ISSN 2007-9737

Web-based Application Layer Distributed Denial-of-Service Attacks: A Data-driven Machine Learning Strategy

K Alluraiah[*], Manna Sheela Rani Chetty

Koneru Lakshmaiah Education Foundation,
Department of Computer Science & Engineering,
India

allura02@gmail.com, sheelarani_cse@kluniversity.in

Abstract. DDoS attacks, which aim to overwhelm a system with requests, are commonplace in the cyber world. In this type of assault, bandwidth and processing resources are deliberately clogged to disrupt legitimate users' interactions. These attacks inundate the victim's system with packets, rendering it inaccessible. Diverging from the singular source of Denial of Service (DoS) attacks, DDoS attacks emanate from many servers, magnifying their impact. Over the last decade, a concentrated effort has been invested in comprehending the orchestration and authentication of DDoS attacks, resulting in valuable insights into discerning attack patterns and suspicious activities. Currently, the focus has shifted towards real-time detection within the stream of network transactions, constituting a critical research domain. Yet, this focus often sidelines the importance of benchmarking DDoS attack assertions within the streaming data framework. As a remedy, the Anomaly-based Real-Time Prevention (ARTP) framework has been formulated and designed specifically to combat application layer DDoS attacks, particularly targeting web applications. Employing advanced machine learning techniques, ARTP offers adaptable methodologies to swiftly and accurately pinpoint application-layer DDoS attacks. Rigorous testing on a representative LLDoS (Low-Level DoS) benchmark dataset has affirmed the resilience and efficiency of the proposed ARTP model, underscoring its capacity to achieve the research objectives set forth.

Keywords. Detection of App-DDoS, denial of service (DoS) attacks, application layer DDoS (App-DDoS), LLDoS dataset, distributed DoS (DDoS) attacks.

1 Introduction

Without question, in the present setting, the Internet has established itself as a crucial component in the realm of daily business activity. The extraordinary evolution of communication methods, trade practices, and individual online presence continues to captivate attention [1]. With this dynamic setting, it's no surprise that Internet vulnerabilities like Denial of Service (DoS) attacks and their more sophisticated sibling, Distributed Denial of Service (DDoS) attacks, have become major problems.

A DoS attack targets a specific system, often a web server, to disrupt its capacity to serve legitimate users without affecting the network or server architecture. DDoS assaults use a network of devices spread over the Internet to flood targeted systems with excessive traffic, draining their resources and disrupting network integrity, making the systems unavailable to legitimate users.

Network layer attacks that utilize DDoS and application layer HTTP attacks are the primary forms of distributed loss of service attacks. Both forms are common in the realm of decentralized denial of service attacks. One type of attack involves an adversary utilizing methods such as IP spoofing to flood the system in question with fake data packets [2]. Application layer DDoS assaults, also known as App-DDoS attacks, flood the system being attacked with numerous legitimate requests to disrupt its functioning.

The attacker's infected machines must create legitimate TCP links with the victim's system to avoid the connections being terminated [3]. One instance is the Internet flood, which uses increased HTTP requests to overwhelm the targeted servers. Operating at the program level with the ground, these assaults target the hosting server of websites by overwhelming it with a large number of simultaneous and ostensibly valid requests, ultimately making the service inaccessible to genuine users.

Recent studies, like [4], explore new methods to evade DDoSdefences and reveal methodologies that enable more effective assault partnerships. The HTTP flood method increases the number of HTTP requests to attack weaknesses in the targeted servers' design. During this flood of data, the servers cannot distinguish between genuine payload information and the overwhelming amount of seemingly normal HTTP requests. The main contribution of the paper is:

- Desigining the Anomaly-based Real-Time Prevention (ARTP) framework for prediction DDoS attacks in targeting web applications.
- Employing advanced machine learning techniques, ARTP offers adaptable methodologies to swiftly and accurately pinpoint application-layer DDoS attacks.
- Experimental results have been implemented and the suggested ARTP model increases efficiency, accuracy and prediction ratio compared to existing models.

2 Literature Review

Certain DDoS protection methods designed to combat HTTP floods rely on application layer knowledge, as a recent survey shows [5]. One example is the DDoS shield [6], which uses the detection of session beginning timings and control of time delays between arrivals to prevent HTTP floods.

The study "Using Page Access Leads to Counter the HTTP Flood" explores the complex connection between data sizes and surfing durations. However, these methods are ineffective when dealing with the high packet transmission rates enabled by the organized actions of malevolent Botnets [7].

Implementing a CAPTCHA-based probability validation method may be beneficial in some instances, but it might overly complicate the user experience [8]. This might lead to an unintentional Denial of Service situation, especially when it is crucial to answer practical complaints [9].

The Concealed semi-Markov modelling (HSMM) is presented in reference [10] to use the characteristics of transactional demand instances to decipher common client access patterns. This model assesses the continuous increase of customers utilizing HSMM-derived data. This method results in many false warnings due to the variety in how users access the system, particularly given the different ways individuals browse.

Clients interacting with external internet interfaces, entering URLs for searches, or using various browsers may unintentionally cause false alerts, introducing confusion. The main goal of the suggested technique is to evaluate transactional relationships using regular and flood data. Unlike traditional benchmarking methods, our suggested model is based on thorough observations of the request stream over a long period.

Eric Muhati and DandaRawat [11] suggested the Data-Driven Network Anomaly Detection with Cyber Attack and Defense Visualization. The author employs a data analytics-based intrusion detection system that examines all network connections; this innovative method is based on network scanning tools and network discovery services, which enable us to view the network according to the number of active IP-based networking devices.

The author then takes precautions by graphically differentiating between safe and dangerous connections, using red for the former and blue for the latter. The experimental assessment reveals that the model outperforms previous research in this field with an impressive F1 score of 97.9% and a low false positive rate of 0.3%.

Talha Farid and MaheyzahSirat [12] recommended the Hybrid of Supervised Learning and Optimization Algorithm for Optimal Detection of IoT Distributed Denial of Service Attacks. An authenticated and verified real-time IoT traffic dataset is used to train machine learning classification algorithms, and their prediction performance is first evaluated in this work.

With a prediction accuracy of 97%, the findings show that Logistic Regression (LR) is the most effective supervised machine learning classifier for identifying IoT DDoS assaults. As a result, Grasshopper Optimizer Algorithms (GOA) emerge

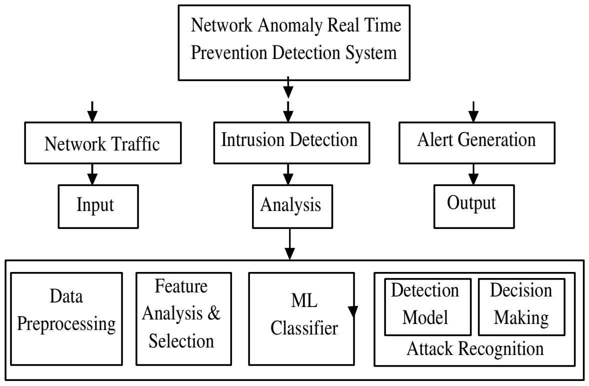

Fig. 1. The architecture of network anomaly real time prevention detection system

as the top optimizer for enhancing LR's prediction accuracy to 99% in a subsequent study on LR hybridization with optimization algorithms.

Therefore, the best option for detecting IoT DDoS attacks is the LR hybridized by GOA. Therefore, the paper lays forth A data-driven strategy to counter new forms of malicious IoT DDoS attacks, such zero-day attacks.

Wubetu Barud Demilie and Fitsum Gizachew Deriba [13] discussed the structured query language injection (SQLI) attack detection using machine learning and hybrid techniques.

The Keras library was used to implement the ML algorithms, while the Tensor FlowLearn package was used to implement the classical techniques.

We used 54,306 bits of data from weblogs, cookies, session use, and HTTP (S) request flows to train and test our model for this suggested research study.

In terms of performance assessment findings for training sets, measures like f1-score (99.35% and 99.57%), accuracy (99.20% and 99.60%), recall (99.65% and 99.61%), and hybrid technique (ANN and SVM) outperform other ML approaches.

The next parts of this work carefully adhere to a structured path. Section 3 explains the ARTP model, and Section 4 thoroughly analyzes a test study to assess the effectiveness of ARTP. Section 5 concludes this academic pursuit by providing definitive insights.

3 Web-based Applications and Regional Denial-attacks: Utilizing a Data-driven Learning-based Approach

We have developed an anomaly-based Actual Time Preventive (ARTP) framework to address the threat of Application Layer DDoS assaults on the World Wide Web. The multitude of demands prompted the creation of this framework. The system's prompt and early identification of application-level DDoS assaults has been a major success, establishing it as one of its most notable accomplishments.

The ARTP model's core resides in its multifarious entity metrics, including important aspects like request chain length, packet count variations, packet sorting variants, request intervals, and the contextual background of transactional chains. Many common benchmarking approaches rely on sessions or requests as main inputs; however, they unintentionally overlook the essential premise of this methodology.

Comprehensive testing was conducted using the LLDoS dataset to assess the usefulness and strength of the proposed model, highlighting its flexibility and resilience. The anticipated approach demonstrates exceptional effectiveness in dealing with application layer DDoS assaults. With the current environment of online applications, the volume of received requests has increased to the size of petabytes, far beyond the gigabyte-level load of prior web request systems.

Using labelled datasets, various supervised learning algoithms, including Neural Networks, Support Vector Machines (SVMs), and Random Forests, are learned to distinguish between valid and dangerous inflow requests.

These algorithms can accurately identify and classify recognized DDoS attacks because they learn torecognize patterns indicative of these attacks using characteristics taken from the traffic data. This study uses supervised and unsupervised learning methodsto spot suspicious behaviours in network data, which might indicate new types of distributeddenial of serviceattacks.

To be more precise, thisstudy finds clusters of requeststhat are much out of the ordinary by using

ISSN 2007-9737

Table 1. Specifics on the normal and DDoS attack time frames, and the training and testing datasets used

	Number of SuccessfulTransactions (CS)			2,29,386
		Training (60%)	Testing (40%)	Total
N (Normal) CS_N	Theoperations	74,260	49,520	123,780
	Sessions	2872	1838	4710
	Groups	287	175	462
	tf: thelength of the time frame	608	614	1222
	Many time frames	242	147	389
D (DDoSAttack) CS_D	Theoperations	56,440	35,961	92,401
	Sessions	2548	1533	4081
	Groups	253	154	407
	tf: thelength of the time frame	744	759	1503
	Many time frames	264	167	431

a clustering-based anomaly detection method. As part of ourun supervised learning strategy, this study clusters incoming requests according to variables, including request frequency, pay load characteristics, and IP addresses.

Our approach can adapt to new threats and identify attack patterns not present in the training data since this study used unsupervised learning methods. Because of this, ourDDoS detection techniqueis more resilient and can with stand new attacks at the web applicationlayer.

4 Determining the Duration of a Period

The abbreviation CS represents a collection of Client Sessions denoted as {s1, s2,…,sn}. Every session in this set, which is labelled as {llsi^siCS}, includes transactions classified as either N (typical) or D (disorganized, indicating a DDoS assault) [14].

The aggregate count of transactions in CS encompasses both regular (CSN) and anomalous (CSD) transactions, forming the composite volume of transactions associated with DDoS attacks.

The heuristic measures outlined in Section 3 will undergo validation through experimentation by employing these datasets. To clarify further, the dataset CS, which encompasses both CSN and CSD instances, undergoes a crucial partitioning process into distinct CSN (normal) and CSD (DDoS attack) subsets.

Subsequently, harnessing the K-Means technique [15], the sessions within these subsets are individually clustered. This strategic clustering aids in the determination of the optimal number of clusters within the CSN and CSD sets. This culminates in the simultaneous computation of time frames for both CSN and CSD, adding a layer of precision to the analysis process.

Selecting characteristics to detect and respond to DDoS attacks effectively includes the web applications' session behaviour, request patterns, and pay load content . Features that are important for distributed denial of service attacks are given priority. These features include request mazda mx5rates at peak hours, pay load sizes larger than typical, and sudden increases in session creation.

These aspects may capture the unique patterns of traffic from DDoS attacks. In Figure 1, the web-based application layer works on a network Anomaly real-time Prevention Detection System; here, the ARTP system can be categorized into three network system models.

The first model will be designed to detect the DDoS attack on the network traffic models that can be represented as input models, and the second intrusion detection system model will be represented as the network analysis model [16].

The third network model will be represented as a DDoS attack detection generation alert message; this means whenever the attack is detected from the network application layer, an alert message will be generated and sent to the ARTP system, So this is the output model. In the analysis model, feature

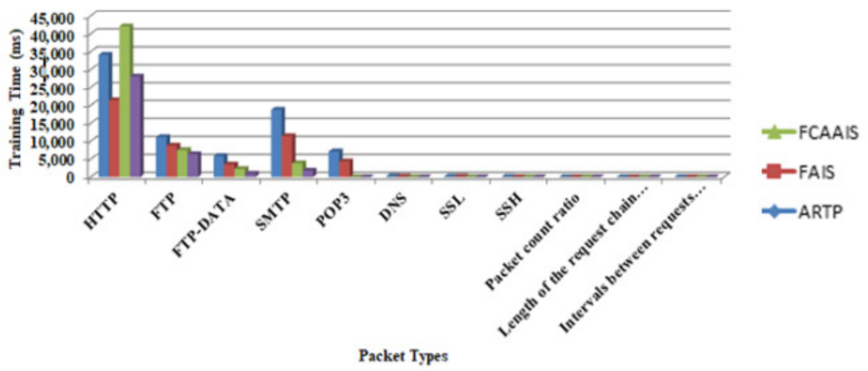

Fig. 2. Comparative analysis of the request for CSN tests and CSD training as a proposition

selections include data pre-processing, feature analysis and selection, machine learning classifier and attack recognition, i.e., DDoS attack detection model and decision making.

The above network models will be given the detection of DDoS attack rate and reduced attack rate by machine learning classifiers.

This paper outlines the procedures used to pre-process test data to train machine learning models to identify distributed denial-of-service (DDoS) attacks at the web-based application layer. Data formatting, normalization, and optimization for analysis are all aspects of pre-processing that guarantee accurate results.

Within the context of the amalgamated regular transaction group CS_N, let $C=\{c_1, c_2,\ldots,c_m\}$ denote clusters characterized by a spectrum of K values. Within the realm of each cluster $C_j\{c_j\exists c_j\in C \wedge j= 1, 2, 3,\ldots M\}$, the determination of time frames entails a calculation rooted in the discrepancy between the completion time of the most extensive session and the initiation time of the most succinct session.

Considering the cluster C_j, let $SB_N(C_j) = \{sb_1, sb_2, \ldots sb_{|cj|}\}$ represent an array of session start times arranged in ascending order. Here, sb1 corresponds to the session with the earliest start time, while $sb_{|cj|}$ corresponds to the one with the latest initiation time. Additionally, denote $SE_N(c_i)= \{se_1, se_2,\ldots se_{|ci|}\}$ as a sequence encompassing session end times within cluster c_i, with se1 marking the termination of the shortest session and $se_{|ci|}$ denoting the end of the longest one. The ensuing expression encapsulates the time frame of cluster c_j ($tf(c_j)$):

$$tf(c_j)=\sqrt{(se_1 - se_2)}^2. \tag{1}$$

The standard length of a time frame is computed as follows, taking into account all clusters:

$$<tf(c_j)>=\frac{\sum_{j=1}^{M} \quad tf(cj)}{M}. \tag{2}$$

The Absolute Deviation (tfAD) of time frames for all clusters is defined as:

$$tfAD=\frac{\sqrt{\sum_{j=1}^{M}} \quad (<tf(C)>-tf(Cj))2}{M}. \tag{3}$$

Finally, the time frame (tf) is determined by multiplying the average length of the time frame (<tf (C) >) with the Absolute Deviation of the time frame (tfAD):

$$tf=<tf(C)>+tfAD. \tag{4}$$

5 The Heuristics of Empirical Metrics

5.1 Request Chain Length (RCL)

In the context of a DDoS attack, the assailants orchestrate numerous queries directed at the Web server. Conversely, a tactic involving substantial HTTP requests aims at compromising websites; furthermore, such attacks can arise when genuine users inadvertently submit requests that vastly surpass typical sizes.

To preempt possible memory-related challenges on the server end, the range of requests within specific time frames is delineated

Table 2. The metric's values, as well as packettype ratios for different packet types

Types of packets	The ratio between packet types					
	N (normal) CS_N			D (DDoS Attack) CS_D		
	Training		Test	Training		Testing
	Minimum limit	Maximum limit		Minimum limit	Maximum limit	
HTTP	0.4384	0.46	0.36	0.68	0.75	0.77
FTP	0.1348	0.16	0.15	0.12	0.13	0.12
FTP-DATA	0.0627	0.07	0.07	0.02	0.05	0.02
SMTP	0.2368	0.28	0.23	0.053	0.07	0.07
POP3	0.0982	0.11	0.08	0.0	0.0	0.0
DNS	0.0025	0.004	0.03	0.0	0.0	0.0
SSL	0.0009	0.003	0.03	0.03	0.05	0.47

Table 3. Metrics for ARTP performance and actual out comes

True positive (tp)	The "normal" transactions are routine ones.				36,531
False Positive (fp)	The total number of legitimate transactions that were incorrectly marked as intrusions.				4244
True Negative (tn)	How many spoofed transactions are spoofed.				45,144
False Negative (fn)	The percentage of unapproved transactions that are mistakenly classified as regular.				541
Precision	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$	0.897	Accuracy	$\frac{(\mathrm{tp}+\mathrm{tn})}{(\mathrm{tp}+\mathrm{tn}+\mathrm{fp}+\mathrm{fn})}$	0.944
Recall/sensitivity	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$	0.985	F-Measure	2× (precision * recall)/ precision + recall	0.938
Specificity	$\frac{\mathrm{tn}}{\mathrm{fp}+\mathrm{tn}}$			0.914	

for both N (normal) and D (DDoS attack) scenarios, culminating in the creation of CSN and CSD.

Within CSN, $TS(CS_N) = \{ts_1, ts_2, \ldots ts_{|TS(CSN)|}\}$ dissects the progression of transactions within the normal set into distinct time frames. Here, ts_j represents the number of transactions received during the jth time frame, while tf denotes the temporal extent of the time frame, as elucidated in Section 3.1. The following model is utilized to calculate the average chain length of transactions observed across all time frames within CSN:

$$<\mathrm{TS}(\mathrm{CS_N})> = \sum_{\mathrm{j}=1}^{|\mathrm{TS(CSN)}|}\{|ts_j| \exists\, ts_j \in \mathrm{TS(CSn)}\}. \quad (5)$$

Within every CSN time frame cluster, the Absolute Deviation (clAD) of request chain length is computed using the following formula:

$$\mathrm{clAD} = \frac{\sqrt{\sum_{\mathrm{j}=1}^{|\mathrm{TS(CSn)}|} \left(<\mathrm{TS(CSn)}> - \left(|\mathrm{tsj}| \exists \mathrm{tsj} \in \mathrm{TS(CSn)}\right)\right)2}}{|\mathrm{TS(CSn)}|}. \quad (6)$$

Within each time frame of the CS collection, the Request Chain Length (RCL) is determined by amalgamating the average transaction chain <TS(CSN)> with the Absolute Deviation of RCL (clAD):

$$\mathrm{rcl}(CS_N) = <TS(CS_N)> + \mathrm{clAD}. \quad (7)$$

5.2 Packet Count Ratio

DDoS attacks deploy substantial bandwidth volumes to inundate their targets, rendering them inaccessible.Packet computing is crucial in data flow for tasks such as detecting counterfeit attacks,

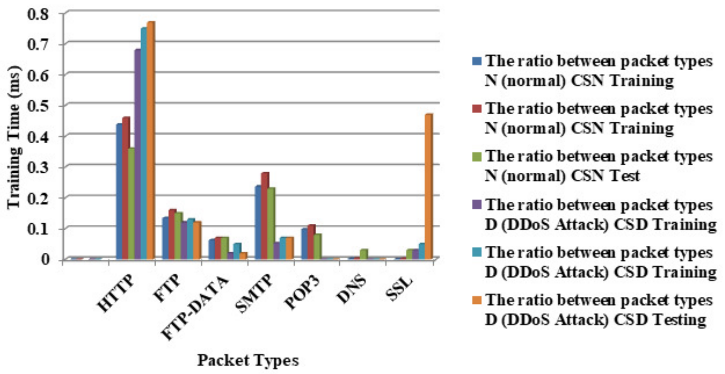

Fig. 3. Different packet types of metric values and packet type ratios

identifying unspoofedDDoS attacks, and functioning as a meter for flow coherence.

This calculating method assesses the packet processing level in the order of requests in normal and DDoS attack scenarios by analyzing the message volume during each period.

Determining if a DDoS assault has occurred relies on analyzing the volume of packets for each time increment.

The computation is governed by the number of packets per time frame, represented as tsj, which belongs to the set TS(CSN) and ranges from 1 to the total number of elements in TS(CSN). This reveals the complex relationship between particles and query patterns:

$$\text{rp}(\text{tj}_i)= \sum_{j=1}^{|TS(CSn)|} \frac{|P(tsj)|}{\sum_{j=1}^{|TS(CSn)|} |P(tsk)|}. \quad (8)$$

The period Floor Packets Support Relative Difference (tflpsAD) duration is calculated for every frame in the Argument Space Net (CSN) using the approach outlined below:

$$\text{tflpsAD}=\frac{\sqrt{\sum_{j=1}^{|TS(CSn)|} (1-rp(tsj))2}}{|TS(CSn)|}. \quad (9)$$

The total amount of packets in a CSN reflects the network's activities throughout all periods, while the Time Frame Tier Messages Support Relative Deviation offers a thorough assessment of particle intensity.

The time frame for Levels Packets Supported Absolute Deviation is obtained by adding these numbers:

$$\text{rpc}(\text{TS}(\text{CS}_\text{N}))=\frac{\sum_{j=1}^{|TS(CSn)|} rp(tsi)}{|TS(CSn)|}+tflpsAD. \quad (10)$$

5.3 Intervals between Request Ratios

The time elapsed between consecutive requests in order within a single session is calculated as the access time, starting from creating training-focused transactional sets. Each session produces a series of time frames for assessing the time intervals between requirements in regular situations and during DDoS assaults. Within CSN, every frame includes an Interval Total Deviation (iAD), calculated as follows:

$$\text{iAD}=\frac{\sqrt{\sum_{j=1}^{|TS(CSn)|} (gm(CSn)-lm(tsj))2}}{|TS(CSn)|}. \quad (11)$$

In conclusion, the degree of interval is ascertained by aggregating the mean value computed across all CSN intervals, along with including the Interval Absolute Deviation (iAD) concerning intervals within the training set CSN. This holistic approach encapsulates a comprehensive evaluation of interval dynamics:

$$\text{ri}(\text{CS}_\text{N})=\frac{\sum_{j=1}^{|TS(CSn)|} lm(tsj)}{|TS(CSn)|}+\text{iAD}. \quad (12)$$

5.4 Packet Type Ratio in a Predetermined Time Frame

The term "packet ratio threshold" refers to the fraction of unique packet types, such as DNS,

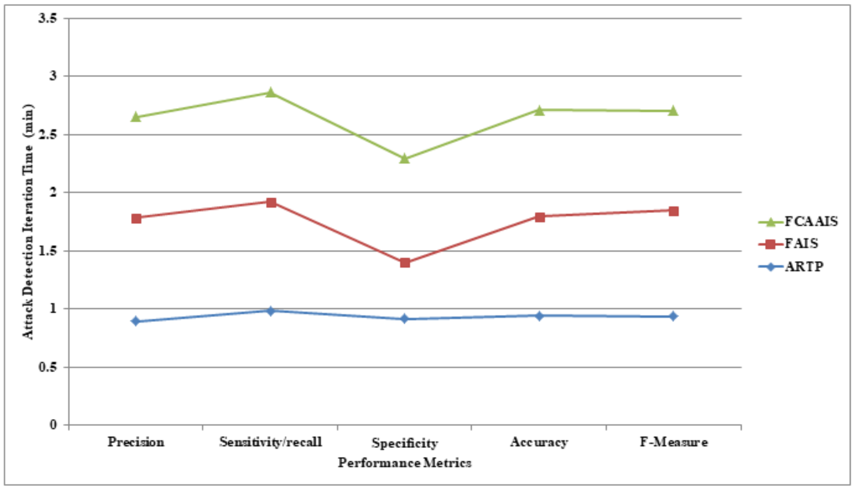

Fig. 4. A comparison is presented between FCAAIS, FAIS and ARTP

SMTP, FTP-DATA, SSL, POP3, HTTP, and FTP, that occur within a given period that is less than the fraction of unique packet types, such as FTP-DATA, POP3, HTTP, DNS, SMTP, and FTP, that occur within the transactions of N (normal) as CSN and D (DDoS attack) as CSD.

For each packet type included in the order of requests analyzed within ts_l, the local support within CSN, represented by ts_i $\{ts_i \exists\ ts_i \in TS(CS_N)$ ^ $i= 1,2,3,\ldots|TS(CS_N)|\}$ of CS_N, establishes the threshold. Moving on, for every type of packet, represented by pt_k $\{pt_k \exists\ pt_k \in PT$^ $k= 1,2,\ldots,|PT|\}$, the Absolute Deviation (ptsAD) of packet type support is investigated across all time frames within CS_N, juxtaposing local and global support. This evaluation unfolds in the following manner:

$$\text{ptsAD(ptk)} = \frac{\sqrt{\sum_{i=1}^{|TS(CSn)|} \quad (gs(ptk)-lstsi(ptk))2}}{|TS(CSn)|}. \tag{13}$$

In the concluding phase, the degree of ptk (rpt(ptk))is determined through the summation of the average packet threshold type and the Absolute Deviation of packet type support (ptsAD). This computation follows the sequence of time lengths within the training set CSN, encompassing various packet types:

$$\text{rpt}(\text{pt}_k)= \frac{\sum_{i=1}^{|TS(CSn)|} \quad \text{lstsi(ptk)}}{|TS(CSn)|} + \text{ptsAD}(\text{pt}_k). \tag{14}$$

5.5 Order of Requests or Request Chain Context

Given the ensemble of generated transactions designated for training, the progression involves segmenting the requests into discrete time frames corresponding to regular and DDoS attack instances. a request pair set rps_N is formulated within the training transaction set CSN, encompassing pairs $rps_N = \{p_1, p_2,\ldots,p_{|rpsN|}\}$, where each pair p_i represents the immediate two consecutive requests within the CSN sequence.

The local support ls_{tsj} (pi) attributed to ls_{tsj} (p_i) of p_i $(p_i \exists\ p_i \in rps_N$ ^ $i=1, 2,\ldots, |rps_N|)$ quantifies the occurrences of the pair pi within time frame ts_j.

The training dataset CS_N includes all recorded sequences of requests, and for each pair p_i $(p_i \exists\ p_i \in rps_N$ ^ $i=1,2,\ldots,|rps_N|)$, the cumulative support $gs(p_i)$ includes all instances of the pair.

This iterative process culminates in a comprehensive understanding of request pair dynamics. When do these conditions shift from supporting pi locally to supporting it globally?

The Absolute Deviation (rpsAD), an abbreviation of relative standard deviation, is calculated for each pi $(p_i \exists\ p_i \in rps_N$ ^ $i=1, 2,\ldots,|rps_N|)$, This calculation is expressed as follows:

Table 4. FAIS and FCAAIS are use DTO compare the suggested ARTP technique

	ARTP	FAIS	FCAAIS
Precision	0.938	0.911	0.855
Sensitivity/recall	0.944	0.851	0.917
Specificity	0.915	0.496	0.885
Accuracy	0.985	0.935	0.942
F-Measure	0.895	0.889	0.869

$$\text{rpsAD}=\sqrt{\frac{\sum_{j=1}^{|TS(CSn)|}(gs(pi)-lstsj(pi))2}{|TS(CSn)|}}. \quad (15)$$

The mean value derived from all the pair carry values is computed across the sequence of time frames within the training set CS_N. Moreover, for every possible pairing that arises within the scope of this paper, the Absolute Deviation (rpsAD) of request pair support is explored. The sequence ($p_i\exists$ $p_i\in rps_N$ ^ i=1,2,…,$|rps_N|$) is being evaluated in parallel on the request chain context rcc(p_i):

$$\text{rcc}(p_i)=\frac{\sum_{j=1}^{|TS(CSn)|} lstsj(pi)}{|TS(CSn)|}+\text{rpsAD}. \quad (16)$$

Organizations can enhance their ability to detect and mitigate web-based application layer DDoS attacks, minimize disruption to their online services, and protect themselves from potential security threats by utilizing attack recognition techniques, deploying detection models based on machine learning, and building adaptive decision-making processes.

6 Outcomes of Experiments and Performance Evaluation

The tests have come to a close to evaluating the suggested model ARTP's, resilience, process complexity, Scalability and detection accuracy.

6.1 Experimental Results

To simulate application layer DDoS attack scenarios, the framework LLDOS 2.0.2 [17] is harnessed under normal and attack conditions. For the testing phase, 229,386 transactions are processed, encompassing N (normal) and D (disruptive) transactions representative of DDoS attacks.

60% of this dataset has been set aside for training, while the remaining 40% will be used for testing.Our study uses synthetic transactional data imitating web application traffic as test data. A custom-built data-generating tool was used to produce these transactions, which were meant to simulate user interaction swith web-based applications.

Assessed independently using the CS dataset, each metric is generated, comprising CSN (normal) and CSD (DDoS attack) instances. CSN comprises a total of 123,780 transactions, out of which 60 percent (74,260) are dedicated to training, and the remaining 40 percent (49,520) are reserved for testing.

Sessions in the CSN [18] training and testing datasets are evenly divided into 1-minute and 10-second intervals. As a next step, the K-Means technique is applied separately to both the training and testing phases to establish the locations of any existing clusters.

The proposed K means technique's efficiency is compared with other unsupervised learning techniques, such as Principle Component Analysis and the Apriori algorithm.

Similarly, the assault dataset CSD is divided into sessions and clusters using the same manner to facilitate training and testing. Additionally, temporal frames (as detailed in Sect. 3.1) are synthesized from the stream of sessions, subsequently defining the temporal scope of the CSN and CSD training and evaluation phases in Table 1.

6.2 Request Chain Length (RCL)

The request chain length is characterized by the maximum count of requests originating from a transaction. This concept is expanded in the CSN and CSD training environment to incorporate the typical length of requests recorded from clients within a time frame described as either an attack or a normal scenario.

The goal is to better equip pupils to deal with real-world difficulties. A comprehensive overview of these details is meticulously presented in Table 2.

6.2.1 Packet Count Ratio

Within the training set, the ratio of packet counts is calculated by tallying the total number of packets received during each interval along the request chain, which is split into CSN and CSD. A chain of requests undergoes this analysis. All the while the CSN and CSD teams are being trained and tested, a thorough examination of the packet calculating process is being conducted.

FTP-DATA, SSL, HTTP, POP3, DNS, and SMTP packets are only a few that fall inside this inquiry's scope. These computations encompass calculating the packet ratio across all time frames within CSN and CSD. The detailed results of these computations are meticulously presented and recorded in Table 2.

6.2.2 The Ratio of Request Intervals

The table 2 summarises extracted metrics from the training sets in depth. Request Chain Length (RCL), Request Interval Ratio, Packet Count, and Packet Count Ratio are all metrics that may be measured across CSN and CSD.

These metrics have been extracted from the designated training datasets and are presented in Tables 2(a) & (b). "approach time" is used in CSN and CSD transactions to describe the time between the first and last requests made within a given session. For each period, the ratio of intervals between requests is calculated by painstakingly calculating the approach time for each pair of requests within the same session.

CSN and CSD packet types, request chain length (RCL), packet count ratio, and RCL are all measured using the provided training sets. In Figure 2, the representations of types of packets were tested on normal attack of CSN packet count of N and DDoS attack of CSD packet count D. The resultant graph will be analyzed on training and testing of CSN packet count of N and CSD packet count of D.

6.2.3 Ratio of Packets Types

Training and testing procedures that use the CSN and CSD datasets allow for examining ratios associated with threshold creation for each packet type. Internet Protocol (HTTP), File Transfer Protocol (FTP) (including FTP-DATA), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and Secure Sockets Layer (SSL) packets are all included here.

The required ratios are detailed in Table 2 below. These tables owe a great deal to the CSN and CSD training sets. This set of ratios encapsulates dissimilar packet types constituting a substantial portion of commonly employed packet types within the application layer.

6.3 Performance Analysis

Precision, sensitivity (true positive rate), specificity (true negative rate), accuracy, and F-measure are some of the key parameters that can be used to assess the usefulness of ARTP's detection capabilities. Quantitative parameters can include accuracy, sensitivity (the proportion of correct diagnoses), and specificity (the proportion of incorrect diagnoses).

Similarly, a quantitative parameter is specificity. As showcased in Table 4, this procedure's statistics offer insights into its performance. Predicted requests are accurately classified as normal when they are, and expected attacks are correctly labelled as attacks.

As determined in the experiments, sensitivity guides the identification of true positives projected as positive outcomes. Meanwhile, true negatives that are accurately anticipated as negatives are attributed to specificity. Accuracy is also notable, signifying the duration required for precise request classification.

In other words, the evaluation shows a lower risk of misidentifying a normal condition as an attack than misclassifying an attack configuration as normal; hence the sensitivity rating is higher than the specificity rating. The evaluation shows that the sensitivity is greater than the specificity.

An attack misclassified as a normal request has a 1-sensitivity rate of 0.0145, whereas misclassifying a normal request as an attack has a 1-specificity rate of 0.0859. There has been a significant performance boost for each of these numbers.

Consequently, it is reasonable to assert that the suggested ARTP demonstrates increased effectiveness in detecting attacks, as evident from the numerical performance metrics and practical observations in Table 3. In contrast, the Fuzzy Artificial Immune System FAIS [19] and feature correlation analysis and association impact scale (FCAAIS) [20-21] models are proposed for DDoS attack detection.

Conducted on the same dataset, these models exhibit scalability and resilience in forecasting the scope of network DDoS attacks, achieving an approximate detection accuracy of 91%. However, when juxtaposed with the proposed ARTP model, these approaches encounter challenges stemming from process complexity, particularly regarding the statistical metrics employed for performance calculation. Notably, as displayed in Table 4 and Fig. 4, the precision of our proposed ARTP model surpasses both FAIS and FCAAIS, exemplifying higher predictive accuracy.

In Figure 3, the resultant graph will depict different packet types of metric values and packet type ratios. All packets compared to the ratios between the packet types of normal attack CSN and DDoS attack CSD with minimum and maximum limits will be done in the testing phase. Finally, the resultant graph will showthe accuracy of all types of packets.

Figure 4 compares the ARTP with FCAAIS and FAIS based on precision, recall, specificity, F-measure and accuracy. These metrics, including precision, recall, specificity, F-measure, and accuracy, were employed to evaluate the efficacy of each model in various classification or pattern recognition tasks [22, 23].

7 Conclusion

This work suggests an approach to machine learning based on empirical data to identify, protect against, and stop denial of provider (DDoS) assaults at the app level in actual time on various web services. The author of this essay diligently attempts to classify the article's contributions into three separate categories.

The initial contribution is to evaluate particular metrics of a demand stream to determine if the intent is malicious or benign. Experimental threshold values derived from the measurements in the ARTP framework are used to determine if a stream's behaviour meets the criteria for being classified as a flood.

The second contribution focuses on observing ARTP behaviour, supported by thorough testing on the LLDoS dataset. This component combines perceived procedural complexities with appropriate speed to enhance detection accuracy. The results show that the indications derived from the training data set, which is used to assess the status of the request stream, are very promising and essential.

The final addition uses threshold values from the training set Report Phrase to identify if a request stream could indicate a software layer DDoS assault over time. ARTP's resilience is thoroughly tested under simple settings and high speeds. The technique outlined in this research shows a surprising capacity to maintain high forecast accuracy while also reducing computing workload, proving its efficiency in operation.

References

1. **Mahdavi-Hezavehi, S., Rahmani, R. (2020).** An anomaly-based framework for mitigating effects of DDoS attacks using a third party auditor in cloud computing environments. Cluster Computing, Vol. 23, No. 4, pp. 2609–2627. DOI: 10.1007/s10586-019-03031-y.
2. **Krishna-Kishore, P., Ramamoorthy, S., Rajavarman, V. (2023).** ARTP: Anomaly based real time prevention of distributed denial of service attacks on the web using machine learning approach. International Journal of Intelligent Networks, Vol. 4, pp. 38–45. DOI: 10.1016/j.ijin.2022.12.001.
3. **Byers, S., Rubin, A. D., Kormann, D. (2004).** Defending against an internet-based attack on the physical world. ACM Transactions on Internet Technology, Vol. 4, No. 3, pp. 239–254. DOI: 10.1145/1013202.1013203.
4. **Estevez-Tapiador, J., Garcia-Teodoro, P., Diaz-Verdejo, J. (2005).** Detection of web-based attacks through Markovian protocol parsing. 10th IEEE symposium on computers and communications, Vol. 2396, pp. 457–462. DOI: 10.1109/iscc.2005.51.
5. **Kishore, P. K., Rajavarman, R. V. (2023).** Detection of DDoS enabled flood attacks using an ensemble classifier in distributed networks. International Journal of Intelligent Systems

and Applications in Engineering, Vol. 11, No. 3, pp. 578–590.

6. **Yatagai, T., Isohara, T., Sasase, I. (2007).** Detection of HTTP-GET flood attack based on analysis of page access behavior. IEEE Pacific rim conference on communications, computers and signal processing, pp. 232–235. DOI: 10.1109/pacrim.2007.4313218.
7. **Sivatha-Sindhu, S. S., Geetha, S., Kannan, A. (2012).** Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with Applications, Vol. 39, No. 1, pp. 129–141. DOI: 10.1016/j.eswa. 2011.06.013.
8. **Shevtekar, A., Ansari, N. (2009).** Is it congestion or a DDoS attack? IEEE Communications Letters, Vol. 13, No. 7, pp. 546–548. DOI: 10.1109/lcomm.2009.090628.
9. **Fouladi, R. F., Ermiş, O., Anarim, E. (2022).** A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN. Computer Networks, Vol. 214, pp. 109140. DOI: 10.1016/j.comnet. 2022.109140.
10. **Kishore, P. K., Ramamoorthy, D. S., Rajavarman, D. (2019).** Detection, defensive and mitigation of DDoS attacks through machine learning techniques: A literature. International Journal of Recent Technology and Engineering, Vol. 8, No. 4, pp. 2719–2725. DOI: 10.35940/ijrte.d7335.118419.
11. **Muhati, E., Rawat, D. (2024).** Data-driven network anomaly detection with cyber attack and defense visualization. Journal of Cybersecurity and Privacy, Vol. 4, No. 2, pp. 241–263. DOI: 10.3390/jcp4020012.
12. **Farid, T., Sirat, M. (2023).** Hybrid of supervised learning and optimization algorithm for optimal detection of IoT distributed denial of service attacks. International Journal of Innovative Computing, Vol. 13, No. 1, pp. 1–12. DOI: 10.11113/ijic.v13n1.329.
13. **Demilie, W. B., Deriba, F. G. (2022).** Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. Journal of Big Data, Vol. 9, No. 1. DOI: 10.11 86/s40537-022-00678-0.
14. **Wisanwanichthan, T., Thammawichai, M. (2021).** A double-layered hybrid approach for network intrusion detection system using combined Naive Bayes and SVM. IEEE Access, Vol. 9, pp. 138432–138450. DOI: 10.1109/access.2021.3118573.
15. **Xie, Y., Shun-Zheng, Y. (2009).** A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors. IEEE/ACM Transactions on Networking, Vol. 17, No. 1, pp. 54–65. DOI: 10.1109/tnet.2008. 923716.
16. **MIT (1998).** 1998 DARPA intrusion detection evaluation dataset. Darpa Intrusion Detection Evaluation. https://www.ll.mit.edu/ideval/data/ 1998data.html.
17. **Hartigan, J. A., Wong, M. A. (1979).** Algorithm AS 136: a K-means clustering algorithm. Applied Statistics, Vol. 28, No. 1, pp. 100. DOI: 10.2307/2346830.
18. **Powers, D. M. (2006).** Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. 23rd international conference on machine learning, pp. 37—63. DOI: 10.48550/arXiv.2010. 16061.
19. **Jyothsna, V., Rama-Prasad, V. (2016).** FCAAIS: Anomaly based network intrusion detection through feature correlation analysis and association impact scale. ICT Express, Vol. 2, No. 3, pp. 103–116. DOI: 10.1016/j.icte. 2016.08.003.
20. **SenthamilSelvan, R., Wahidabanu, R. S. D., Karthik, B. (2020).** Intersection collision avoidance in dedicated short-range communication using vehicle ad hoc network. Concurrency and Computation: Practice and Experience, Vol. 34, No. 13. DOI: 10.1002/ cpe.5856.
21. **Gelbukh, A., Pérez-Alvarez, D. A., Kolesnikova, O., Chanona-Hernandez, L., Sidorov, G. (2024).** Multi-instrument based N-grams for composer classification task. Computación y Sistemas, Vol. 28, No. 1. DOI: 10.13053/cys-28-1-4903.
22. **Adebanji, O. O., Ojo, O. E., Calvo, H., Gelbukh, I., Sidorov, G. (2024).** Adaptation of transformer-based models for depression

detection. Computación y Sistemas, Vol. 28, No. 1. DOI: 10.13053/cys-28-1-4691.

Article received on 28/02/2024; accepted on 15/05/2024.
**Corresponding author is K Alluraiah.*