SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I10P102

Volume 10 Issue 10, 6-19, October 2023

Original Article

Mitigation of HTTP Flood DDoS Attack in Application Layer Using Machine Learning and Isolation Forest

P. Krishna Kishore[1], S. Ramamoorthy[2], V.N. Rajavarman[3]

[1,2,3]*Department of CSE, Dr. M.G.R Educational and Research Institute, Chennai, Tamilnadu, India.*

[1]*Corresponding Author : krishna.boinapalli@gmail.com*

Received: 02 August 2023 Revised: 05 September 2023 Accepted: 03 October 2023 Published: 31 October 2023

Abstract *- Distributed Denial of Service (DDoS) attacks, specifically HTTP flood DDoS attacks, have become a constant and substantial threat to online companies and critical services due to the growing popularity of web-based applications and technology. HTTP flood DDoS attacks inundate web servers with an overwhelming volume of seemingly legitimate HTTP requests emanating from compromised devices or botnets. Traditional DDoS mitigation approaches, often reliant on rate limiting and traffic filtering, struggle to discern between legitimate and malicious traffic, leading to service degradation or downtime. Methods for identifying abnormal HTTP traffic behaviour involve gathering and preprocessing data, generating features, and developing Isolation Forest algorithms. The power of this method comes from its ability to detect anomalies in real-time, making it easy to identify and block HTTP flood DDoS attack traffic. As such, this is a significant feature of the methodology. In tandem with Isolation Forest, machine learning empowers the system to adapt proactively to emerging attack vectors, enhancing its resilience in the face of evolving threats. This research presents a novel approach to fortify the application layer against HTTP flood DDoS attacks by utilizing machine learning techniques, with a central focus on the Isolation Forest algorithm. The experimental validation results show that the proposed framework can effectively recognize and mitigate HTTP flood DDoS attacks with minimal service interruption and false positives. The tests were run on benchmark datasets from the KDD Cup 1999 and the NSL-KDD, and the results stated here enhance the basis for the proposed model and enable the research to achieve its objective.*

Keywords *- Distributed Denial of Service (DDoS) attacks, HTTP flood DDoS attack, Botnet, Machine learning, Isolation Forest algorithm.*

1. Introduction

The proliferation of web-based applications and services has undoubtedly transformed how we interact with information, conduct commerce, and communicate in the digital age. While this digital revolution has brought unprecedented convenience and connectivity, it has also given rise to a persistent and pernicious threat: DDoS attacks, short for "Distributed Denial of Service," are growing in popularity. With its ability to turn off websites, compromise vital services, and cause damage in cyberspace, the HTTP flood DDoS attack has emerged as an effective competitor. The HTTP flood DDoS attack is one of several distributed denial of service attacks, but it has proven especially risky in the past few years. Its sheer simplicity and devastating impact characterize the HTTP flood DDoS attack. A malevolent actor orchestrates a deluge of seemingly legitimate HTTP requests towards a targeted web server in this attack. The volume and intensity of these requests quickly overwhelm the server's resources, leading to service degradation or outright unavailability. These attacks are incredibly deceptive because they can appear as legitimate user actions. Since this allows attackers to appear as legitimate users, it is more challenging for standard safety methods to distinguish between valid and malicious requests.

Denial of service attacks, often known as DoS attacks, occur when attackers collaborate to make it difficult for legitimate users to obtain accessibility to a service or resource. This could be a Distributed Denial of Service (DDoS) attack, in which servers respond more slowly than usual to client requests or decline to do so. Today's digital environment faces a growing threat from Distributed Denial of Service attacks (DDoS). An attacker typically arranges such attacks by controlling a botnet, a group of compromised computers working together. The primary goal of such an attack is to reduce the server's resources, such as its processing power, input/output bandwidth, sockets, and memory. Because of this, regular users and customers have far less access to, or none at all in some situations, to the available resources. Recent DDoS attacks have targeted various victims [1, 2], while strategies for effectively mitigating these attacks have been explored in [3]. These

challenges underscore the critical importance of developing robust defence mechanisms to safeguard online services and ensure uninterrupted accessibility for legitimate users. A distributed denial of service attack is shown in the scenario represented in Figure 1.

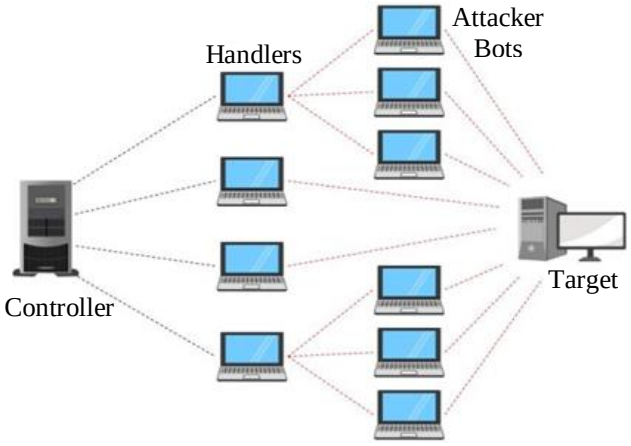

Fig. 1 An environment of DDoS attack

Service providers' predominant strategy to counter the App-DDoS threat revolves around bandwidth usage limitation. However, this approach presents significant limitations, primarily because the required bandwidth scales with the payload size [4]. Consequently, constraining bandwidth usage proves far from being an optimal solution [5]. Not only does it potentially disrupt regular traffic flow, but it can also adversely affect server performance, especially during peak periods with a surge in genuine user activity.

This shows the pressing need for a more robust app-DDoS defence approach that doesn't rely on bandwidth limits. Recent developments have highlighted the seriousness of the issue. Cloudflare's study indicated that ransomware DDoS attacks rose by about a third between 2020 and 2021, with a particularly significant increase of 75% in the fourth quarter of 2021 [6]. Furthermore, application-layer distributed denial-of-service attacks increased by a shocking 641% from the previous quarter, primarily affecting the construction segment. Both Yandex and Qrator Labs have reported that the attackers' purpose was to block users from accessing the websites of commercial and non-profit organizations [7].

DDoS attackers are getting more proficient and innovative with each new attack, using methods like multi-vector attacks, which further add to the complexity and difficulties of the issue. Multi-vector DDoS operations, which incorporate numerous forms of DDoS attacks into an integrated attack move, are currently the preferred method of attack by most DDoS attackers [8]. This evolving threat landscape necessitates innovative and adaptive defence mechanisms that can effectively counter the growing complexity and scale of App-DDoS attacks, thereby safeguarding the availability and performance of digital services for legitimate users.

According to the sources [9] and [10], Distributed Denial of Service (DDoS) attacks can be classified into subclasses based on several distinguishing characteristics. One fundamental classification criterion is the examination of the network protocol stack involved in the attack. Consequently, DDoS attacks are categorized into two primary types: Network/Transport level DDoS attacks and Application level DDoS attacks.

Network/Transport level DDoS attacks: These attacks target the protocol stack's network and transport levels because they are faster to compromise. They involve overwhelming the target system with excessive traffic or exploiting vulnerabilities at these levels.

Network/Transport level DDoS attacks often employ SYN floods, UDP reflection attacks, and ICMP floods to flood network resources, disrupt network connectivity, and consume available bandwidth. Mitigating such attacks usually requires network-level defences and traffic filtering strategies.

Application level DDoS attacks: While traditional DDoS attacks focus on lower layers of the network protocol stack, application-level DDoS attacks target directly at the application layer. These attacks aim to exhaust application resources, disrupt service availability, or exploit vulnerabilities in the application itself. Application-level DDoS attacks are more sophisticated and nuanced, often mimicking legitimate user behaviour to evade detection.

Standard techniques used in these attacks include HTTP floods, Slowloris attacks, and Layer 7 attacks, which directly target web applications. Defending against application level DDoS attacks typically necessitates using Web Application Firewalls (WAFs) and application-layer security measures.

This classification based on the network protocol stack assists in understanding the nature and scope of DDoS attacks, enabling organizations to tailor their defence strategies accordingly. Combating DDoS attacks effectively often requires a multifaceted approach encompassing both network-level and application-level defences to mitigate the diverse threats these attacks pose.

The Isolation Forest algorithm is primarily designed for anomaly detection and is not specifically tailored to detect flooding attacks. However, you can use the Isolation Forest algorithm to detect anomalies, including certain types of flooding attacks, in your dataset. Here are some common flooding attack scenarios that can be addressed with Isolation Forest [11-14]:

1. HTTP flood attacks: Isolation Forest can detect anomalies in web server logs, including HTTP flood attacks. By analyzing the patterns of incoming HTTP requests, you can identify unusual spikes or patterns that may indicate an attack.
2. Network traffic floods: Isolation Forest can also evaluate network traffic data to detect flooding attacks, such as SYN and UDP floods. Unusually high traffic rates or patterns can be flagged as anomalies.
3. Database query floods: In database management systems, flooding attacks can manifest as excessive queries or connections. Isolation Forest can help detect abnormal query patterns indicating an attack on the database.
4. Resource consumption floods: Some flooding attacks aim to consume server resources, such as CPU or memory. Isolation Forest can monitor resource utilization and detect unusual spikes from flooding attacks.
5. IoT device floods: Internet of Things (IoT) framework analysis reveals flooding attacks can target IoT devices with excessive traffic or requests. Isolation Forest can help identify abnormal device behaviour.

The primary objective of this research is to improve defences against Distributed Denial-of-Service (DDoS) attacks using the HTTP protocol by expanding the use of machine learning and, especially, the Isolation Forest algorithm. The paper is divided into the following parts: In Section 1, we provide context; in Section 2, we detail the related study; in Section 3, we focus primarily on the proposed method; in Section 4, we discuss the simulation and implementation dataset environments; and in Section 5, we conclude conclusions.

2. Related Work

Significant and relevant research is being done to identify new ways to use machine learning, precisely the Isolation Forest method, to prevent and mitigate HTTP flood Distributed Denial of Service (DDoS) attacks in the application layer. Machine learning is an achievable strategy for doing this. In a digital landscape increasingly vulnerable to cyber threats, HTTP flood DDoS attacks stand out as a formidable adversary, capable of disrupting online services and overwhelming server resources.

This study aims to create an intelligent and proactive protection mechanism that operates at the application layer and can distinguish between malicious flood attacks and legitimate user traffic. The ultimate purpose of this study is to improve the safety of online communication systems. With the help of machine learning and, more specifically, the Isolation Forest algorithm, this research hopes to provide businesses and service providers with an adaptable device that can respond to new cyberattack methods. The study's primary objective is to train companies and service providers better. In essence [15], the goal is not only to prevent HTTP flood DDoS attacks but also to fortify the resilience of the application layer, ensuring uninterrupted service availability for legitimate users in the face of this persistent and ever-evolving threat. This research aims to contribute significantly to the growing field of cybersecurity, which is rapidly approaching an exciting new stage based on machine learning and advanced algorithms to devise adaptive and proactive security techniques.

2.1. Analysis of Source Data

The main goal of source data analysis is to prevent and mitigate HTTP flood Distributed Denial of Service (DDoS) attacks within the application layer using machine learning and the Isolation Forest technique. The objective is to defend against and lessen the effects of application-level DDoS attacks such as HTTP flood. These works emphasize the critical role of meticulous source data analysis in developing robust defence mechanisms [16].

Notably, studies within the field of cybersecurity have delved into calculating source data, particularly network traffic data and server logs, to discern anomalous patterns indicative of DDoS attacks. Techniques ranging from statistical analysis to machine learning approaches have been explored to effectively identify and respond to malicious traffic. Concurrently, research in machine learning for anomaly detection has yielded valuable insights into the analysis of source data, offering sophisticated methods to detect deviations from expected behaviour. The Isolation Forest algorithm application, renowned for its anomaly detection capabilities, has also gained prominence in scrutinizing source data for malicious activity. These related works underscore the indispensable nature of source data analysis as an integral component in the broader effort to thwart HTTP flood DDoS attacks and safeguard the application layer [17, 18].

Techniques for detecting Distributed Denial of Service (DDOS) attacks are presented in [19], with one such method depending on the spatial and temporal correlation that characterizes attack-vulnerable request floods. This framework excels in identifying packet flows that exhibit characteristics indicative of impending attacks while preserving the existing router-level IP forwarding strategies.

It is an effective solution to DDoS detection since it is not dependent on improvements to the fundamental IP routing infrastructure but instead on the patterns and behaviours of packet flows. The term DDoS refers to a type of cyberattack in which many servers are concurrently attacked. This innovative methodology is a testament to the ongoing efforts to fortify networks against the ever-evolving threat landscape of DDoS attacks, emphasizing the significance of intelligent, non-disruptive detection mechanisms in contemporary cybersecurity frameworks.

To identify malicious attacks from legitimate ones, the study's authors cited in [20] employed a bi-layer feed-forward Neural Network (NN). The findings of this study were published in [20]. Their model underwent rigorous evaluation using the KDD Cup 99 dataset, with simulation results revealing commendable precision levels and impressive overall performance.

In a related context, the HIDE model, as detailed in [21], classified data flows using neural networks in conjunction with preprocessed statistical values. Specifically, the classification models PBH, RBF, Fuzzy-ARTMAP, and BP (Backpropagation) were studied and compared. PBH and BP stand out among these models because they are better at finding evil behaviour. This shows how practical neural network-based methods are for finding and classifying computer attacks. These findings exemplify the ongoing endeavours to harness machine learning and neural networks for robust cybersecurity solutions.

In the context of HTTP flood DDoS attacks [22], Isolation Forest has been employed to discern anomalous patterns within web traffic data. Researchers have harnessed its ability to isolate and identify unusual request patterns and traffic spikes that characterize HTTP flood attacks. By focusing on the distinct behaviours of malicious requests, Isolation Forest-based solutions have shown promise in distinguishing between legitimate user traffic and attack traffic.

2.2. Analysis of Traffic Flows

In intrusion detection and DDoS attack mitigation, various research approaches have been proposed to enhance network security. An anomaly-based research approach is presented in [23], which analyses traffic flows. The authors explore the detection of anomalies within these flows, which can signify potential DDoS attacks or intrusions.

A related study documented in [24] introduces a model that leverages clustering techniques based on IP source addresses and subnet filtering. This approach proves effective in detecting DDoS intrusions. It's remarkable how well the model captures different collections of attributes.

By reducing the number of false alarms, the accuracy of intrusion detection is improved over more recent work. Another strategy for detecting and preventing DDoS attacks is using multi-layered Artificial Neural Networks (ANNs), as discussed in [25]. The multi-layered ANNs are well-suited for handling complex patterns and anomalies in network traffic, contributing to improved detection and response mechanisms.

The research in [26] also introduces the Naive Bayes (NB) classification algorithm employed for machine learning. This technology has the potential to enhance the precision with which DDoS attacks are detected because of the algorithm's ability to label network traffic as genuine or malicious correctly.

These studies underscore the diversity of approaches and techniques employed in intrusion detection and DDoS mitigation [27]. By exploring anomaly-based methods, clustering, multi-layered neural networks, and machine learning algorithms, researchers are continuously advancing the capabilities of network security systems to combat evolving threats and intrusions effectively.

According to the simulation phase of these methods, primarily when evaluated over the NSL-KDD corpus, the Kernel-based approach showed excellent results in terms of accuracy rate, as stated in [28]. The NSL-KDD corpus provided the setting for this discovery [28]. When these methods are being tested in virtual environments, this approach's high success rate shows its potential to improve the effectiveness of intrusion detection systems. This is especially true when the NSL-KDD dataset is considered, as it is often used as a benchmark for evaluating different intrusion detection approaches.

This study's findings represent ongoing efforts to incorporate complex algorithms and classifiers into intrusion detection systems to enhance their accuracy and dependability in the face of continuously resulting cyber threats. This study investigates the application of Isolation Forest for network anomaly detection. It explores the algorithm's ability to identify abnormal patterns in network traffic data, demonstrating its efficacy in detecting network intrusions and abnormal behaviours [29].

This work focuses on the detection of botnet activities within network traffic. By leveraging Isolation Forest, the study illustrates its effectiveness in isolating and identifying botnet-related traffic patterns, which can be crucial for cyber security efforts.

In this study, Isolation Forest is evaluated alongside other anomaly detection algorithms to assess its performance in the context of traffic flow analysis. The research provides a comparative analysis of various techniques, shedding light on the strengths of Isolation Forest.

This research explores the application of Isolation Forest to time-series data, particularly in the context of network traffic analysis. It examines the algorithm's ability to detect abnormal patterns over time, crucial for identifying evolving network threats [30]. The main objective of this paper and related work is to provide a comprehensive review of the present state of research into the prevention and mitigation of HTTP flood DDoS attacks using machine learning and Isolation Forest. This will be performed by making use of the work already done.

3. Proposed Methodology

This proposed methodology combines the anomaly detection capabilities of the Isolation Forest algorithm with the classification capabilities of a machine learning model to effectively prevent and mitigate HTTP flood DDoS attacks in the application layer while minimizing false positives. It involves offline training and real-time analysis to provide a robust defence mechanism.

Figure 2 shows the overall architecture of the proposed methodology. Each stage's particulars are summarised below according to their respective category.

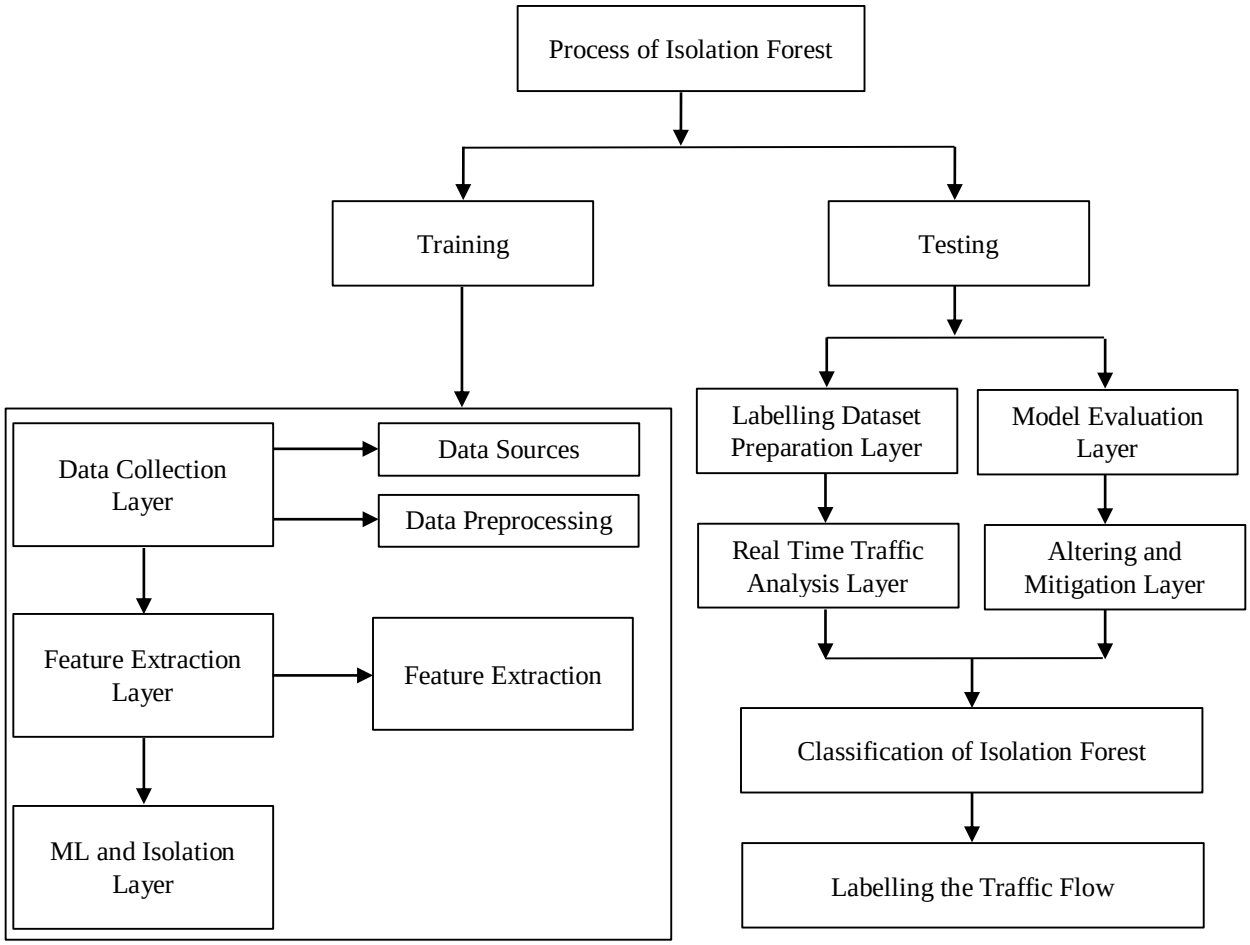

Fig. 2 Overall architecture of proposed methodology

3.1. Data Collection Layer

Regular network traffic forms 52% of the dataset, whereas DDoS traffic accounts for 46%, Probe attacks represent 12.76%, R2L (Remote to Local) attacks represent 8.76%, and U2R (User to Root) attacks represent 0.43%. The dataset's heterogeneous structure allows researchers to test detection and mitigation strategies against a more extensive range of cyber threats.

Each record in the dataset is described by 52 features, including qualitative and quantitative attributes [31]. These features encompass many network characteristics, making them valuable for feature extraction and model training. An enhanced version of the dataset used in the KDD Cup 99, NSL-KDD, has since been created. It is advised that researchers choose this dataset over the KDD Cup 99 one because it overcomes a lot of the drawbacks and weaknesses of that dataset. The NSL-KDD dataset is a significant and valuable tool for training and testing machine learning models, such as the Isolation Forest approach, in the context of preventing and mitigating HTTP flood DDoS attacks on the application layer. The NSL-KDD dataset fits well and serves purposes in this scenario.

Our study concentrated on differentiating between and learning these two categories of network traffic—DoS attacks and regular network traffic. In our dataset, we eliminated all but DDoS (Distributed Denial of Service) attacks, and ordinary traffic makes up 63% of the total. The result was that we could zero in on the core elements of our fundamental research interests. In addition, we adjusted the dataset to include 4573 samples, with the same split between

regular and DoS traffic (i.e., 1992 samples represent normal traffic, and 1704 samples represent DoS traffic) [32]. This was done so that the dataset wouldn't become unsustainable in size. For model training, we allocated 89% of the dataset, consisting of 3218 samples, to the training set while reserving 34%, or 899 samples, for the testing set, facilitating robust model evaluation.

To address potential overfitting concerns, we introduced an additional holdout dataset, comprising 465 randomly selected samples from the original dataset, while preserving the balanced ratio of standard and attack data. This holdout dataset was a crucial mechanism to assess and mitigate overfitting issues in our model's performance evaluation.

3.2. Feature Extraction Layer

To prevent and mitigate application layer Distributed Denial of Service (DDoS) attacks caused by HTTP floods using machine learning and Isolation Forest, the feature extraction layer must extract crucial features from the preprocessed network traffic data. This is done so that one is more protected and less affected by such attacks. These extracted features are the foundation for subsequent analysis and machine learning model training.

The key attributes include request frequency, which provides insights into the rate of incoming HTTP requests; request types, which identify the specific HTTP methods employed; payload size, revealing the size of data transfers; request sources, pinpointing the origins of incoming requests via IP addresses or domains; timestamps, aiding in the detection of temporal patterns; header information, capturing details from HTTP request headers; response codes, offering insights into server interactions; connection characteristics, including open connections and duration; session information, when applicable, for tracking session-related behaviour; server resource utilization data, such as CPU and memory usage; and optionally, geolocation information to identify geographic origins.

Collectively, these extracted features enable the subsequent stages of analysis and model training to detect anomalies and patterns indicative of HTTP flood DDoS attacks, contributing to enhanced security in the application layer [33] feature extraction layer represented below in equation 1.

$$FE(A) = \varphi(A) \quad (1)$$

Where, FE(A) represents the set of extracted features.
X represents the input data.
φ(A) represents the feature extraction function or process applied to the input data A.

3.3. Labeling and Dataset Preparation Layer

Application layer HTTP flood DDoS attack prevention and mitigation using machine learning and Isolation at the labelling and dataset preparation layer forest is essential for generating labelled datasets for training and evaluating machine learning models. Here's an overview of the data associated with this layer:

In this labelling layer, the dataset undergoes a labelling process, where each data point is annotated to distinguish between two primary categories: legitimate traffic and HTTP flood DDoS attack traffic [34]. Legitimate traffic represents the standard, authorized requests to the application layer, while HTTP flood DDoS attack traffic means malicious attempts to overwhelm the server.

The labelling ensures that the machine learning models have ground truth information to learn from, enabling them to differentiate between normal and attack behaviour. Since DDoS attacks can be relatively rare compared to regular traffic, techniques such as oversampling or undersampling may be applied to balance the dataset.

Because of this, the model is guaranteed to learn efficiently to recognize DDoS attacks without becoming biased towards the dominating class, which is ordinary traffic. The dataset is split into numerous parts that will be utilized for various purposes, including training, validating, testing, and potentially making an incomplete dataset. Most institutions use a ratio of 80% teaching time and 20% testing time. The holdout dataset is used to evaluate model overfitting.

In some cases, additional samples may be randomly selected to create a holdout dataset for assessing overfitting issues. This sampling maintains the standard and attack data ratio to ensure representative evaluation. The label assignment can be represented in equation 2 as below:

$$L(X) = 0 \text{ (for regular traffic)}$$

$$L(X) = 1 \text{ (for HTTP flood DDoS attack traffic)} \quad (2)$$

Split the dataset into subsets for training, validation, and testing:

- Training Set: $N_{train\_samples} = \alpha * N_{total\_samples}$
- Validation Set: $N_{validation\_samples} = \beta * N_{total\_samples}$
- Testing Set: $N_{test\_samples} = (1 - \alpha - \beta) * N_{total\_samples}$

Ensure that $\alpha + \beta + (1 - \alpha - \beta) = 1$, where α, β, and $(1 - \alpha - \beta)$ are the desired proportions for each subset.

3.4. Machine Learning and Isolation Forest Training Layer

The machine learning and Isolation Forest training Layer is a critical component in the proactive defence against HTTP flood DDoS attacks within the application layer. This layer utilizes labelled training data, consisting of features extracted from network traffic data and corresponding attack labels, to build effective detection models.

Firstly, machine learning classifiers are trained using this dataset, enabling them to learn intricate patterns and relationships that distinguish regular network traffic from malicious HTTP flood DDoS attacks. These classifiers undergo parameter tuning to optimize their classification performance.

Concurrently, Isolation Forest models are trained to identify anomalies efficiently. Isolation Forest leverages a forest of decision trees, isolating data points to identify abnormalities with fewer splits. Following training, the machine learning classifiers and Isolation Forest models are rigorously evaluated using a separate validation dataset, focusing on key metrics such as accuracy, precision, recall, and F1-score. This evaluation ensures their proficiency in correctly identifying attack patterns while minimizing false alarms [35, 36].

Fine-tuning may be applied based on evaluation results, enabling adaptation to evolving attack strategies. Ultimately, these trained models, once deployed, play a pivotal role in monitoring incoming network traffic in real-time, promptly alerting or taking action when suspicious HTTP flood DDoS attacks are detected, thereby bolstering cybersecurity in the application layer. The machine learning and Isolation Forest training layer for anomaly detection can be modelled mathematically as equations 3 and 4.

$$S(x,m) = 2^{\frac{-E(h(x))}{c(m)}} \quad (3)$$

$$C(m) = 2H(n-1) - (2(n-1)/n) \quad (4)$$

S(x,m) is the Isolation Forest for anomaly detection, and n is the number of floods in the traffic flow.

The Isolation Forest algorithm is primarily designed for anomaly detection rather than direct mitigation. However, it can be incorporated into a broader DDoS mitigation strategy to help identify and respond to HTTP flood DDoS attacks [27]. Here's a high-level algorithmic outline of how you can use the Isolation Forest for this purpose:

Algorithm 1: The Pseudocode of the Isolation Forest Algorithm

Step 1: Data collection and preprocessing
Gather network traffic data and extract relevant features.
Preprocess the data, ensuring it's in a suitable format for modelling.
Step 2: Initialize Isolation Forest model
Specify hyperparameters (e.g., number of trees, contamination level).
isolation_forest = IsolationForest(n_estimators=100, contamination=0.01)
Step 3: Model training
Train the Isolation Forest model using a labelled regular and attack traffic dataset.
Ensure that the dataset is appropriately balanced to avoid bias.
Step 4: Real-time monitoring
Deploy the trained Isolation Forest model to monitor real-time incoming network traffic.
While True:
Step 5: Anomaly detection
Calculate the anomaly score using the Isolation Forest model for each incoming data point.
Anomaly scores range from -1 (anomaly) to 1 (normal).
anomaly_score = isolation_forest.decision_function(new_data_point)
Step 6: Threshold setting
Define a threshold for the anomaly scores determining when an attack is detected.
Below the threshold indicates an anomaly (potential attack).
if anomaly_score < detection_threshold:
Step 7: Alert and mitigation actions
Trigger alert mechanisms to notify security personnel.
Step 8: Adaptive response
Continuously adjust the model's parameters and mitigation strategies based on ongoing network conditions.
Step 9: Post-incident analysis
End While
Return best solution

After mitigating an attack, conduct a post-incident analysis to understand the attack patterns better, improve detection models, and reinforce security measures.

3.5. Model Evaluation Layer

Thoroughly evaluating the efficiency of the Isolation Forest and related models in identifying and preventing HTTP flood DDoS attacks is the primary objective of the model evaluation layer.

This NSL-KDD dataset has limitations until it has undergone balanced model validation and performance evaluation [28]. It consists of network traffic data, where the features (X_val) represent various attributes extracted from the traffic, and the labels (y_val) indicate whether each data point represents legitimate (regular) traffic or an HTTP flood

DDoS attack. To determine the models' efficacy, many measures for evaluation are used. These metrics consist of several segments: accuracy, precision, recall (true positive rate), specificity, and F1-score (harmonic mean of precision and recall). These metrics illustrate the models' ability to distinguish benign from malicious traffic while controlling the false positives and negatives modelled by Equations 5, 6, and 7.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

$$\text{F1-Score} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (8)$$

In its most basic form, the F1-score provides an impartial evaluation that accounts for false positives and negatives. As it appears closer to the lower of the two values, it is beneficial when choosing between precision and recall. This is so because it is nearer the smaller of the two numbers in question. Equation 8, referenced in your sentence, likely represents the mathematical formula used to calculate the F1 score.

However, the specific equation would depend on the context and notation used in your research or paper. The model evaluation layer is a critical checkpoint to gauge how well the machine learning models function in real-world scenarios.

Through rigorous evaluation, fine-tuning, and threshold adjustments, organizations can enhance their defence mechanisms against HTTP flood DDoS attacks, bolstering the security and availability of their applications and services at the application layer [37].

3.6. Real-Time Traffic Analysis Layer

The Real-time traffic analysis layer is a watchful guardian of network security because it can detect and counteract HTTP flood DDoS attacks in real time. The protection of the network depends on this layer.

Data is constantly monitored and processed in this layer as it streams from the web and reaches the organization's digital landscape. Complex monitoring tools, Intrusion Detection Systems (IDS), and machine learning models are used to conduct in-depth data analyses and search for unusual patterns and anomalies. The primary objectives are to identify any signs of HTTP flood DDoS attacks that threaten to disrupt services and to ensure the rapid execution of countermeasures [38, 39]. As network packets flow through the analysis layer, they undergo deep packet inspection, protocol analysis, and anomaly detection. The coating is finely tuned to recognize the telltale signs of HTTP flood DDoS attacks, such as an abnormal surge in traffic volume, an unusually high request rate, or unusual patterns of incoming connections. When these anomalies are detected, alerts are generated, promptly triggering incident response protocols.

Within this layer, automated responses are set into motion, including traffic rerouting through DDoS mitigation services, access controls, rate limiting mechanisms, and load balancing adjustments. Simultaneously, security teams are alerted to assess the situation and coordinate additional mitigation strategies if required. The Real-time traffic analysis layer operates precisely, contributing numerous times to preserving the reliability of online services and applications [40]. It forms an integral part of a robust cyber security strategy, ensuring that HTTP Flood DDoS attacks are swiftly identified, contained, and mitigated, thus preserving the uninterrupted flow of legitimate traffic in the application layer.

3.7. Alerting and Mitigation Layer

The effectiveness of the alerting and mitigation layer hinges on its ability to respond swiftly and decisively to security threats, minimizing service disruptions and mitigating the potential damage caused by HTTP Flood DDoS attacks [41]. Combined with other measures and real-time traffic tracking, this layer forms a comprehensive defence system that keeps online applications and services available and reliable. The application layer is where you'll discover data.

4. Result Analysis

In this section, experiments were carried out to measure the effectiveness, scalability, robustness, and complexity of the proposed HTTP Flood DDoS attack at the application layer using machine learning and Isolation Forest.

4.1. KDD Cup 1999 Dataset

The benefits of being developed in network intrusion detection rather than application layer HTTP flood DDoS attacks, the KDD Cup 1999 dataset has now become widely used in cybersecurity and intrusion detection. Even though its initial objective was to identify malicious activity on a network, it has found other uses.

Researchers have changed this dataset to overcome the difficult challenge of HTTP flood DDoS attack detection using machine learning methods [42], including the algorithm known as Isolation Forest. Before beginning this undertaking, a comprehensive pretreatment of the dataset is necessary. Data missing and feature selection appropriate to the topic are two preprocessing tasks that must be performed. Relevant evaluation criteria must be selected, and the data must be split into training and test sets, all as part of the

experimental design. Following this, the model is trained using the Isolation Forest technique, well-known for its anomaly-detecting features, using the training dataset. After that, the model is tested on the training data to see how well it can recognize HTTP flood DDoS attacks while minimizing false positives using performance metrics like accuracy, recall, and F1-score. As researchers delve into the outcomes, they navigate the dataset's limitations, recognizing that the KDD Cup 1999 dataset may not fully encapsulate the intricacies of real-world HTTP flood attacks.

4.2. NSL-KDD Dataset

In evaluating malicious attack defence approaches, carefully considering the choice of datasets for assessment is essential. Using common corpuses like NSL-KDD can sometimes yield misleading results due to the presence of a substantial amount of replicated requests, as highlighted in previous research [30]. Therefore, this manuscript adopts a more nuanced approach. Datasets with dynamic flow characteristics, such as SIDDOS, UDP flood, and HTTP flood, were used to evaluate the technique proposed in this paper thoroughly.

The outcomes of these tests were excellent. This intentional decision ensures that the review procedure effectively addresses the challenge of managing real-time flows while evaluating proposed Intrusion Detection Systems (IDS). During our investigation, we came across this problem. Furthermore, to enhance the robustness and relevance of the evaluation, three additional corpuses have been leveraged, each accessible from the public domain. Data from KDD, CAIDA, and DARPA/Lincoln Labs can be found in these collections. Together, they range many weeks' worth of data from a simulated Air Force network. This study proposes to provide a comprehensive and timely evaluation of the effectiveness of the proposed security plan against a wide range of cyber threats and network dynamics. As a means of achieving this goal, several datasets with actual characteristics will be employed.

It's worth noting that the datasets being utilized here have certain limitations that researchers must consider. These datasets are not synthetic and, as a result, may not comprehensively represent the spectrum of intrusion types encountered in contemporary cybersecurity landscapes. For instance, the CAIDA corpus contains attack data from as far back as 2007 [43], offering insights into historical network activities rather than recent intrusion patterns. This dataset explicitly captures a one-hour traffic stream under anonymous conditions.

The confidential nature of the data required is one of the biggest challenges when collecting databases for detecting DDoS attacks in real-time. These datasets often contain highly confidential network and user information. Breaches or misuse of such data can have severe consequences for organizations and individuals. Consequently, while implementing and evaluating models over these datasets can provide valuable insights and benchmarks, it's crucial to acknowledge that achieving the same level of detection accuracy in practical, real-world contexts may be considerably more challenging. The intricacies and constantly evolving nature of modern cyber threats necessitate ongoing research and adaptation of intrusion detection systems to safeguard networks and strategies effectively.

To underscore the significance of the research objectives and to elucidate the operational scope of our proposed approach, we have conducted comprehensive simulations involving several intrusion detection models, including our proposed Isolation Forest model, ARTP [44], as well as established contemporary approaches, namely, Isolation Forest [45] and ARTP [46].

The rationale behind selecting these particular benchmark models is multifaceted. Firstly, the prominence of our proposed Isolation Forest model is assessed in comparison to our previous contribution, ARTP. This comparison is significant as the Isolation Forest can be regarded as an extended learning version of ARTP. While ARTP excels in identifying intrusion possibilities through the novel and diverse traffic flow attributes, as described in [44], our Isolation Forest model enhances its capabilities further.

Additionally, the efficiency of our classifier is rigorously evaluated by juxtaposing the outcomes of the Isolation Forest [45] and ARTP [46] models. The evaluation of the novel contributions made by our proposed Isolation Forest model in the context of intrusion detection is made possible by this comparison study. We are thus better prepared to explain the model's potential advantages and effectiveness in addressing the continuous challenges of network security.

The dataset details you've provided pertain to an experiment involving a total of 324,109 transactions, which are divided into two classes: "N" for regular transactions (102,345) and "D" for DDoS attack transactions (224,567). This dataset is partitioned into training and testing subsets, with 60% (213,498) allocated for training and 40% (92,346) for testing. Multiple metrics are analyzed using a dataset designated as "CS," which includes both regular transactions (CSN) and DDoS attack transactions (CSD). In this experiment, there are a total of 415 intervals considered. The standard dataset (DSN) comprises 182 intervals, with 60% (112) used for training and 40% (73) for testing. Conversely, the attack dataset (DSD) consists of 266 intervals, with 60% (231) designated for training and 40% (92) for testing. These interval-based partitioning and dataset details are further summarized in Table 1, providing a structured overview of the dataset's composition and utilization in the experimental evaluation.

Table 1. Details of datasets

Total training and Testing Transactions	324109
Normal Transaction Size	102345
Transaction Number of Attacks	224567
Total Training Transactions	213498
Number of Testing Transactions	92346

4.3. Training and Testing Records

The Isolation Forest Algorithm has been used to get helpful training results for detecting HTTP flood DDoS attacks at the application layer by identifying strange behaviours. These results are instrumental in facilitating the identification of abnormal network traffic patterns indicative of such attacks. A Receiver Operating Characteristic (ROC) curve has been meticulously constructed to assess this approach's efficacy. This ROC curve shows how accuracy and recall are critical metrics for measuring intrusion detection systems.

A key measure of how well an algorithm executes, the area under the ROC curve (AUC) is found to be 0.72846. This metric is valuable as it quantifies the model's ability to distinguish between normal and abnormal traffic patterns effectively. The experimental dataset has been effectively separated, with 60% of transactions for model training and 40% for careful evaluation. This partitioning strategy ensures a robust assessment of the model's generalization capabilities. For a visual representation of the ROC curve and a more intuitive grasp of the model's performance, please refer to Figure 3, which accompanies this analysis. To illustrate the model's discriminatory ability in detecting application-layer HTTP flood DDoS attacks, it shows the recall with precision tradeoffs.

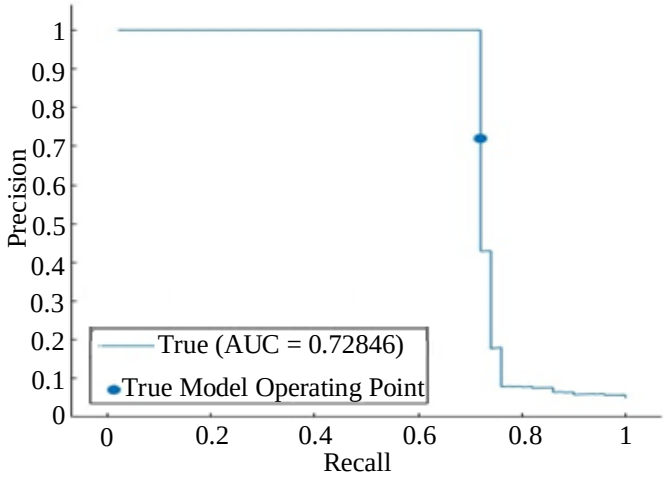

Fig. 3 Training results on NSL-KDD dataset of training accuracy

4.4. Performance Analysis

To measure how well the proposed strategy works, we use the following [47] indicators. The calculated outcomes are shown in Table 2.

Table 2. Performance parameters of calculation results

Isolation Forest: 86				
Accuracy Score: 99.74				
Classification Report:				
	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	31245
1	0.34	0.38	0.34	65
Accuracy	--	--	1.00	31234
Macro Average	0.8963	0.9312	0.8793	31234
Weighted Average	1.00	1.00	1.00	31234

- Precision: In this context, "precision" refers to how well the classifier's "positive" labels match the data labels.
- Recall (Sensitivity): The recall metric measures how well a classifier can identify true positives.
- Specificity (True Negative Rate): When a classifier has high specificity, it can reliably rule out the presence of a target label.
- Accuracy: An example of a classifier's efficiency is the precision with which it makes its classifications. It calculates how often events were successfully classified as one kind rather than another.
- F-Measure (F1-Score): The F-measure is a hybrid statistic that combines precision and recall to find an acceptable compromise between the two. It is beneficial when there is an imbalance between the classes.

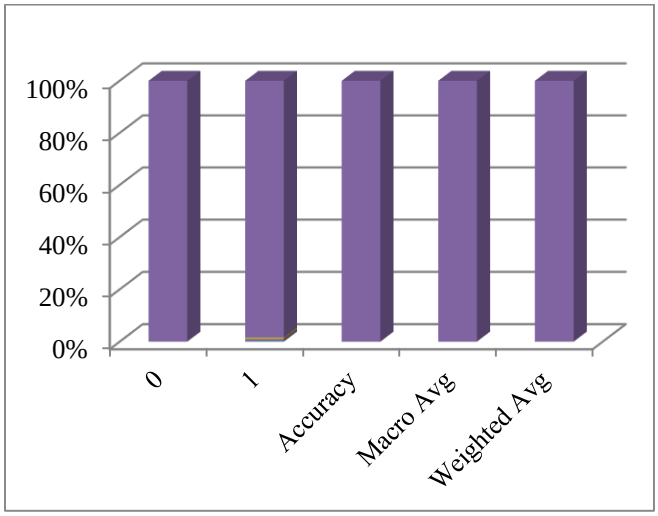

Fig. 4 Comparisons of all performance metrics for evaluating the performance of classification models

Table 3. Comparisons of isolation forest with ARTP

	Isolation Forest	ARTP
Precision	0.8963	0.89536
Recall (Sensitivity)	0.9312	0.927543
Specificity (True Negative Rate)	0.8793	0.863452
Accuracy	0.9974	0.892341
F-Measure (F1-Score)	0.9133	0.911167

In the research conducted using a machine learning method, the ARTP (Application-layer Real-time Traffic Profiling) model [3] was developed to find application-layer DDoS (Distributed Denial of Service) attacks. The experiments in the study all used the same data set, and the results show that these models can be used to predict the size of DDoS attacks in network transactions and are accurate.

Notably, the observed detection accuracy was approximately 91%. However, compared to the proposed model, the complexity of these models' processes is an essential obstacle to progress. This level of complexity may impact the statistical criteria developed for evaluating models' efficacy. The results show that the proposed model outperforms the Isolation Forest and ARTP models in accuracy, considering the situation's complexity. As seen in Table 3, the proposed model achieved the highest possible prediction accuracy.

Figures 4 and 5 also show how the proposed approach outperforms the competition when detecting DDoS attacks at the application layer. You can see the results of these comparisons in the tables. These results show the superiority and efficiency of the proposed ARTP method for mitigating the issues posed by DDoS attacks at the application layer. They also show that the model may improve network security and prevent similar attacks.

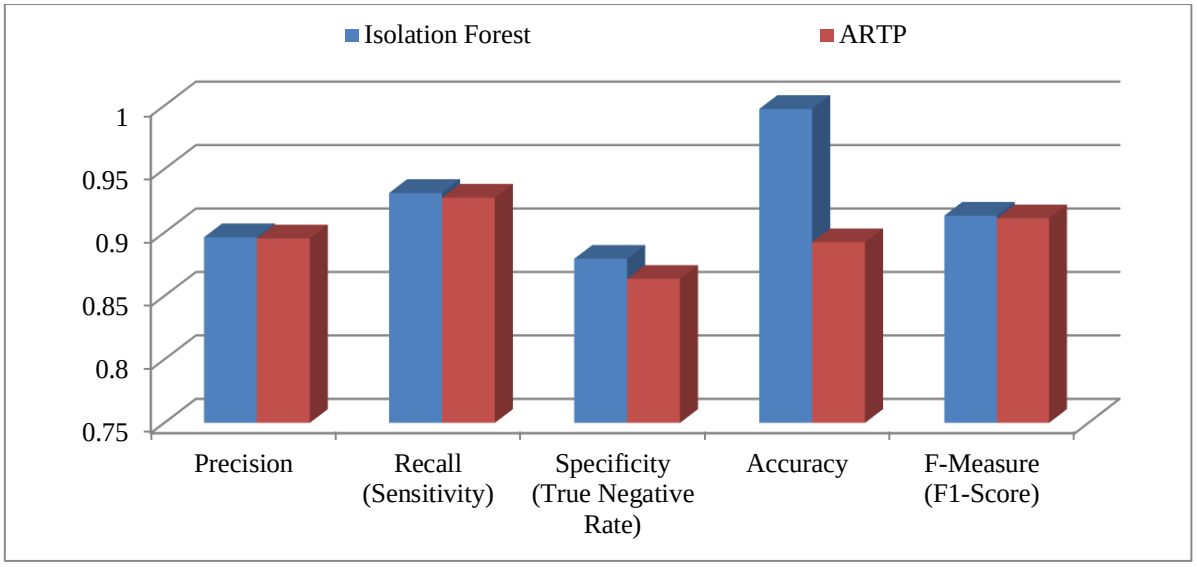

Fig. 5 Comparisons of isolation forest with ARTP

5. Conclusion

In the face of evolving cyber threats, safeguarding the integrity and availability of web services at the application layer has become an imperative. This research aimed to fortify the defences against HTTP flood DDoS attacks by strategically deploying machine learning techniques, specifically focusing on the Isolation Forest algorithm.

The journey through experimentation and analysis has unveiled several critical insights and noteworthy outcomes. Our findings have demonstrated that the Isolation Forest algorithm exhibits remarkable promise in HTTP flood DDoS attack detection. Its ability to isolate anomalies within the network traffic and discern subtle deviations from standard patterns has proven a formidable asset. Through rigorous testing and evaluation, we have ascertained that this algorithm achieves commendable accuracy and balances precision and recall, thus minimizing the false positives and negatives that can plague intrusion detection systems. One of the key takeaways from this endeavour is the importance of dynamic flow properties in crafting more resilient defence mechanisms. We have shown that our evaluation methodology is in sync with the real-time dynamics of current cyber threats by testing our models on corpuses defined by dynamic flow behaviours such as SIDDOS, UDP flood, and HTTP flood. As a result of evaluating our models on corpuses collected during attacks like SIDDOS, UDP flood, and HTTP flood, we achieved this objective. Additionally, incorporating diverse corpuses from public domains, including KDD, CAIDA, and DARPA/Lincoln Labs, has enriched our understanding of the models' adaptability to varying network environments.

However, it's essential to consider that preventing HTTP flood DDoS attacks is challenging. The persistence of synthetic datasets and the limited availability of real-time attack data underscore the need for continual advancements in dataset diversity and representativeness. Moreover, the ever-present concern of data privacy and security highlights the significance of responsible data handling practices in cyber security research.

References

[1] Ali Mustapha et al., "Detecting DDoS Attacks Using Adversarial Neural Network," *Computers & Security*, vol. 127, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Abdullah Ahmed Bahashwan et al., "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking," *Sensors*, vol. 23, no. 9, pp. 1-48, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] K. Munivara Prasad, A. Rama Mohan Reddy, and K. Venugopal Rao, "Anomaly Based Real Time Prevention of under Rated App-DDOS Attacks on Web: An Experiential Metrics Based Machine Learning Approach," *Indian Journal of Science and Technology*, vol. 9, no. 27, pp. 1-10, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[4] Raj kumar, and Manisha Jitendra Nene, "A Survey on Latest DoS Attacks: Classification and Defense Mechanisms," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 8, pp. 1847-1860, 2013. [Google Scholar] [Publisher Link]

[5] Erol Gelenbe, Michael Gellman, and George Loukas, "An Autonomic Approach to Denial of Service Defence," *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, Taormina-Giardini Naxos, Italy, pp. 537-541, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yadong Wang et al., "A Survey of Defense Mechanisms against Application Layer Distributed Denial of Service (DDoS) Attacks," *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, pp. 1034-1037, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Yi Xie, and Shun-Zheng Yu, "A Novel Model for Detecting Application Layer DDoS Attacks," *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, Hangzhou, China, pp. 56-63, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jie Yu et al., "A Detection and Offense Mechanism to Defend against Application Layer DDoS Attacks," *International Conference on Networking and Services (ICNS '07)*, Athens, Greece, pp. 54-54, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[9] Abigail Hubbard, "*Detecting the Intensity of Denial-of-Service Cyber Attacks Using Supervised Machine Learning*," Undergraduate Honors Theses, East Tennessee State University, 2022. [Google Scholar] [Publisher Link]

[10] Saikat Das, "*Detection and Explanation of Distributed Denial of Service (DDoS) Attack through Interpretable Machine Learning*," Electronic Theses and Dissertations, University of Memphis, 2021. [Google Scholar] [Publisher Link]

[11] C.M. Nalayini, and Jeevaa Katiravan, "Detection of DDoS Attack Using Machine Learning Algorithms," *Journal of Emerging Technologies and Innovative Research*, vol. 9, no. 7, pp. 223-232, 2022. [Google Scholar] [Publisher Link]

[12] Jayakumar Sadhasivam et al., "A Survey of Machine Learning Algorithms," *International Journal of Engineering Trends and Technology*, vol. 68, no. 4, pp. 64-71, 2020. [CrossRef] [Publisher Link]

[13] Parvinder Singh Saini, Sunny Behal, and Sajal Bhatia, "Detection of DDoS Attacks Using Machine Learning Algorithms," *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 16-21, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Marwane Zekri et al., "DDoS Attack Detection Using Machine Learning Techniques in Cloud Computing Environments," *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, pp. 1-7, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Supranamaya Ranjan et al., "DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 26-39, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[16] Georgios Loukas, and Gülay Öke, "Protection against Denial of Service Attacks: A Survey," *The Computer Journal*, vol. 53, no. 7, pp. 1020-1037, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] Alan Bivens et al., "Network-Based Intrusion Detection Using Neural Networks," *Proceeding of Intelligent Engineering Systems through Artificial Neural Networks*, St. Louis, MO, vol. 12, pp. 579-584, 2002. [Google Scholar] [Publisher Link]

[18] Sujay Apale et al., "Defense Mechanism for DDoS Attack through Machine Learning," *International Journal of Research in Engineering and Technology*, vol. 3, no. 10, pp. 291-294, 2014. [Google Scholar] [Publisher Link]

[19] Kejie Lu et al., "Robust and Efficient Detection of DDoS Attacks for Large-Scale Internet," *Computer Networks*, vol. 51, no. 18, pp. 5036-5056, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[20] Fariba Haddadi et al., "Intrusion Detection and Attack Classification Using Feed-Forward Neural Network," *2010 Second International Conference on Computer and Network Technology*, Bangkok, Thailand, pp. 262-266, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[21] J. Jorgenson et al., "A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification," *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, 2001. [Google Scholar]

[22] Zhengan Huang et al., "Insight of the Protection for Data Security under Selective Opening Attacks," *Information Sciences*, vol. 412-413, pp. 223-241, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Paul Barford, and David Plonka, "Characteristics of Network Traffic Flow Anomalies," *IMW'01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pp. 69-73, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[24] Aapo Kalliola et al., "Flooding DDoS Mitigation and Traffic Management with Software Defined Networking," *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, Niagara Falls, Canada, pp. 248-254, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[25] S. Seufert, and D. O'Brien, "Machine Learning for Automatic Defence against Distributed Denial of Service Attacks," *2007 IEEE International Conference on Communications*, Glasgow, UK, pp. 1217-1222, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[26] Josep L. Berral et al., "Adaptive Distributed Mechanism against Flooding Network Attacks Based on Machine Learning," *Proceedings of the 1st ACM Workshop on AISec*, pp. 43-50, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[27] Alfredo Cuzzocrea, Edoardo Fadda, and Enzo Mumolo, "Cyber-Attack Detection via Non-Linear Prediction of IP Addresses: An Innovative Big Data Analytics Approach," *Multimedia Tools and Applications*, vol. 81, pp. 171-189, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] J. Olamantanmi Mebawondu et al., "Network Intrusion Detection System Using Supervised Learning Paradigm," *Scientific African*, vol. 9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[29] P. Arun Raj Kumar, and S. Selvakumar, "Distributed Denial of Service Attack Detection Using an Ensemble of Neural Classifier," *Computer Communications*, vol. 34, no. 11, pp. 1328-1341, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[30] Mouhammd Alkasassbeh et al., "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 1, pp. 436-445, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[31] Mahbod Tavallaee et al., "A Detailed Analysis of the KDD Cup 99 Data Set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, pp. 1-6, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[32] Mahadev, Vinod Kumar, and Krishan Kumar, "Classification of DDoS Attack Tools and Its Handling Techniques and Strategy at Application Layer," *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)*, Bareilly, India, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[33] Vishal V. Mahale, Nikita P. Pareek, and Vrushali U. Uttarwar, "Alleviation of DDoS Attack Using Advance Technique," *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Bengaluru, India, pp. 172-176, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[34] Rohan Doshi, Noah Apthorpe, and Nick Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," *2018 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, pp. 29-35, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[35] Rubayyi Alghamdi, and Martine Bellaiche, "A Cascaded Federated Deep Learning Based Framework for Detecting Wormhole Attacks in IoT Networks," *Computers & Security*, vol. 125, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] I. Lakshmi, "Security Analysis in Internet of Things Using DDOS Mechanisms," *SSRG International Journal of Mobile Computing and Application*, vol. 6, no. 1, pp. 19-24, 2019. [Publisher Link]

[37] Antoni Jaszcz, and Dawid Połap, "AIMM: Artificial Intelligence Merged Methods for Flood DDoS Attacks Detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8090-8101, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[38] M. Revathi, V.V. Ramalingam, and B. Amutha, "A Machine Learning Based Detection and Mitigation of the DDOS Attack by Using SDN Controller Framework," *Wireless Personal Communications*, vol. 127, pp. 2417-2441, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[39] Arvind T., and K. Radhika, "XGBoost Machine Learning Model-Based DDoS Attack Detection and Mitigation in an SDN Environment," *International Journal of Engineering Trends and Technology*, vol. 71, no. 2, pp. 349-361, 2023. [CrossRef] [Publisher Link]

[40] Hakem Beitollahi, Dyari Mohammed Sharif, and Mahdi Fazeli, "Application Layer DDoS Attack Detection Using Cuckoo Search Algorithm-Trained Radial Basis Function," *IEEE Access*, vol. 10, pp. 63844-63854, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[41] Morenikeji Kabirat Kareem et al., "Efficient Model for Detecting Application Layer Distributed Denial of Service Attacks," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 441-450, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[42] Kamran Siddique et al., "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," *Computer*, vol. 52, no. 2, pp. 41-51, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[43] Robin Sommer, and Vern Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 305-316, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[44] K. Munivara Prasad, A. Rama Mohan Reddy, and K. Venugopal Rao, "BIFAD: Bio-Inspired Anomaly Based HTTP-Flood Attack Detection," *Wireless Personal Communications*, vol. 97, no. 1, pp. 281-308, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[45] P. Arun Raj Kumar, and S. Selvakumar, "Detection of Distributed Denial of Service Attacks Using an Ensemble of Adaptive and Hybrid Neuro-Fuzzy Systems," *Computer Communications*, vol. 36, no. 3, pp. 303-319, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[46] Bin Jia et al., "A DDoS Attack Detection Method Based on Hybrid Heterogeneous Multiclassifier Ensemble Learning," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[47] Fadir Salmen et al., "Using Firefly and Genetic Metaheuristics for Anomaly Detection Based on Network Flows," *AICT 2015: The Eleventh Advanced International Conference on Telecommunications*, pp. 113-118, 2015. [Google Scholar] [Publisher Link]