Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

WEB-BASED APPLICATION LAYER DISTRIBUTED DENIAL-OF-SERVICE ATTACKS: A DATA-DRIVEN MACHINE LEARNING STRATEGY

K ALLURAIAH
Research Scholar, Department of Computer Science & Engineering from Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, A.P.-522302, India. Email: allura02@gmail.com

Dr. MANNA SHEELA RANI CHETTY
Professor, Department of Computer Science & Engineering from Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, A.P.-522302, India. Email: 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 in order to disrupt the interactions of legitimate users. These attacks operate by inundating the victim's system with a deluge of packets, rendering it inaccessible. Diverging from the singular source of Denial of Service (DoS) attacks, DDoS attacks emanate from a multitude of 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, 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. 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 come to the fore as major problems. A DoS attack orchestrates a direct attack on a chosen target system, often a web server, with the intent of incapacitating its ability to serve legitimate users, all while refraining from intruding upon the underlying network or server infrastructure. Conversely, DDoS attacks employ a distributed array of devices scattered across the Internet, collectively generating a torrent of traffic that inundates the targeted systems, depleting their resources and distorting network integrity, thereby rendering the systems inaccessible to genuine users.

Network layer DDoS attacks and application layer DDoS attacks are the two main types of distributed denial of service attacks [1]. Both types are typical in the world of distributed denial of service attacks. The first form of attack occurs when an adversary uses techniques like IP spoofing to bombard the target system with counterfeit data packets. However, application layer DDoS attacks, or App-DDoS attacks as they are more commonly known, involve overwhelming the targeted system with a large number of valid requests. In these attacks, the assailant's compromised computers need to establish valid TCP connections with the victim's system, else the connections would be severed. A notable example here is the HTTP flood, a method that leverages surges in HTTP requests to strain the targeted servers' capabilities. Functioning at the application level, these attacks aim to besiege the host server of online applications by inundating it with an avalanche of simultaneous and seemingly legitimate requests, culminating in rendering the server unreachable for legitimate clients.

In an effort to address and comprehend these intricate tactics, recent works, such as [2], delve into novel strategies to bypass DDoS defenses and uncover tactics that facilitate more efficient attack collaborations. Notably, the HTTP flood technique amplifies the volume of HTTP requests, orchestrating a surge in transactional interactions to exploit the vulnerabilities in the targeted servers' architecture [3, 4]. Amidst this inundation, the targeted servers grapple with the challenge of differentiating between legitimate payload data and the deluge of seemingly ordinary HTTP requests.

2. RELATED WORK

Some of the DDoS defense solutions tailored for countering HTTP floods exhibit a reliance on application layer insights, as indicated by a recent survey [5]. An example of such is the DDoS shield [6], which employs the detection of session initiation times and the manipulation of time intervals between arrivals to thwart HTTP floods. An additional noteworthy approach, investigated in the work "Using Page Access Lead to Fight HTTP Flood" [7, 8], delves into the intricate relationship between data sizes and browsing durations. Yet, these methodologies fall short when confronted with the formidable packet transmission rates facilitated by the coordinated efforts of malicious Botnets [9]. Introducing a CAPTCHA-based probabilistic validation system, while effective in rare cases, risks unduly burdening the user experience. This could potentially translate into an inadvertent Denial of Service scenario, particularly when addressing practical grievances is paramount. The Hidden semi-Markov model (HSMM) is introduced in [10] for this purpose; it utilises the specifics of transactional request instances to decode typical client access patterns. This model further evaluates the ongoing influx of clients using HSMM-derived data. However, this approach yields a substantial volume of false alerts, attributable to the variability in invocation patterns, especially considering the diverse browsing behaviors of actual users. Instances where clients engage external online interfaces, input URLs for potential queries, or navigate with a medley of browsers can inadvertently trigger false alarms, adding an element of uncertainty to the equation. The primary objective of the proposed methodology centers on assessing transactional correlations, drawing from both regular and flood data. In contrast to conventional benchmarking techniques, the focal points of this proposed model derive from extensive

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

observations of the request stream across an extended timeframe. The subsequent sections of this paper meticulously follow a well-organized trajectory: Section 3 elaborates on the proposed ARTP model, while Section 4 delves into a comprehensive examination of a test study and evaluates the efficacy of ARTP. Finally, Section 5 culminates this scholarly endeavor by offering conclusive insights.

3.WEB-BASED APPLICATION LAYER DISTRIBUTED DENIAL-OF-SERVICE ATTACKS: A DATA-DRIVEN MACHINE LEARNING STRATEGY

To answer the danger posed by Application Layer DDoS attacks on the internet, we have designed and fine-tuned an anomaly-based Real-Time Prevention (ARTP) framework [11]. The overwhelming number of requests at once inspired the development of this framework. The system's "speedy and early detection" of Application Layer DDoS attacks has been a huge success, making that goal one of its most significant achievements. The essence of the ARTP model lies in its multifaceted entity metrics, encompassing crucial facets such as "request chain length, variations in packet count, permutations in packet sorting, intervals between requests, and the contextual backdrop of transactional chains." Importantly, a multitude of prevalent benchmarking methods, often hinged on sessions or requests as primary inputs, inadvertently bypass the fundamental basis of this methodology. To evaluate the effectiveness and potency of the proposed model, comprehensive testing was undertaken utilizing the LLDoS dataset, underscoring both its adaptability and robustness. It is noteworthy that the envisaged model exhibits remarkable efficacy within the realm of application layer DDoS attacks. This holds especially true given the contemporary landscape of web applications, where the sheer magnitude of incoming requests has escalated to the scale of petabytes, dwarfing the gigabyte-level loads of previous web request paradigms.

3.1 Finding the Length of a Time Frame

The acronym $CS=\{s_1, s_2,\ldots,s_n\}$ encapsulates an assemblage of Client Sessions. Each session within this collection, denoted as $\{l \exists l \in s_i \wedge s_i \in CS\}$, encompasses transactions categorized as either N (normal) or D (disordered, signifying a DDoS attack). The aggregate count of transactions in CS encompasses both regular (CS_N) and anomalous (CS_D) transactions, jointly forming the composite volume of transactions associated with DDoS attacks. The heuristic measures, outlined in Section 3, will undergo validation through experimentation employing these datasets. To provide further clarification, the dataset CS, which encompasses both CS_N and CS_D instances, undergoes a crucial partitioning process into distinct CS_N (normal) and CS_D (DDoS attack) subsets. Subsequently, harnessing the K-Means technique [12], the sessions within these subsets are individually clustered. This strategic clustering aids in the determination of the optimal number of clusters within the CS_N and CS_D sets. This culminates in the simultaneous computation of time frames for both CS_N and CS_D, adding a layer of precision to the analysis process.

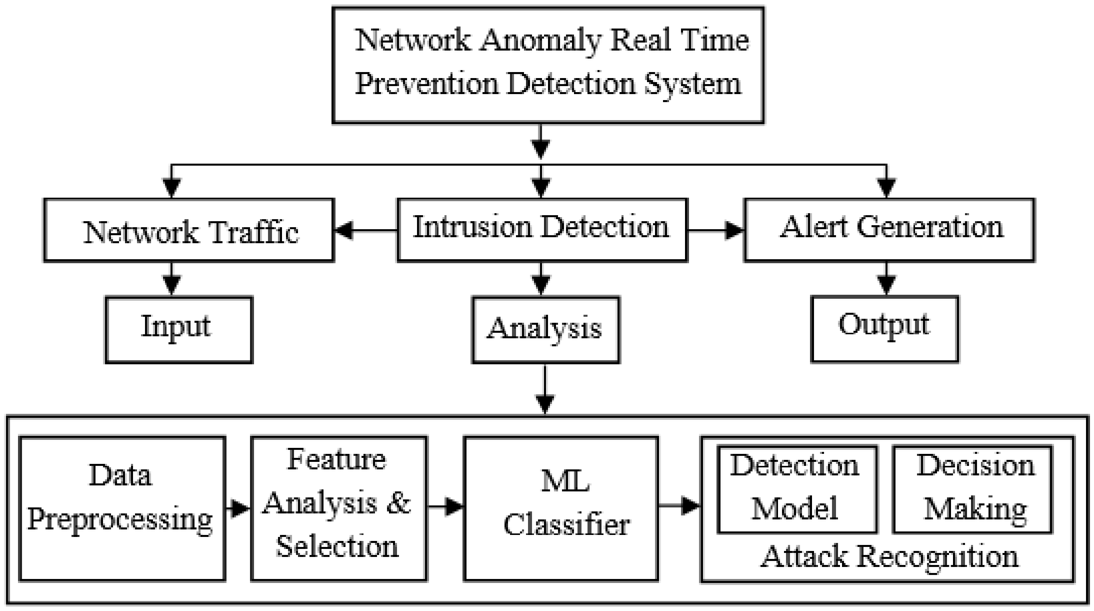

Figure 1: The architecture of Network Anomaly Real time Prevention Detection System

In fig. 1, web based application layer works on network Anomaly Real time Prevention Detection System, here the ARTP system can be categorized into three network systems models, the first model will be designed for detecting 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 network analysis model. The third network model will be represented as DDoS attack detection generation alert message this means whenever getting the attack detection from the network application layer automatically alert message will be generated and sent to the ARTP system, So this is the output model. In the analysis model, there are some feature selections like data preprocessing, feature analysis and selection, machine learning classifier and attack recognition i.e., DDoS attack detection model and decision making. The above all network models will be given detection of DDoS attack rate and the will be reduced attack rate by machine learning classifiers.

Within the context of the amalgamated regular transaction group CS_N, let C= {c_1, c_2… cm} denote a collection of clusters characterized by a spectrum of K values. Within the realm of each cluster C_j{$c_j \exists c_j \in C$^ j= 1, 2, 3,…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 $se_{|ci|}$

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

denoting the end of the longest one. The ensuing expression encapsulates the time frame of cluster c_j ($tf(c_j)$):

$$tf(c_j)=\sqrt{(se1-se2)^2} \quad (1)$$

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

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

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

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

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).

$$tf=<tf(C)>+tfAD \quad (4)$$

4. THE HEURISTICS OF EMPIRICAL METRICS

4.1 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, instances of 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 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 j^{th} time frame, while tf denotes the temporal extent of the time frame, as elucidated in Section 3.1.

The following algorithm is utilized to calculate the average chain length of transactions observed across all time frames within CSN:

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

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

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

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).

$$rcl(CS_N) = <TS(CS_N)> + clAD \quad (7)$$

4.2 Packet Count Ratio

DDoS attacks deploy substantial bandwidth volumes to inundate their targets, rendering them inaccessible. Packet computation stands as a pivotal facet of any data flow, serving multifaceted roles such as counterfeit attack identification, unspoofed DDoS attack recognition, and most notably, acting as a flow coherence metric. Based on an analysis of the packet volume for each time period, this calculation approach evaluates the level of packet computation within the sequence of requests for both the N (normal) CSN and D (DDoS attack) CSD situations. In other words, the foundation for establishing whether or not a DDoS attack has happened is the analysis of packet volume for each time interval. Within this framework, the count of packets per time frame, denoted as ts_j $\{ts_j \exists\ ts_j \in TS(CS_N) \wedge =1,2,\ldots|TS(CS_N)|\}$, governs the computation, revealing the intricate interplay between packets and request patterns.

$$rp(tj_i)= \sum_{j=1}^{|TS(CSn)|} \frac{|P(tsj)|}{\sum_{j=1}^{|TS(CSn)|} |P(tsk)|} \tag{8}$$

Time Frame Level Packets Support Absolute Deviation (tflpsAD) is determined for each time frame within the Contention Space Network (CSN) in the method described below.

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

The total number of packets in a CSN is representative of the network's activity because it represents the average over all time periods, whereas the Time Frame Level Packets Support Absolute Deviation provides a comprehensive measure of packet intensity. We get the Time Frame Level Packets Support Absolute Deviation by adding these two quantities together.

$$rpc(TS(CS_N))= \frac{\sum_{j=1}^{|TS(CSn)|} rp(tsi)}{|TS(CSn)|} + tflpsAD \tag{10}$$

4.3 Intervals between Request Ratios

The interval between successive requests within a sequence, pertaining to the same session, is computed as the access time, initiating with the compilation of training-oriented transaction sets. For every session, a succession of time frames is generated, aiming to appraise the temporal span between requisites, both within normal contexts and during DDoS attacks. Within the scope of CS_N, each time frame encompasses an Interval Absolute Deviation (iAD), computed in the ensuing manner:

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

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

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

$$ri(CS_N)= \frac{\sum_{j=1}^{|TS(CSn)|} lm(tsj)}{|TS(CSn)|} + iAD \tag{12}$$

4.4 Packet Type Ratio in a Predetermined Time Frame

The term "packet ratio threshold" refers to the fraction of unique packet types, such as DNS, SMTP, FTP-DATA, SSL, POP3, HTTP, and FTP, that occur within a given time 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_I, the local support within CSN, represented by ts_i $\{ts_i \exists\ ts_i \in TS(CS_N) \wedge i= 1,2,3,\ldots|TS(CS_N)|\}$ of CS_N, establishes the threshold. Moving on, for every type of packet, represented by pt_k $\{pt_k \exists\ pt_k \in PT \wedge k= 1,2,\ldots,|PT|\}$, 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:

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

In the concluding phase, the degree of pt_k $(rpt(pt_k))$is determined through the summation of the average packet threshold type and the Absolute Deviation of packet type support (ptsAD). This computation transpires in accordance with the sequence of time lengths within the training set CS_N, encompassing various packet types.

$$rpt(pt_k)= \frac{\sum_{i=1}^{|TS(CSn)|} lstsi(ptk)}{|TS(CSn)|} + ptsAD(pt_k) \tag{14}$$

4.5 Order of Requests or Request Chain Context

Given the ensemble of generated transactions designated for training, the progression involves segmenting the sequence of requests into discrete time frames corresponding to both 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 $(p_i \exists\ p_i \in rps_N \wedge i=1, 2,\ldots, |rps_N|)$ 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 $(p_i \exists\ p_i \in rps_N \wedge i=1,2,\ldots,|rps_N|)$ 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 start to shift from supporting pi locally to supporting it globally? The Absolute Deviation *(rpsAD)*, an abbreviation of relative standard deviation, is calculated for each *pi* $(p_i \exists\ p_i \in rps_N \wedge i=1, 2,\ldots,|rps_N|)$, This calculation is expressed as follows:

$$rpsAD= \sqrt{\frac{\sum_{j=1}^{|TS(CSn)|}\left(gs(pi)-lstsj(pi)\right)2}{|TS(CSn)|}} \tag{15}$$

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

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 *($p_i \exists\ p_i \in rps_N$ ^ $i=1,2,\ldots,|rps_N|$)* is being evaluated in parallel on the request chain context rcc(p_i).

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

5. OUTCOMES OF EXPERIMENTS AND PERFORMANCE EVALUATION

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

5.1 Experimental Results

To simulate application layer DDoS attack scenarios the framework LLDOS 2.0.2 [13, 14] is harnessed under both normal and attack conditions. For the testing phase, a total of 229,386 transactions are processed, encompassing both N (normal) and D (disruptive) transactions representative of DDoS attacks. 60% of this dataset has been set aside for training purposes, while the remaining 40% will be used for testing.

Assessed independently using the CS dataset and each metric is generated, which comprises CS_N (normal) and CS_D (DDoS attack) instances. CS_N 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 [15] 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. 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 CS_N and CS_D training and evaluation phases in table 1.

5.2 Request chain length (RCL)

The request chain length is characterized by the maximum count of requests originating from a transaction. In fact, 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 difficulties in the actual world. A comprehensive overview of these details is meticulously presented in Table 2.

a. Packet Count Ratio

Within the training setting, 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

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

into two groups: CSN and CSD. There is a chain of requests that 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 few of the many that fall inside the scope of this inquiry. These computations encompass the calculation of 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.

b. The Ratio of Request Intervals

Extracted metrics from the training sets are also summarised in depth in the table. Request Chain Length (RCL), Request Interval Ratio, Packet Count, and Packet Count Ratio are all metrics that may be measured across both CSN and CSD. All of these metrics have been extracted from the designated training datasets and are presented in Table 2. The term "approach time" is used in the context of CSN and CSD transactions to describe the duration of time between the first and last requests made within a given session. For each time period, the ratio of intervals between requests is calculated by painstakingly calculating the approach time for each pair of requests within the same session.

Table 1: CSN and CSD packet types, request chain length (RCL), packet count ratio, and RCL are all measured using the provided training sets

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

Table 2: Shows the distribution of requests for CSN testing and CSD training as a proportion. The 11th and 14th hours were very different from one another.

Types of packets	***CS_N*packet count of N (normal)**		***CS_D* packet count D (DDoS Attack)**	
	Training	**Testing**	**Training**	**Testing**
HTTP	34,117	21,470	42,088	28,106
FTP	11,146	8798	7540	6410
FTP-DATA	5857	3527	2308	982
SMTP	18,813	11,397	3836	1846
POP3	7168	4362	0.0	0.0
DNS	285	207	0.0	0.0
SSL	143	138	0.0	0.0
SSH	57	23	0.0	0.0
Packet count ratio	0.767	0.507	0.534	0.307
Length of the request chain (RCL)	22.254	21.735	28.408	29.10
Intervals between requests ratio	0.351	0.140	0.223	0.155

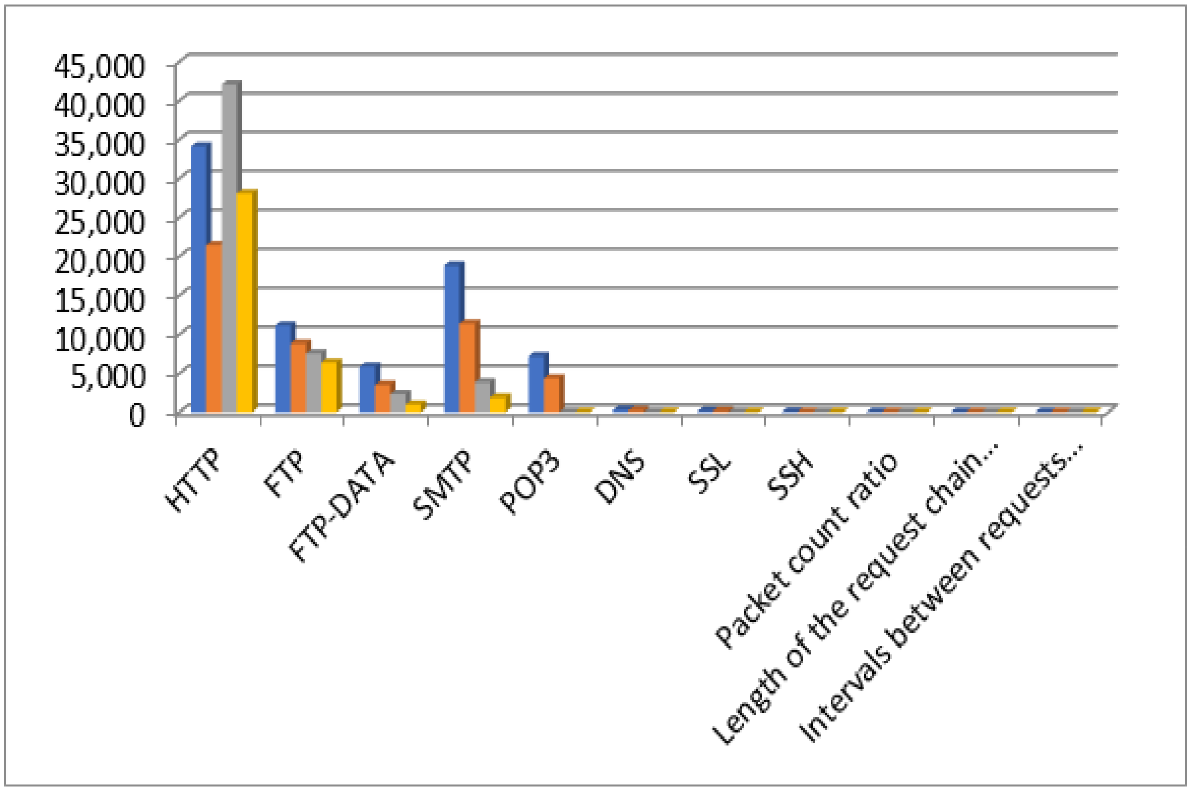

Figure 2: Comparative analysis of request for CSN tests and CSD Training as a Proposition

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.

c. Ratio of Packets Types

Training and testing procedures that use the CSN and CSD datasets allow for the examination of 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 3 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.

5.3 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 things like accuracy, sensitivity (the proportion of correct diagnoses), and specificity (the proportion of incorrect diagnoses). Similarly, a quantitative parameter is specificity. As showcased in Table 5, this procedure's statistics offer insights into its performance. Predicted requests are accurately classified as normal when they indeed are, and expected attacks are correctly labeled as attacks as well. Sensitivity, as determined in the experiments, guides the identification of true positives projected as positive outcomes. Meanwhile, true negatives that are accurately anticipated as negatives are attributed to specificity. Accuracy, signifying the duration required for precise request classification, is also a notable measure.

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 across the board 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 documented in Table 4.

In contrast, the FAIS [16] and FCAAIS [17] 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 in terms of the statistical metrics employed for performance calculation. Notably, as displayed in Table 5 and Fig. 4, the precision of our proposed ARTP model surpasses both FAIS and FCAAIS, exemplifying higher predictive accuracy.

Table 3: The Metric's Values, as well as Packet type 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

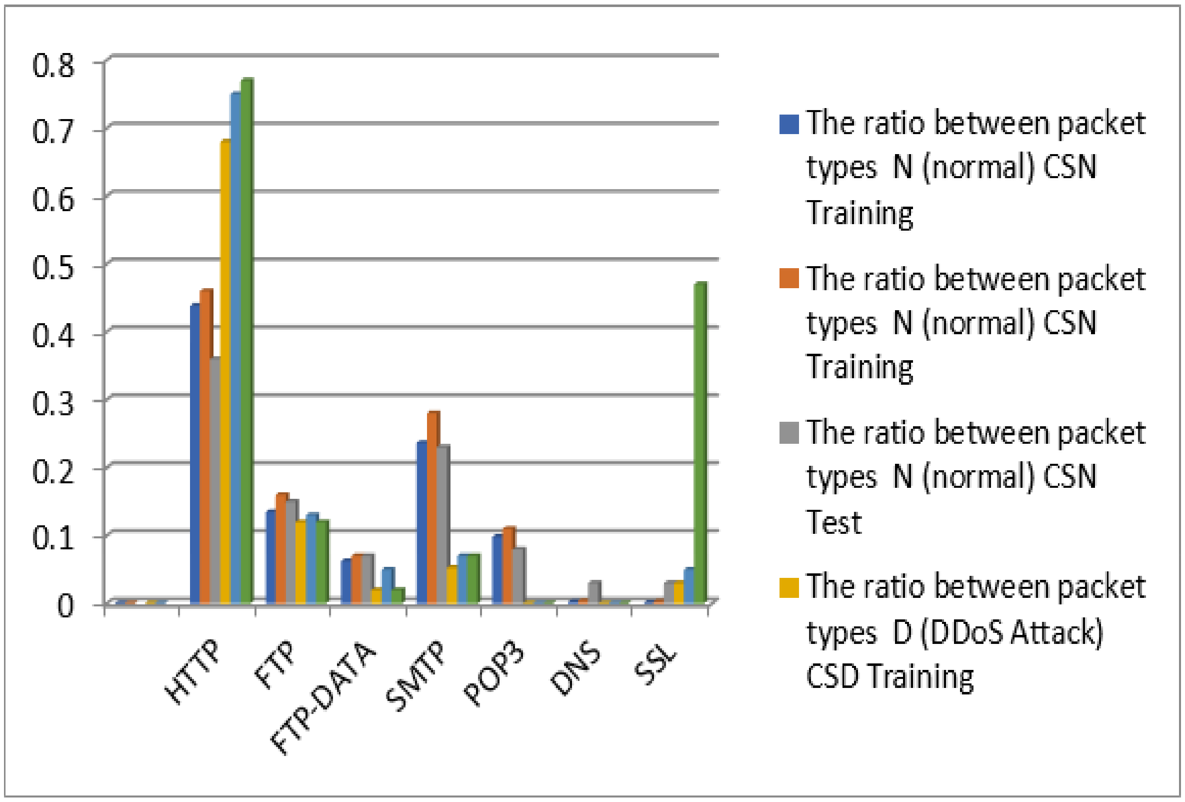

Figure 3: Different Packet Types of Metric Values and Packet Type Ratios

In figure 3, different packet types of metric values and packet type ratios will be depicted in resultant graph. All types of 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 on testing phase. Finally the resultant graph will be shown as accuracy of all types of packets.

Table 4: Metrics for ARTP Performance and Actual Outcomes

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 11-2023
DOI: 10.5281/zenodo.10205690

True positive *(tp)*	The "normal" transactions really 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 actually spoofed.				45,144
False Negative (*fn*)	The percentage of unapproved transactions that are mistakenly classified as regular.				541
Precision	$\frac{tp}{tp+fp}$	0.897	Accuracy	$\frac{(tp+tn)}{(tp+tn+fp+fn)}$	0.944
Recall/sensitivity	$\frac{tp}{tp+fn}$	0.985	F-Measure	2× (precision * recall)/ precision + recall	0.938
Specificity	$\frac{tn}{fp+tn}$	0.914			

Table 5: FAIS and FCAAIS are used to 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

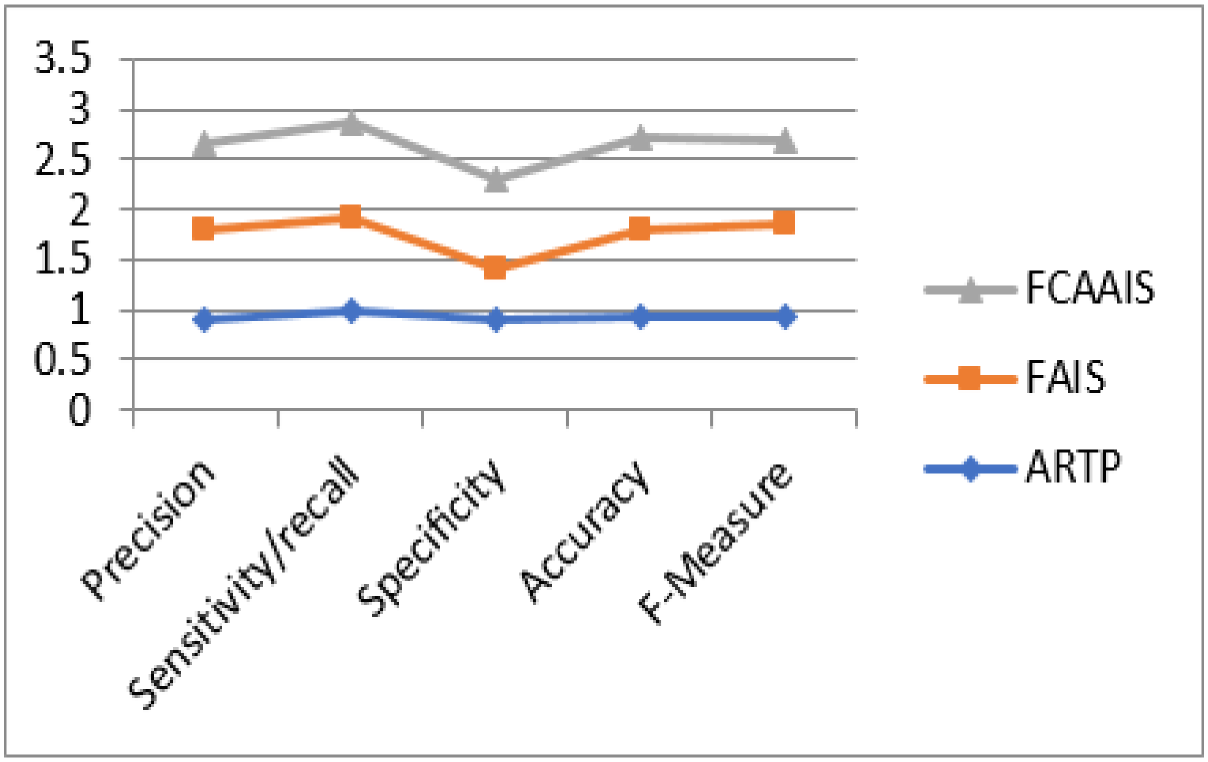

Figure 4: A Comparison is Presented between FCAAIS, FAIS and ARTP

In figure 4, a comparison is depicted between 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 algorithm in various classification or pattern recognition tasks.

6. CONCLUSION

To detect, defend against, and prevent distributed denial of service (DDoS) attacks at the application layer in real time across different types of online services, this study proposes a machine learning approach that is inspired by empirical measurements. The author of this piece makes an earnest effort to categorise the article's contributions along three distinct directions. Firstly, the initial contribution revolves around scrutinizing feature metrics intrinsic to the demand stream, thereby discerning whether the intent is aligned with an attack or a benign state. Moreover, experimental threshold values, stemming from the metrics defined within the ARTP framework, are employed to ascertain whether a given stream's behavior qualifies as a flood or not.

The second contribution centers on the observation of ARTP behavior, substantiated through rigorous experimentation on the LLDoS dataset. This component operates by amalgamating perceived procedural intricacies with optimal swiftness to augment the precision of detection. The outcomes highlight that the indicators extrapolated from the training dataset, serving to evaluate the state of the request stream, hold significant promise and indispensability. Furthermore, the third contribution involves the utilization of threshold values derived from the training dataset, allowing the determination of whether a request stream is potentially indicative of an application layer DDoS attack throughout the entire temporal span. The robustness of ARTP is rigorously tested under straightforward conditions and at elevated velocities. Consequently, the strategy delineated in this paper demonstrates a remarkable ability to uphold peak predictive accuracy, concurrently minimizing computational overhead—a testament to its effective operational efficacy.

References

1) Mahdavi Hezavehi, S., Rahmani, R. An anomaly-based framework for mitigating effects of DDoS attacks using a third party auditor in cloud computing environments. Cluster Comput 23, 2609–2627 (2020). https://doi.org/10.1007/s10586-019-03031-y

2) P. Krishna Kishore, S. Ramamoorthy, V.N. Rajavarman, ARTP: Anomaly based real time prevention of Distributed Denial of Service attacks on the web using machine learning approach, International Journal of Intelligent Networks, Volume 4, 2023, Pages 38-45, ISSN 2666-6030, https://doi.org/10.1016/j.ijin.2022.12.001.

3) S. Byers, A.D. Rubin, D. Kormann, Defending against an internet based attack on physical world. ACM Trans. Internet Technol. 239–254 (2004)

4) J.M. Estevez-Tapiador, P. García-Teodoro, J. Díaz-Verdejo, in *Detection of Web-Based Attacks Through Markovian Protocol Parsing*, 10th IEEE symposium on computers and communications (2005), pp. 457–462

5) P Krishna Kishore, S. Ramamoorthy, V.N. Rajavarman, (2023). Detection of DDoS Enabled Flood Attacks using an Ensemble Classifier in Distributed Networks. International Journal of Intelligent Systems and Applications in Engineering, 11(3), 578–590. https://ijisae.org/index.php/IJISAE/article/view/3260

6) T. Yatagai, T. Isohara, I. Sasase, in *Detection of HTTP-GET Flood Attack Based on Analysis of Page Access Behaviour*, Proceedings IEEE Pacific RIM conference on communications, computers, and signal processing (2007), pp. 232–235

7) S.S. Sindhu, Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst. Appl. 129–141 (2012)

8) A. Shevtekar, N. Ansari, Is it congestion or a DDoS attack? IEEE Commun. Letters 546–548 (2009)

9) S. Kandula, D. Katabi, M. Jacob, A. Berger, in *Botz-4-Sale: surviving Organized DDoS Attacks That Mimic Flash Crowds*, Proceedings of the 2nd conference on symposium on networked systems design & implementation (2005), pp. 287–300

10) P Krishna Kishore, S Ramamoorthy, V. N Rajavarman "Detection, Defensive and Mitigation of DDoS Attacks through Machine learning Techniques: A Literature" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, pages. 2719-2725, November 2019.

11) C. Katar, Combining multiple techniques for intrusion detection. Intern. J. Comput. Sci. Netw. Secur. 208–218 (2006)

12) Y. Xie, S.Z. Yu, A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors. IEEE/ACM Trans. Netw. 54–65 (2009)

13) J.A. Hartigan, Algorithm AS 136: "A k-means clustering algorithm". J. Roy. Stat. Soc.: Ser C (Appl. Stat.) 100–108 (1979)

14) M.I. MIT, in Darpa Intrusion Detection Evaluation. Retrieved from Lincoln Laboratory: https://www.ll.mit.edu/ideval/data/1998data.html

15) D.M. Powers, in Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation, 23rd international conference on machine learning (Pitsburg, 2006)

16) V. Jyothsna, V.V. Rama Prasad, Anomaly based network intrusion detection through assessing feature association impact scale (FAIS). Intern. J. Inform. Comput. Secur. (IJICS) (*in forthcoming article). Inderscience (2016)

17) V. Jyothsna, V.V. Rama Prasad, FCAAIS: anomaly based network intrusion detection through feature correlation analysis and association impact scale (ICT Express, The Korean Institute of Communications Information Sciences, Elsevier, 2016)