VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Information Security Analysis and Audit

J component – Final Review

Title

Rule-Based Classifier on Entropy for efficient Stress Test and detection against DoS attacks

Members

Reg No	Name
19BIT0156	Akshat Mishra
19BIT0179	Saksham Arora
19BIT0181	Pranav Gupta

Abstract

One of most frequent types of attack nowadays seems to be on web servers and web applications. The fundamental reason for this is because the majority of web applications and services are open to exploits and may be readily exploited. DDoS is one of the most common attacks.

The term DDoS stands for Distributed Denial of Service attack. The majority of modern websites and web servers are incapable of withstanding powerful attacks such as DDOS attacks. These online services are vulnerable to simple vulnerabilities and can be quickly exploited. However, we can perform this form of attack in penetration testing as well, to assess server stress and aid to improve security by determining the website's ability to survive such attacks.

The goal of this study is to test various online applications for DDoS attacks and to assess the extent to which servers can protect themselves from malicious attacks. We'll use a basic application layer denial of service (DoS) tool that can be deployed from a single PC and will target servers directly. It won't use a botnet, and all connections will come from the same place. The application will initiate a TCP connection flood to its target after execution, exhausting session table resources and effectively crashing the server.

Introduction

DDoS assaults occur on a daily basis, and it is sometimes difficult to distinguish malicious traffic from legitimate traffic. We may, however, be better prepared to fight DDoS attacks and have precautions in place to protect our website.

Any component of our network and IT infrastructure might be targeted by a DDoS attack. Attackers are looking for weaknesses in different layers of our network to exploit. The following are some of the most prevalent DDoS attacks we see:

- Application Layer Attacks:

By providing malicious HTTP traffic load to our network's application layer, these attacks basically target our network's application layer. When an HTTP request is sent to the server, the server must conduct a number of activities, including loading files, querying the database, calculating the request, producing the answer, and so on. With so much traffic, the server becomes overburdened, exhausting infrastructure resources and eventually shutting down. The application layer DDoS assaults are difficult to avoid since it is hard to recognize these requests as malicious owing to their nature being similar to that of real users.

- Protocol Attacks:

These attacks cause the service to go down by depleting intermediary resources such as state table capacity, load balancers, firewalls, and TCP handshakes, among other things. For example, attackers can initiate a connection by sending a TCP handshake request to the server, which the server responds to then waits for confirmation from the client. However, the client never sends the confirmation, and the server continues to wait for it, exhausting the server's resources. State-exhaustion assaults are another name for these types of attacks.

Need of DDoS Stress Testing

Stress testing our website for DDoS scenarios will provide us with enough data to keep us on our toes. Here's why we should use a stress testing method to test our website for DDoS protection.

- Prior to DDoS assaults, identify and fix website infrastructure faults and bottlenecks.
- Determine your website's breaking threshold under load conditions and optimize for sturdiness.
- Planning for an incident response procedure.
- Developing mitigation and preventive techniques for DDoS attacks.
- Examining third-party services in the context of a DDoS attack.

Literature Survey

Sr. No	Journal	Year	Author	Title	Mechanism	Accomplishments/ Advantages	Limitations or Future work
1	MDPI	2016	Xuan Dau Hoang et al.	A Review on Hot-IP Finding Methods and Its Application in Early DDoS Target Detection	Count-min and sliding window methods are used in the detector	Count-Min gives the best overall performance both in terms of space and time complexities	Optimizing is still needed to monitor large bandwidth network connections
2	CSIS	2017	Dezhi Han et al.	A DDoS Attack Detection System Based on Spark Framework	A novel DDoS attack detection system based on Spark framework	K-Means parallelization algorithm detects all kinds of attacks with low false rate	In-depth research in parameter tuning of the Spark framework is still required
3	ScienceDirect	2017	K.J. Singh et al.	MLP-GA based algorithm to detect application layer DDoS attack	Multilayer Perceptron with a Genetic Algorithm	MLP-GA has minimum false positive when compared to traditional classifiers such as Naive Bayes, RBF, etc.	Couldn't differentiate an application layer DDoS attack from that of a flash event
4	IEEE Xplore	2020	Amir Djenna et al.	A Pragmatic Cybersecurity Strategies for Combating IoT-Cyberattacks	Artificial Neural Network classification	Proposed a novel approach for detection and mitigation DDoS on IoT	-
5	ScienceDirect	2019	Indraneel Sreeram et al.	HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm	Bio-Inspired Anomaly based HTTP-Flood Attack Detection (BIFAD)	The devised Bat algorithm amplified the detection accuracy with minimal process complexity.	-
6	Springer	2018	D. Arivudainambi et al.	LION IDS: A meta-heuristics approach to	Lion optimization algorithm	Model works better than ACO+CNN and	-

				detect DDoS attacks against Software-Defined Networks		BCO+CNN models	
7	Elsevier	2021	Abdullah Emir Cil et al.	Detection of DDoS attacks with feed forward based deep neural network model	Deep Neural Network	Detection accuracy = 99.99% Attack type classification = 94.57%	Need for a larger dataset observed
8	Springer	2020	Omer Elsier Tayfour et al.	Collaborative Detection and Mitigation of Distributed Denial-of-Service Attacks on Software-Defined Network	Combination of sflow-RT and Snort NIDS	Method allows the RSMQ mechanism to update the Ryu controllers and allows fast detection of any DDoS attacks in other SDN sites	The scalability of the this method is yet to be analyzed
9	IEEE Xplore	2020	R.B. Shohani et al.	Introducing a New Linear Regression Based Method for Early DDoS Attack Detection in SDN	Linear Regression	Early detection of DDoS regardless of the sort of DDoS	-
10	IEEE Xplore	2020	Ali Muhammad et al.	Robust Early Stage Botnet Detection using Machine Learning	Random Forest, Multilayer Perceptron	Accuracy of 99%, True Positive Rate of 0.99%, and False Positive Rate of 0.007%	need to discover the detection of botnet to deal with P2P based botnets.
11	Springer	2018	Anteneh Girma et al.	An Efficient Hybrid Model For Detecting Distributed Denial Of Service (Ddos) Attacks In Cloud Computing Using Multivariate Correlation And Data Mining Clustering Techniques	Multivariate Correlation, DBSCAN Clustering	false alarm rate is significantly reduced to zero	Needs improvement in handling big data

12	IEEE Xplore	2021	B.S.Kiruthika Devi et al.	Cloud DDoS Detection and Defense System using Complex Event Processing	Complex event processing	System immediately acts on the attack sources to take remedial actions and protects the cloud from DDoS attacks	the detection of unknown patterns causing zero day attacks
13	IEEE Access	2020	Bruno Martins Rahal et al.	A Distributed Architecture for DDoS Prediction and Bot Detection	Two-tier distributed architecture using α-investing^{+}	volume of analyzed data occurs per cluster of nodes, without losing detection efficiency	Comparison with other ML models still needed
14	IEEE Xplore	2019	Ming Xuanyuan et al.	Detection and Mitigation of DDoS Attacks Using Conditional Entropy in Software-defined Networking	Conditional entropy	the proposed method has a high average detection rate of 99.372%.	Detection has been done but filtering still needs to be implemented
15	IEEE Access	2019	Luis A. Trejo et al.	DNS-ADVP: A Machine Learning Anomaly Detection and Visual Platform to Protect Top-Level Domain Name Servers Against DDoS Attacks	DNS-ADVP	A defensive system dedicated to DNS servers against DDoS attacks	Heavy computation for which GPUs are required
16	IEEE Xplore	2020	Bavani K et al.	Statistical Approach Based Detection of Distributed Denial of Service Attack in a Software Defined Network	Statistical Approach: determining the mean entropy and the rate of percentage drop	97% success in detecting a 25% attack and 100% success in detecting a 50% attack	Designing a multi controller environment by identifying the source of attack can be considered as future work
17	Hindawi	2021	Hassan Mahm	S-DPS: An SDN-Based DDoS Protection	Software Defined Networkin	Minimal CPU utilization with	Vulnerable to computational or bandwidth

			ood et al.	System for Smart Grids	g-based DDoS Protection System	high Detection Rate	bottlenecks for very large networks
18	IEEE Xplore	2021	Abdullah Yasin Nur	Combating DDoS Attacks with Fair Rate Throttling	collaborative fair rate throttling mechanism	Reduction with minimal overhead to routers	-
19	IEEE Xplore	2018	Kapil Juneja et al.	A Rule Framed SVM Model for Classification of Various DDOS Attack in Distributed Network	SVM – support vector machines	the proposed method has improved the recognition rate for authenticated KDD dataset	Lacks to be proven with other datasets. It's possible for the model to be biased for this particular dataset
20	IEEE Xplore	2021	Kseniya Yu. Nikolskaya et al.	Development of a Mathematical Model of the Control Beginning of Ddos-Attacks and Malicious Traffic	K-means clustering	A novel methodology on which further improvement could be done	Is mostly theoretical as of now and is yet to be implemented
21	IEEE Xplore	2020	Runyu Li et al.	Early Detection of DDOS based on φ-entropy in SDN networks	Lightweight DDOS early detection method	Φ-entropy has better detection effect than Shannon entropy.	The calculation cost of the controller in this detection scheme has not been thoroughly studied.
22	IEEE Xplore	2018	Neelam Dayal et al.	An RBF-PSO Based Approach for Early Detection of DDoS Attacks in SDN	detection of DDoS attacks implemented with SDN controller-based om RBF network.	RBF network with PSO has better convergence rate than that of Neural Networks and Neural Network with PSO optimization	the flows with similar destination IP address and protocol as of victim IP address and attack protocols gets affected

23	IEEE Xplore	2018	Yaokai Feng et al.	Feature Selection for Machine Learning-Based Early Detection of Distributed Cyber Attacks	Detection using 55 features. SVM and PCA are utilized for feature selection and SVM and RF are for building the classifier	detection performance is generally getting better if more features are utilized	after the number of 40 features, the detection performance will not change.
24	IEEE Xplore	2019	Sanjeetha. R et al.	Early Detection and Diminution of DDoS attack instigated by compromised switches on the controller in Software Defined Networks	SDN Controller	Attack is detected instantly, highly efficient solution	Switch is compromised by manipulating the hard and idle timeout of the flow entry by the controller
25	Science Direct	2019	Nilesh patil et al.	E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks	Hadoop based distributive system E-HAD.	100% detection accuracy, 89% precision, 88%TNR, 100% NPV, 94% F-Measure, and 94% classification rate	Explore the use of more advanced/-Entropy and other information theory-based divergence measures in detecting different types of DDoS attacks
26	Science Direct	2018	Sunny Behal et al.	D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events	D-FACE defense system	outperformed the existing entropy-based defense systems in terms of high values of detection rate, precision.	impart more efforts to realize the proposed distributed model using software defined networking (SDN) technique to reduce cost

27	Scopus	2017	S. Moussavi et al.	Early Detection of DDoS Attacks Against Software Defined Network Controllers	Attack Detection algorithm	detection rate for this threshold was 96%.	an attack against an entire network may not be detected.
28	Complementary Index	2021	Reza et al.	A Statistical Model for Early Detection of DDoS Attacks on Random Targets in SDN	network traffic anomaly detection framework	successful detection of DDoS attacks as an anomalous deviation from our derived reference model.	-
29	IEEE Xplore	2019	Tamara .R et al.	Entropy Analysis Method for Attacks Detection	network traffic analysis method / maximum entropy method	Can detect various attacks with a probability of about 94%, while false-positive values did not exceed 10%.	it is necessary to conduct a comparative analysis of the speed and quality of the proposed method with other methods.
30	Science Direct	2019	Indraneel. S et al.	HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm	Bio-Inspired Anomaly based application layer DDoS attack	accuracy is approx. 91%	major obstacle observed these models are that compared to the proposed model is process complexity.
31	IEEE Xplore	2018	Bineet .K et al.	Early Detection of Distributed Denial of Service Attack in Era of Software-Defined Network	SDN	Highly efficient	in near future more of such techniques will evolved and used by researchers to detect the DDoS in early stages to either minimize its effects or to

							mitigate it completely.
32	IEEE Xplore	2018	D.C Grant et al.	Distributed Detection and Response for the Mitigation of Distributed Denial of Service Attacks	open-source honeypots with opensource intrusion prevention system/ TCP dump	feasible and relatively efficient	the third phase of this project is to integrate these remote sensors with a similarly funded project on Intelligent Intrusion Detection Systems.
33	IEEE Xplore	2020	Shahzeb. H et al.	A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks	Convolutional neural network (CNN)	CNN attains highest percentage in prediction of the PPV among the other ensemble and hybrid approaches	Time consuming
34	IEEE Xplore	2020	Jiabin li et al.	RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things	RTVD method/ QuinDC algo	a lowlatency and accurate performance	use LSTM or GRU for directly IoT DDoS detection, and focus on the Slow DDoS, as well as real-time volumetric DDoS taxonomy.
35	IEEE Xplore	2021	Bilal.H et al.	Deep Learning-Based DDoS-Attack Detection for Cyber–Physical System Over 5G Network	CNNs	This framework achieved higher than 91% normal and underattack cell detection accuracy by utilizing the DRC model for	the computation requirements need to be investigated keeping in view the online and offline settings

						silent call, signaling, and SMS flooding attacks	
36	IEEE Xplore	2020	Hwang. R et al.	An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection	traffic detection mechanism D-PACK/CNN/Deep Learning	100% accuracy, low false-positive rate	Time consuming
37	IEEE Xplore	2019	Ayush kumar et al.	EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques	EDIMA/ deep learning	High accuracy, high security	working on the software-based implementation of EDIMA and its performance evaluation
38	Science Direct	2018	Sagar.S et al.	An early detection of low-rate DDoS attack to SDN based data center networks using information distance metrics	Generalized Entropy (GE) based metric	High detection accuracy, highly efficient.	Use of (GE) technique for high-rate DDoS attack and try to set the threshold more dynamic way in a real traffic scenario.
39	IEEE Xplore	2017	Vaishali .K et al.	Proactive DDoS Attack Detection and Isolation	EDIP	Less number of proxies required	-
40	Science Direct	2021	Jisa.D et al.	Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm	computation of attributes and dynamic thresholding algorithm.	higher detection rate and lower false positives	It neither locates nor mitigates the attack flows.
41	ScienceDirect	2017	K. Munivara Prasad et al.	BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web	Bio-Inspired Anomaly based Real Time Detection of under	Significantly lower processing time in BARTD detector for higher load as compared to	Could only be used for HTTPS flood attacks, and error margin is quite high when

					rated App-DDoS Attack on Web using Cuckoo and bat/firefly search algorithms	Firefly and Bat algorithms	compared to others
42	ScienceDirect	2020	Agathe Blaise et al.	Detection of zero-day attacks: An unsupervised port-based approach	Use of Split-Merge detection techniques to detect port based anomaly.	Showed very promising detection model, even better than MAWILab, and the quantity of false-positive is really low	Can also implement this model in an SDN environment, using a controller and several switches running this algorithm.
43	IEEEAccess	2020	Shahzeb Haider et el.	A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks	Use of CNN based classifier in an SDN environment to detect DDoS attacks	Comparision between various Ensemble deep learning models like CNN, RNN, LSTM and RL	Hybrid models could be introduced to more efficient detection of attacks.
44	IEEE TrustCom	2016	Jian Zhang el al.	A Hadoop based analysis and detection model for IP Spoofing typed DDoS attack	Use of Hadloop based model to identify abnormal packets	Efficient detection is done through harnessing HBASE- a bloom filter based mapping mechanism named TCP2HC/UDP2HC	Further optimization of MapReduce function can happen, to perform real time packet analysis under high-speed links.
45	ICMETE	2018	Kapil Juneja et al.	SVM Model for Detection of Various DDOS Attack in Social Distributed Network	Detection of relevant features through probabilistic analysis	Calculating weights for different attributes before passing it through SVM model to	Discarding accuracy and efficiency to get faster predictions.

					then detection and classification through SVM based model.	perform faster analysis of the packets	
46	IEEEXplore	2021	Dr. R. Venkatesan et al.	A NOVEL APPROACH TO DETECT DDOS ATTACK THROUGH VIRTUAL HONEYPOT	Creation of virtual honeypot to prevent intrusion into the network at ISP level	Discussed mitigative capacity of virtual honeypots as a protective measure against DDoS Attacks	Theoretical consequences have been discussed with no practical example and demonstration
47	IEEEXplore	2017	Suman Sankar Bhunia et al.	Dynamic Attack Detection and Mitigation in IoT using SDN	They have proposed An SDN based secure IoT framework called SoftThings to detect abnormal behaviours	Establishment of SDN based framework to perform early detection of anomaly in attack prone IoT devices	Hardware prototype could be developed using IoT devices and Switches to get more insight in case of practical deployment.
48	MDPI	2019	Pedro Manso et al.	SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks	Designed a Software Defined Intrusion Detection System (IDS)	Their IDS reactively impairs the attacks at its origin, ensuring the normal operation of network infrastructure	A rule-based machine learning model could be incorporated to achieve effective detection.
49	IEEEXplore	2017	Xiaxia Zhou et al.	An adaptive filtration based defense framework against DDoS	They presented a spark-streaming based online DDoS defense	Novelty lies in joint online detection and mitigation schemes, to check abnormal inbound and	Further studies on growth of abnormal data flow using more spark operation to optimize

					Framework	outbound traffics	detection performance of model by real time stream processing can happen
50	IEEEXplore	2020	Monika Khatkar et al.	An overview of distributed denial of service and internet of things in healthcare devices	Detection and response at source hosts are deployed	Easier and cheaper than other DDoS attack prevention systems	High overhead storage and loading on routers
51	IEEEXplore	2019	N. S. Vishnu et al.	DENIAL OF SERVICE: TYPES, TECHNIQUES, DEFENCE MECHANISMS AND SAFE GUARDS	Provided broad description of DDoS attacks.	Segregated DoS based attacks in 3 types, volumetric, state-exhaustion and application layer oriented.	Provided only basic description of DDoS attacks, and their effects.
52	IEEEXplore	2020	Ren-Hung Hwang et al.	An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection	Used D-PACK a CNN based Unsupervised deep learning algorithm to detect anomalies	D-PACK only scans first few bits of the first few packets in each flow for early and efficient detection of DDoS attacks	Further optimization by exploiting advantages of deep learning models could be achieved to build an effective online anomaly detection system
53	Springer	2017	K. Munivara Prasad et al.	BIFAD: Bio-Inspired Anomaly Based HTTP-Flood Attack Detection	Use of Cuckoo and other Bio Inspired search algorithms	Use of Jaccard index to reduce dimensionality of the attributes	Further studies may include a hybrid strategy that uses more than one bio inspired technique to define anomaly detection

54	IEEEXplore	2020	Giovane C. M. Moura et al.	Into the DDoS maelstrom: a longitudinal study of a scrubbing service	Discussed about inner workings of DDoS scrubber NaWas	Determined that the collateral damage incurred by DDoS attacks per victim is at least quadratically larger than targeted addresses	They discussed how DDoS footprint scan over DNS authoritative traffic could result in early detection.
55	IEEEXplore	2016	Suzan Almutairi et al.	Peer to Peer Botnet Detection Based on Network Traffic Analysis	This paper addressed the problem of detecting P2P botnet flow records from P2P applications	Their algorithm extracted features to differentiate between different botnet behavior among legitimate and anomalous	To further improve accuracy metric, proper preprocessing algorithm could be developed.
56	IEEEXplore	2017	Wei Wei et al.	Research and Simulation of Queue Management Algorithms in Ad Hoc Networks Under DDoS Attack	They introduced three queue management algorithms to detect and handle DDoS attacks	Active queue management algorithms such as REM and RED exhibited stronger defensive abilities	Due to inherent shortcomings of queue management algorithms, they have to be integrated with other strategies
57	IEEEXplore	2020	Jiabin Li et al.	RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things	They had used an entropy-based detection method	Entropy calculation through Sliding time window.	Future studies could focus on Slow DDoS detection using LSTM or GRU
58	MDPI	2019	Ruomeng Xy et al.	A DRDoS Detection and Defense Method Based on Deep	Used HDTI feature to build a DRDoS detection	Their model was based on XGBoost forest estimator, and a combination	Future works may include fusion based detection and

				Forest in the Big Data Environment	and defense model	of 2 random forest estimators slightly different to each other, at different intervals to efficiently detect	defense method.
59	Hindawi	2018	Jieren Cheng et al.	Adaptive DDoS Attack Detection Method Based on Multiple-Kernel Learning	This paper focuses on an adaptive DDoS attack detection method (ADADM) based on multiple kernel learning (MKL)	They have used an ensemble learning framework to calculate weight of each dimension and then adapt their model through gradient descent to establish early identification of DDoS attacks	Further studies may include better transform of the multidimensional weight through a convex optimization technique and hence improve detection rate and convergence speed of the method
60	IEEEXplore	2019	Luis A. Trejo et al.	DNS-ADVP: A Machine Learning Anomaly Detection and Visual Platform to Protect Top-Level Domain Name Servers Against DDoS Attacks	Use of DNS-ADVP a DNS anomaly Detection Visual Platform, which in an integrated manner will provide proper visualization of the online DNS traffic	They have also included a once-class specifier which will classify anomalous packets on basis of this visualization and hence will regularly update what counts as normal behavior	Future studies may reduce latency of the system through implementing it with GPUs, to accelerate numerical computations and also take advantage of the inherent parallelism.

Comparative Study

- *Genetic Algorithms:*

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. Various such algorithms are used in DDoS Detection, this could be further read upon from [6], [30], [53] etc.

- *Machine Learning Based Algorithms:*

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. DDoS Detection employs a variety of such methods, as demonstrated by [9], [45], [52], and others.

- *IoT & SDN based Networks:*

Software-Defined Networking (SDN) is an approach to networking that uses software-based controllers or application programming interfaces (APIs) to communicate with underlying hardware infrastructure and direct traffic on a network. SDN separates control plane and data forwarding plane by centralizing control and programmability of network. As proven by [17], [38], [48], and others, DDoS Detection utilizes a number of similar approaches.

Proposal

Step 1: sandbox a DDoS attack using a CLI tool

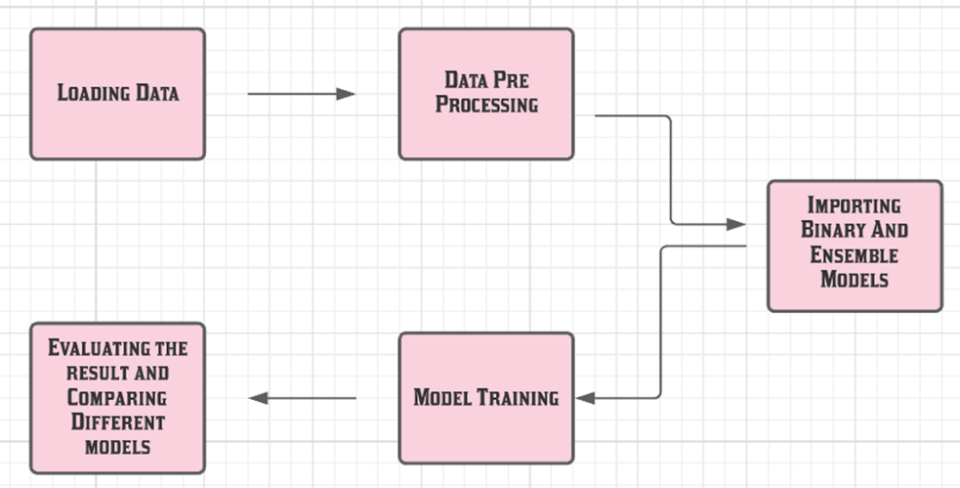
DDoS attacks target websites and online services. Their primary goal is to overburden the server with get requests. As a result, Server is unable to react to incoming requests and, this results in, services getting interrupted. And hence an effective attack is conducted.

So, to achieve this sandbox we'll be using a Linux based CLI tool which will flood the server with multiple requests and hence make it incapable to respond to any other requests.

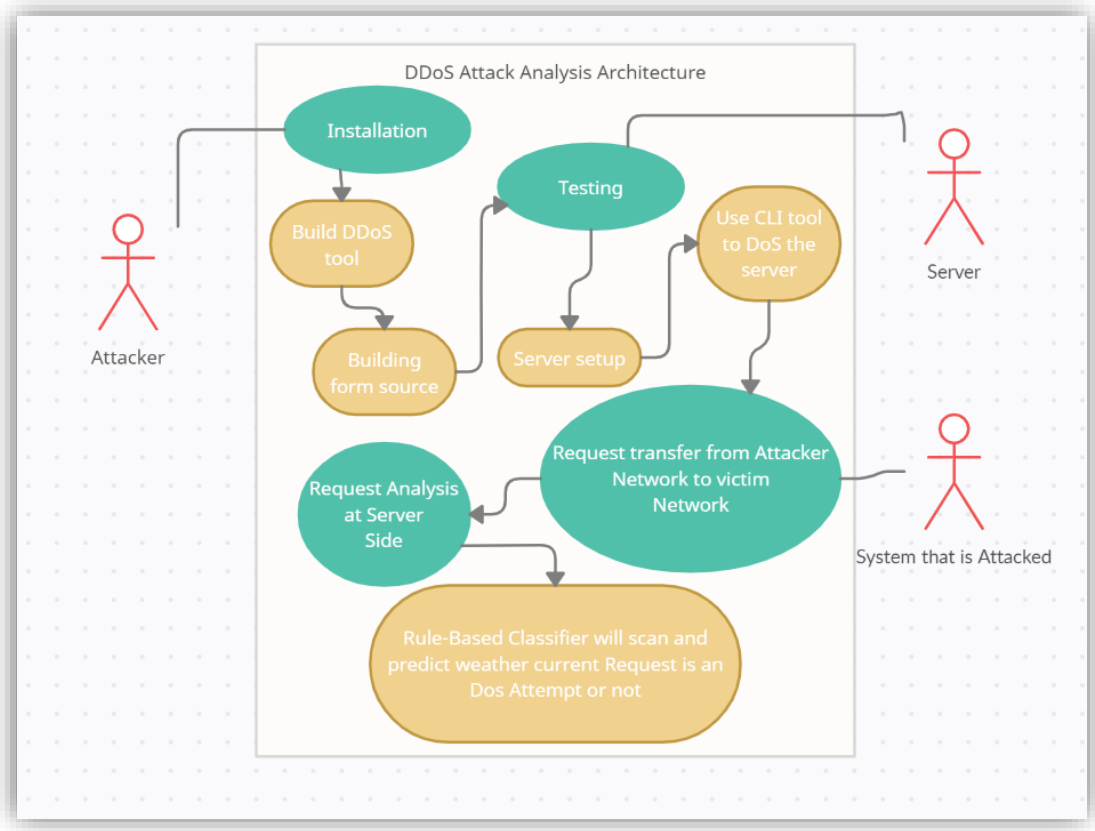
Step 2: analyze the data through rule-based machine learning algorithm

Every minute that the server is running, it creates data. This information may be gathered and saved as a CSV file for further examination. Wireshark would be used to collect information about the packet requests that the server receives when it is under siege in this scenario. After exporting the data as a CSV file, we'll import it as a dataset and use Python to do some basic preprocessing. We will apply a Rule-based Machine Learning Classifier to the data after it has been prepared. This algorithm will classify websites to determine whether or not they are vulnerable to DDOS attacks.

Low Level Diagram



Architecture diagram & Use Case Diagram



In the above given method, we have added the dataset to our data frames and then we have preprocessed the data in our data frames.

After this we have imported ensemble models. After this we have trained these models based on our training dataset. After this we have tested the data based on our testing data, based on which we have calculated the accuracy and precisions of our model.

Dataset Description

The raw network packets of the UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors.

Tcpdump tool is utilised to capture 100 GB of the raw traffic (e.g., Pcap files). This dataset has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The Argus, Bro-IDS tools are used and twelve algorithms are developed to generate totally 49 features with the class label.

The number of records in the training set is 175,341 records and the testing set is 82,332 records from the different types, attack and normal.

```
df1.columns
```

```
Index(['id', 'dur', 'proto', 'service', 'state', 'spkts', 'dpkts', 'sbytes',
       'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss', 'dloss',
       'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb', 'dwin',
       'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth',
       'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm',
       'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm',
       'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm',
       'ct_srv_dst', 'is_sm_ips_ports', 'attack_cat', 'label'],
      dtype='object')
```

Experimental Setup and Result Analysis

• **Phase 1**: Simulation of DDoS attack in kali linux using xerxes tool

During our research we found that a simple DDOS attack is not sufficient to flood the server effectively to enhance the attack we can use the xerxes tool.

Xerxes is an extremely powerful and efficient dos tool. It provides the capacity to launch multiple independent attacks against several target sites without necessarily requiring a botnet. At the core, it converts a single threaded application to multi-threaded and sends multiple requests to the server, thus flooding it and making it incapable to respond to any request.

So here we use xerxes tool to enhance our DDoS attack so that we can analyze the worst condition.

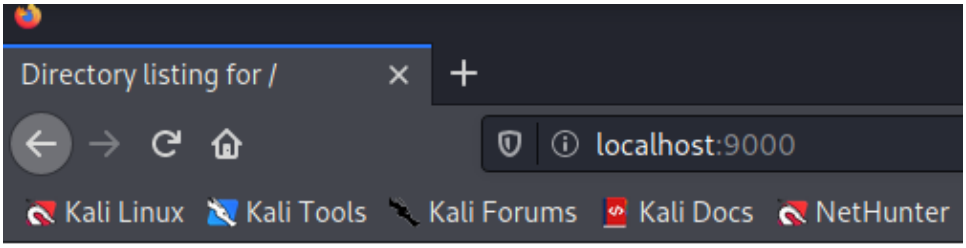
First we'll build our copy of the Xerxes tool.

```
┌──(kali㉿kali)-[~/Desktop/Xerxes]
└─$ cd Xerxes && mkdir build && cd build && cmake .. && make
-- The C compiler identification is GNU 10.2.1
-- The CXX compiler identification is GNU 10.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
```

Afte creating Xerxes, we'll host our local python HTTP server

```
┌──(kali㉿kali)-[~]
└─$ python -m SimpleHTTPServer 9000
Serving HTTP on 0.0.0.0 port 9000 ...
```

Basic status of our server before DDoS



Directory listing for /

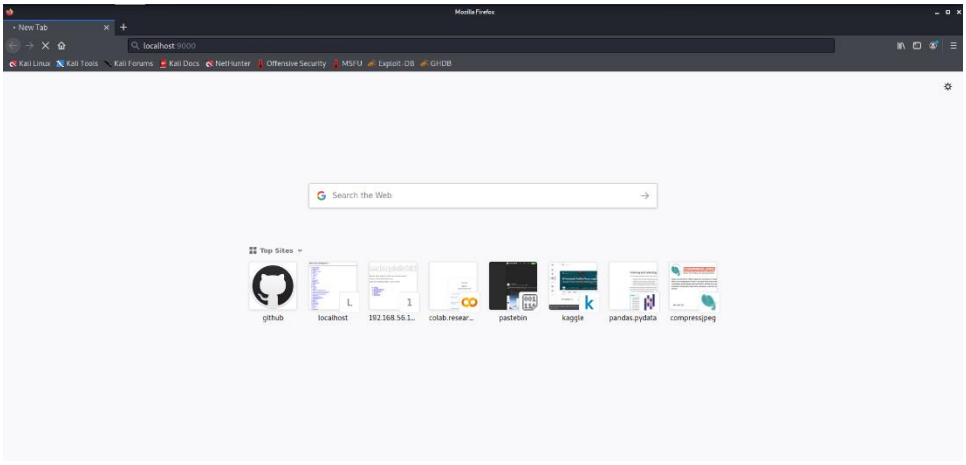
- .bash_history
- .bash_logout
- .bashrc
- .bashrc.original
- .bully/
- .BurpSuite/
- .cache/
- .config/
- .dmrc
- .face
- .face.icon@
- .gnupg/
- .ICEauthority

Then initiating DDoS through Xerxes

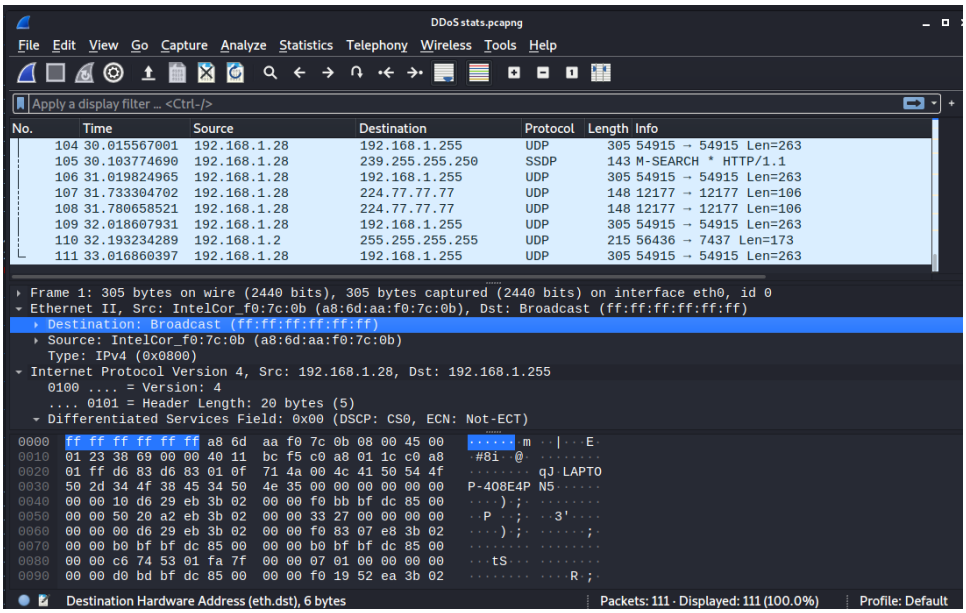
```
┌──(kali㉿kali)-[~/…/Xerxes/Xerxes/build/bin]
└─$ ./Xerxes -h 127.0.0.1 -p 9000
--==[ Xerxes enhanced by Sepehrdad Sh ]==--

[info] Connected → 127.0.0.1:9000
[info] Voly Sent
[info] Connected → 127.0.0.1:9000
[info] Connected → 127.0.0.1:9000
[info] Voly Sent
[info] Connected → 127.0.0.1:9000
[info] Voly Sent
[info] Connected → 127.0.0.1:9000
[info] Voly Sent
[info] Connected → 127.0.0.1:9000
```

And Server status during DDoS



Simultaneously scanning through Wireshark



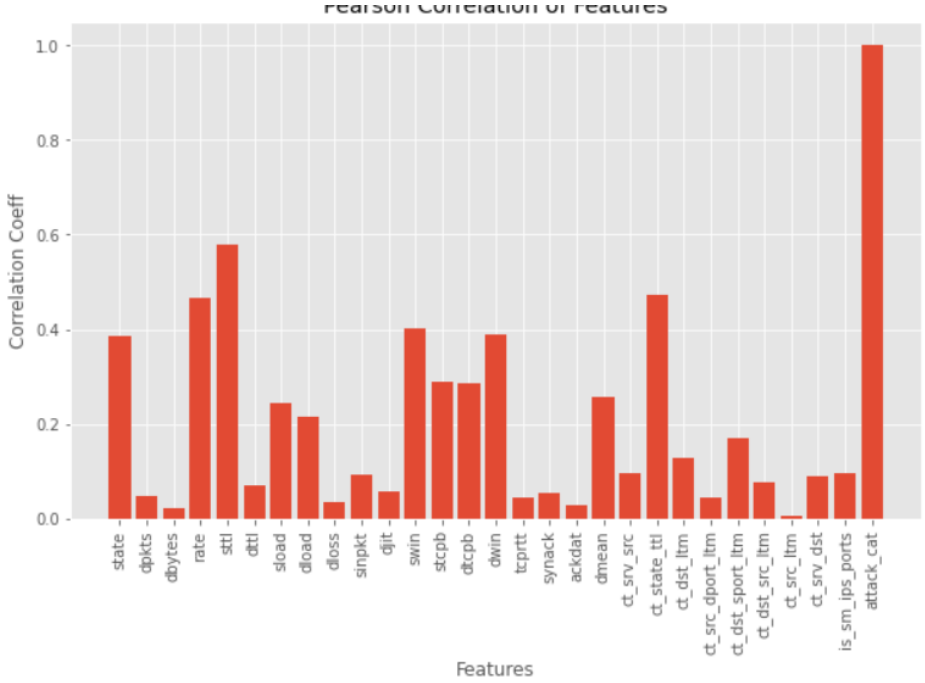
To gain the dataset in sample format for ML analysis

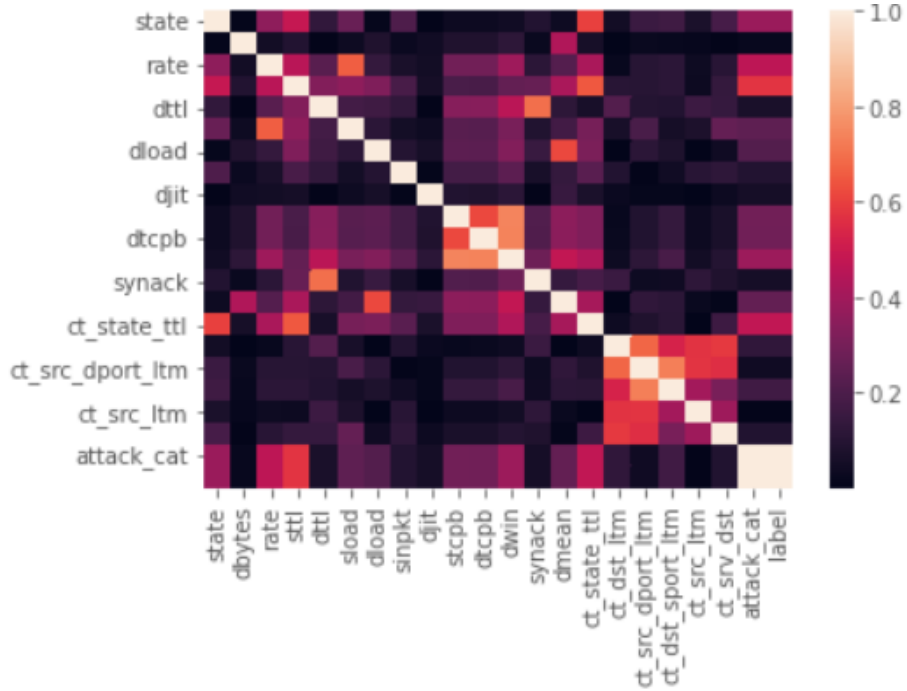
• **Phase 2**: Analyzing the data generated by the server under attack by using various Machine Learning algorithm.

The server under attack generates several MB every minute that it runs. This data can be collected and stored in a CSV file for further analysis. Here, we use Wireshark as our scanning tool to collect information about the packet requests that the server receives when it is under attack. After saving this data as a CSV file, we import this as a dataset on Google Colaboratory and conduct some basic preprocessing using the python numpy and pandas library.

After preparing the data, we apply various Machine Learning Algorithms. Then these algorithms classify websites to see whether they are being attacked by a DDoS attack or not.

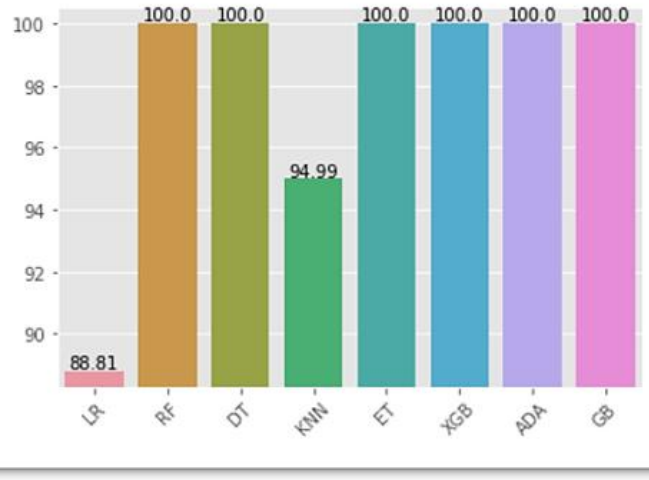
Pearson Correlation and Heat Map of our Various Features with Attack Categories





Basic Accuracy results for our analysis between various ensemble models, including entropy based, Gini-based models and classical logistic regression.

```
plot_scores(model_names, scores, True)
```



We had also prepared Confusion matrices for our results which constitutes of True Positive, True Negative, False Positive and False Negative values for our testing set, with Precision, Recall and F-Score of our Models, and time taken to predict 41089 testing set values.

```
LR
[[33810  3190]
 [ 1407  2682]]
Precision:  0.91
Recall:  0.96
F-score: 0.9363446279961782
Time to predict:  0.011507158279418945
```

```
RF
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.2359306812286377


DT
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.009255409240722656


KNN
[[36258   742]
 [ 1315  2774]]
Precision:  0.98
Recall:  0.97
F-score: 0.9724162900781784
Time to predict:  49.6117970943451


ET
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.23770928382873535


XGB
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.04407310485839844
```

```
ADA
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.013460397720336914


GB
[[37000     0]
 [    0  4089]]
Precision:  1.00
Recall:  1.00
F-score: 1.0
Time to predict:  0.030294418334960938
```

Tabulated Result:

Models	Precision	Recall	F-score
Logistic Regression	0.91	0.96	0.936
Random Forest Classifier	1.00	1.00	1.00
Decision Tree Classifier	1.00	1.00	1.00
K Neighbours Classifier	0.98	0.97	0.972
Extra Trees Classifier	1.00	1.00	1.00
XGB Classifier	1.00	1.00	1.00
Ada Boost Classifier	1.00	1.00	1.00
Gradient Boosting Classifier	1.00	1.00	1.00

Conclusion & Future Enhancements

From the above result table, we can see that ensemble models have performed far better than the classical models. Their accuracy and precision are far better than that of the classical models, in addition to which the prediction time is also less than those of the classical models.

Future enhancements could include physical hardware implementation of such Entropy-Based algorithms, where hardware will automatically flag suspicious packets and then inform the user/organization of such malicious activity.

List of References

1. A Review on Hot-IP Finding Methods and Its Application in Early DDoS Target Detection
2. A DDoS Attack Detection System Based on Spark Framework
3. MLP-GA based algorithm to detect application layer DDoS attack
4. A Pragmatic Cybersecurity Strategies for Combating IoT-Cyberattacks
5. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm
6. LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks
7. Detection of DDoS attacks with feed forward based deep neural network model
8. Collaborative Detection and Mitigation of Distributed Denial-of-Service Attacks on Software-Defined Network
9. Introducing a New Linear Regression Based Method for Early DDoS Attack Detection in SDN
10. Robust Early Stage Botnet Detection using Machine Learning
11. An Efficient Hybrid Model For Detecting Distributed Denial Of Service (Ddos) Attacks In Cloud Computing Using Multivariate Correlation And Data Mining Clustering Techniques
12. Cloud DDoS Detection and Defense System using Complex Event Processing
13. A Distributed Architecture for DDoS Prediction and Bot Detection
14. Detection and Mitigation of DDoS Attacks Using Conditional Entropy in Software-defined Networking
15. DNS-ADVP: A Machine Learning Anomaly Detection and Visual Platform to Protect Top-Level Domain Name Servers Against DDoS Attacks
16. Statistical Approach Based Detection of Distributed Denial of Service Attack in a Software Defined Network
17. S-DPS: An SDN-Based DDoS Protection System for Smart Grids
18. Combating DDoS Attacks with Fair Rate Throttling
19. A Rule Framed SVM Model for Classification of Various DDOS Attack in Distributed Network
20. Development of a Mathematical Model of the Control Beginning of Ddos-Attacks and Malicious Traffic
21. Early Detection of DDOS based on ϕ-entropy in SDN networks
22. An RBF-PSO Based Approach for Early Detection of DDoS Attacks in SDN
23. Feature Selection for Machine Learning-Based Early Detection of Distributed Cyber Attacks
24. Early Detection and Diminution of DDoS attack instigated by compromised switches on the controller in Software Defined Networks
25. E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks
26. D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events
27. Early Detection of DDoS Attacks Against Software Defined Network Controllers
28. A Statistical Model for Early Detection of DDoS Attacks on Random Targets in SDN
29. Entropy Analysis Method for Attacks Detection
30. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm

31. Early Detection of Distributed Denial of Service Attack in Era of Software-Defined Network
32. Distributed Detection and Response for the Mitigation of Distributed Denial of Service Attacks
33. A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks
34. RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things
35. Deep Learning-Based DDoS-Attack Detection for Cyber–Physical System Over 5G Network
36. An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection
37. EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques
38. An early detection of low-rate DDoS attack to SDN based data center networks using information distance metrics
39. Proactive DDoS Attack Detection and Isolation
40. Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm
41. BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web
42. Detection of zero-day attacks: An unsupervised port-based approach
43. A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks
44. A Hadoop based analysis and detection model for IP Spoofing typed DDoS attack
45. SVM Model for Detection of Various DDOS Attack in Social Distributed Network
46. A NOVEL APPROACH TO DETECT DDOS ATTACK THROUGH VIRTUAL HONEYPOT
47. Dynamic Attack Detection and Mitigation in IoT using SDN
48. SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks
49. An adaptive filtration based defense framework against DDoS
50. An overview of distributed denial of service and internet of things in healthcare devices
51. DENIAL OF SERVICE: TYPES, TECHNIQUES, DEFENCE MECHANISMS AND SAFE GUARDS
52. An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection
53. BIFAD: Bio-Inspired Anomaly Based HTTP-Flood Attack Detection
54. Into the DDoS maelstrom: a longitudinal study of a scrubbing service
55. Peer to Peer Botnet Detection Based on Network Traffic Analysis
56. Research and Simulation of Queue Management Algorithms in Ad Hoc Networks Under DDoS Attack
57. RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things
58. A DRDoS Detection and Defense Method Based on Deep Forest in the Big Data Environment
59. Adaptive DDoS Attack Detection Method Based on Multiple-Kernel Learning
60. DNS-ADVP: A Machine Learning Anomaly Detection and Visual Platform to Protect Top-Level Domain Name Servers Against DDoS Attacks
61. UNSW_NB15 Dataset
62. Xerxes Tool

Code

```
!pip install kaggle

!mkdir ~/.kaggle

from google.colab import files

files.upload()

!cp kaggle.json ~/.kaggle

!chmod 600 ~/.kaggle/kaggle.json

!kaggle datasets download mrwellsdavid/unsw-nb15

!unzip unsw-nb15.zip

import pandas as pd
import numpy as np
from numpy import mean,std
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')

import warnings
warnings.filterwarnings("ignore")

df1 = pd.read_csv("UNSW_NB15_testing-set.csv")
df1.dataframeName = 'wiresharkdata.csv'
nRow,nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')

df1.head()

df1.columns

df2 = pd.read_csv("UNSW_NB15_training-set.csv")
df2.dataframeName = 'wiresharkdatatest.csv'
nRow,nCol = df2.shape
print(f'There are {nRow} rows and {nCol} columns')

df2.head()

df1=df1.drop(['id'],axis=1)

df2=df2.drop(['id'],axis=1)

for x in df1.columns:
  print(df1[x].isnull().values.any())

df1
```

```
plt.xticks(rotation=90)
sns.countplot(df1['attack_cat'])

df1.drop(df1[(df1['attack_cat'] != 'DoS') & (df1['attack_cat'] != 'Normal')].index,
inplace = True)
df2.drop(df2[(df2['attack_cat'] != 'DoS') & (df2['attack_cat'] != 'Normal')].index,
inplace = True)

df1

plt.xticks(rotation=90)
sns.countplot(df1['state'])

plt.xticks(rotation=90)
sns.countplot(df1['service'])

from sklearn.preprocessing import LabelEncoder
from sklearn.utils import column_or_1d

class MyLabelEncoder(LabelEncoder):

    def fit(self, y, arr=[]):
        y = column_or_1d(y, warn=True)
        if arr == []:
            arr=y
        self.classes_ = pd.Series(arr).unique()
        return self

le = MyLabelEncoder()

df1.select_dtypes("object").info()

for feature in df1.select_dtypes("object").columns:
    df1[feature]=le.fit_transform(df1[feature])

df1

for feature in df2.select_dtypes("object").columns:
    df2[feature]=le.fit_transform(df2[feature])

df2

pcorr=df1.drop('label',1).corrwith(df1['label'])
plt.figure(figsize=(10,6))
plt.title("Pearson Correlation of Features")
plt.xlabel("Features")
plt.ylabel("Correlation Coeff")
plt.xticks(rotation=90)
plt.bar(pcorr.index, list(map(abs,pcorr.values)))

df1=df1.drop(['dur','proto','service','spkts','sbytes','dinpkt','response_body_len','
sloss','sjit','smean','trans_depth','is_ftp_login','ct_ftp_cmd','ct_flw_http_mthd'],a
xis=1)

df2=df2.drop(['dur','proto','service','spkts','sbytes','dinpkt','response_body_len','
```

```
sloss','sjit','smean','trans_depth','is_ftp_login','ct_ftp_cmd','ct_flw_http_mthd'],a
xis=1)

pcorr=df1.drop('label',1).corrwith(df1['label'])
plt.figure(figsize=(10,6))
plt.title("Pearson Correlation of Features")
plt.xlabel("Features")
plt.ylabel("Correlation Coeff")
plt.xticks(rotation=90)
plt.bar(pcorr.index, list(map(abs,pcorr.values)))

sns.heatmap(df1.corr().apply(abs))

df1 = df1.drop(['dpkts', 'dloss', 'is_sm_ips_ports', 'tcprtt', 'ackdat',
'ct_srv_src', 'ct_dst_src_ltm', 'swin'], axis=1)

df2 = df2.drop(['dpkts', 'dloss', 'is_sm_ips_ports', 'tcprtt', 'ackdat',
'ct_srv_src', 'ct_dst_src_ltm', 'swin'], axis=1)

sns.heatmap(df1.corr().apply(abs))

df1.info()

y_train=df1['label']
X_train=df1.drop('label',1)

y_test=df2['label']
X_test=df2.drop('label',1)

seed= 42

from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier

models = {
    'LR':LogisticRegression(random_state=seed),
    'RF':RandomForestClassifier(random_state=seed,criterion='entropy'),
    'DT':DecisionTreeClassifier(random_state=seed, criterion='entropy'),
    'KNN':KNeighborsClassifier(),
    'ET':ExtraTreesClassifier(random_state=seed,criterion="entropy"),
    'XGB':XGBClassifier(random_state=seed),
    'ADA':AdaBoostClassifier(random_state=seed),
    'GB':GradientBoostingClassifier(random_state=seed)
    }

def plot_scores(xval,yval,show_value=False):
    plt.ylim(ymax = max(yval)+0.5, ymin = min(yval)-0.5)
    plt.xticks(rotation=45)
    s = sns.barplot(xval,yval)
    if show_value:
```

```
        for x,y in zip(range(len(yval)),yval):
            s.text(x,y+0.1,round(y,2),ha="center")

models =[(key,value) for key,value in models.items()]

print(models)

import time
scores=[]
preds=[]
t=[]
for model in models:
    model[1].fit(X_train,y_train)
    print(model[0],"trained.")
    scores.append(model[1].score(X_test,y_test))
    start_time = time.time()
    preds.append(model[1].predict(X_test))
    t.append((time.time()-start_time)/(y_test.size))
print("Results are ready.")

from sklearn.metrics import confusion_matrix

model_names= [i[0] for i in models]
scores = list(map(lambda x: x*100, scores))

index = 0
#41089
for i in preds:
  print(model_names[index])
  cm = confusion_matrix(y_test, i)
  print(cm)
  p = cm[0][0]/(cm[0][0] + cm[0][1])
  r = cm[0][0]/(cm[0][0] + cm[1][0])
  print("Precision: ", '%.2f'%p)
  print("Recall: ", '%.2f'%r)
  print("F-score:", 2*p*r/(p+r))
  print("Time to predict: ", t[index]*y_test.size)
  print("\n")
  index = index + 1

plot_scores(model_names, scores, True)
```