

# **Mapping Network Features to Attack profiles to Enhance the Real Time Intrusion Detection**



**MCS**

Author

**Joveria Rubaab**

Registration number

**00000317611**

Supervisor

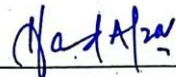
**Professor Dr. Hammad Afzal**


A thesis submitted to the faculty of Department of Computer Software Engineering,  
Military College of Signals, National University of Sciences and Technology (NUST),  
Rawalpindi in partial fulfillment of the requirements for the degree of MS in Computer  
Software Engineering.


(August 2023)

**THESIS ACCEPTANCE CERTIFICATE**

Certified that final copy of MS Thesis written by Mr. Joveria Rubab, Registration No. 00000317611, of Military College of Signals has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

Signature:   
Name of Supervisor: Assoc Prof Dr. Hammad Afzal  
Date: 31/8/2023

Signature (HOD):   
Date: 21/9/23

Signature (Dean/Principal)   
Date: 21/9/23  
Brig  
Dean, MCS (NUST)  
(Asif Masood, Phd)

# Declaration

I, Joveria Rubaab declare that this thesis “**Mapping Network Features to Attack profiles to Enhance the Real Time Intrusion Detection**” and the work presented in it are my own and has been generated by me as a result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST.
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated.
3. Where I have consulted the published work of others, this is always clearly attributed.
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help.
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself Joveria Rubab , 00000317611.



Joveria Rubab

NUST 00000317611 MSSE26

# **Dedication**

I thank Almighty Allah, The Most Gracious and The Most Merciful.

This thesis is dedicated to my beloved parents and my son Muhammad Ibrahim. Their unwavering support and encouragement have been the cornerstone of my success. Their love and sacrifices have inspired me to strive for excellence in my studies and in all aspects of my life.

# Abstract

The immeasurable amount of data in network traffic has increased its vulnerability. Therefore, monitoring and analyzing traffic for threat hunting is inevitable. Analyzing and capturing real-time network traffic is challenging due to privacy and space concerns. However, many simulated datasets are available. Machine-learning based intrusion detection systems are trained on these datasets for attack detection. Selection of correct features has significant importance in determining the efficiency of various ML-based algorithms. Hence, this paper provides a literature survey of the various machine learning based IDS. Features, attacks, machine learning algorithms and their corresponding datasets are identified in the survey. The survey may help researchers in identifying benchmark features correlated to network attacks. After a comprehensive survey, we selected one of the papers and did our experimentation on the feature set advised by the author. We reduced the feature set further and defined unique datasets corresponding to each attack. The reduced dataset further enhanced efficiency of the model by reducing execution time and improving space complexity. At the time of writing this thesis paper there is no such IDS that associates network features to attacks.

**Keywords:** IDS-Intrusion Detection System, DoS- Denial of Service, Cyber space, NetFlow

# Acknowledgments

All praises to Allah for the strengths and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor, Dr. Hammad Afzal, PhD, for his supervision and constant support. His priceless help of constructive comments and suggestions throughout the experimental and thesis works are major contributions to the success of this research. Also, I would thank my teacher Dr Waleed bin Shahi for his support and guidance throughout.

Lastly, I am highly thankful to my parents for their constant support. I would like to thank them for their patience, cooperation and motivation in times of stress and hard work.

# Contents

## Mapping Network Features to Attack profiles to Enhance the Real Time Intrusion Detection

.....	i
<b>Chapter 1 .....</b>	<b>1</b>
Introduction.....	1
1.1 Overview .....	1
1.2 Motivation and Problem Statement.....	2
1.3 AIMS and Objectives.....	2
1.4 Research Contribution.....	3
1.5 Thesis Organization .....	3
<b>Chapter 2 .....</b>	<b>4</b>
2.1 Network Attacks .....	4
2.1.1 DDoS Attack:.....	4
2.1.2 Man in the Middle Attack:.....	5
2.1.3 Passive Scanning:.....	5
2.1.4 Malware:.....	5
2.1.5 Social Engineering Attacks: .....	5
2.1.6 Security Breach:.....	6
<b>Chapter 3 .....</b>	<b>7</b>
3.1 Literature Survey .....	7
3.2 Shortcoming of Existing Literature Network .....	13
<b>Chapter 4 .....</b>	<b>14</b>
4.1 Methodology and Framework .....	14
4.1.1 Data Collection .....	14
4.1.2 Features .....	17
4.1.3 Data Cleaning: .....	19
4.1.4 Feature Selection.....	20
<b>Chapter 5 .....</b>	<b>33</b>

5.1 Results and Evaluation:.....	33
5.1.1 Metrics.....	33
5.1.2 Time .....	33
5.1.3 Space complexities.....	34
5.2 Analysis of Experimental Results.....	34
<b>Chapter 6 .....</b>	<b>41</b>
6.1 Conclusion and Future Work.....	41
6.2 Limitations .....	42
6.3 Future work .....	42

## List of Figures

Figure 1 .....	4
Figure 2 .....	6
Figure 3 .....	18
Figure 4 .....	21

## List of Tables

Table 1 .....	13
Table 2 .....	14
Table 3 .....	15
Table 4 .....	16
Table 5 .....	17
Table 6 .....	19
Table 7 .....	31
Table 8 .....	33
Table 9 .....	41
Table 10 .....	41



# List of Abbreviations

<b><i>SVM</i></b>	<i>Support Vector Machine</i>
<b><i>MITM</i></b>	<i>Man in the Middle</i>
<b><i>DoS</i></b>	<i>Denial of Service</i>
<b><i>NIDS</i></b>	<i>Network Intrusion Detection Systems</i>
<b><i>ICMP</i></b>	<i>Internet Control Message Protocol</i>
<b><i>NF</i></b>	<i>Netflow</i>

# Chapter 1

## Introduction

### 1.1 Overview

The 21st century has revolutionized human society. Electronic devices on the internet have been growing at a rapid pace. Advent of COVID-19 has increased human's dependence on electronic devices many fold. According to research by the International Data Corporation (IDC), it is estimated that the number of IoT devices will reach 41.6 billion till 2025 [1]. The increase in the number of devices connected with the internet has exposed the threat of intrusion as well. The recently discovered spyware Pegasus [2], happened to have intruded into many cell phones via a single text. A cyber arm of Israeli company Niv, Shalev and Omri (NSO), established the spyware to exploit versions of iOS and Android cell phones. The same spyware has been used by different governments for clandestine operations.

Cyber security has become an ever growing challenge for researchers, due to an increased number of attacks. Network Intrusion Detection Systems (NIDSs) play a vital role. NIDS senses network attacks and preserves the three principles of information security: confidentiality, integrity, and availability [3]. Signature-based NIDSs compare attack's signatures to detect traffic, giving high detection accuracy to identified attacks. However, for zero-day attack machine learning algorithms are comparatively more reliable. Researchers are working to upgrade performance of NIDSs to detect any unforeseen malicious activities. [2] is evidence that network security can be compromised by novel methods. Machine learning models can detect malicious patterns that may threaten the security. Each intrusion leaves a unique set of patterns that assists in its classification. To apply a supervised learning model a labeled dataset is required, which labels network data flows as benign or attack.

Real-network traffic is hard to obtain because of privacy concerns; as a result, researchers have designed network test-beds to generate synthetic datasets. Research community has applied many models; however, deployment of such models practically is still scarce. Reason is lack of common set of network features across all datasets, hence, a set of common features is required. The University of Queensland Australia has suggested 4 new datasets with 12 and 43 common features, to achieve this objective. These datasets are publically available to detect intrusions [4][5]. PCAP files contain huge amounts of data, therefore, they are converted to NetFlows.

NetFlow is an industry standard protocol for network traffic collection [6]. Its practical and scalable properties enhance the deployment feasibility of ML-based NIDSs. NetFlow features key

security events that are crucial in the identification of network attacks. Application of NetFlow-based features set will facilitate the successful deployment of ML-based NIDS.

Four widely used NIDSs datasets, referred as UNSW-NB15, CSE-CIC-IDS2018, BoT-IoT and ToNc are characterized into attack or benign class [4][5]. Although the researchers have done a commendable job to identify key features and have attained good accuracy; however, they are unable to identify domain knowledge and reasons for selection of specified features.

Hence, I plan to use the model proposed in the [4][5] to examine live traffic. The result will help us analyze its applicability in the real world. I would also design a Taxonomy of features affiliated to relevant attacks. Domain knowledge and feature info gain can be used to design the taxonomy. After experimentation it will be decided to enhance the model that would allow us to design NIDS with increased performance in real time.

## **1.2 Motivation and Problem Statement**

According to the Washington Post, estimated global losses from cybercrime are projected to hit just under a record \$1 trillion for 2020[16]. Likewise, future wars might not be fought on battle ground but in cyberspace. Cyberspace is a fifth-generation warfare domain and has recently attracted attention of many developed and developing countries. The reports of government websites being hacked, and sensitive data being stolen by foreign groups are not new anymore. Effective cyber-attacks not only compromise personal user information but can also cripple an entire nation's infrastructure. Cyber Security has been therefore recognized as a global problem, transcending national boundaries. Therefore, a country with an effective cyber security system and solution may be considered fittest for the war. Network intrusion detection being of prime facets of this domain, enjoys great importance in the research arena.

## **1.3 AIMS and Objectives**

Following are the objectives of the proposed research:

- To devise a machine learning/deep learning-based methodology to propose a real time network traffic analysis framework that is capable of performing in real time.
- To study the effects of various existing feature engineering techniques and propose an optimum set of features that can achieve the best performance.
- To design a taxonomy of related features corresponding with relevant attacks using domain knowledge or feature info gain.

## 1.4 Research Contribution

A strong cyber defense system can make a country safe and secure. Intrusion detection is the first step towards cyber security. The research may help communities to effectively secure their systems.

According to The News [17] Pakistani intelligence agencies have tracked a major security breach by Indian hackers whereby phones and other gadgets of government officials and military personnel were targeted. According to a statement by the Inter-Services Public Relations (ISPR), the cyber-attack by Indian intelligence agencies involved "a range of cybercrimes including deceitful fabrication by hacking personal mobiles and technical gadgets". "Pakistan Army has further enhanced necessary measures to thwart such activities including action against violators of standing operating procedures (SOPs) on cybersecurity," added the statement. It also said that an advisory is being sent to all government departments so they may identify security lapses and enhance cybersecurity measures. Under such circumstances an efficient, effective, and feasible intrusion detection system can be helpful.

## 1.5 Thesis Organization

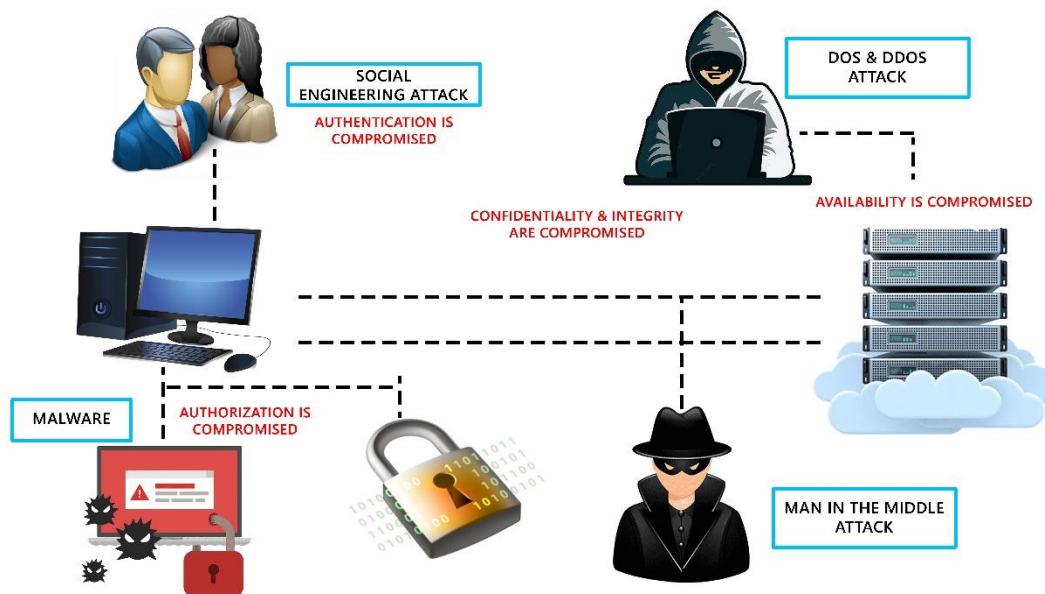
The thesis has been organized as follows:

- Chapter 2 gives an overview of all the network attacks.
- Chapter 3 literature survey related to all the network intrusion detection systems.
- Chapter 4 covers methodology and framework of for experimentation. Collection of data set, data cleaning, feature selection and extraction and classification model are defined in the chapter 4. The comparative analysis of previous studies is also presented which shows how classification has improved with newly identified features.
- Chapter 5 discusses results of experimentation performed in Chapter 4. It presents accuracy, time and space complexity of the model with new identified features.
- Chapter 6 is conclusion and future work which summarizes the research work, presents the limitations of the study and the proposed framework with respect to the network attack identification It also suggests future direction in the corresponding domain.

## Chapter 2

### 2.1 Network Attacks

The emergence of newer technologies has enhanced communication and proved advantages. However, it has increased the risk of attacks. For designing a secure network confidentiality, integrity, non-repudiation, access control and availability must be ensured [5]. Network attacks are illegal actions aimed at disrupting the regular functioning of network. Motivation behind network attacks can be economic benefits, clandestine operations, destroying someone's reputation, revenge, or intellectual challenge [6]. Fig 1 shows a summary of network attacks.



Summary of Network Attack

Figure 1

#### 2.1.1 DDoS Attack:

Denial of service is either classified as denial of resources or bandwidth. It is usually targeted at corporate organizations like banks, universities, or government websites. DOS and DDOS are two major types of denial of service. In DOS malicious node consumes bandwidth of network node and makes it unavailable. DOS includes Hulk, Goldeneye, Slowloris, and Slowhttpstest. Application layer DOS attacks are executed through Ddossim, Goldeneye, Hulk, RUDY Slowhttpstest, Slowloris [7]. DDOS, works like an army of zombies. A master node recruits handlers, handlers in return recruit agents and they finally attack the victim. Handlers and agents are also referred to as Botnets. Botnets are silent malwares activated on

requirement bases. The larger the army of botnets the greater the magnitude of attack [8]. Various DDOS attacks are Botnets (Menti, Murlo, Neris, NSIS, Robot, Sogou, Strom, Virat, Zeus), Botnets (Menti, Murlo, Neris, NSIS, Robot, Sogou, Virat), DDOS (Executed Through LOIC), DDOS flood, SIDDONS and ICMP (flood, UDP flood).

### **2.1.2 Man in the Middle Attack:**

Man in the Middle attack interferes communication between two ends of network. Various steps involved in MITM are interception, interruption, modification, and fabrication [9]. The science of cryptography ensures confidentiality, and integrity of information being shared. MITM, however can jeopardize the encryption process, by getting hold of public key of anyone of the communication parties [10]. It is used for information gathering, sniffing and eaves dropping. Attacks in literature relating to MITM are privilege escalation (remote-to-local and user-to-root), probing DNS Attacks, XSS/SQL injection backdoors, infiltration, and crypto Ransome.

### **2.1.3 Passive Scanning:**

Passive scanning scans open ports and sees which ports are vulnerable for the attacks. Port scanning, IP address scanning, version scanning, eaves dropping, OS finger printing, traffic analysis and port scans (PingScan, SYN-Scan) are various ways in which passive scanning be done. The purpose is to steal information. Such attacks are hard to detect as they are done passively without knowledge of the machine being attacked [11]. Once information is gathered it is then used for required purpose.

### **2.1.4 Malware:**

Malware is a malicious software downloaded on computer system to steal, misuse, damage or destroy information. According to 2021 global threat report the eCrime index has increased from 129.96 percent to 328.36 percent from 2020 to 2021 [12]. The increase in this number is mainly due to malwares. Malwares were particularly used in this year by governments to get hold of vaccine research done by adversary government. Major types of malware attacks are backdoor, virus, worms, trojan horses attack, ransomware, heartbleed, rootkits, logicbombs, keyloggers, scans, etc.

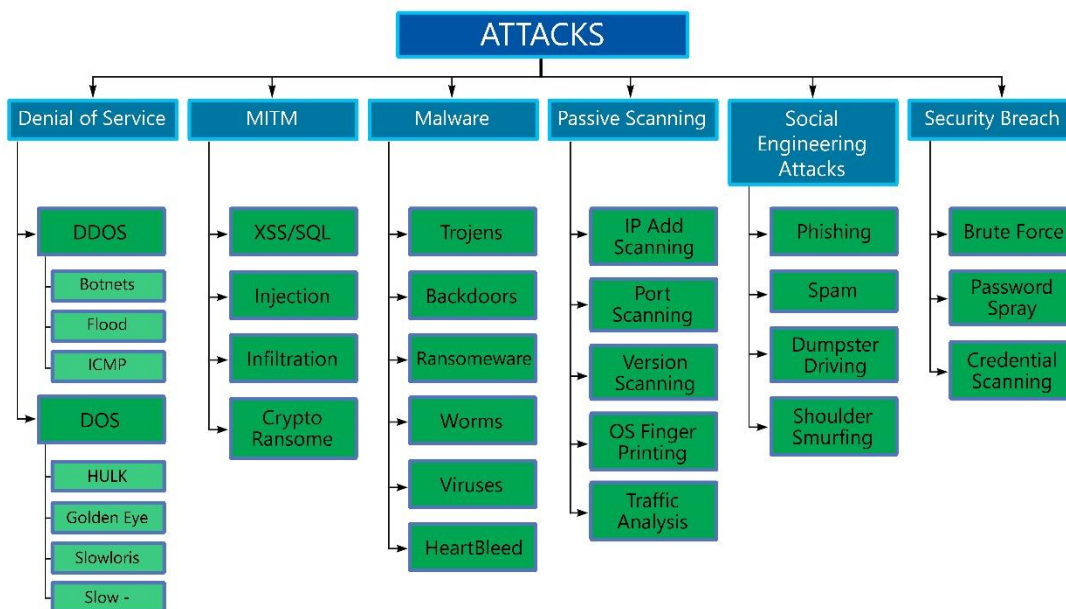
### **2.1.5 Social Engineering Attacks:**

A popular attack used by hacker to bypass authorization and authentication mechanism. They are business for hacker. 4 J.R, H.A, W.S Novice, uninformed and uneducated users

may accidentally download or click on links that might trap them into a dangerous territory. User might add personal information, banking credentials or other sensitive information without having an iota of doubt that they are being attacked. Some common types of social engineering attacks are phishing emails, spam and spear phishing. [13].

### 2.1.6 Security Breach:

Network systems requires physical and non-physical security systems. Physical security can be ensured by keeping servers under strict security surveillance. However non-physical security is harder to maintain. Credential surfing, brute force, password spray etc can breach security of a system and make the system vulnerable for other attacks [14]. Fig 2 shows a list of attacks



Network Attacks

Figure 2

## Chapter 3

### 3.1 Literature Survey

Network traffic has enormous amount of data being transmitted in limited time. To monitor and analyze this traffic for threats and vulnerabilities machine learning provide efficient and effective solutions. They process enormous amount of data in limited amount of time. It helps to create patterns and classifies traffic anomaly or normal traffic. Various machine algorithms being used in literature are Bayesian network, artificial neural networks, clustering, association rules and fuzzy association rules, decision trees, evolutionary computation, ensemble learning, hidden Markov models, Naive Bayes, sequential pattern mining, inductive learning, and support vector machine [15]. Machine learning classification algorithms work on features. The choice of accurate features affects the results of classification. Various intrusion detection systems have used different number of features. Generally detection systems, have used similar number of features, for detection of multiple attacks.

Table 1 shows a summary of all research work done. Vilhelm uses the CICIDS2017 dataset and Zeek tool to covert PCAP files into log files [21]. Initially, the author uses two sets of features for the detection of 14 network attacks. A set of simple and complex features containing 10 and 50 features was designed respectively. Complex set of features was less efficient and time consuming. Therefore the author selected simpler set of 10 features, for improved speed and better accuracy. KNN, RF and DT were applied to attained an accuracy of 97 %.

Sarhan converted four widely known datasets into NetFlow [22]. NetFlow is an industry-standard protocol for network traffic collection [23]. Zeek extracted 49 and 44 features from UNSW-NB15 and ToN-IoT respectively. Argus extracted 42 features from BOT-IOT while 73 features were extracted from CSE-CICIDS2018 using CICFlowMeter-V3. 8 common features, were selected from all the datasets. Forward packet length, backward packet length, total bytes, avg forward length, avg backward length, duration, bytes , forward bytes and backward bytes were the selected features.

Random forest was used to perform binary and multiclass classification. The binary classifiers performed better on all four datasets. However, the multi-class classifier did not perform well on some attacks i.e. fuzzers, analysis, exploits.

The authors therefore decided to increase the number of features, for better performance. The authors in [22] was not satisfied multi-class classification results. As a result, they increased the number of features from 12 to 43. [24]. Increased number of features improved classification results. The authors claim, these 43 features may prove to be a “benchmark feature set” lacking for NIDS, previously. They believe these common feature sets may help in the practical



deployment of the intrusion detection systems which currently is scarce. However, no justification for the selection of these features is provided in the paper.

Jian applied a gradient descent boosting algorithm to detect DDoS attacks based on TCP and UDP protocol [25]. WIDE 2017 dataset was used having a total of 102 and 49 features for TCP and UDP respectively. The authors used a blend of Random Forest and Pearson correlation coefficient RFPW for the selection of features. RFPW is then compared to traditional feature selection algorithms, PCA principal component analysis, SVD (singular value decomposition) and LDA (linear discriminant analysis). PCA, SVD and LDA used 33,31, and 1 feature to attain an accuracy of 92.5 percent, 93.2 percent, and 88.5 percent respectively for detecting TCP based DDoS. Whereas, RFPW uses 10 features to attain an accuracy of 95 percent. 10 TCP features were: Syninpps, Pushinpps, Pushinpp, Pushoutpps, shakehdspps, Window average flowing into the packet, Window average of outgoing packets, Outbound packet TCP header square size, Outbound packet TCP header square size, The total size of the packets flowing in per second, Flow into port size and Outgoing packet interval maximum. For UDP based DDoS attacks PCA, SVD and LDA used 11,11, and 1 feature to attain an accuracy of 88.1 percent, 89.3 percent, and Title Suppressed Due to Excessive Length 7 75 percent respectively. While RFPW used 6 features to attain an accuracy of 93 percent. 6 UDP features used were: Outgoing UDP tagged TCP packet per second, Flowing into unmarked TCP packets per second, Handshake times, Outgoing urg tag TCP packet per second, Duration of a flow and Average packet size.

Sarhan assesses the performance of three datasets proposed in [26] with 83 CICFlowMeter generated features on the same dataset. The paper evaluates three datasets (CSE-CIC-IDS2018, CIC-BoT-IoT, and CIC-ToNIoT) with 83 features CICFlowMeter generated and their respective datasets in NF-CSE-CICIDS2018-v2, NF-BoT-IoT-v2, and NF-ToN-IoT-v2 with 43 features. For assessing the performance of selected features Deep Feed Forward and Random Forest were used. The dataset with 43 Netflow standard features performed better as compared to 83 CICFlowMeter generated features. The authors also claim that the gap between research and practical deployment of ML-based intrusion detection systems is a “black box” of ML based models. To fulfill this gap the author explains classification results achieved by the ML classifiers by assessing the Shapley values of each feature using the SHapley Additive exPlanations (SHAP) methodology.

This will aid in recognition of key features utilized in the model’s predictions. Mean Shapley value is the average of all test samples. A greater mean Shapley value indicates a stronger influence of feature in classification. The author believes provision of common dataset and explanation of features relevance will help in industrial deployment of such systems.

Hashem has used Bro for converting four datasets: information Security and Object Technology (ISOT) dataset, CTU-50, CTU-51, CTU-52, and CTU-53 from CTU datasets, Alfaisal University, Prince Sultan College Jeddah (PSCJ), into NetFlow [27]. Bro (currently known as

Zeek) is the world's leading platform for network security monitoring [18]. It creates logs files that can be transformed using custom scripts. The paper is a comparison between NetFlow and packetbased intrusion detection. Seven classification algorithms were applied to renewed dataset. A set of 8 features was selected for classification. Features selected were: Flow duration, Protocol, Destination Port, Source Port ,Number of Packets per flow, Number of bytes per flow, Number of Bytes per packet and Number of Bytes per second. Although only three attributes were used per classification algorithm for detection of respective malicious activity. During experimentation, no single false negative was reported on all classification algorithms. However, false-positive alerts were generated as compared to packet-based detection. Iman creates a new dataset of updated attacks [28].

CICFLowMeter is utilized to extract 80 network features. The dataset is labeled for all attacks. The attacks identified for the dataset are: DoS, GoldenEye, DoS Hulk, DoS Slowhttp, SSHPatator, FTP-Patator, DoS slow loris, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet. The dataset is generated in a lab environment for 5 days. After the generation of the dataset, RandomForestRegressor is used for feature selection. The selected features are tested on seven machine learning classification algorithms. The algorithms applied are KNN, RF, ID3, Adaboost, MLP, Naïve-Bayes, and QD. ID3 and random forest achieved highest accuracy, while Adaboost has minimum accuracy with maximum execution time.

Mahmoud assess machine learning approaches for detection of attacks in Software Defined Network and used CICDDoS2019 dataset. It's a novel dataset containing 12 DDOS attacks. Attacks included in dataset are UDP, SNMP, NetBIOS, LDAP, TFTP, NTP, SYN, WebDDoS, MSSQL, UDP-Lag, DNS, and SSDP. The dataset contains a total of 80 features, however socket features like IP address of source and destination are removed. The attacker and normal user might have same IP addresses and it may create over-fitting problem for deep learning model. The network features are different for all networks, therefore only packet features are considered, reducing the number from 80 to 77. Deep learning model based on RNN auto encoder is applied to CCIDDoS2019 dataset to classify traffic into normal and benign. The model attained an accuracy of 99 percent. The author believes attack scenarios change, therefore associating specific features to attacks is not necessary [29].

Aditya[30] uses network simulation environment to create simulated dataset. 6000 data samples are collected using floodlight controller. DDoS attacks is performed in a simulated environment using Hping 3 tool. Classification models are constructed used KNN, SVM and Naïve Bayes. A total of 6 features were used for the model: number of Packets, Protocol, Delay, Bandwidth, Source IP and Destination IP. Naïve Bayes gives an accuracy of 83 percent, KNN gives an accuracy of 97 percent and SVM gives an accuracy of 82 percent. . Dong li also uses SVM, to identify DDoS attacks [31].

Simulated traffic is generated using Openflow protocol. Before testing the simulated traffic SVM is trained on DARPA 1999. For training SVM entropy distribution of 5 features is calculated. This entropy distribution is then used in SVM model training. The 5 key features identified are such as source IP address, source port, destination IP address and destination port. Simulated traffic is then tested on trained SVM to attain an accuracy of 97 percent. [32] The author has used N-BaIoT benchmark dataset having 115 features. Bashlite and Mirai attacks are used in the dataset which are further divided into 5 classes each. 115 features are difficult to manage, they hamper efficiency of the classification algorithm. In order to reduce number of features, reduction algorithm are used i.e. PCA, MI and ANOCA f-test. Amongst all three, MI, fine granulated mutual information and aggregated mutual information provides best result. The reduced resultant features are produced using minimum, maximum and average aggregation function. The number of features selected by three functions were 15, respectively.

Feature selection algorithms were tested on four different classifiers XGB, GNB, k-NN, LR and SVM. MI(MIN and Average) gave best results on, XGB and K-NN classifiers achieving 99.19 and 98.28 accuracy respectively, while k-NN performed better with MAX aggregation function[33]. There are many meta heuristic algorithms in literature for example Grasshopper optimization system (GOA), seagull optimization algorithm (SOA), harmony search (HS) etc. However, most of them suffer from a balance between exploration and exploitation, hence trapping them in local minima or global maxima. To overcome this problem author suggested using a combination of Gorilla Troop Optimize GTO and Bird swarp algorithm BSA.

A four step strategy is used to enhance searching abilities of the algorithm i.e. control randomization parameter, advance nonlinear transfer function, various phases transition of GTO exploration phase and a up-to-date local revising positing strategy, dependent on BSA algorithm is used. The proposed model was tested using four datasets: NSL-KDD, CICIDS-2017, UNSWNB-15, and Bot-IoT Dataset. All datasets are compared on meta heuristic algorithms, GTO, BSA and Harrold-Gupta-Soffa HGS , Multiverse optimization MVO, Harris Hawks Optimization (HHO and Particle Swarm Optimization PSO algorithm. The ratio of total features and reduced mean features as a result of GTOBSA is - NSL-KDD 41: 14.75, CICIDS-2017 78: 10, UNSWNB-15 49:16.6 , and Bot-IoT Dataset 43: 2.533. The results show that proposed algorithms gave best results on NSL-KDD, CICID2017, UNSW-NB, WITH 95.5, 98.7, 81.5 and second best on 81.5 on BoT-IoT dataset. [34] is extensive review of various intrusion detection systems. Wrapper feature selection method is common across all selected NIDS. Wrapper feature technique uses optimization and classification technique simultaneously, to attain the desired results A taxonomy is created by the author classifying wrapper techniques, design, structure, and application. NIDS are reviewed based on this Taxonomy. The selection techniques are classified as machine learning (ML), statistical (St) or meta-heuristic (MH). These techniques are further identified as bio inspired and non bio inspired. Wrapper class is labeled as bACP and mcACP, binary attack class detection and

multi class attack class detection respectively. The architecture is categorized as hierarchical, general and specialized.

Wrapper class is labeled as bACP and mcACP, binary attack class detection and multi class attack class detection respectively. An extensive survey is done on various NIDS, using above mentioned taxonomy. NIDS are scrutinized over TPR, Acc, FPR, FNR, Prec Spec and F1, and author then identifies few best approaches. 6 best approaches are selected for bACP, three of them used UNSW-NB15 dataset and the rest three used NSL-KDD, KDD99 and DARPA respectively. Similarly, five approaches are selected for mcACP. Two of them used NSL-KDD, rest used UNSW-NB15, KDD99 and DARPA respectively. The author believes this study may help researchers working in feature selection domain for NIDS.

<b>Research Work</b>	<b>DataSet</b>	<b>Attacks</b>	<b>Features</b>	<b>Algorithm</b>
[21]	CICIDS2017 dataset	DoS, Heartbleed, Infiltration, Botnet, Web, Bruteforce, DDoS, FTP, Port, Scan, SQL, Injection, SSH, XSS	10 Features	KNN, RF, DT
[22]	NF-UNSWNB15, NF-BoNIoT, NF-TOTIOT, NF-CSE-CICIDS2018	Fuzzers, Analysis, Backdoor, DoS, Exploits, Reconnaissance, Shellcode, Worms, DDoS, Theft	8 Features	Extra Tree Ensemble
[24]	NF-UNSW-NB15- v2, NF-BoN-IoT-v2, NF-TOT-IOT-v2, NF-CSE-CICIDS2018-v2, NF-UQNIDS-v2	Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, DDoS, Theft, MITM, Password, Ransomware, Scanning, XSS, BurteForce, Web Attacks	43 Features	Extra Tree Ensemble
[25]	WIDE 2017	DDOS attacks Random Forest	TCP 10 features ,06 features for UDP	Pearson Correlation coefficient

[26]	[CICDDoS2019]	DDOS	CISCO Flowmeter features	Deep Feed-Forward Random Forest
[27]	(ISOT) dataset, CTU-50, CTU-51, CTU-52, CTU-53 from CTU datasets	[Spam-bot, IRC-Bot, P2P-Bot]	8 Features	J48, Random Rep BF Tree, PART, JRIB, DTNB
[28]	CICIDS2017	DoS Goldeneye, HEARTBLEED, DoS Hulk, DoS slowlorris, SSH-patotor, FTPPatotor, Web attack, Infiltration, Bot, PortScan, DDoS	30 features	KNN, RF, ID3, Adaboost, MLP, NaiveBaye, QD, ID
[29]	CCIDDoS2019	DDoS	77 Features	RNN auto encoder
[30]	Simulation Based	DDoS	6 features	SVM, Naive Bayes, KNN
[31]	DDoS 5 Features SVM	DDoS	5 Features	SVM
[32]	N-BaIoT	Mirai and Bashlite	15 Features	XGB, GNB, k-NN, LR and SVM
[33]	NSL-KDD, CICIDS2017, UNSWNB-15, and Bot-IoT Dataset	Mutiple attacks	NSL-KDD 41: 14.75, CICIDS-2017 78: 10, UNSWNB-15 49:16.6	Gorilla Troop Optimize GTO and Bird swarp algorithm BSA

			and Bot-IoT Dataset	
--	--	--	---------------------	--

## Network Intrusion Detection Systems

*Table 1*

### 3.2 Shortcoming of Existing Literature Network

Network attack occur at different layers of the network. For identification of each attack unique network features are required. For example, for DDoS “destination IP” and “timtout” are required, “active minutes” and “total length of forward packets” is not important. Similarly, for PortScan “initial forward bytes” and “backward packet” are important, but “flow duration” and “total length of forward packets” are unnecessary. Title Suppressed Due to Excessive Length 11 Our survey shows that similar features are being used for multiple attacks by most NIDS. The results of classification, therefore, can either be biased or uncertain.

Also, little or no justification is provided by many of the authors for selection of feature space. As [22] has selected 8 features, for 11 network attacks and also [24] has selected 43 features, for all network attacks. Whereas, network attacks have different nature, therefore applying equal and similar features for all attacks may not be a good idea. Also, the attack dimensions are evolving rapidly, identification of common features set for multiple attacks is necessary.

A common feature set with proper justification, associated to individual attacks, will enhance deployment of machine learning based NIDs. The current research uses same mirror to for all attacks. Different attacks are targeted at different layers therefore similar features cannot be used for all attacks. Researchers need to identify unique features correlated to each attack for better performance of machine learning algorithms. Domain knowledge must be included for the recognition of features. A unique set of features correlated to each attack will help in the practical and physical deployment of NIDs. It will increase the confidence of users on the system. Network attacks are a serious threat to computer systems. They cause monetary, emotional, and national damage. The selection of correct features provides effective and efficient classification results. The study is a step towards finding and benchmarking features relevant to various attacks. Experimentation may allow researchers to make correct profiling of features relevant to corresponding attacks.

## Chapter 4

### 4.1 Methodology and Framework

#### 4.1.1 Data Collection

For experimentation, we have used dataset used by [35]. There are five datasets that are used for our experiments NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-BoT-IoT-v2 and NF-CSE-CIC-IDS2018-v2.

##### 4.1.1.1 NF-UNSW-NB15-v2

The NetFlow-based format of the UNSW-NB15 dataset, named NF-UNSW-NB15, has been expanded with additional NetFlow features and labelled with its respective attack categories. The total number of data flows is 2,390,275 out of which 95,053 (3.98%) are attack samples and 2,295,222 (96.02%) are benign. The attack samples are further classified into nine subcategories, Table 2 below represents the NF-UNSW-NB15-v2 dataset's distribution of all flows.

Class	Count	Description
Benign	2295222	Normal unmalicious flows
Analysis	2299	A group that presents a variety of threats that target web applications through ports, emails, and scripts.
Backdoor	2169	A technique that aims to bypass security mechanisms by replying to specific constructed client applications.
DoS	5794	Denial of Service is an attempt to overload a computer system's resources with the aim of preventing access to or availability of its data.
Exploits	31551	Are sequences of commands controlling the behavior of a host through a known vulnerability
Generic	16560	A method that targets cryptography and causes a collision with each block-cipher.
Reconnaissance	12779	A technique for gathering information about a network host and is also known as a probe.
Shellcode	1427	A malware that penetrates a code to control a victim's host.
Worms	164	Attacks that replicate themselves and spread to other computers.
Fuzzers	22310	An attack in which the attacker sends large amounts of random data which cause a system to crash and also aim to discover security vulnerabilities in a system.

**NF-UNSW-NB15-v2**

*Table 2*

#### 4.1.1.2 NF-ToN-IoT-v2

The publicly available pcaps of the ToN-IoT dataset are utilized to generate its NetFlow records, leading to a NetFlow-based IoT network dataset called NF-ToN-IoT. The total number of data flows is 16,940,496 out of which 10,841,027 (63.99%) are attack samples and 6,099,469 (36.01%), Table 3 below lists and defines the distribution of the dataset.

Class	Count	Description
Benign	6099469	Normal unmalicious flows
Backdoor	16809	A technique that aims to attack remote-access computers by replying to specific constructed client applications.
DoS	712609	An attempt to overload a computer system's resources with the aim of preventing access to or availability of its data.
DDoS	2026234	An attempt similar to DoS but has multiple different distributed sources.
Injection	684465	A variety of attacks that supply untrusted inputs that aim to alter the course of execution, with SQL and Code injections two of the main ones.
MITM	7723	Man In The Middle is a method that places an attacker between a victim and host with which the victim is trying to communicate, with the aim of intercepting traffic
Password	1153323	covers a variety of attacks aimed at retrieving passwords by either brute force or sniffing.
Ransomware	3425	An attack that encrypts the files stored on a host and asks for compensation in exchange for the decryption technique/key.
Scanning	3781419	A group that consists of a variety of techniques that aim to discover information about networks and hosts, and is also known as probing.
XSS	2455020	Cross-site Scripting is a type of injection in which an attacker uses web applications to send malicious scripts to end-users.

#### NF-ToN-IoT-v2 Dataset

Table 3

#### 4.1.1.3 NF-BoT-IoT-v2

An IoT NetFlow-based dataset was generated by expanding the NF-BoT-IoT dataset. The



features were extracted from the publicly available pcap files and the flows were labelled

## NF-BoT-IoT-v2

Table 4

with their respective attack categories. The total number of data flows is 37,763,497 out of which 37,628,460 (99.64%) are attack samples and 135,037 (0.36%) are benign. There are four attack categories in the dataset, Table 4 represents the NF-BoT-IoT-v2 distribution of all flows.

### 4.1.1.4 NF-CSE-CIC-IDS2018-v2

Class	Count	Description
Benign	135037	Normal unmalicious flows
Reconnaissance	2620999	A technique for gathering information about a network host and is also known as a probe.
DDoS	18331847	Distributed Denial of Service is an attempt similar to DoS but has multiple different distributed sources.
DoS	16673183	An attempt to overload a computer system's resources with the aim of preventing access to or availability of its data.
Theft	2431	A group of attacks that aims to obtain sensitive data such as data theft and keylogging

The original pcap files of the CSE-CIC-

IDS2018 dataset are utilised to generate a NetFlow-based dataset called NF-CSE-CIC-IDS2018-v2. The total number of flows is 18,893,708 out of which 2,258,141 (11.95%) are attack samples and 16,635,567 (88.05%) are benign ones, the Table 5 represents the dataset's distribution.

Class	Count	Description
Benign	16635567	Normal unmalicious flows
BruteForce	120912	A technique that aims to obtain usernames and password credentials by accessing a list of predefined possibilities
Bot	143097	An attack that enables an attacker to remotely control several hijacked computers to perform malicious activities.
DoS	483999	An attempt to overload a computer system's resources with the aim of preventing access to or availability of its data.
DDoS	1390270	An attempt similar to DoS but has multiple different distributed sources.

Infiltration	116361	An inside attack that sends a malicious file via an email to exploit an application and is followed by a backdoor that scans the network for other vulnerabilities
Web Attacks	3502	A group that includes SQL injections, command injections and unrestricted file uploads

## NF-CSE-CIC-IDS2018-v2

*Table 5*

### 4.1.2 Features

Network Intrusion systems work on netflows instead of PCAP. Large size of PCAP files, makes processing ineffective and slow. PCAP files can be converted to flow features using various soft-wares Suricata [16], Snort [17], ZEEK [18], nProbe [19], CISCO flowmeter [20] etc. Flow is a distinct stream of packets that has an own ID which consists of the features source and destination IP, source and destination ports and protocol. Flow features are identified in Fig 3.

The unique set of 43 features identified in [22] are used in our experimentation. Features are mentioned in table 6.

PKTS	PKTS	SRC&DST	PROTOCOLS
NUM_URGENT	FIN_IN_PPS	SCR_TO_DST_AVG_THROUGHPUT	FTP_COMMAND_RET_CODE
NUM_TOS	FIN_OUT_PPS	DST_TO_SCR_AVG_THROUGHPUT	TCPCAPLEN_IN_MEAN
NUM_PKTS_UP_TO_128_BYTES	OTHER_IN_PPS	L4_SCR_PORT	TCPCAPLEN_OUT_MEAN
NUM_PKTS_128_TO_256_BYTES	SHAKEHDS_PPS	L4_DST_PORT	TCPCAPLEN_IN_MAX
NUM_PKTS_256_TO_512_BYTES	CRW_IN_PPS	PORT_IN_SIZE	TCPCAPLEN_OUT_MAX
NUM_PKTS_512_TO_1024_BYTES	CRW_OUT_PPS	PORT_OUT_SIZE	TCPCAPLEN_IN_VAR
NUM_PKTS_1024_TO_1514_BYTE	CRW_IN_PPS/IN_PPS	SIZE OF PKT	TCPCAPLEN_OUT_VAR
ECN_IN_PPS	CRW_IN_PPS/ACK_IN_PPS+CRW_IN_PPS	SIZE_SEQ_MEAN	TCP_FLAGS
ECN_OUT_PPS	SHORTEST_FLOW_PKT	SIZE_SEQ_MAX	CLIENT_TCP_FLAGS
ECN_IN_PPS/IN_PPS	LONGEST_FLOW_PKT	SIZE_SEQ_MIN	SERVER_TCP_FLAGS
ECN_IN_PPS/(ECN_IN_PPS+ACK_IN_PPS)	PUSH_IN_PPS	SIZE_SEQ_VAR	TCP_WIN-MAX_IN
ECN_IN_PPS/(URG_IN_PPS+ACK_IN_PPS)	PUSH_OUT_PPS	SIZE_IN_PPS	TCP_WIN-MAX_OUT
IN_PPS	TIME	SIZE_OUT_PPS	ICMP_TYPE
OUT_PPS	TTL	SIZE_IN_MAX	ICMP_IPV4_TYPE
IN_PPS/(OUT_PPS+IN_PPS)	MIN_TTL	SIZE_OUT_MAX	PROTOCOL
RST_IN_PPS	MAX_TTL	SIZE_IN_MIN	L7_PROTO
RST_OUT_PPS	DNS_TTL_ANSWER	SIZE_OUT_MIN	IPV4_SCR_ADDR
MIN_IP_PKT_LEN	FLOW_DURATION_MILLISECONDS	SIZE_IN_MEAN	IPV4_DST_ADDR
MAX_IP_PKT_LEN	DURATION	SIZE_OUT_MEAN	BYTES
ACK_IN_PPS	INTERVAL_IN/OUT_MAX	SIZE_IN_VAR	RETRANSMITTED_IN_BYTES
ACK_OUT_PPS	INTERVAL_IN/OUT_MIN	SIZE_OUT_VAR	RETRANSMITTED_OUT_BYTES
URG_IN_PPS	INTERVAL_IN/OUT_MEAN	SIZE_IN_MEDIAN	IN_BYTES
URG_OUT_PPS	INTERVAL_IN/OUT_VAR	SIZE_OUT_MEDIAN	OUT_BYTES
RETRANSMITTED_IN_PKTS	INTERVAL_OUT_14	SIZE_IN_14	SCR_TO_DST_SECOND_BYTES
RETRANSMITTED_OUT_PKTS	INTERVAL_OUT_34	SIZE_OUT_14	SCR_TO_SRC_SECOND_BYTES
SYN_IN_PPS		SIZE_IN_34	
SYNACK_OUT_PPS		SIZE_OUT_34	
SYN_IN_PPS/IN_PPS		SIZE_IN_MEAN	
		SIZE_OUT_MEAN	
		WINSIZE_IN_MEAN	
		WINSIZE_OUT_MEAN	

### NetFlow Features

Figure 3

1. IPV4_SRC_ADDR	IPv4 source address	
2. IPV4_DST_ADDR	IPv4 destination address	
3. L4_SRC_PORT	IPv4 source port number	
4. L4_DST_PORT	IPv4 destination port number	
5. PROTOCOL	IP protocol identifier byte	
6. L7_PROTO	Layer 7 protocol (numeric)	
7. IN_BYTES	Incoming number of bytes	
8. OUT_BYTES	Outgoing number of bytes	
9. IN_PKTS	Incoming number of packets	
10. OUT_PKTS	Outgoing number of packets	
11. FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds	
12. TCP_FLAGS	Cumulative of all TCP flags	
13. CLIENT_TCP_FLAGS	Cumulative of all client TCP flags	
14. SERVER_TCP_FLAGS	Cumulative of all server TCP flags	
15. DURATION_IN	Client to Server stream duration (msec)	
16. DURATION_OUT	Client to Server stream duration (msec)	
17. MIN_TTL	Min flow TTL	
18. MAX_TTL	Max flow TTL	
19. LONGEST_FLOW_PKT	Longest packet (bytes) of the flow	

20. SHORTEST_FLOW_PKT	Shortest packet (bytes) of the flow	
21. MIN_IP_PKT_LEN	Len of the smallest flow IP packet observed	
22. MAX_IP_PKT_LEN	Len of the largest flow IP packet observed	
23. SRC_TO_DST_SECOND_BYTES	Src to dst Bytes/sec	
24. DST_TO_SRC_SECOND_BYTES	Dst to src Bytes/sec	
25. RETRANSMITTED_IN_BYTES	Number of retransmitted TCP flow bytes (src->dst)	
26. RETRANSMITTED_IN_PKTS	Number of retransmitted TCP flow packets (src->dst)	
27. RETRANSMITTED_OUT_BYTES	Number of retransmitted TCP flow bytes (dst->src)	
28. RETRANSMITTED_OUT_PKTS	Number of retransmitted TCP flow packets (dst->src)	
29. SRC_TO_DST_AVG_THROUGHPUT	Src to dst average thpt (bps)	
30. DST_TO_SRC_AVG_THROUGHPUT	Dst to src average thpt (bps)	
31. NUM_PKTS_UP_TO_128_BYTES	Packets whose IP size <= 128	
32. NUM_PKTS_128_TO_256_BYTES	Packets whose IP size > 128 and <= 256	
33. NUM_PKTS_256_TO_512_BYTES	Packets whose IP size > 256 and <= 512	
34. NUM_PKTS_512_TO_1024_BYTES	Packets whose IP size > 512 and <= 1024	
35. NUM_PKTS_1024_TO_1514_BYTES	Packets whose IP size > 1024 and <= 1514	
36. TCP_WIN_MAX_IN	Max TCP Window (src->dst)	
37. TCP_WIN_MAX_OUT	Max TCP Window (dst->src)	
38. ICMP_TYPE	ICMP Type * 256 + ICMP code	
39. ICMP_IPV4_TYPE	ICMP Type	
40. DNS_QUERY_ID	DNS query transaction Id	
41. DNS_QUERY_TYPE	DNS query type (e.g. 1=A, 2=NS..)	
42. DNS_TTL_ANSWER	TTL of the first A record (if any)	

### Features used for Experimentation.

Table 6

## 4.1.3 Data Cleaning:

### 4.1.3.1 Removal of Rows containing NaN values

Data cleaning is a crucial step in research data preparation, aiming to enhance data quality and reliability. The removal of rows containing NaN (Not a Number) values plays a fundamental role in addressing missing or undefined data in the DataFrame. NaN values often arise due to various reasons, such as measurement errors, data collection issues, or incomplete records, however getting rid of them is mandatory before starting any processing.

#### **4.1.3.2 Replacing NaN with the mean of each Column:**

It is necessary to replace missing data in a dataset by replacing NaN values with the mean of each column. Filling missing values with the mean is a common strategy for dealing with missing data, especially when the data is assumed to be approximately normally distributed. This approach helps to retain the overall data structure and avoid the loss of valuable information that might occur if rows with missing data were removed entirely.

#### **4.1.3.3 Scaling the data**

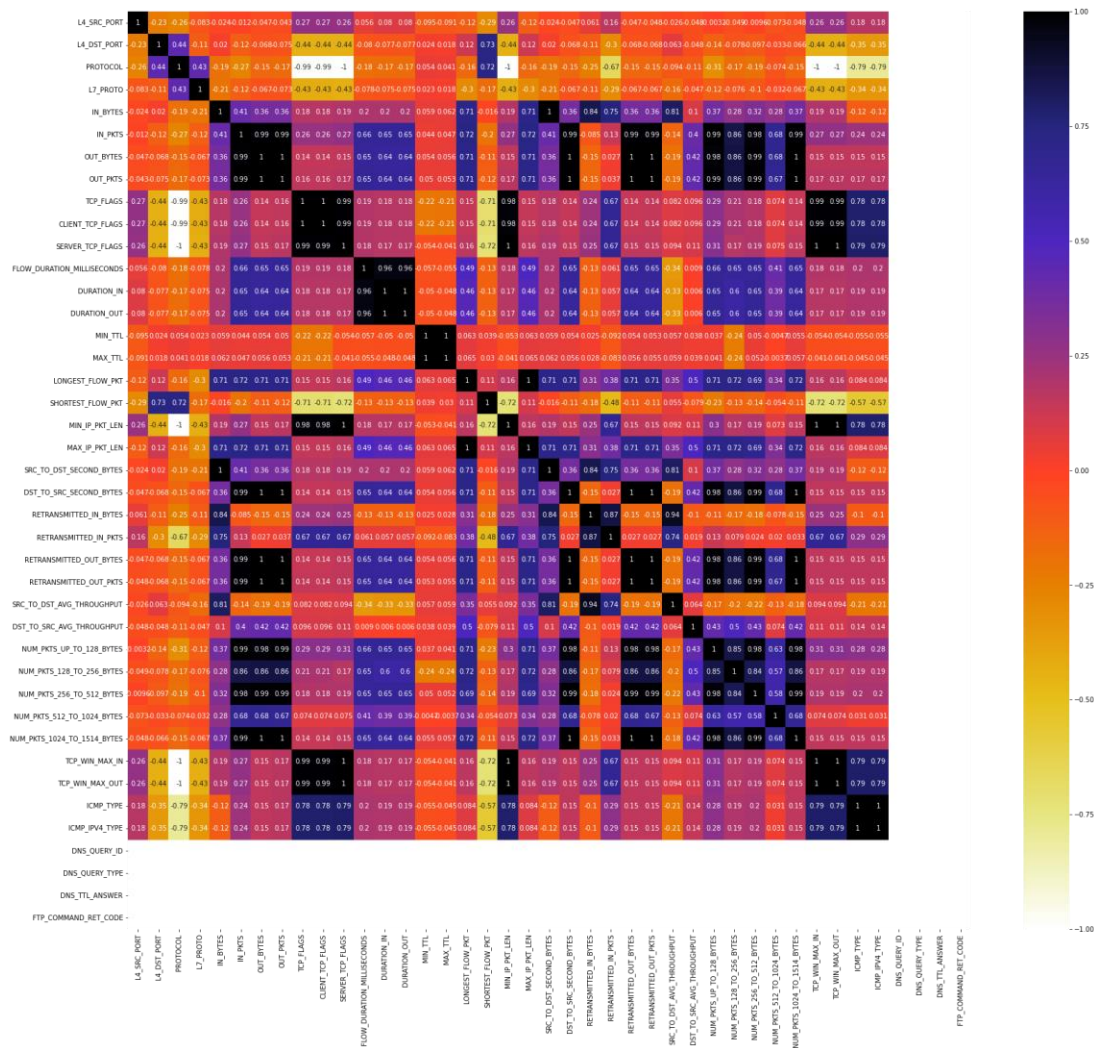
We have scaled the dataset using Min-Max scaling technique. The MinMaxScaler is used to scale features to a specified range, typically between 0 and 1. Then, we use the fit method to calculate the minimum and maximum values of the Normalize\_DATA dataset. This step computes the minimum and maximum values required for scaling the data using the Min-Max scaling method.

#### **4.1.4 Feature Selection**

##### **4.1.4.1 Pearson Correlation coefficient**

A few features were removed because they had no important in identification of attacks and could be not used in a machine learning algorithm due to nature of their type i.e. IPV4\_SRC\_ADDR, IPV4\_DST\_ADDR and L7\_PROTOCOL

We used highly correlated features in our dataset using the correlation function and a threshold value of 0.85. The features that were highly correlated were not selected as part of our feature set. For experimentation, each dataset was used and for each attack features that correlated were identified. We segregated the datasets based on network attacks. Figure 4 shows a heat map of correlated features, for NF-UNSW-NB15-v2.



Heatmap of correlated features ( NF-UNSW-NB15-v2)

Figure 4

#### 4.1.4.2 Importance of feature in Literature

In our literature survey we mentioned features previously selected were purely on experimentation basis, however we have also done some theoretical study about each feature and identified, which attacks could be possibly identified using the feature.

Table 7 represents theoretical explanation of feature and attacks that could possibly be identified used that feature (the attacks are supported by experiments done using Pearson correlation coefficient).

Feature	Attack
<p><b>DST_TO_SRC_AVG_THROUGHPUT</b></p> <p>The data that is transferred from destination to source is identified using this feature. A sudden change in size of data indicates an existence of an attack.</p>	<ul style="list-style-type: none"> <li>• DDoS</li> <li>• Injection</li> <li>• XSS</li> <li>• Scanning</li> <li>• Ransomware</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Shell code</li> <li>• DoS</li> </ul>
<p>The feature <b>DST_TO_SRC_SECOND_BYTES</b> represents the number of bytes sent from the destination to the source in the second time frame. This information can be used to identify patterns and anomalies in network traffic, which can be used to detect network attacks. However, it's just one feature among many others and should be considered in the context of the entire dataset to get an accurate understanding of network behavior.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• Ransomware</li> <li>• BOT</li> <li>• Shell code</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Theft</li> </ul>
<p>The feature '<b>ICMP_IPV4_TYPE</b>' is a numerical representation of the type of Internet Control Message Protocol (ICMP) used in an IP version 4 (IPv4) network. ICMP is a network-layer protocol used to send messages about network conditions, such as errors or congestion. The '<b>ICMP_IPV4_TYPE</b>' feature could be used to identify certain types of network attacks, such as denial-of-service (DoS) attacks, where the attacker floods the network with ICMP packets. Again, this feature should be considered in the context of the entire dataset to accurately identify network attacks.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• Ransomware</li> <li>• BOT</li> <li>• Shell code</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Password</li> <li>• DDoS</li> <li>• Analysis</li> <li>• Brute force</li> <li>• Backdoor</li> <li>• Generic</li> <li>• Exploits</li> <li>• Worms</li> </ul>

<p>The feature '<b>ICMP_TYPE</b>' likely refers to the type of Internet Control Message Protocol (ICMP) packet that is being transmitted. ICMP is used to transmit error messages, control messages, and test packets to other devices on a network. The type field in an ICMP packet specifies the type of message being transmitted. Having this feature in a network attack detection dataset can be useful for identifying specific types of attacks that utilize ICMP packets.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• DOS</li> <li>• Reconnaissance</li> <li>• DDoS</li> <li>• Analysis</li> <li>• Brute Force</li> <li>• BOT</li> </ul>
<p>The "<b>IN_PKTS</b>" feature represents the number of inbound packets. In the context of network attack detection, this feature could provide information about the volume of incoming traffic to a system or network. However, it would need to be considered in conjunction with other features to determine if an attack is taking place.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Injection</li> <li>• MITM</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Backdoor</li> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Theft</li> </ul>
<p><b>MAX_IP_PKT_LEN</b> is a feature that can be used in detecting network attacks. The change in the size of packet length may indicate anomaly.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Password</li> <li>• Scanning</li> <li>• XSS</li> <li>• Injection</li> <li>• MITM</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Theft</li> <li>• DoS</li> </ul>
<p><b>MAX_TTL</b> refers to the maximum value of the time-to-live (TTL) field in the IP header of a packet. The TTL value</p>	<ul style="list-style-type: none"> <li>• Password</li> <li>• Ransomware</li> <li>• XSS</li> </ul>



<p>represents the maximum number of hops a packet can take before it is discarded.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• MITM</li> <li>• DDoS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• DoS</li> <li>• Exploits</li> <li>• Generic</li> <li>• Shellcode</li> <li>• Reconnaissance</li> <li>• Worms</li> <li>• Scanning</li> </ul>
<p><b>NUM_PKTS_1024_TO_1514_BYTES</b> is a feature that refers to the number of packets that have a size between 1024 and 1514 bytes. This feature can be useful in network intrusion detection as it can provide information about the size of the packets in the network and help identify unusual patterns that might indicate a potential attack. However, it is just one of many features that can be used in the detection process, and its usefulness would depend on the specific dataset and the type of attack being detected.</p>	<ul style="list-style-type: none"> <li>• Scanning</li> <li>• Password</li> <li>• Ransomware</li> <li>• XSS</li> <li>• Injection</li> <li>• MITM</li> <li>• Brute force</li> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Worms</li> <li>• Reconnaissance</li> </ul>
<p>The feature <b>NUM_PKTS_UP_TO_128_BYTES</b> is a measure of the number of packets in a network flow that have a size of 128 bytes or less. This feature can be used in network security to detect certain types of attacks, where an attacker sends a large number of small packets to overwhelm a target system. By examining the distribution of packet sizes in a network flow, it may be possible to identify unusual patterns of traffic that are indicative of malicious activity.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Injection</li> <li>• DDOS</li> <li>• BoT</li> <li>• Brute Force</li> <li>• Generic</li> <li>• Exploits</li> <li>• Worms</li> <li>• Shellcode</li> <li>• Reconnaissance</li> </ul>

<p><b>OUT_PKTS</b> is a feature used in network intrusion detection systems to detect attacks. It refers to the number of packets sent from the destination to the source in a given time interval. This feature can be used to determine the volume of traffic generated by a source and could be useful in identifying malicious activities.</p>	<ul style="list-style-type: none"> <li>• XSS</li> <li>• Password</li> <li>• Ransomware</li> <li>• Injection</li> <li>• MITM</li> <li>• DDOS</li> <li>• BOT</li> <li>• Brute force</li> <li>• Analysis</li> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Shell Code</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Theft</li> <li>• Backdoor</li> </ul>
<p>The feature '<b>RETRANSMITTED_IN_PKTS</b>' can be used in detecting network attacks by analyzing the number of inbound packets that have been retransmitted. A high number of retransmitted packets could indicate that the network is experiencing congestion or some other issue, which could be a sign of a network attack. It's important to note that this feature should be used in combination with other features and analysis methods to accurately detect network attacks.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Injection</li> <li>• MITM</li> <li>• DDoS</li> <li>• BOT</li> <li>• Brute force</li> <li>• Analysis</li> <li>• DOS</li> <li>• Backdoor</li> <li>• Exploits</li> <li>• Shellcode</li> <li>• Generic</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Theft</li> </ul>
<p>The feature "<b>SERVER_TCP_FLAGS</b>" is a numerical representation of the flags in the TCP header of a packet in a network communication. The values of this feature can</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Scanning</li> <li>• XSS</li> </ul>

<p>indicate various aspects of the communication, such as whether a packet is a SYN (Synchronize) packet, an ACK (Acknowledgment) packet, etc. This feature can help in detecting network attacks by giving an understanding of the type of communication that is happening in the network.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• MITM</li> <li>• DDoS</li> <li>• Bot</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Back door</li> <li>• DOS</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Reconnaissance</li> </ul>
<p><b>DNS_QUERY_ID</b> is a field in the header section of a DNS (Domain Name System) packet. It is a 16-bit identifier assigned by the DNS client to identify the DNS query it is making. When a DNS server responds to a query, it includes this identifier in the response packet so that the client can match the response to the query.</p> <p>The <b>DNS_QUERY_ID</b> field is important because it allows for multiple concurrent DNS queries to be made by a single client or multiple clients without confusion. It is randomly generated by the client, and this helps to ensure that each query has a unique identifier</p>	<ul style="list-style-type: none"> <li>• Password</li> </ul>
<p>In network performance monitoring, "<b>DURATION_OUT</b>" could refer to the time it takes for a packet or message to leave a system and reach its destination. This metric can be used to track the performance of a network or application and identify potential bottlenecks or areas for optimization.</p>	<ul style="list-style-type: none"> <li>• Password</li> <li>• MITM</li> <li>• BOT</li> <li>• Brute force</li> <li>• Shell code</li> <li>• Worms</li> </ul>
<p><b>RETRANSMITTED_OUT_PKTS</b> refers to the number of packets that have been retransmitted by a network device or application. Specifically, this metric tracks the number of packets that have been sent from a device or application to a remote system or device, but for which no acknowledgment or response has been received within a certain time. In such</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Password</li> <li>• DDOS</li> <li>• Bot</li> <li>• Brute Force</li> <li>• Analysis</li> <li>• Backdoor</li> </ul>

<p>cases, the device or application will retransmit the packets in order to ensure reliable delivery.</p>	<ul style="list-style-type: none"> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• DOS</li> </ul>
<p><b>L4_DST_PORT</b>, also known as Layer 4 destination port, is a feature in network communications that identifies the specific port number to which a packet is destined. It is commonly used in transport layer protocols such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).</p>	<ul style="list-style-type: none"> <li>• Analysis</li> </ul>
<p>The <b>PROTOCOL</b> feature in networks refers to the specific communication protocols being used, such as TCP, UDP, ICMP, or others</p>	<ul style="list-style-type: none"> <li>• Theft</li> </ul>
<p><b>IN_BYTES</b>, also known as input bytes, is a feature in networks that represents the total number of bytes received by a network device or interface. This information can be helpful in detection of multiple attacks.</p>	<ul style="list-style-type: none"> <li>• DDoS</li> <li>• Brute Force</li> <li>• Analysis</li> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Reconnaissance</li> <li>• Theft</li> </ul>
<p>The "<b>CLIENT_TCP_FLAGS</b>" feature in networks refers to the TCP flags observed in the packets sent by the client during a TCP connection. <b>CLIENT_TCP_FLAGS</b>" feature to identify unusual or malicious flag combinations exhibited by the client during the TCP communication. Certain flag patterns, such as sending repeated SYN or FIN packets without following the expected TCP protocol behavior, can indicate potential attack attempts or abnormal behavior.</p>	<ul style="list-style-type: none"> <li>• MITM</li> <li>• DDoS</li> <li>• Bot</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Backdoor</li> <li>• DoS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Reconnaissance</li> </ul>

	<ul style="list-style-type: none"> <li>• Theft</li> </ul>
<p>The "<b>FLOW_DURATION_MILLISECONDS</b>" feature in networks represents the duration of a network flow, which is the period between the start and end of a communication session between two network entities. Rapid connections to different ports or IP addresses within a short time frame can indicate scanning attempts and help in the early detection of potential attacks</p>	<ul style="list-style-type: none"> <li>• Bot</li> <li>• Brute Force</li> <li>• Reconnaissance</li> <li>• Theft</li> </ul>
<p>The "<b>MIN_TTL</b>" feature in networks represents the minimum Time-to-Live (TTL) value observed in network packets. The TTL field in IP packets indicates the maximum number of network hops (routers) that a packet can traverse before being discarded.</p>	<ul style="list-style-type: none"> <li>• Brute Force</li> <li>• Analysis</li> <li>• Reconnaissance</li> <li>• DDoS</li> </ul>
<p>The "<b>LONGEST_FLOW_PKT</b>" feature in networks represents the size (in bytes) of the longest packet observed in a network flow, which is the sequence of packets exchanged between two network entities during a communication session. By examining the "LONGEST_FLOW_PKT" feature, an AI algorithm can identify flows with unusually large fragmented packets, which may indicate attempts to disrupt network services or exploit fragmentation-related vulnerabilities.</p>	<ul style="list-style-type: none"> <li>• DDoS</li> <li>• Brute Force</li> <li>• Reconnaissance</li> </ul>
<p>The "<b>SHORTEST_FLOW_PKT</b>" feature in networks represents the size (in bytes) of the shortest packet observed in a network flow, which is the sequence of packets exchanged between two network entities during a communication session. When used by an artificial intelligence (AI) algorithm for attack detection, the "SHORTEST_FLOW_PKT" feature can provide valuable insights and contribute to the detection of potential attacks.</p>	<ul style="list-style-type: none"> <li>• DDOS</li> <li>• Brute Force</li> <li>• DOS</li> </ul>
<p>The "<b>MIN_IP_PKT_LEN</b>" feature in networks represents the minimum size (in bytes) of IP packets observed in network traffic. By analyzing the "MIN_IP_PKT_LEN" feature, an AI algorithm can identify IP packets with</p>	<ul style="list-style-type: none"> <li>• Scanning</li> <li>• XSS</li> <li>• DDOS</li> <li>• Brute force</li> </ul>

<p>unusually small sizes, suggesting the presence of malformed or anomalous packets that may indicate attack attempts.</p>	<ul style="list-style-type: none"> <li>• Analysis</li> <li>• Shell code</li> <li>• Reconnaissance</li> <li>• Dos</li> <li>• Theft</li> </ul>
<p>The "<b>RETRANSMITTED_IN_BYTES</b>" feature in networks represents the number of bytes that have been retransmitted during network communication. Retransmission of bytes often occurs due to network congestion, packet loss, or other performance-related issues. By analyzing the "RETRANSMITTED_IN_BYTES" feature, an AI algorithm can identify flows or connections that experience a high number of retransmitted bytes.</p>	<ul style="list-style-type: none"> <li>• Brute Force</li> <li>• Analysis</li> <li>• DOS</li> <li>• Generic</li> <li>• Exploits</li> <li>• Theft</li> </ul>
<p>The "<b>SRC_TO_DST_AVG_THROUGHPUT</b>" feature in networks represents the average throughput (data transfer rate) from the source to the destination in a network flow. A sudden drop or spike in the "SRC_TO_DST_AVG_THROUGHPUT" feature may suggest a network-based attack or an attempt to exfiltrate data.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• Brute Force</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Theft</li> </ul>
<p>The "<b>DST_TO_SRC_AVG_THROUGHPUT</b>" feature in networks represents the average throughput (data transfer rate) from the destination to the source in a network flow. By comparing the average throughput from the destination to the source with the expected or established baseline, the algorithm can identify flows that exhibit significant disparities in throughput. Such asymmetrical traffic patterns may indicate suspicious or malicious activity.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Scanning</li> <li>• XSS</li> <li>• Injection</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Shellcode</li> <li>• DOS</li> </ul>
<p>The "<b>NUM_PKTS_128_TO_256_BYTES</b>" feature in networks represents the number of packets with a size between 128 and 256 bytes observed in network traffic. "NUM_PKTS_128_TO_256_BYTES" feature, an AI algorithm can identify flows or connections with a significant number of packets within the specified size range. This may</p>	<ul style="list-style-type: none"> <li>• Brute force</li> <li>• BOT</li> <li>• Analysis</li> <li>• Worms</li> <li>• Dos</li> <li>• Theft</li> </ul>

<p>indicate the presence of attack payloads that are being transmitted using small-sized packets.</p>	<ul style="list-style-type: none"> <li>• DDOS</li> </ul>
<p>The "<b>NUM_PKTS_256_TO_512_BYTES</b>" feature in networks represents the number of packets with a size between 256 and 512 bytes observed in network traffic. This can help in identifying network flows or connections that exhibit a significant number of packets with sizes between 256 and 512 bytes. Unusual patterns or deviations from normal traffic can indicate potential malicious activities.</p>	<ul style="list-style-type: none"> <li>• Ransomware</li> <li>• Scanning</li> <li>• XSS</li> <li>• Injection</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• BoT</li> <li>• DOS</li> <li>• Exploits</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Theft</li> </ul>
<p>The "<b>NUM_PKTS_512_TO_1024_BYTES</b>" feature in networks represents the number of packets with a size between 512 and 1024 bytes observed in network traffic. Unusual patterns or deviations from normal traffic can indicate potential malicious activities that utilize packet sizes in this range.</p>	<ul style="list-style-type: none"> <li>• Injection</li> <li>• Ransomware</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Generic</li> </ul>
<p>The "<b>NUM_PKTS_1024_TO_1514_BYTES</b>" feature in networks represents the number of packets with a size between 1024 and 1514 bytes observed in network traffic. . Large payloads are often associated with specific types of attacks, such as data exfiltration, file transfers, or the transmission of malicious payloads. By detecting flows with an abnormal number of packets in this size range, the AI algorithm can raise alerts for further investigation.</p>	<ul style="list-style-type: none"> <li>• Password</li> <li>• Ransomware</li> <li>• Scanning</li> <li>• XSS</li> <li>• Injection</li> <li>• MITM</li> <li>• Brute force</li> <li>• Backdoor</li> <li>• Dos</li> <li>• Generic</li> <li>• Exploits</li> <li>• Worms</li> <li>• Recco</li> </ul>

<p>TCP window size is an important parameter that determines the amount of data a sender can transmit before receiving an acknowledgment from the receiver. By analyzing the "<b>TCP_WIN_MAX_IN</b>" feature, an AI algorithm can monitor the maximum window size observed in incoming TCP connections. Unusual or unexpected window sizes may indicate malicious activities or anomalies in the network traffic.</p>	<ul style="list-style-type: none"> <li>• DOS</li> <li>• Theft</li> <li>• MITM</li> <li>• DDOS</li> <li>• Analysis</li> <li>• Dos</li> <li>• Generic</li> <li>• Exploits</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Reconnaissance</li> <li>• Dos</li> </ul>
<p>The "<b>TCP_WIN_MAX_OUT</b>" feature in networks represents the maximum TCP window size observed in outgoing network traffic. By analyzing the "<b>TCP_WIN_MAX_OUT</b>" feature, an AI algorithm can monitor the maximum window size observed in outgoing TCP connections. Unusual or unexpected window sizes may indicate malicious activities or anomalies in the network traffic</p>	<ul style="list-style-type: none"> <li>• MITM</li> <li>• DDOS</li> <li>• Brute force</li> <li>• Analysis</li> <li>• Backdoor</li> <li>• Dos</li> <li>• Generic</li> <li>• Exploits</li> <li>• Shellcode</li> <li>• Worms</li> <li>• Reconnaissance</li> </ul>
<p>The "<b>DNS_QUERY_IND</b>" feature in networks represents the indication of DNS (Domain Name System) queries observed in network traffic.</p>	
<p>The "<b>DNS_QUERY_TYPE</b>" feature in networks represents the type of DNS (Domain Name System) queries observed in network traffic.</p>	
<p>The "<b>DNS_TTL_ANSWER</b>" feature in networks represents the Time-to-Live (TTL) value of DNS (Domain Name System) answers observed in network traffic.</p>	

### Theoretical Explanation of Features

Table 7



#### 4.1.4.3 Finally selected features

After identifying features from the using pearson correlation coefficient and theory, we have selected new a unique set of features for our experimentation. The feature have reduced from 43 to 36. Although the feature set for each attack has reduced significantly. Table 8 shows finally selected features.

1. L4_SRC_PORT	IPv4 source port number	
2. L4_DST_PORT	IPv4 destination port number	
3. PROTOCOL	IP protocol identifier byte	
4. L7_PROTO	Layer 7 protocol (numeric)	
5. IN_BYTES	Incoming number of bytes	
6. OUT_BYTES	Outgoing number of bytes	
7. OUT_PKTS	Outgoing number of packets	
8. FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds	
9. TCP_FLAGS	Cumulative of all TCP flags	
10. CLIENT_TCP_FLAGS	Cumulative of all client TCP flags	
11. SERVER_TCP_FLAGS	Cumulative of all server TCP flags	
12. DURATION_IN	Client to Server stream duration (msec)	
13. DURATION_OUT	Client to Server stream duration (msec)	
14. MIN_TTL	Min flow TTL	
15. MAX_TTL	Max flow TTL	
16. LONGEST_FLOW_PKT	Longest packet (bytes) of the flow	
17. SHORTEST_FLOW_PKT	Shortest packet (bytes) of the flow	
18. MIN_IP_PKT_LEN	Len of the smallest flow IP packet observed	
19. MAX_IP_PKT_LEN	Len of the largest flow IP packet observed	
20. SRC_TO_DST_SECOND_BYTES	Src to dst Bytes/sec	
21. DST_TO_SRC_SECOND_BYTES	Dst to src Bytes/sec	
22. RETRANSMITTED_IN_BYTES	Number of retransmitted TCP flow bytes (src->dst)	
23. RETRANSMITTED_IN_PKTS	Number of retransmitted TCP flow packets (src->dst)	
24. RETRANSMITTED_OUT_BYTES	Number of retransmitted TCP flow bytes (dst->src)	
25. RETRANSMITTED_OUT_PKTS	Number of retransmitted TCP flow packets (dst->src)	
26. SRC_TO_DST_AVG_THROUGHPUT	Src to dst average thpt (bps)	
27. DST_TO_SRC_AVG_THROUGHPUT	Dst to src average thpt (bps)	
28. NUM_PKTS_UP_TO_128_BYTES	Packets whose IP size <= 128	
29. NUM_PKTS_128_TO_256_BYTES	Packets whose IP size > 128 and <= 256	
30. NUM_PKTS_256_TO_512_BYTES	Packets whose IP size > 256 and <= 512	

31. NUM_PKTS_512_TO_1024_BYTES	Packets whose IP size > 512 and <= 1024
32. NUM_PKTS_1024_TO_1514_BYTES	Packets whose IP size > 1024 and <= 1514
33. TCP_WIN_MAX_IN	Max TCP Window (src->dst)
34. TCP_WIN_MAX_OUT	Max TCP Window (dst->src)
35. ICMP_TYPE	ICMP Type * 256 + ICMP code
36. ICMP_IPV4_TYPE	ICMP Type

### Finally Selected Features

Table 8

#### 4.1.5 Application of decision trees on Network intrusion datasets

To test our newly identified features on previously identified datasets we have used decision, as used by author in [35], to keep the comparison same. Decision trees were applied on each dataset, using newly identified features to test the accuracy of the model. Binary and multiclass classification were done, using decision trees. For binary classification 36 set of unique features were used, however, for multiclass classification we used newly identified feature corresponding to each attack.

## Chapter 5

### 5.1 Results and Evaluation:

#### 5.1.1 Metrics

The metrics for evaluating the results of the models used in this study are Accuracy. Among all the data points, accuracy is the proportion of data points that were properly anticipated shown in 5.1.2.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (5.1.2)$$

Accuracy considers precision and recall where precision is characterized as the proportion of accurately identified positive samples. The recall is determined as the proportion of Positive samples that were properly identified as Positive among all Positive samples. In our case, precision accounts for each class where e.g., ratio of correctly predicted attacks class out of all the samples predicted is the precision of the class attack.

#### 5.1.2 Time

In our study, we recognized that time is a crucial metric to evaluate the goodness of the machine learning classification model. While accuracy and other traditional performance metrics provide valuable insights into the model's ability to classify data correctly, they may not provide a

comprehensive picture of its practicality in real-world scenarios. The consideration of time as a metric aims to assess the model's efficiency and responsiveness, which are essential factors in many applications, especially in the context of network security and real-time decision-making.

Incorporating time as a metric in the evaluation allows for a more well-rounded analysis of the classification model's goodness. By understanding its computational efficiency and responsiveness, we gain valuable insights into its practicality and potential utility in real-world applications. Consequently, in my thesis, I emphasized the significance of considering time as a key performance metric, alongside traditional accuracy and other evaluation metrics, to ensure a comprehensive evaluation of the machine learning classification model's overall goodness and practical applicability.

### **5.1.3 Space complexities**

By considering space complexity as a metric, I aimed to strike a balance between model performance and resource utilization. A model with low space complexity is preferable, as it not only reduces memory overhead but also allows for faster model loading and inference times, leading to more efficient and responsive applications.

Throughout the evaluation process, I assessed the trade-off between model performance and space complexity. I compared various model architectures and techniques to identify the most efficient and compact solution that met the desired performance criteria. By incorporating space complexity as a metric in my thesis, I aimed to ensure that the classification model's goodness is not solely evaluated based on its predictive accuracy but also on its ability to utilize computational resources efficiently, making it practical and viable for real-world deployment in diverse computing environments.

## **5.2 Analysis of Experimental Results**

Table 9 identifies dataset, corresponding feature and accuracy of decision trees. Our experimentation shows that as identified by [35] using a unique set of features for each attack overfits the classification model and hence provides an accuracy of 1 or more than 99.78 percent. However, using different set of features for each attack gives a more realistic accuracy and hence can be used in a real-world system with live network attacks.

The execution time and space complexity of classification have reduce significantly since the number of features have reduced in many cases. Table 10 identifies the difference in processing speed, space complexity and number of features reduced. Data for only one dataset NF-UNSW-NB15-v2.csv, has been displayed in the table. Experiment should an ~50% increase in execution time, with reduced dataset. Hence it improves the efficiency of the model and gives better results, with less processing time.



Attack	Features	Accuracy
<b>NF-ToN-IoT-v2.csv</b>		
Password	All features	99.97
Password	{'DURATION_OUT','ICMP_IPV4_TYPE','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','OUT_PACKETS','RETRANSMITTED_OUT_PACKETS'}}	98.92
Ransomware	All features	1
Ransomware	{'DST_TO_SRC_AVG_THROUGHPUT','DST_TO_SRC_SECOND_BYTES','ICMP_IPV4_TYPE','IN_PACKETS','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_256_TO_512_BYTES','NUM_PKTS_512_TO_1024_BYTES','NUM_PACKETS_UP_TO_128_BYTES','OUT_PACKETS','RETRANSMITTED_IN_PACKETS','RETRANSMITTED_OUT_PACKETS','SERVER_TCP_FLAGS','SRC_TO_DST_SECOND_BYTES'}}	98.76
Scanning	All features	
Scanning	{'DST_TO_SRC_AVG_THROUGHPUT','ICMP_IPV4_TYPE','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_256_TO_512_BYTES','SERVER_TCP_FLAGS'}}	91.20
XSS	{'DST_TO_SRC_AVG_THROUGHPUT','ICMP_IPV4_TYPE','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_256_TO_512_BYTES','SERVER_TCP_FLAGS'}}	97.78
XSS	All features	99.89
Injection	{'DST_TO_SRC_AVG_THROUGHPUT','DST_TO_SRC_SECOND_BYTES','ICMP_IPV4_TYPE','ICMP_TYPE','IN_PACKETS','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','NUM_PACKETS_256_TO_512_BYTES','NUM_PACKETS_512_TO_1024_BYTES','NUM_PACKETS_UP_TO_128_BYTES','OUT_PACKETS','RETRANSMITTED_IN_PACKETS','SERVER_TCP_FLAGS','SRC_TO_DST_AVG_THROUGHPUT'}}	90.68

Injection	All features	99.73
MITM	{'CLIENT_TCP_FLAGS','DURATION_OUT','ICMP_IP_V4_TYPE','IN_PKTS','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','OUT_PKTS','RETRANSMITTED_IN_PKTS','SERVER_TCP_FLAGS','TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	95.98
MITM	All features	99.98
<b>CSE-CIC-IDS2018-v2</b>		
HOIC DOSS attack	All features	1
	{'DST_TO_SRC_AVG_THROUGHPUT','DURATION_IN','ICMP_IPV4_TYPE','ICMP_TYPE','NUM_PKTS_256_TO_512_BYTES','OUT_PKTS','RETRANSMITTED_IN_PKTS','SERVER_TCP_FLAGS','TCP_FLAGS'}	99.99
LOIC Doss Attack	All features	1
	All features	1
BOT	{'CLIENT_TCP_FLAGS','IN_BYTES','IN_PKTS','LONGEST_FLOW_PKT','MAX_IP_PKT_LEN','MAX_TTL','IN_IP_PKT_LEN','NUM_PKTS_512_TO_1024_BYTES','NUM_PKTS_UP_TO_128_BYTES','OUT_PKTS','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','SHORTEST_FLOW_PKT','TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	94.97
	All features	1
Brute force	{'CLIENT_TCP_FLAGS','DST_TO_SRC_SECOND_BYTES','DURATION_IN','DURATION_OUT','FLOW_DURATION_MILLISECONDS','ICMP_IPV4_TYPE','ICMP_TYPE','IN_PKTS','MAX_IP_PKT_LEN','NUM_PKTS_128_TO_256_BYTES','NUM_PKTS_256_TO_512_BYTES','NUM_PKTS_UP_TO_128_	97.99

	BYTES','OUT_PKTS','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS'}	
	All features	98.92
<b>NF-UNSW-NB15-v2.csv</b>		
Analysis	{'CLIENT_TCP_FLAGS','DST_TO_SRC_AVG_THROUGHPUT','ICMP_IPV4_TYPE','ICMP_TYPE','IN_PKTS','L4_DST_PORT','L7_PROTO','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','MIN_TTL','NUM_PKTS_128_TO_256_BYTES','NUM_PKTS_256_TO_512_BYTES','OUT_PKTS','RETRANSMITTED_IN_BYTES','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	98.83
	All features	99.71
Backdoor	All	99.89
	{'CLIENT_TCP_FLAGS','ICMP_IPV4_TYPE','IN_PACKETS','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_256_TO_512_BYTES','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	97.86
DoS	All features	99.76
	'CLIENT_TCP_FLAGS','ICMP_IPV4_TYPE','IN_PACKETS','MAX_IP_PKT_LEN','MAX_TTL','NUM_PACKETS_1024_TO_1514_BYTES','NUM_PACKETS_256_TO_512_BYTES','OUT_PACKETS','RETRANSMITTED_IN_BYTES','RETRANSMITTED_IN_PACKETS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PACKETS','SERVER_TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	98.26

Generic	All features	1
	{'CLIENT_TCP_FLAGS','ICMP_IPV4_TYPE','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_512_TO_1024_BYTES','NUM_PKTS_UP_TO_128_BYTES','OUT_PKTS','RETRANSMITTED_IN_BYTES','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	99.67
Exploits	All features	99.82
	{'CLIENT_TCP_FLAGS','ICMP_IPV4_TYPE','IN_PKTS','MAX_IP_PKT_LEN','MAX_TTL','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_256_TO_512_BYTES','NUM_PKTS_UP_TO_128_BYTES','OUT_PKTS','RETRANSMITTED_IN_BYTES','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	98.80
Shell Code	All features	99.96
	{'CLIENT_TCP_FLAGS','DST_TO_SRC_AVG_THROUGHPUT','DST_TO_SRC_SECOND_BYTES','DURATION_IN','DURATION_OUT','ICMP_IPV4_TYPE','IN_PKTS','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','NUM_PKTS_UP_TO_128_BYTES','OUT_BYTES','OUT_PKTS','RETRANSMITTED_IN_PKTS','SERVER_TCP_FLAGS','SRC_TO_DST_AVG_THROUGHPUT','TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	96.91
worms	{'CLIENT_TCP_FLAGS','DST_TO_SRC_SECOND_BYTES','DURATION_IN','DURATION_OUT','ICMP_IPV4_TYPE','MAX_IP_PKT_LEN','MAX_TTL','MIN_IP_PKT_LEN','NUM_PKTS_1024_TO_1514_BYTES','NUM_PKTS_128_TO_256_BYTES','NUM_PKTS_256_TO_512_BYTES','NUM_PKTS_UP_TO_128_BYTES','OUT_BYTES','OUT_PKTS','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','SRC_TO_DST_AVG_THROUGHPUT','SRC_TO_DST_SECOND_BYTES','TCP_FLAGS','TCP_WIN_MAX_IN','TCP_WIN_MAX_OUT'}	99.99
	All feature	99.99



Reconnaissance	'ICMP_IPV4_TYPE', 'ICMP_TYPE', 'IN_PKTS', 'MAX_IP_PKT_LEN', 'MAX_TTL', 'MIN_IP_PKT_LEN', 'NUM_PKTS_1024_TO_1514_BYTES', 'NUM_PKTS_256_TO_512_BYTES', 'NUM_PKTS_UP_TO_128_BYTES', 'OUT_PKTS', 'RETRANSMITTED_OUT_PKTS', 'SERVER_TCP_FLAGS', 'TCP_WIN_MAX_IN', 'TCP_WIN_MAX_OUT'}	98.95
	All features	1
<b>NF-BoT-IoT-v2</b>		
Reconnaissance	All features	1
	{'ICMP_IPV4_TYPE', 'ICMP_TYPE', 'IN_PKTS', 'MAX_IP_PKT_LEN', 'MAX_TTL', 'MIN_IP_PKT_LEN', 'NUM_PKTS_1024_TO_1514_BYTES', 'NUM_PKTS_256_TO_512_BYTES', 'NUM_PKTS_UP_TO_128_BYTES', 'OUT_PKTS', 'RETRANSMITTED_OUT_PKTS', 'SERVER_TCP_FLAGS', 'TCP_WIN_MAX_IN', 'TCP_WIN_MAX_OUT'}	99.63
DOS	'DST_TO_SRC_AVG_THROUGHPUT', 'ICMP_IPV4_TYPE', 'ICMP_TYPE', 'IN_PKTS', 'L7_PROTO', 'MAX_IP_PKT_LEN', 'MAX_TTL', 'MIN_IP_PKT_LEN', 'NUM_PKTS_1024_TO_1514_BYTES', 'NUM_PKTS_128_TO_256_BYTES', 'RETRANSMITTED_IN_PKTS', 'RETRANSMITTED_OUT_PKTS', 'SERVER_TCP_FLAGS', 'SHORTEST_FLOW_PKT', 'TCP_FLAGS', 'TCP_WIN_MAX_IN'}	99.99
	All features	1
Theft	{'CLIENT_TCP_FLAGS', 'DST_TO_SRC_SECOND_BYTES', 'DURATION_IN', 'FLOW_DURATION_MILLISECONDS', 'ICMP_IPV4_TYPE', 'MAX_IP_PKT_LEN', 'MAX_TTL', 'MIN_IP_PKT_LEN', 'NUM_PKTS_1024_TO_1514_BYTES', 'NUM_PKTS_128_TO_256_BYTES', 'NUM_PKTS_256_TO_512_BYTES', 'NUM_PKTS_UP_TO_128_BYTES', 'OUT_PKTS',	99.99

		'PROTOCOL','RETRANSMITTED_IN_BYTES','RETRANSMITTED_IN_PKTS','RETRANSMITTED_OUT_BYTES','RETRANSMITTED_OUT_PKTS','SERVER_TCP_FLAGS','SRC_TO_DST_AVG_THROUGHPUT', 'TCP_FLAGS', 'TCP_WIN_MAX_IN', 'TCP_WIN_MAX_OUT']					
Attack	Previous All features	Reduced features	Last processing time	Current Processing time	Previous Space Complexity	Current Space Complexity	
DDOS	43	15	28.55 seconds	8.56 seconds	99.99 bytes	99.57 bytes	
Analysis	43	21	28.55 seconds	8.56 seconds	10450971 bytes	1047257 bytes	
Backdoor	43	13	19.98 seconds	5.90 seconds	10038012 bytes	1036441 bytes	
DoS	43	15	35.93 seconds	11.00 seconds	10043774 bytes	1082985 bytes	
Generic	43	15	38.37 seconds	12.40 seconds	1076953 bytes	864937 bytes	
Exploits	43	13	20.03 seconds	7.04 seconds	8645085 bytes	1078809 bytes	
Shell Code	43	18	21.31 seconds	8.33 seconds	86551333 bytes	1071801 bytes	
worms	43	21	28.55 seconds	10.76 seconds	10450979 bytes	1069353 bytes	
Reconnaissance	43	13	19.99 seconds	5.90 seconds	1045097 bytes	1098777 bytes	

Table 9

Table 10

## Chapter 6

### 6.1 Conclusion and Future Work

In this pivotal chapter, we meticulously delineate the contributions, limitations, and prospective avenues for future research, all of which emanate from the rigorous exploration of our thesis. Our focal endeavors encompass the elucidation of our innovative model, comprehensive experimentation, and a thorough review of the attained findings.

Foremost, our novel model has yielded substantial enhancements in both processing time and space complexity. By judiciously reducing the number of features, we achieved remarkable reductions in processing time, rendering our model significantly more efficient and responsive. Moreover, through

prudent management of memory resources, we successfully curtailed the space complexity, affording a more streamlined and economical utilization of computational resources.

One of the paramount challenges we encountered pertained to the accuracy of our multi-class classifier, which was previously susceptible to overfitting. However, with the adoption of our newly proposed featureset we triumphantly surmounted this limitation. The refined model now exhibits substantially improved accuracy, bringing us closer to the decisive resolution of real-world predicaments in network security and classification.

## **6.2 Limitations**

In the course of conducting my thesis, I acknowledge that there are certain limitations to the research that warrant consideration. One significant limitation is that the experimentation phase of this study did not involve real-time network attack data. Instead, the research heavily relied on previously defined datasets to assess the performance of the classification model. While these datasets have been widely used in the cybersecurity community and are well-established benchmarks, they might not fully capture the dynamic and evolving nature of real-time network attacks.

By utilizing pre-existing datasets, the study might not have accounted for the latest and emerging cyber threats, which could potentially lead to a lack of representation of the current threat landscape. Moreover, since real-time attacks are often sophisticated and ever-changing, the model's performance in a live environment could differ from the results obtained through the use of static datasets.

Another limitation to consider is the potential bias or incompleteness present in the selected datasets. These datasets might not encompass all possible attack scenarios or might be skewed towards specific attack types. Consequently, the classification model's generalization and ability to accurately detect novel or less-represented attacks may be affected.

Furthermore, the lack of real-time data limits the evaluation of the model's responsiveness and adaptability to sudden variations in attack patterns or network behavior. In a live environment, the model's ability to promptly detect and respond to emerging threats is critical to maintaining network security.

## **6.3 Future work**

To address these limitations and enhance the practical applicability of the research, future work should incorporate real-time data collection and experimentation with a diverse range of attack scenarios. Integrating real-world data streams and employing techniques such as data augmentation and online learning could contribute to a more comprehensive and dynamic evaluation of the classification model's performance. By embracing these improvements, the research findings would be better suited

to address the challenges posed by contemporary cybersecurity threats in real-world network environments.

## References

1. <https://www.globaldots.com/resources/blog/41-6-billion-iot-devices-will-begenerating-79-4-zettabytes-of-data-in-2025/>
2. <https://www.theverge.com/22589942/nso-group-pegasus-project-amnestyinvestigation-journalists-activists-targeted>
3. <https://www.washingtonpost.com/politics/2020/12/07/cybersecurity-202-globallosses-cybercrime-skyrocketed-nearly-1-trillion-2020/>
4. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf/>
5. pawar2015network, title=Network security and types of attacks in network, author=Pawar, Mohan V and Anuradha, J, journal=Procedia Computer Science, volume=48, pages=503–506, year=2015, publisher=Elsevier
6. Tasnuva Mahjabin and Yang Xiao and Guang Sun and Wangdong Jiang, title =A survey of distributed denial-of-service attack, prevention, and mitigation techniques, journal = International Journal of

Distributed Sensor Networks, volume = 13, number = 12, pages = 1550147717741463, year = 2017, doi = 10.1177/1550147717741463,

7. aljabri2021intelligent, title=Intelligent Techniques for Detecting Network Attacks: Review and Research Directions, author=Aljabri, Malak and Aljameel, Sumayh S and Mohammad, Rami Mustafa A and Almotiri, Sultan H and Mirza, Samiha and 12 J.R, H.A, W.S Anis, Fatima M and Aboulmour, Menna and Alomari, Dorieh M and Alhamed, Dina H and Altamimi, Hanan S, journal=Sensors, volume=21, number=21, pages=7070, year=2021, publisher=Multidisciplinary Digital Publishing Institute

8. deshमुख2015understanding, title=Understanding DDoS attack & its effect in cloud environment, author=Deshmukh, Rashmi V and Devadkar, Kailas K, journal=Procedia Computer Science, volume=49, pages=202–210, year=2015, publisher=Elsevier

9. mallik2019man, title=Man-in-the-middle-attack: Understanding in simple words, author=Mallik, Avijit, journal=Cyberspace: Jurnal Pendidikan Teknologi Informasi, volume=2, number=2, pages=109–134, year=2019

10. rahim2017man, title=Man-in-the-middle-attack prevention using interlock protocol method, author=Rahim, Robbi, journal=ARN J. Eng. Appl. Sci, volume=12, number=22, pages=6483–6487, year=2017

11. eian2020wireless, title=Wireless Networks: Active and Passive Attack Vulnerabilities and Privacy Challenges, author=Eian, Isaac Chin and Lim, Ka Yong and Yeap, Majesty Xiao Li and Yeo, Hui Qi and Fatima, Z, year=2020, publisher=Preprints

12. <https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2021GTR.pdf>

13. krombholz2015advanced, title=Advanced social engineering attacks, author=Krombholz, Katharina and Hobel, Heidelinde and Huber, Markus and Weippl, Edgar, journal=Journal of Information Security and applications, volume=22, pages=113–122, year=2015, publisher=Elsevier

14. eian2020wireless, title=Wireless Networks: Active and Passive Attack Vulnerabilities and Privacy Challenges, author=Eian, Isaac Chin and Lim, Ka Yong and Yeap, Majesty Xiao Li and Yeo, Hui Qi and Fatima, Z, year=2020, publisher=Preprints

15. @inproceedingsbanerjee2019impact, title=Impact of machine learning in various network security applications, author=Banerjee, Jayashree and Maiti, Sumana and Chakraborty, Sumalya and Dutta, Surajit and Chakraborty, Arpita and Banerjee, Jyoti Sekhar, booktitle=2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pages=276–281, year=2019, organization=IEEE

16. <https://suricata.io/>

17. <https://www.snort.org/>

18. <https://zeek.org/>

19. <https://www.ntop.org/products/netflow/nprobe/>

20. <https://www.cisco.com/c/en/us/support/security/stealthwatch-flow-sensorseries/series.html>

21. gustavsson2019machine, title=Machine Learning for a Network-based Intrusion Detection System: An application using Zeek and the CICIDS2017 dataset, author=Gustavsson, Vilhelm, year=2019
22. sarhan2020netflow, title=Netflow datasets for machine learning-based network intrusion detection systems, author= Sarhan, Mohanad and Layeghy, Siamak and Moustafa, Nour and Portmann, Marius, journal=arXiv preprint arXiv:2011.09144, year=2020
23. <https://www.eginnovations.com/blog/what-is-netflow/>
24. sarhan2021towards, title=Towards a standard feature set of nids datasets, author= Sarhan, Mohanad and Layeghy, Siamak and Moustafa, Nour and Portmann, Marius, journal=arXiv preprint arXiv:2101.11315, year=2021
25. zhang2019feature, title=A feature analysis based identifying scheme using GBDT for DDoS with multiple attack vectors, author=Zhang, Jian and Liang, Qidi and Jiang, Rui and Li, Xi, journal=Applied Sciences, volume=9, number=21, Title Suppressed Due to Excessive Length 13 pages=4633, year=2019, publisher=Multidisciplinary Digital Publishing Institute .
26. sarhan2021explainable, title=An Explainable Machine Learning-based Network Intrusion Detection System for Enabling Generalisability in Securing IoT Networks, author= Sarhan, Mohanad and Layeghy, Siamak and Portmann, Marius, journal=arXiv preprint arXiv:2104.07183, year=2021
27. alaidaros2017flow, title=Flow-based approach on bro intrusion detection, author=Alaidaros, Hashem and Mahmuddin, Massudi, journal=Journal of Telecommunication, Electronic and Computer Engineering, volume=9, number=2-2, pages=139–145, year=2017, publisher=Universiti Teknikal Malaysia Melaka
28. sharafaldin2018toward, title=Toward generating a new intrusion detection dataset and intrusion traffic characterization., author=Sharafaldin, Iman and Lashkari, Arash Habibi and Ghorbani, Ali A, journal=ICISSp, volume=1, pages=108–116, year=2018
29. elsayed2019machine, title=Machine-learning techniques for detecting attacks in SDN, author=Elsayed, Mahmoud Said and Le-Khac, Nhien-An and Dev, Soumyabrata and Jurcut, Anca Delia, journal=arXiv preprint arXiv:1910.00817, year=2019
30. @inproceedingsprakash2018intelligent, title=An intelligent software defined network controller for preventing distributed denial of service attack, author=Prakash, Aditya and Priyadarshini, Rojalina, booktitle=2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pages=585–589, year=2018, organization=IEEE
31. @inproceedingsli2018using, title=Using SVM to detect DDoS attack in SDN network, author=Li, Dong and Yu, Chang and Zhou, Qizhao and Yu, Junqing, booktitle=IOP Conference Series: Materials Science and Engineering, volume=466, number=1, pages=012003, year=2018, organization=IOP Publishing
32. Al-Sarem, Mohammed et al. “An Aggregated Mutual Information Based Feature Selection with Machine Learning Methods for Enhancing IoT Botnet Attack Detection.” Sensors (Basel, Switzerland) vol. 22,1 185. 28 Dec. 2021, doi:10.3390/s22010185

33. Javier Maldonado, Mar'ia Cristina Riff, Bertrand Neveu, A review of recent approaches on wrapper feature selection for intrusion detection, Expert Systems with Applications, Volume 198, 2022, 116822, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2022.116822>. (<https://www.sciencedirect.com/science/article/pii/S0957417422002780>)
34. @articleMALDONADO2022116822, title = A review of recent approaches on wrapper feature selection for intrusion detection, journal = Expert Systems with Applications, volume = 198, pages = 116822, year = 2022, issn = 0957-4174, doi = <https://doi.org/10.1016/j.eswa.2022.116822>, url = <https://www.sciencedirect.com/science/article/pii/S0957417422002780>, author = Javier Maldonado and Mar'ia Cristina Riff and Bertrand Neveu, article
35. <https://doi.org/10.1007/s11036-021-01843-0>