DDOS Detection using Behavioral Dynamics in web applications

MANCHALA ASHOKNAGASAI[1] VADLAMOODI MAHESH KUMAR [2] SOMU SATISH KUMAR [3] VEMULA NAGARJUNA [4]

[1]ASST.PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[2] ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[3]ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[4] ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[1,2,3,4] SRI MITTAPALLI COLLEGE OF ENGINEERING

Abstract:

One kind of very damaging attack is the Distributed Denial of Service (DDoS) attack, which targets web applications. assaults that aim to overwhelm a server's network capacity are known as network layer DDoS assaults. In these attacks, called volumetric attacks, a large number of packets at the network layer are used to limited bandwidth. Network infrastructure, on the other hand, got more robust and attacks on the network layer became more complex over time. Advanced denial-of-service attacks are increasingly targeting applications. In contrast to attacks on the network layer, they may be executed with very little attack volume. Because they use legitimate application-layer queries, existing defensive mechanisms have a tougher time detecting them. Because application-layer DDoS attacks target a wider variety of resources, they may potentially knock down a server more quickly and covertly than network-layer DDoS attacks. Application layer distributed denial of service (DDoS) attacks have been the subject of specialised research for the better part of a decade. In order to better understand how application layer DDoS attacks work, this study will look at all of them using important criteria. Defensive systems against different forms of attacks are also covered, with an emphasis on the traits that assist recognise them. Researchers hope that by having such a discussion, they will have a better grasp of the reasons why a certain combination of traits is useful for detecting a particular kind of attack.

1. Introduction

The advent of online apps has liberated users from the constraints of space and time. From the comfort of one's own home, one may engage in online banking, social networking, shopping, and buying and selling. Companies large and small have tried to capitalise on this trend by putting their services online. In addition, many countries' governments have shifted their focus to providing more services online, and not only to keep up with the competition. It is critical for businesses and governments to maintain constant client access to their websites and the services they provide as many modern businesses and services rely on web applications. Companies and banks are increasingly using web applications, which makes them a more tempting target for cybercriminals. A lot of cybercriminals are in it for the money, and that's why they target applications that save sensitive information like credit card numbers. However, social and political issues may also lead to attacks against online apps, in addition to business competition and policy disagreements. Attacks like this aim to disrupt online applications so that people can't access their services, which means money lost for the business. In only one minute, a business may lose as much as $22,000 due to downtime [1]. Much worse would be if consumers stopped trusting the product or if the value of the company's brand dropped. Customers will stop caring about a company's products or services if its website is always unavailable. These attacks, sometimes called Denial of Service (DoS) Attacks, have been around for a long time and are still a major worry due to how they evolve and propagate. Arbour networks recorded an average of 1,24,000 DDoS attacks per week throughout the 18-month period beginning in January 2015 and ending in June 2016 [2].When there are disagreements over policy, the perpetrators often target government websites with these attacks. The global hacker collective Anonymous has launched distributed denial of service (DDoS) attacks on many government websites, including the Spanish government's support for Catalan independence [3]. This kind of hack has recently targeted the governments of many countries, including Brazil [7], Ireland [5], India [6], and the United States [4]. These attacks not only expose vulnerabilities, but they also show that security is lacking. Online stores and banks are common targets of distributed denial of service attacks. As a result of a complete shutdown of online banking in the event of a hack on a bank's website, the economy might collapse. This is a major issue in this day and age when more and more individuals are willing to buy and sell things online. In 2012, customers were unable to complete transactions for hours at a time due to distributed denial of service attacks on US-based institutions [11]. A similar hack struck the HSBC bank in the UK as late as January 2016 [10]. Websites associated with bitcoin have also been targeted, similar to banking websites, by those who doubt the practicality of a currency without a physical representation. In the event of a distributed denial of service attack, an enormous number of users' Internet connections would be compromised, perhaps affecting a whole country. When a DNS server goes down, numerous websites go black as users can't resolve domain names, as shown by the 2016 attack on Dyn [12]. Internet connectivity issues might arise if parts of the network infrastructure are disabled, particularly if other means of accessing the internet are not available. The majority of Liberia lost internet connectivity due to attacks on the country's internet infrastructure [15]. The proliferation of linked gadgets means that DDoS attacks affect more than just computers. Systems in Finland [13] and Sweden [14] for heating and transportation both came to a standstill after DDoS assaults rendered them useless. For context, research out of network security company Corero found that 51 percent of UK critical infrastructure organisations were ignoring the risk of distributed denial of service attacks in 2017. Recent years have seen an uptick in high-profile distributed denial-of-service (DDoS) attacks. The evolution and escalation in complexity of distributed denial of service attacks is a contributing factor. Devices were the first targets of these attacks, which included flooding them with network layer packets very quickly. As infrastructure evolved and protections at the network layer got better at thwarting attacks, the emphasis of attackers moved to the application layer. More and more distributed denial of service attacks are targeting the application layer as of late.In their DDoS Threat Landscape Report 2015–16, Imperva Incapsula found that apps were the target of half of the attacks [18]. This also means that DDoS attacks on the application layer will likely get more complex over time. Application layer attacks are more complex forms of distributed denial of service (DDoS) attacks because they are difficult to detect since they seem so much like normal user traffic. Since these attacks leverage legitimate user requests, there is no way to inspect a packet to identify maliciousness. This means that some modern Web Application Firewalls (WAF) and network layer defences are both helpless against this kind of attack. The fact that these assaults may be launched across a range of connection-oriented and connectionless application-layer protocols further adds to the danger.

2 . Related Work

Many earlier denial of service (DoS) attacks target internet services. It wasn't until the late 1990s that a Denial-of-Service attack was first used [19]. They used to be common, but they've evolved into one of the most common ways to compromise web apps nowadays. Distributed denial of service (DDoS) attacks vary in severity depending on the attacker count. One hacker is usually responsible for a Denial-of-Service attack. DDoS attacks may include thousands of attackers per large-scale operation. In most cases, human attacks are few and infrequent, and they aren't always required. The "attackers" here are the computer systems that the humans are using to launch their attacks. Bots and zombies are maliciously compromised computer systems that the real bad guys use as a springboard for their attacks. A large number of these bots would typically be used by an attacker to construct a botnet.The goal of a distributed denial of service (DDoS) attack is to make a web server unreachable by legitimate users. There are a lot of ways to do this, but ultimately, you want to exhaust every resource that the server has. The use of central processing unit (CPU) and database cycles, memory, socket connections, and network bandwidth are all instances of such resources. Attackers may do this by repeatedly accessing the server's features or by taking advantage of security holes in the system or protocol. Sending a large amount of packets to the server may rapidly exhaust the network capacity, which is why many DDoS attacks aim at it. At the network layer, protocols such as User Datagram Protocol and Internet Control Message Protocol are used for this purpose. With the passage of time, two things changed. In order to identify DDoS attacks at the network layer, servers and networks became more robust, and servers had access to better and more bandwidth. Although it was getting more difficult, a server might still be brought to a standstill by a huge surge of requests that overwhelmed the network. The invaders retaliated by climbing the stack to the transport tier. SSL renegotiation attacks[20] took use of transport-layer weaknesses; however, servers eventually built defences to counter these attacks. These attacks were obviously malicious since they followed a pattern that was both noticeable and transferable between platforms. In recent years, attackers have resorted to climbing the stack, resulting in a new trend called DDoS attacks at the application layer. In contrast to network bandwidth throttling, these attacks target servers in an effort to exhaust their resources, such as socket connections, CPU, database, and memory. Attacks against the application layer have increased in frequency and severity throughout the last several years. Network layer assaults fell for the fourth consecutive quarter, according to Imperva Incapsula's Global DDoS Threat Landscape Report 2017 [21], but application layer attacks rose. "The crème of the cybercriminal groups are now moving to Application Layer DDoS assaults," according to Kaspersky. There are a few quirks that set application-layer attacks apart from others.Application-layer distributed denial-of-service attacks propagate over legitimate HTTP communications. There is almost any difference between a normal request and an attack request. A difference in intent, rather than actual content, is the only difference. Most application-layer firewalls and network-level packet filters so miss these attacks. Whereas network-layer DDoS attacks encounter bandwidth limitations, application-layer attacks see server resources as the bottleneck. With fewer requests, it may be cut down, which in turn reduces the traffic volume. Most current distributed denial of service detection solutions rely on very high traffic volumes to detect attacks. Due to the ineffectiveness of most modern DDoS detection technologies, this strategy is unable to identify application-based DDoS assaults.Because of this, application-layer DDoS assaults are very targeted. Attackers might aim for the central processing unit, database, memory, or even socket connections. The other resources will remain unaffected by an attack on a single resource, but the whole system will become useless. Consequently, a defence mechanism that is tailored to one resource will not work when faced with another. Real people's visits to a website skyrocket all of a sudden because of a major event or an online sale, which is comparable to Flash Crowds. Very much like Flash Crowds: Some people mistakenly believe that an application layer DDoS attack is different from a flash crowd since both include an influx of legitimate HTTP requests to a website. Since flash crowds provide valuable traffic to a website, it would be a shame to block them since defensive systems couldn't tell the difference between a DDoS attack and flash crowds.

3. Methodology

The capacity of the attackers to automate queries to the web server is the primary cause for a denial-of-service assault. As a means to this end, botnets or simple DDoS programs or tools are used. You can find both of them easily online. As a first line of defence against distributed denial of service (DDoS) attacks, limiting automated requests is essential. One simple way to do this is by using user puzzles. An automated system will struggle to do a job that a human could do effortlessly. User riddles may be as basic as CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart) or AYAHs (Are You A Human?). Even if bots may bypass them using suitable image processing methods, this primary defence against DDoS attacks is still quite effective [49], [50], [51]. Part of the reason for this is because the majority of distributed denial of service attacks employ simplistic techniques that can be executed using easily available tools, but do not possess the necessary processing power to effectively circumvent these challenges.

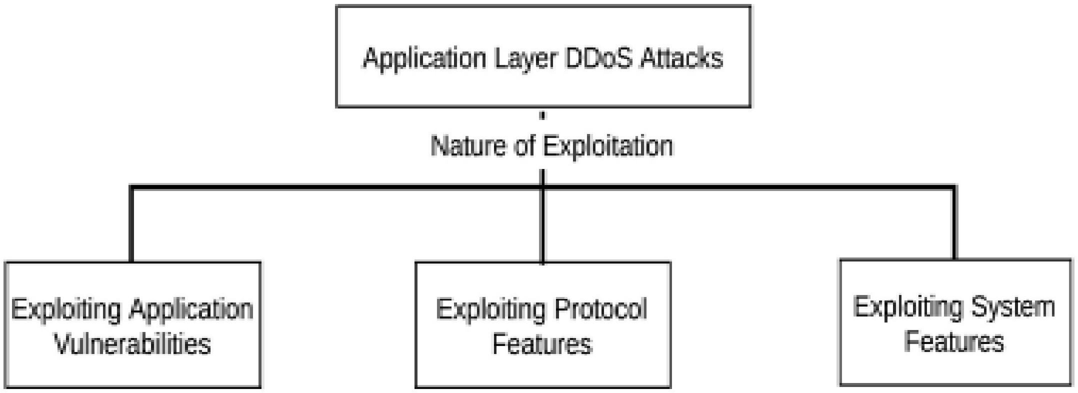

Application Layer DDoS Attacks
Nature of Exploitation
Exploiting Application Vulnerabilities
Exploiting Protocol Features
Exploiting System Features

Fig. 1. Classification of Application Layer DDoS attacks based on Nature of Exploitation

The question of which users get to utilise the challenge-response system arises with this protective tactic. Use the challenge-response approach if you want to simplify things for your customers. It's not ideal, but it's a workaround anyway. The user's overall satisfaction with the website is severely diminished when they encounter user puzzles. No matter what website a person visits, they would rather not be asked to fill out an unnecessary form. Very few individuals in the future will be able to solve the puzzles.To ensure that only a tiny fraction of requests are really challenged, it employs a moderator module. The defensive system takes workload into account while deciding how to sample requests. The request becomes more burdensome as the problem becomes more difficult. Selecting at random individuals to distribute the riddles is the last remaining step. But maybe the wisest course of action isn't to serve a challenge to everyone. It becomes harder to spot suspicious users in this scenario. Research by Sivabalan and Radcliffe [53] suggests that tracking the frequency and domains of suspicious visits can help identify them. As long as the server load remains below appropriate limits, users will not have any issues with their response. When the server demand became too high, they would serve AYAHs to those they suspected of using the service. If one person whose signature is dubious manages to fix the issue, it will verify all other users whose signatures are similar.This allows them to calibrate the process of creating their signature. Conversely, users are banned while the AYAH continues to be unsolved.Web services also use comparable safety protocols. In the hash-based computation-bound problem proposed by Suriadi et al.[54], clients are tasked with finding partial preimages in cryptographic hash functions. They proposed an approach that would combine the issue and solution by adding a SOAP header containing both. By using the nonce approach, they successfully prevented the client from re-creating the solution. To detect HTTP DDoS attacks on online services, Karnwal et al. [55] utilised a similar method. Further varieties of attacks, such as forcible parsing, are also under development. Although CAPTCHAs and AYAHs effectively prevent denial-of-service attacks by using puzzle-based tactics, they degrade the user experience in the process. Because of this, the majority of research is on discovering other ways to fix the problem. There are typically four paths that an assault detection system may take: 1) Requests tailored to certain templates 2) By monitoring enquiries Thirdly, by studying the dynamics of the request stream Fourthly, by examining the semantics of the request stream 1) Template Matching: DDoS attacks that use SOAP often try to take advantage of the wiggle room in the various security standards. Strict constraints that incoming requests should respect, such as the right degree of nesting or usage of external entities, are a popular defence mechanism against these assaults. Schema hardening describes this phenomenon [26]. Verifying the schema requirements in incoming packets is a necessary step after schema hardening. Put simply, DDoS assaults based on SOAP or XML may be significantly mitigated by comparing each incoming request to a template. 2) Request Tracking: This feature goes beyond just looking at each request and comparing them to a template. The term "request tracking" describes systems that keep tabs on the total number of requests and replies issued and received, with the goal of finding patterns among them. A sluggish DDoS assault or one that uses an underlying UDP connection may be easily detected with the use of such techniques. 3) Analysing Request Stream Dynamics: Request dynamics examines a request stream "numerically." Rather than trying to make sense of the requests stream, this kind of analysis is mainly focused with statistics. Factors such as request rate, dispersion of source IP addresses, amount and kind of requests, etc. fall under this category. This tier of monitoring doesn't care about the user's experience with a web app and instead concentrates on the nitty-gritty of a request stream. When it comes to identifying HTTP flooding assaults, these approaches are quite useful. 4) Analysing Request Stream Sematics: These detection algorithms aim to capture characteristics that show how a user uses the online application. The average user has no idea what his request rate or session rate are or how to change them. To these users, the most important thing is figuring out which resources or web sites to visit and in what sequence.

3. Results and Discussion

Using POX as the network controller, the tests are conducted in a Mininet environment. The computer is set up with an Intel Core i5-7300HQ processor, 8GB of RAM, and the Ubuntu 5.4.0-6ubuntu1 operating system. Within the TensorFlow framework, the convolution neural network is put into action. The detection evaluation dataset is selected as CICIDS2017 [9]. A controller, six switches, and a deep detection server make up the experimental topology. Figure 3 depicts the topology. A script generates regular traffic, and Hping3 is used to mimic DDoS assaults. Assault hosts are selected as Host1, Host3, and Host 5. Accuracy and recall are inversely proportional to the threshold value. A threshold somewhat higher than the typical packet rate is chosen to get high recall, since the convolutional neural network—on which the deep detection is built—has the ability to guarantee high accuracy. The experiment began with 0 seconds of normal traffic and continued with 25–50 seconds of attack flow. The rate of PACKET_IN messages received by the controller is shown in Fig. 4, with a sample interval of 1 second.

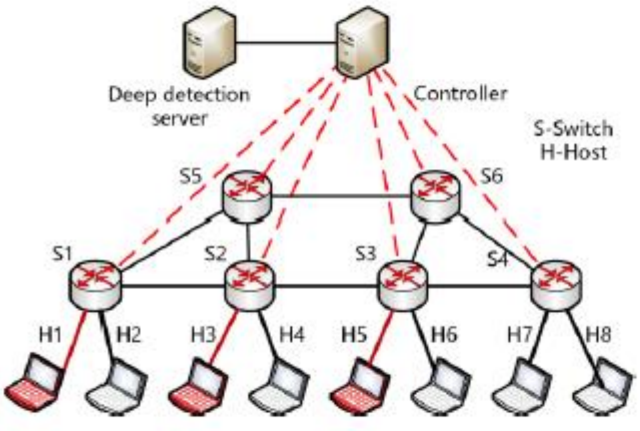


Fig. 3. Experimental topology.

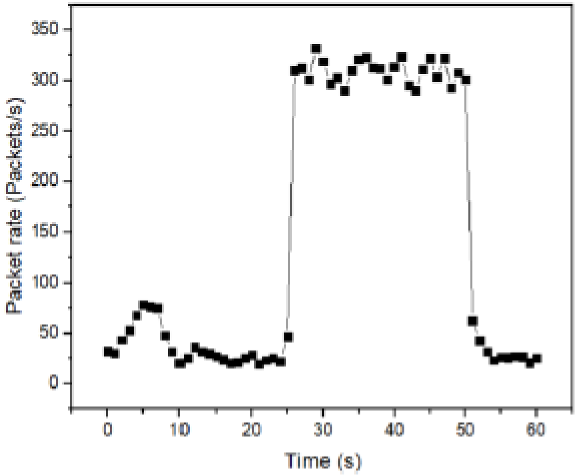


Fig. 4. PACKET_IN message rate of the switch.

The PACKET_IN message rate varied by about 50 packets per second during the zeroth and twenty-fifth seconds. Due to the initial occurrence of regular traffic entering the network, a considerable amount of PACKET_IN messages were sent to create flow entries between the 0th and 10th seconds. The rate stabilized at less than 50 packets per second once flow entries were generated. There was a dramatic spike in the packet rate after the 25th second, reaching a maximum of approximately 350 packets per second. The packet number rose because malicious activity entered the network. The rate started to slowly drop after the 50th second and got back to normal in the 55th second. Normal traffic may create a maximum packet rate of about 90 packets per second, as seen in the image. The 100 Packets/s criterion was therefore chosen because of its ability to guarantee a strong recall. Different forms of distributed denial of service attacks (DDoS) have different characteristics in their traffic. We will choose common elements of various DDoS attack traffic in order to properly portray the distinction between attack traffic and regular traffic. The experiment makes use of four common vectors of distributed denial of service attacks: HTTP flooding, UDP flooding, ICMP flooding, and SYN flooding. In Transmission Control Protocol/Internet Protocol, a flow is denoted by the five-tuple: Protocol, Source Port, Destination Port, Source IP Address, and Port. We use Packet Length to differentiate between the two types of traffic since the length of attack packets created by the same script is rather comparable. We compare the assault traffic's six-tuple entropy to that of regular traffic. Figure 5 displays the outcomes for various entropy settings. The entropy value of attack traffic and regular traffic differs dramatically, as seen in Figure 5. Hence, information entropy allows for their separation. Experience has shown that features are more successful at differentiating between attack and regular traffic when there is a significant gap between their entropies. Following the preceding concept, the picture depicts the selection of the Source IP Address, Packet Length, and Protocol.

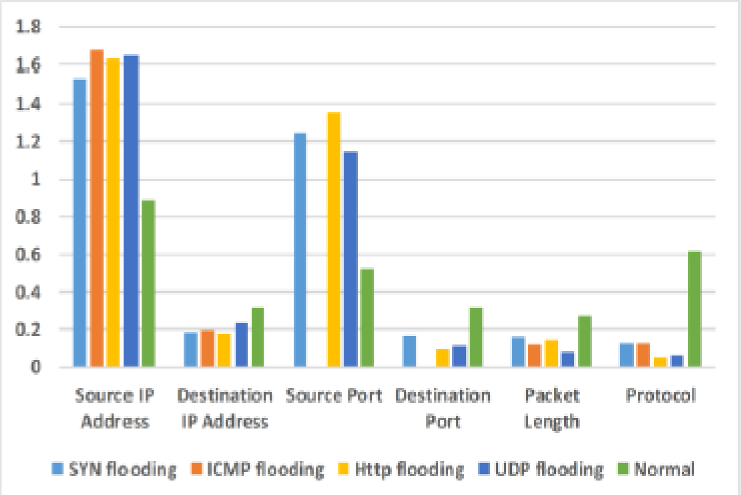
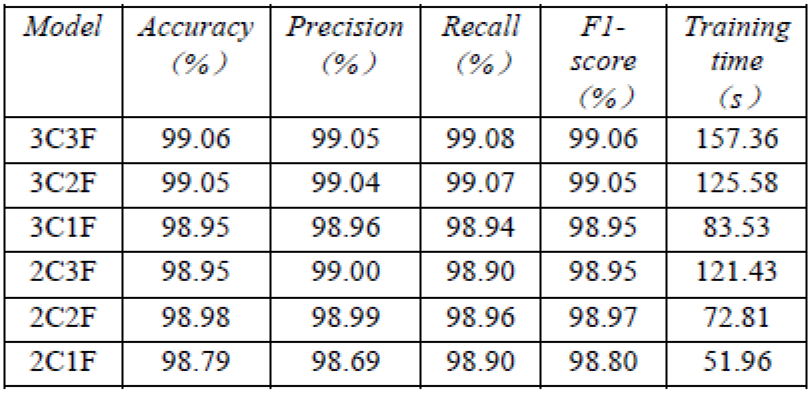


Fig. 5. Six-tuple information entropy of attack traffic and normal traffic.

We compute the entropy of three traffic characteristics and compare it against thresholds in turn. Additional deep detection is required in the event that the threshold value is surpassed, since it is assumed that an attack may occur. While deciding on an entropy threshold, it is important to keep memory and accuracy in mind. With deep detection likely to maintain high accuracy, information entropy detection should guarantee maximum recall. That is why, when one of the three cutoffs is crossed, the traffic will be considered suspect.The amount of layers in the model of a convolutional neural network: The more layers a neural network has, the more accurate its classifications will be. In general, the more layers a neural network has, the more accurate it might be. However, the model convergence time will increase geometrically with a deep model. For this reason, if you want better accuracy and less training time, go for a simpler model with fewer layers.It is at the convolution and full connection layers that the bulk of the convolution neural network's computation occurs. Therefore, training time and classification accuracy are strongly affected by the number of these layers. The variation across experimental models is mostly expressed by the quantity of these two components. A model with one, two, or three complete connection layers and two or three convolution layers is used in the experiment. Two pooling layers are used. The experiment establishes six distinct models. The accuracy, precision, recall, F1-score, and training duration are the metrics used to assess the performance of models. In Table I, you can see how the six models compare in terms of performance. Two convolutional layers and a fully connected layer make up the model, as shown by the 2C1F. Most assessment metrics, as shown in Table I, go up in direct proportion to the number of convolution and complete connection layers. Despite a 20.85-second increase in training time, 2C2F achieves an accuracy that is approximately 0.2 percentage points greater than 2C1F. Although 2C2F has a lesser accuracy, 2C3F and 3C1F have a considerably longer training duration.Training with 3C2F takes 52.77 seconds more time, but the accuracy is 0.07% better than with 2C2F. With 3C3F, it's the same. Based on the comparison, it seems that the 2C2F model may train with little time investment and yet get good results. Hence, we will be doing more evaluations using this approach.

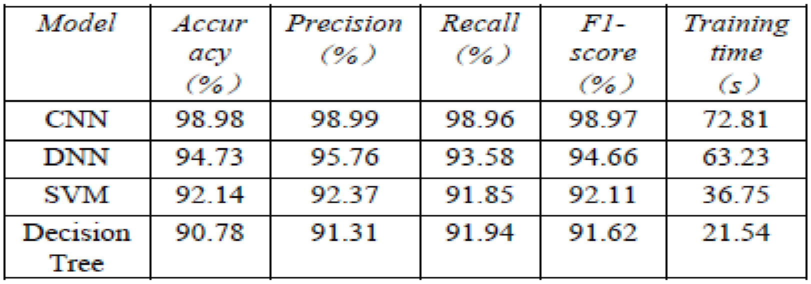
TABLE I. ACCURACY METRICS OF DIFFERENT LAYERS



Model	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*	*Training time (s)*
3C3F	99.06	99.05	99.08	99.06	157.36
3C2F	99.05	99.04	99.07	99.05	125.58
3C1F	98.95	98.96	98.94	98.95	83.53
2C3F	98.95	99.00	98.90	98.95	121.43
2C2F	98.98	98.99	98.96	98.97	72.81
2C1F	98.79	98.69	98.90	98.80	51.96

In order to evaluate the 2C2F model's performance to three common machine learning approaches, we used the experimental data from the parameter section as our basis. Training time, accuracy, ROC curves, and area under the ROC curve are the evaluation measures. True positive rate (TPR) is vertically shown on the ROC curve, whereas false positive rate (FPR) is horizontally represented on the same curve. A steeper ROC curve indicates that the model is doing better. Therefore, a higher AUC indicates a more effective classifier. Therefore, it may be able to depict the model's performance. Compared to SVM, DNN, and decision tree algorithm, CNN achieves an accuracy that is 4.25% to 8.20% higher; nevertheless, its performance in recall, F1-score, and precision are comparable. This information is shown in Table II. Comparing CNN to the other three algorithms, its training time is much longer. A little longer training period is tolerable in the event of good accuracy since the detection model is typically developed offline and not updated often. Of the four machine learning methods, CNN has the steepest ROC curve. CNN has an area under the curve (AUC) of 0.949, which is 0.081 more than the runner-up. It proves the convolutional neural network model is better at detecting traffic.

TABLE II. ACCURACY COMPARISON OF DIFFERENT MODELS



Model	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*	*Training time (s)*
CNN	98.98	98.99	98.96	98.97	72.81
DNN	94.73	95.76	93.58	94.66	63.23
SVM	92.14	92.37	91.85	92.11	36.75
Decision Tree	90.78	91.31	91.94	91.62	21.54

4. Conclusion

This work will help researchers understand and mitigate the dangers posed by application layer distributed denial of service attacks. There has also been an analysis of the present research directions and protective systems to highlight the different traits and detecting methods used to spot these attacks. Although there has been considerable success in identifying and combating application layer denial of service attacks, the challenges in establishing effective countermeasures mean that these assaults continue to provide a significant threat. Our ultimate goal is for our study to spark new lines of inquiry and discussion.

References

[1] Radware, Cyber Security on the Offense - A Study of IT Security Experts, 2012. [Online]. Available: https://security.radware.com/uploadedfiles/resourcesandcontent/attacktools/cybersecurityontheoffense.pdf

[2] A. Networks, 2016 DDoS Attack Statistics, 2016. [Online]. Available: https://www.arbornetworks.com/arbor-networks-releasesglobal-ddos-attack-data-for-1h-2016

[3] I. S. Magazine, Anonymous Attacks Spanish Government Sites, 2017. [Online]. Available: https://www.infosecurity-magazine.com/ news/anonymous-attacks-spanish/

[4] Incapsula, Analysis of Vikingdom DDoS Attacks on U.S. Government Sites, 2015. [Online]. Available: https://www.incapsula.com/blog/ vikingdom-ddos-attacks-us-government.html

[5] Silicon, Irish Government Websites Taken Down By DDoS Attacks, 2017. [Online]. Available: http://www.silicon.co.uk/e-regulation/irishgovernment-websites-ddos-184428

[6] Register, Anonymous turns its DDoS cannons on India, 2012. [Online]. Available: https://www.theregister.co.uk/2012/05/18/ anonymousddosindiasites/

[7] Corero, DDoS Attacks Plague Olympic & Brazilian Government Websites, 2016. [Online]. Available: https://www.corero.com/blog/749- ddos-attacks-plague-olympic--brazilian-government-websites.html

[8] Register, Gits club GitHub code tub with record-breaking 1.35Tbps DDoS drub, 2018. [Online]. Available: https://www.theregister.co.uk/ 2018/03/01/githubddosbiggestever/

[9] Coindesk, Bitcoin Gold Website Down Following DDoS Attack, 2017. [Online]. Available: https://www.coindesk.com/bitcoin-gold-websitefollowing-massive-ddos-attack/

[10] Guardian, HSBC suffers Online Banking Cyber Attack, 2016. [Online]. Available: https://www.theguardian.com/money/2016/jan/29/ hsbc-online-banking-cyber-attack

[11] A. Networks, Leading US Banks targeted in DDoS Attacks, 2012. [Online]. Available: https://nakedsecurity.sophos.com/2012/09/ 27/banks-targeted-ddos-attacks/

[12] Dyn, Dyn Analysis Summary Of Friday October 21 Attack, 2016. [Online]. Available: https://dyn.com/blog/dyn-analysis-summary-offriday-october-21-attack/

[13] I. Times, Hackers leave Finnish residents cold after DDoS attack knocks out heating systems, 2016. [Online]. Available: http://www.ibtimes.co.uk/hackers-leave-finnish-residentscold-after-ddos-attack-knocks-out-heating-systems-1590639

[14] SCMagazine, DDoS attacks delay trains, stymie transportation services in Sweden, 2017. [Online]. Available: https://www.scmagazine.com/ddos-attacks-delay-trains-stymietransportation-services-in-sweden/article/700227/

[15] Guardian, Massive cyber-attack grinds Liberia’s internet to a halt, 2016. [Online]. Available: https://www.theguardian.com/technology/ 2016/nov/03/cyberattack-internet-liberia-ddos-hack-botnet

[16] Forbes, Bitcoin Hit By Massive DDoS Attack As Tensions Rise, 2014. [Online]. Available: https://www.forbes.com/sites/leoking/2014/02/12/ bitcoin-hit-by-massive-ddos-attack-as-tensions-rise/

[17] Guardian, Critical infrastructure not ready for DDoS attacks: FOI data report, 2017. [Online]. Available: https://www.scmagazineuk.com/critical-infrastructure-not-readyfor-ddos-attacks-foi-data-report/article/684838/

[18] Incapsula, DDoS Threat Landscape Report 2015-16, 2016. [Online]. Available: https://lp.incapsula.com/rs/804-TEY-921/images/2015-16%
20DDoS%20Threat%20Landscape%20Report.pdf

[19] L. Garber, “Denial-of-service attacks rip the internet,” Computer, vol. 33, no. 4, pp. 12–17, 2000.

[20] I. Ristic. (2011) Tls renegotiation and denial of service attacks. [Online]. Available: https://blog.qualys.com/ssllabs/2011/10/31/tlsrenegotiation-and-denial-of-service-attacks

[21] Incapsula, Global DDoS Threat Landscape Q1 2017, 2017. [Online]. Available: https://www.incapsula.com/ddos-report/ddosreport-q1-2017.html

[22] Kaspersky, Kaspersky DDoS Intelligence Report for Q1 2016, 2016. [Online]. Available: https://securelist.com/kaspersky-ddosintelligence-report-for-q1-2016/74550/

[23] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, K. S. Ehlert, and D. Sisalem, "Survey of security vulnerabilities in session initiation protocol," IEEE Communications Surveys & Tutorials, vol. 8, no. 3, pp. 68–81, 2006.

[24] S. Armoogum and N. Mohamudally, "Survey of practical security frameworks for defending sip based voip systems against dos/ddos attacks," in IST-Africa Conference Proceedings, 2014. IEEE, 2014, pp. 1–11.

[25] I. Hussain, S. Djahel, Z. Zhang, and F. Naït-Abdesselam, "A comprehensive study of flooding attack consequences and countermeasures in session initiation protocol (sip)," Security and Communication Networks, vol. 8, no. 18, pp. 4436–4451, 2015.

[26] M. Jensen, N. Gruschka, and R. Herkenhöner, "A survey of attacks on web services," Computer Science-Research and Development, vol. 24, no. 4, pp. 185–197, 2009.

[27] V. Durcekova, L. Schwartz, and N. Shahmehri, "Sophisticated denial of service attacks aimed at application layer," in 2012 ELEKTRO, May 2012, pp. 55–60.

[28] M. Aamir and M. A. Zaidi, "A survey on ddos attack and defense strategies: from traditional schemes to current techniques," Interdisciplinary Information Sciences, vol. 19, no. 2, pp. 173–200, 2013.

[29] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," IEEE communications surveys & tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[30] Aharonu, Mattakoyya, et al. "Entity linking based graph models for Wikipedia relationships." *Int. J. Eng. Trends Technol* 18.8 (2014): 380-385.

[31] Srinu, Nidamanuri, Sampathi Sivahari, and Mastan Rao Kale. "Leveraging Radial Basis Function Neural Networks for Rainfall Prediction in Andhra Pradesh." *2022 International Conference on Computer, Power and Communications (ICCPC)*. IEEE, 2022.

[32] Muppavarapu, Rajasekhar, and Mastan Rao Kale. "An Effective Live Video Streaming System."