The findings reveal how important it is to maintain excellent data quality during preprocessing before machine learning applications because poor results can waste resources and cause security threats to be missed particularly in cybersecurity. Simple Naïve Bayes models demonstrated successful performance when researchers applied the models to properly prepared data which proves that model complexity is not the essential factor for effective solutions.

Research findings received backup from the literature review which showed ML and DL approaches growing in popularity for DDoS detection. The analyzed studies in the review established how ensemble models along with anomaly detection methods and hybrid methodologies perform effectively in IoT and cloud-based systems. Studies integrating machine learning with Software-Defined Networking (SDN) and transfer learning techniques produced positive findings that provided understanding for creating adaptive defense systems with real-time detection capabilities.

Finally, the research showed that present methods face prolonged understanding deficits. Existing research mainly depends on labeled datasets that stay fixed while enclosing little investigation of dynamic learning environments that can evolve in real time. Few research investigations focus on enabling their models to learn from continuous streaming or unlabeled traffic data and to detect different evolving attack types during minimal resets.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research was funded by the SAUDI ARAMCO Cybersecurity Chair at the College of Computer Science and Information Technology (CCSIT), Imam Abdulrahman Bin Faisal University (IAU), Dammam, Kingdom of Saudi Arabia.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Abbreviations	Definition
AI	Artificial intelligence
DDoS	Distributed Denial of Service
DOS	Denial of Service
ML	Machine learning
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
SYN	Synchronize
ICMP	Internet Control Message Protocol
OSI	Open Systems Interconnection
IPS	Intrusion Prevention Systems
SDN	Software-Defined Networking
DL	Deep Learning
TMR	Triple Modular Redundancy
IoT	Internet of Things
TD	True Positive Detection
FP	False Positive
IXP	Internet Exchange Point
SVM	Support Vector Machine
k-NN	k-Nearest Neighbors
RNN	Recurrent Neural Network
SDN	Software-Defined Networking
CNN	Convolutional Neural Network
DFS	Decision Feedback System
NFs	Network Functions
DNS	Domain Name System
AIMM	AI-based Intelligent Mitigation Model
IDS	Intrusion Detection System
RF	Random Forest
DT	Decision Tree
NB	Naive Bayes
IDPS	Intrusion Detection and Prevention System
IPFIX	Internet Protocol Flow Information Export
XAI	Explainable AI
LSTM	Long Short-Term Memory
TCP	Transmission Control Protocol
CIC	Canadian Institute for Cybersecurity
PCA	Principal Component Analysis
RFE	Recursive Feature Elimination

Table 4: Abbreviations

References

[1] H. Wang and Y. Li, "Overview of DDoS Attack Detection in Software-Defined Networks," *IEEE Access*, vol. 12, pp. 38351-38362, Mar. 2024. doi: 10.1109/ACCESS.2024.3375395.

[2] Z. Mahdi, N. Abdalhussien, N. Mahmood, and R. Zaki, "Detection of Real-Time Distributed Denial-of-Service (DDoS) Attacks on Internet of Things (IoT) Networks Using Machine Learning Algorithms," *Computer. Mater. Contin.*, vol. 80, no. 2, pp. 2139-2151, Aug. 2024. doi: 10.32604/cmc.2024.053542.

[3] C. Singh and A. K. Jain, "A Comprehensive Survey on DDoS Attacks Detection & Mitigation in SDN-IoT Network," *e-Prime - Adv. Electron. Eng. Electron. Energy*, vol. 8, pp. 100543, Apr. 2024. doi: 10.1016/j.prime.2024.100543.

[4] M. E. Manaa, S. M. Hussain, S. A. Alasadi, and H. A. A. Al-Khamees, "DDoS Attacks Detection Based on Machine Learning Algorithms in IoT Environments," *Inteligencia Artificial*, vol. 27, no. 74, pp. 152-165, Apr. 2024. doi: 10.4114/intartif.vol27iss74pp152-165.

[5] Cloudflare, "What is a DDoS attack? | DDoS meaning, types & protection," *Cloudflare*, [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/. [Accessed: Apr. 2, 2025].

[6] P. Reddy, Y. Adetuwo, and A. K. Jakkani, "Implementation of Machine Learning Techniques for Cloud Security in Detection of DDoS Attacks," *Int. J. Comput. Eng. Technol.*, vol. 15, no. 2, pp. 25-34, Apr. 2024.

[7] Islam, T., Jabiullah, M.I., & Abid, M.H. "DDoS Attack Preventing and Detection with the Artificial Intelligence Approach." Conference Proceedings, 2023, pp. 1-12. doi:10.1007/978-3-030-98457-1_3.

[8] Lee, S.-H., Shiue, Y.-L., Cheng, C.-H., Li, Y.-H., & Huang, Y.-F. "Detection and Prevention of DDoS Attacks on the IoT." Applied Sciences, vol. 12, no. 12407, 2022. doi:10.3390/app122312407.

[9] Nadeem, M.W., Goh, H.G., Ponnusamy, V., & Aun, Y. "DDoS Detection in SDN Using Machine Learning Techniques." Computers, Materials & Continua, vol. 71, no. 1, 2022, pp. 772-791. doi:10.32604/cmc.2022.021669.

[10] Bushart, J., & Rossow, C. "Anomaly-based Filtering of Application-Layer DDoS Against DNS Authoritatives." Conference on Network Security, 2021, pp. 1-15.

[11] Alzahrani, R.J., & Alzahrani, A. "Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic." Electronics, vol. 10, no. 2919, 2021. doi:10.3390/electronics10232919.

[12] Wibowo, M.H.S., Al Ayubi, M.D., & Hardiansyah, N.A. "Real-Time Detection and Prevention of Slowloris DDoS Attacks Using Machine Learning." International Journal of Informatics and Communication Technology, vol. 12, 2024, pp. 35-36. doi:10.11591/ijict.v12i1.

[13] Modi, P. "Towards Efficient Machine Learning Method for IoT DDoS Attack Detection." International Journal of Computer Science and Information Technology, 2023, pp. 1-10.

[14] Chakraborty, M., & Gupta, S. "Detection and Investigation of DDoS Attacks in Network Traffic Using Machine Learning Algorithms." International Journal of Innovative Technology and Exploring Engineering, vol. 11, no. 5, 2022. doi:10.35940/ijitee.F9862.0511622.

[15] Kumar, D., Pateriya, R.K., Gupta, R., Dehalwar, V., & Sharma, A. "DDoS Detection Using Deep Learning." Procedia Computer Science, vol. 218, 2023, pp. 2420-2429. doi:10.1016/j.procs.2023.01.217.

[16] Yungaicela-Naula, N.M., Vargas-Rosales, C., & Perez-Diaz, J.A. "SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning." IEEE Access, vol. 9, 2021, pp. 108495-108509. doi:10.1109/ACCESS.2021.3101650.

[17] Al-Shareeda, M.A., Manickam, S., & Saare, M.A. "DDoS Attacks Detection Using Machine Learning and Deep Learning Techniques: Analysis and Comparison." Bulletin of Electrical Engineering and Informatics, vol. 12, no. 2, 2023, pp. 930-939. doi:10.11591/eei.v12i2.4466.

[18] Ismail, M., Mohmand, M.I., Hussain, H., Khan, A.A., Ullah, U., Zakarya, M., Ahmed, A., Raza, M., Rahman, I.U., & Haleem, M. "A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks." IEEE Access, vol. 10, 2022, pp. 21443-21457. doi:10.1109/ACCESS.2022.3152577.

[19] Anley, M.B., Genovese, A., Agostinello, D., & Piuri, V. "Robust DDoS Attack Detection With Adaptive Transfer Learning." Computers & Security, vol. 144, 2024, pp. 103962. doi:10.1016/j.cose.2024.103962.
[20] Wagner, D., Kopp, D., Wichtlhuber, M., Dietzel, C., Hohlfeld, O., Smaragdakis, G., & Feldmann, A. "United We Stand: Collaborative Detection and Mitigation of Amplification DDoS Attacks at Scale." Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 970-988. doi:10.1145/3460120.3485385.
[21] Nadeem, M., Arshad, A., Riaz, S., Band, S.S., & Mosavi, A. "Intercept the Cloud Network From Brute Force and DDoS Attacks via Intrusion Detection and Prevention System." IEEE Access, vol. 9, 2021, pp. 152300-152314. doi:10.1109/ACCESS.2021.3126535.
[22] Jaszcz, A., & Połap, D. "AIMM: Artificial Intelligence Merged Methods for Flood DDoS Attacks Detection." Journal of King Saud University – Computer and Information Sciences, vol. 34, 2022, pp. 8090-8101. doi:10.1016/j.jksuci.2022.07.021.
[23] Najafimehr, M., Zarifzadeh, S., & Mostafavi, S. "DDoS Attacks and Machine-Learning-Based Detection Methods: A Survey and Taxonomy." Engineering Reports, vol. 5, 2023. doi:10.1002/eng2.12697.
[24] Kalutharage, C.S., Liu, X., Chrysoulas, C., Pitropakis, N., & Papadopoulos, P. "Explainable AI-Based DDOS Attack Identification Method for IoT Networks." Computers, vol. 12, no. 32, 2023, pp. 1-16. doi:10.3390/computers12020032.
[25] Alshahrani, M.M. "A Secure and Intelligent Software-Defined Networking Framework for Future Smart Cities to Prevent DDoS Attack." Applied Sciences, vol. 13, no. 9822, 2023. doi:10.3390/app13179822.
[26] Maksimović, A.N., Nikolić, V.R., Vidojević, D.V., Randjelović, M.D., Djukanović, S.M., & Randjelović, D.M. "Using Triple Modular Redundancy for Threshold Determination in DDOS Intrusion Detection Systems." IEEE Access, vol. 12, 2024, pp. 53785-53795. doi:10.1109/ACCESS.2024.3384380.
[27] Bishop, C. M. Neural Networks for Pattern Recognition. New York: Oxford University Press (1995). (This book offers a good coverage of neural networks)
[28] B. Mahesh, "Machine Learning Algorithms—A Review," Int. J. Sci. Res. (IJSR), vol. 9, no. 1, pp. 381-386, Jan. 2020. doi: 10.21275/ART20203995.
[29] T. O. Ayodele, "Types of Machine Learning Algorithms," in New Advances in Machine Learning, IntechOpen, 2010. doi: 10.5772/9385.
[30] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big Data Preprocessing: Methods and Prospects," Big Data Analytics, vol. 1, no. 9, 2016. doi: 10.1186/s41044-016-0014-0.
[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2019, pp. 53–64. doi: 10.5220/0006639805780585.
[32] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS Detection using Deep Learning," Procedia Computer Science, vol. 218, pp. 2420–2429, 2023. doi: 10.1016/j.procs.2023.01.217.
[33] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning," IEEE Access, vol. 9, pp. 108495–108510, 2021. doi: 10.1109/ACCESS.2021.3101650.