1

by Sub1 Upm

Submission date: 03-Feb-2022 03:19AM (UTC+0300)
Submission ID: 1753776976
File name: Capstone_1_Final_Report-2022-2023.docx (2.54M)
Word count: 12448
Character count: 69873

GP Project

CSFC Department

Project ID: UPM-CSFC-F2021-08-Sem1

January 20, 2022

جامعة الأمير مقرن بن عبد العزيز
University of Prince Mugrin

GUARDIAN

Machine learning framework to detect network malicious traffic in cloud environments

2021 – 2022

Dept. of Cyber Security and Forensic Computing

Faculty of Computer Science and Information Technology

University of Prince Mugrin, Madinah KSA

This project report is submitted to the Department of Cyber Security and Forensic Computing at University of Prince Mugrin in partial fulfillment of the requirements for the degree of Bachelor of Science in Cyber Security and Forensic Computing.

Author(s):

- **Omar Belal**

E-mail: 3710137@upm.edu.sa

- **Rafif Altayar**

E-mail: 3910162@upm.edu.sa

- **Bayan Ghannam**

E-mail: 3810127@upm.edu.sa

- **Aghiad Massarani**

E-mail: 3810225@upm.edu.sa

- **Abdullah Basalama**

E-mail: 3810177@upm.edu.sa

Supervisor(s):

- **Mr. Mohamad Al Khatib**

E-mail: m.al-khatib@upm.edu.sa

- **Mr. Naveed Ahmad**

E-mail: n.ahmad@upm.edu.sa

Dept. of Cyber Security and Forensic Computing
Faculty of Computer Science and Information Technology
University of Prince Mugrin, Madinah KSA

Internet: http://upm.edu.sa
Phone: +966 014 - 831 8484

Intellectual Property Right Declaration

This is to declare that the work under the supervision of Cyber Security and Forensic Computing having title "GUARDIAN, Machine learning framework to detect network malicious traffic in cloud environments" carried out in partial fulfillment of the requirements of Bachelor of Science in Cyber Security and Forensic Computing, is the sole property of the University of Prince Mugrin and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

This above statement applies to all students and faculty members.

Date: 20/1/2022

Author(s):

Omar Belal	Signature: OM
Rafif Altayar	Signature: RT
Bayan Ghannam	Signature: BG
Aghiad Massarani	Signature: AM
Abdullah Basalama	Signature: AB

Supervisor(s):

Mr. Mohamad Al Khatib	Signature:
Mr. Naveed Ahmad	Signature:

ABSTRACT

Several solutions have been implemented in the past to provide an effective methodology to detect and classify anomalies in computer networks using machine learning, where most of these solutions had been deployed either using some sort of IDS solutions or a next generation firewall. Most of the previous studies provided a state of art description around anomaly detection and classification. However, the purpose of this study is to provide a fully automated solution that classifies network traffic in various stages. By reducing the downtime of the production environment and keeping it continuously running with the highest effectiveness as possible without interfering with the security poster of the organization. The whole infrastructure is going to be built using one of the well-known cloud service providers (CSPs) such as AWS, Azure, or Google. The proposed system will be implemented and designed using different machine learning classification techniques. We are defining three main classification models, first model as high-level classical machine learning and Deep Packet Inspection (DPI), Second implementation will be conducted using machine learning classifiers alone, both must be accomplished using synthetic network traffic that would be generated by virtual simulated network environment, second implementation will be focusing on using deep neural network (DNN) technique with an online huge ground truth dataset. Finally, a comparison between all three methods to conclude the best accuracy output.

ABBREVIATION

Abbreviation	Definition	Abbreviation	Definition
GNS	Graphical Network Simulator	**NP**	Network Processors
DDoS	Distributed Denial-of-Service	**CAM**	Content Addressable Memory
DoS	Denial-of-Service	**TCAM**	Ternary Content Addressable Memory
ID2T	Intrusion Detection Dataset Toolkit	**RD**	Random Forest
DNS	Domain Name Server	**FPGA**	Field Programmable Gate Array
NTA	Network Traffic Analysis	**NIDS**	Network-based Intrusion Detection System
DPI	Deep Packet Inspection	**HIDS**	Host-based Intrusion Detection System
SPI	Shallow Packet Inspection	**IDS**	Intrusion Detection System
MPI	Medium Packet Inspection	**DT**	Decision Tree
BMHP	The Boyer-Moore-Horspool	**SVM**	Support Vector Machine
FAR	False Alarm Rate	**KNN**	K-Nearest Neighbors
ML	Machine Learning	**IaaS**	Infrastructure-as-a-Service
DL	Deep Learning	**PaaS**	Platform-as-a-Service
FLAME	Flow-Level Anomaly Modeling Engine	**SaaS**	Software-as-a-Service
DNN	Deep Neural Network	**VM**	Virtual Machine

Table Of Content

Table of figures

1 INTRODUCTION

In the age of technology and the explosion of information, it became common and important to use clouds to store data temporarily or permanently. Also, it became important to provide protection and continuous maintenance for these clouds and keep them secure against all new attacks and vulnerabilities that appear every day. One important part of the security process is to detect any malicious attempt to compromise the clouds and exploit any vulnerabilities that might be there. For clouds security it is particularly important to detect network malicious packets that contain some types of attack and attempt to harm the system.

Therefore, this research will illustrate a network traffic classifier using machine learning and traffic pattern analysis, where we try to detect the malicious traffic using machine learning to protect the system. Since this problem includes all kinds of network attacks, this research will only focus on some of the most popular network security attack types such as: (distributed denial-of-service (DDOS) with its multiple various types such as “smurf, syn flood, UDP flood, and TCP rest,” Brute Force attacks, Fuzzing, and propping. We also describe our plan of generating and labeling our own dataset that includes malicious (attack types included as labels) and normal traffic. In addition, we describe the process of data pre-processing and classifiers training to train our machine learning module to detect malicious traffic and classify them.

This research is constructed as following: First section describes where the problem hits and what is the suggested solution. Second, the founded related studies used to build the knowledge required to design the system and found the existing deployed solutions. Third section gives a Background about machine learning and cyber network attacks with some definitions for basic knowledge that the reader may need to understand the proposed methodology. fourth section describes system’s components and requirement to build and design the whole infrastructure. last section, contains Machine learning methodology with full explanation of how ML modules could be created and needed datasets for this project with demonstration of the data pre-processing and ML classification faces.

1.1 Problem Definition

The focus of the problem statement centralizes on providing a new methodology of analyzing the patterns of the captured network traffic by using machine learning classification algorithms that will classify them into classification schemas Binary and multi-classification starting as benign or malicious traffic, then classify the malicious traffic according to the attack type. This approach will help reducing damages that may affect various aspects in the organizations, especially the financial side as this technology will help in reducing detecting malicious attacks in real time with high accuracy even when those packets are encrypted. Solving this kind of problem requires an integration between multiple system types that could be combined with this technology to create a fully dynamic solution.

The following proposed system consists of two main phases.

1- In the first stage a full experimental environment will be constructed to generate, analyze the network traffic, and create the dataset with the appropriate labels.

2- Second phase using machine learning will go through one of the most important security problems nowadays, which is network traffic finger printing where the network packet will be classified into many categories: going from the first one: BINARY classification problem where the network packet will be classified into two labels malicious and benign then convert to a full MULTI-CLASSIFICATION problem where a number of labels would be identified in advanced in the dataset.

2 Related Study

proposed tools and requirements mentioned in [1] for creating a new dataset by generating encrypted network traffic in a real-world environment, and a new dataset developed by generating encrypted network traffic in a real-world environment. The contributions of researchers are divided into two main parts. First, they have proposed several new crucial criteria for developing new datasets. Second, they built a new IDS dataset that includes encrypted traces of network traffic. Attacks such as brute force login and probing are labeled in the dataset. Along with the background traffic (normal) and ground-truth data, packet traces with payloads are provided. Using Zeek [2], an open-source network security monitoring tool, they retrieved and incorporated more than 80 features from the CICIDS-2017 dataset for ground-truth, normal traffic, and malicious traffic.

In [3] they presented a phenomenon methodology on labeling datasets If a supervised technique is selected, the lack of labels in data sets forces researchers to manually examine and attribute data according to the phenomena they are trying to model and identify network packets.Following [4] have proposed a detection system based on a collection of network traffic attributes to feed IDS that uses an ML model to detect malicious traffic. The ISOT-CID dataset was generated by Aldribi et al. [5]. Using Sklearn, Numpy, Matplotlib, and Pandas, the proposed model is prepared, created, fitted, and evaluated in Python. Their appealing model should be easy to build and fit in memory, so it can respond to network traffic features and forecast anomalies in real time.

On creating ML classification methodology [6] proposed a full technical process on going through the traffic classification methodology by providing seven main steps to construct a network traffic classifier: The first step is to precisely state the classification objective, to achieve its goal, traffic classes can be divided into: Protocols, applications (e.g., Skype, WeChat, or Torrent), traffic-types (e.g., browsing, downloading, or video chat), websites, user actions, operating systems, browsers, and so on. Hence, the goal is to categorize each flow with relevant traffic classes: The source IP, destination IP, source port, destination port, and protocol are commonly used to determine a flow, a huge and ground truth dataset, Data cleaning and preprocessing, feature categories, model selection, Training & Validation (tuning), and finally a periodic evaluation for the models.

Due to the huge challenges that may face AI researchers [7] have Proposed a valuable technique that could be used to generate a sufficient amount of Network traffic in order to create novel updated datasets to use it for ML based deployments such as classification and regression

problems. Another considerable approach that may be used to generate and build the dataset was proposed by [8] is using an existing tool named ID2T that may inject Separated attack traffic into the available Background traffic without the need to create simultaneously multiple types of attack's packets, ID2T is a tool that generates and injects bogus attack traffic into normal traffic. ID2T seeks to recreate the properties of the background traffic in the synthetic attack traffic rather than merely naively combining background and attack traffic. The user can choose the level at which properties are duplicated to suit their needs.

In [9] The network traffic generating research activities are then historically examined and classified into three different methods:

- **Network traffic model-based traffic generation**. Poisson model [10] is a common network traffic model used to create traffic.
- **traffic generation based on the features of the traffic.** According to the hierarchy of characteristics, network traffic creation based on traffic characteristics can be classified into two methods: Traffic generation on the network at the flow level and traffic generation at the packet level. for example, using Harpoon [11] which is a new application-independent tool for simulating packet traffic at the IP flow level. Various performance testing is facilitated by the produced traffic based on flow level.
- **traffic generation based on application protocol**. Different traffic generating tools are used for different application protocols. TCP and UDP bandwidth are tested using the network traffic generation tool Iperf [12]for network performance testing. It can report bandwidth, delay, jitter, and packet loss and has a number of options. In the NS-2 simulator, a tool called Tmix [13] was proposed to generate TCP application workload. The tool began by capturing a TCP packet header trace, which was then "reverse compiled" into a TCP connection vector.

3 Background

3.1 Cyber Network Attacks

3.1.1 Probing and Scanning

Probing/Scanning is the main method that is used by most of the attackers to gather and collect information as much as possible about the victim. Scanning can be done using several types of scanning methods, for example:

1- TCP FIN Scan: This type of attack is used mostly to determine if ports are close in the victim's machine. This type of scan works by sending TCP segments along with FIN bit in the packet header. If the port is open no response massage is being sent. However, if the victim sends an RST flag, then the port is closed.

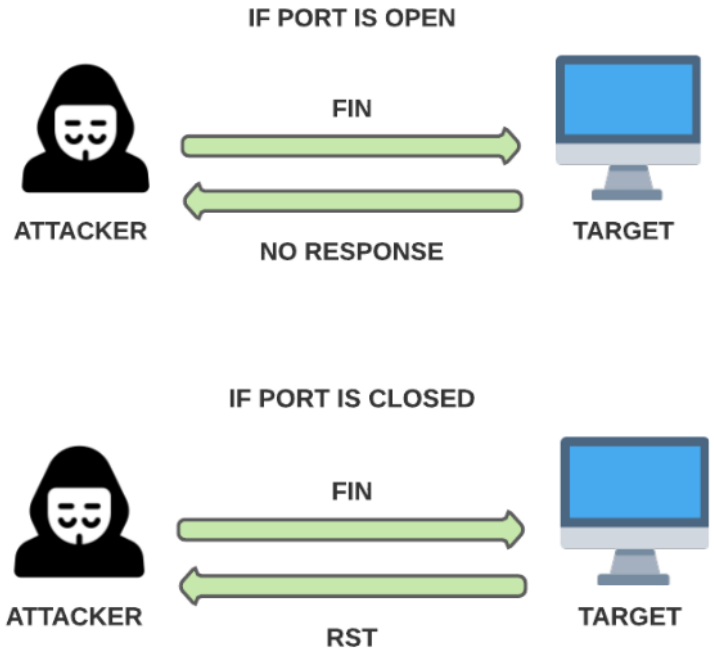

Figure 1 : Two types of responses to FIN scan.

2- TCP ACK Scan: The Ack scan depends on gathering information about the firewall or ACL configuration. Mostly this type of scan is used to discover information about filter configurations, and to check if the port is filtered or unfiltered. If the port is filtered which means the machine is behind firewall or ACL, the target will send error or no respond. On the hand if the port in unfiltered RST massage will be sent from the target's machine.

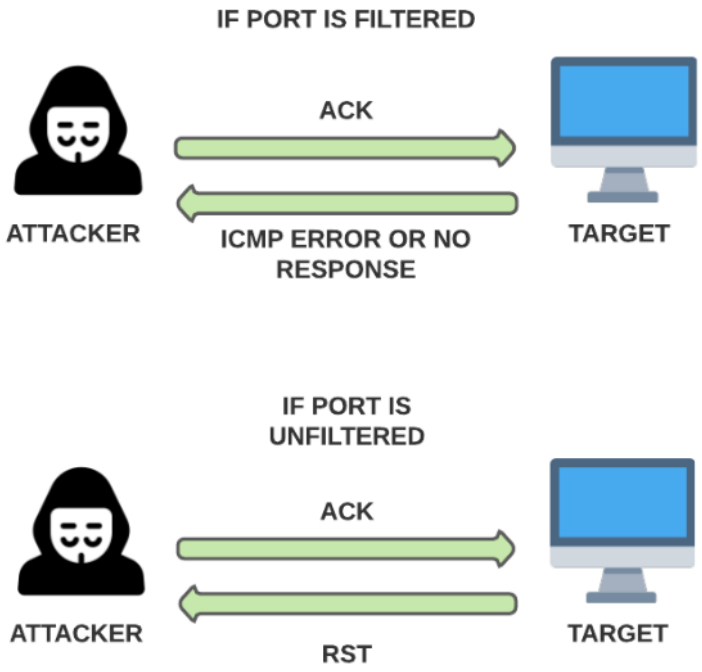

Figure 2: Two types of responses to ACK scan

3.1.2 Distributed denial-of-service (DDoS)

Multiple infected computers, known as bots or zombies, execute a DDoS attack against a single system. The attacker uses these machines (bots and zombies) to attack other computers. There are four essential components that must be present for a DDoS assault to be successful. The first is the genuine attacker, the second is the compromised hosts, which are known as handlers or masters and are capable of commanding multiple agents using software programs. The agent hosts, the third component, are responsible for sending a huge quantity of packets to the victim host. The fourth component is the victim, who is the target host.

One of the most common DDOS attacks is known as Slowloris.

Slowloris Attack

Slowloris is among the most well-known types of DDoS attacks, according to numerous studies. It works by establishing several connections to the web server of the victim and keeps them open for as long as feasible. To keep the connection open, this attack sends partial HTTP requests and subsequent headers for each request, but the requests are never finished. The victim's maximum concurrent connection pool is eventually filled, and valid connection requests are refused.

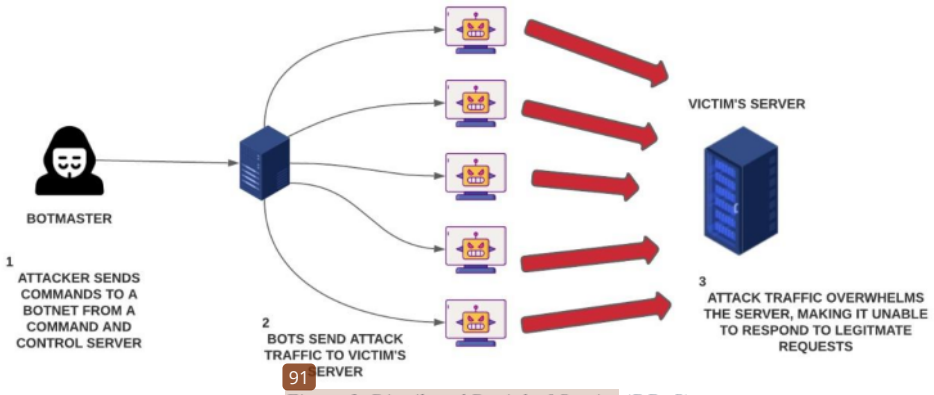

Figure 3: Distributed Denial-of-Service (DDoS)

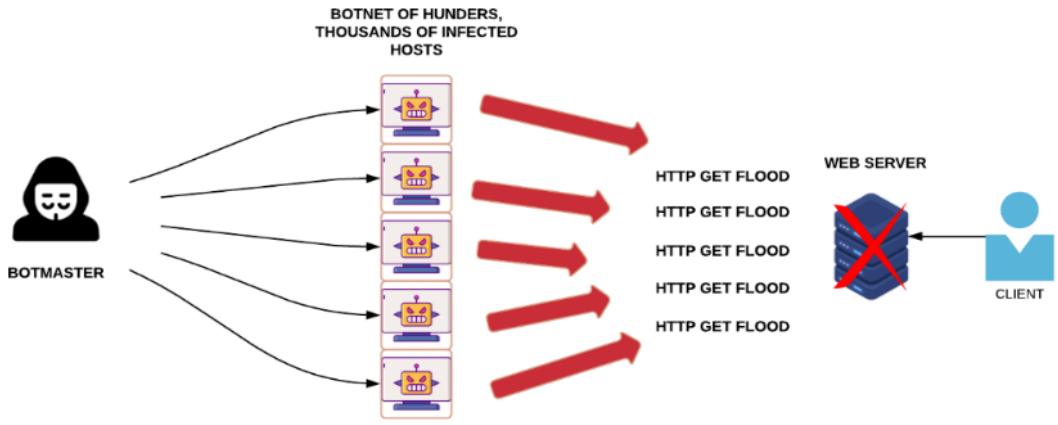

Figure 4: Slowloris Attack

Syn flood

37

TCP SYN flood[14] is a type of Distributed Denial of Service (DDoS) attack that exploits part of the normal TCP three-way handshake to consume resources on the targeted server and render it unresponsive. The attacker creates incomplete SYN packets to slow down the host machine and initiate a connection without finalizing it

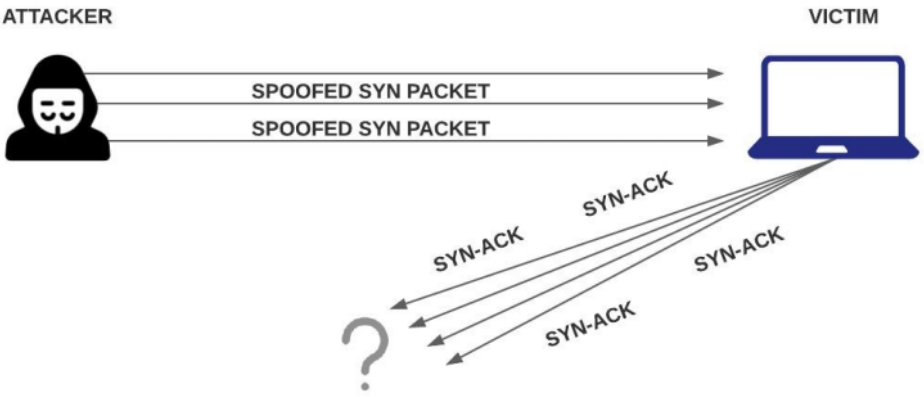

Figure 5: SYN Flood Attack

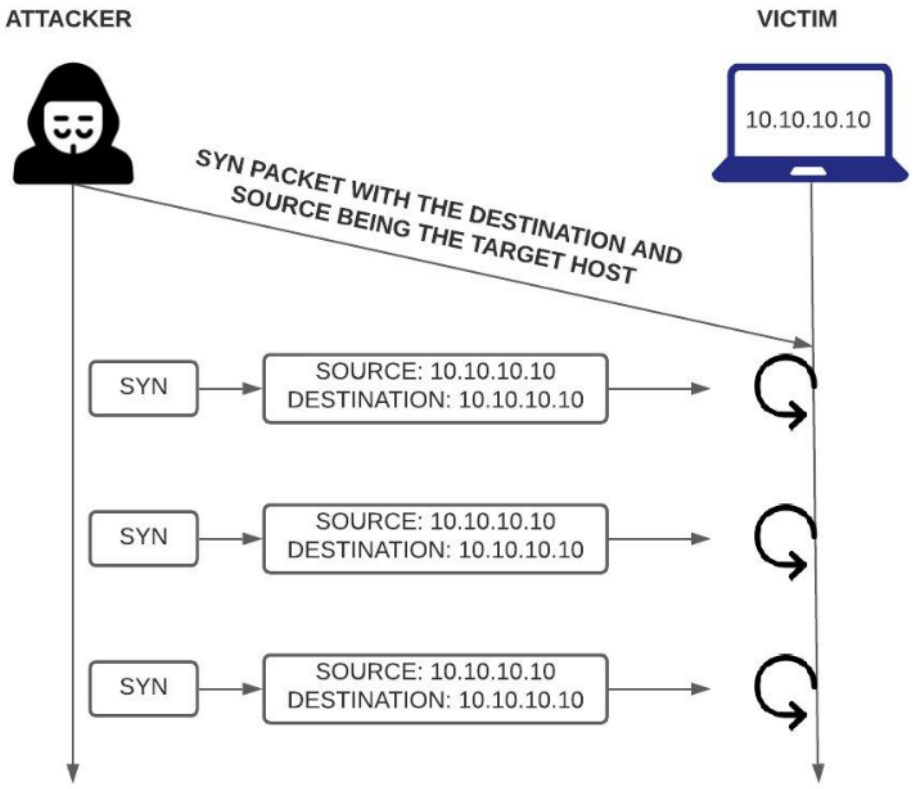

Figure6 : Land Attack

Land attack

In a DoS land (Local Area Network Denial) attack [15], the attacker transmits a faked TCP SYN packet with identical source and destination IPs and ports. When the target system tries to respond, it gets stuck in a loop, sending replies to itself repeatedly, finally crashing the victim machine. This Attack is a Layer 4 Denial of Service (DoS) attack [16]. One of the most popular tools used to generate this attack is Scapy.

3.1.3 DNS Spoofing:

Domain Name System (DNS) is a fundamental component of modern networking that translates between human-readable domain names and the underlying Internet Protocol (IP) addresses that connect devices to talk to each other. Since DNS is all about valuable information, from the attacker's perspective it could be a useful tool for hacking.

- **DNS Spoofing:**

  Because the original DNS protocol lacks authentication, attackers have a major advantage because queries and responses can be altered or falsified at many distinct stages, from the local workstation to global DNS resolvers. DNS spoofing (also known as DNS cache poisoning) is a type of attack in which modified DNS records are used to redirect online traffic to a fake website that looks like the real one.

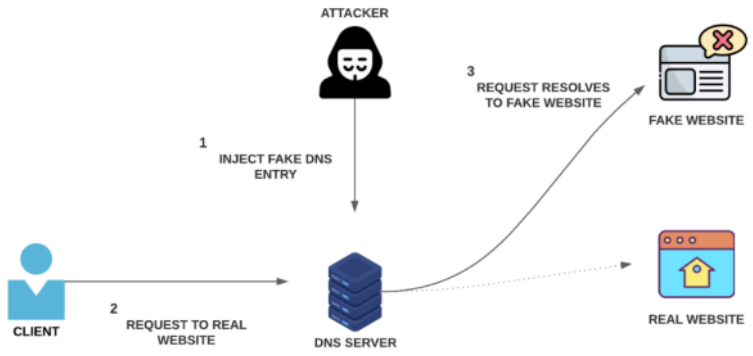

Figure 7: DNS Spoofing Attack Scheme

3.1.4 Brute Forcing Attack

A password is technically defined as secret of characters used to authenticate or gain access to resources. It must be kept secret and hidden from others who are not allowed to access those resources. Passwords have been used with computers since the earliest days of computing.

Brute force attack involves repeatedly trying to login as a user by trying every letter, number, and character combination (using automated tools). This can be done either online (in real-time, by continually trying different username/password combinations on accounts like social media or banking sites) or offline (such as when you have obtained a set of hashed passwords and are trying to crack them offline). For example, if you know that someone is using a 5-character long password, composed only of lowercase letters, the total number of passwords is 26^6 (26 letters to choose from for the first letter, 26 choices for the second letter, etc.), or 11,881,376 combinations.

Types of brute force attacks:

- **Simple brute force attack** — uses a systematic approach to 'guess' that does not rely on outside logic.
- **Hybrid brute force attacks** — starts from external logic to determine which password ...riation may be most likely to succeed.
- **Dictionary attacks** — guesses usernames or passwords using a dictionary of strings or phrases.
- **Rainbow table attacks** — a rainbow table is a precomputed table for reversing cryptographic hash functions.
- **Reverse brute force attack** — uses a common password or collection of passwords against many usernames.
- **Credential stuffing** — uses previously-known password-username pairs, trying them against multiple websites.

Figure 8: Brute Forcing Attack Scheme

3.2 Network Traffic Analysis using Deep Packet Inspection

Network Traffic Analysis (NTA) is the process of detecting, recording, and analyzing communication patterns to detect and respond to security menace, even when messages are encrypted. Traffic analysis is primarily performed to find out the data type, the traffic flowing through a network as well as data sources. However, it is used to discover communication patterns, and break in data over the network. Traffic analysis has many purposes such as evaluating the performance and security of network operations and management. Therefore, network traffic analysis is considered vital for improving network operation and security.

Deep Packet Inspection (DPI)

DPI is a type of network packet analysis that searches the contents of packets for certain patterns. These include the message's headers and data-protocol structures, as well as the message's payload. Advanced network administration, user service, and security features, as well as Internet data mining, eavesdropping, and filtering, are all possible with it. It is presently employed in a variety of applications by businesses, service providers, and governments. A wide range of pattern matching methods can be used to implement DPI [17]. Pattern matching is a fundamental subject in computer science that has been extensively investigated over the last few decades. Traditional algorithms, however, fail to meet current issues when applied to the network domain of recent years.

Traffic pattern analysis purposes:

- **Detect unknown threats:** There are many techniques for detecting threats, traffic pattern analysis has been proven to be the most effective tool. It can be used to detect unknown threats. Furthermore, using this tool, security administrators can configure the IP address range of each server within their local network, thus identifying external IP addresses.
- **Detect malware communication with trusted sources:** For example, every time you open a web browser, malware is registered in your Gmail account. In this case, with the help of traffic pattern analysis, we can monitor the high network traffic of email hosts, which is a starting point for further network inspection.

3.2.1 Different Packet Inspection Levels

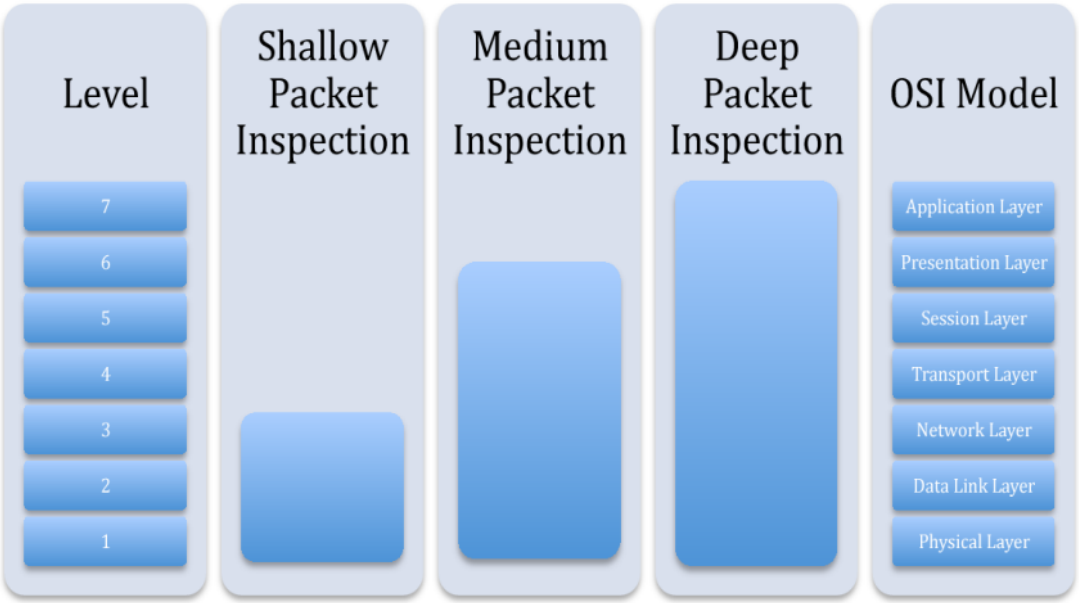

Figure 9: Packet Inspection levels

Figure 9 presents different inspection levels [18] such as Traditional packet forwarding systems such as Shallow Packet Inspection (SPI), Medium Packet Inspection (MPI), and Deep Packet Inspection (DPI) cannot identify the Data section of a packet, but Deep Packet Inspection (DPI) can. Furthermore, Shallow Packet Inspection (SPI) can analyze the Physical and Data Link Layers of Open System Interconnection (OSI), while Medium Packet Inspection (MPI) can detect the Transport, Network, Data Link, and Physical Layers of Open System Interconnection (OSI), and Deep Packet Inspection (DPI) can monitor all of the Layers of Open System Interconnection (OSI) of the reference model [19].

3.2.2 DPI TOOLS

1- nDPI

nDPI [20] [21] was designed to be used by applications that need to detect the application protocol of communication flow. Its focus is on Internet traffic; thus, all the available dissectors support standard protocols (e.g., HTTP and SMTP) or selected proprietary ones (e.g., Skype or Citrix) that are popular across the Internet community [22].

2- Net-Flow

NetFlow technology [23] is made up of three main components. The flow exporter, for example, groups packets into flows and exports flow records to one or more flow collectors. The flow collector, the second component, is in charge of receiving, storing, and pre-processing the flow data obtained from the preceding component. Finally, the received flow data is analyzed using the flow analyzer.

3.2.3 DPI MATCHING ALGORITHMS

- **Bloom Filter Algorithms**

  Machine learning and DPI algorithms [24] are two methods for preventing DDoS attacks; BLOOM FILTER is one of the top DPI categorization algorithms. Machine learning, on the other hand, is good at detecting trends and distinguishing between lawful and malicious requests. Bloom Filter is used to prevent DDoS attacks, which is surprising. Bloom Filter, unlike machine learning algorithms, is a simple data structure that uses very little memory.

- **Aho-Corasick (AC) Algorithm**

  The Aho-Corasick (AC) method[25] is a sophisticated but memory-intensive pattern matching algorithm with properties that make it a good fit for NIDS. The "Split-AC" algorithm, which is a version of the AC algorithm optimized for reconfigurable hardware implementation and uses 28-75 percent less memory than the state-of-the-art AC implementation.

- **Boyer-Moore Algorithm (BM)**

  The Boyer-Moore-Horspool (BMHP) algorithm [26] is one of the quickest extensions of this algorithm. The BMHP algorithm has a linear processing time that is proportional to the length of the string being searched.

  The search procedure takes significantly less time to complete than other string-matching methods. It has a Bad Match table property, which allows BMHP to bypass some of the matching operations. Another feature of BMHP is that when the sub string grows longer, the method becomes faster.

3.2.4 DPI Objectives

1- **Deterministic Performance:** The architecture must be able to run and handle traffic streams regardless of signature or traffic characteristics. As a result, even in the worst-case scenario of software and hardware-based systems, the system must manage traffic.

2- **Memory Efficiency:** In software implementations of DPI systems, memory access time is one of the key bottlenecks. In the meantime, it is important in hardware design because of access time and memory scarcity. As a result, a design with high memory efficiency is preferred.

3- **Dynamic update:** this objective is one of the main objectives in DPI design, specifically in hardware design, where there should be a way of removing and adding signatures of intruder to the system while maintaining the system operation

4- **Signature:** Fixed intruder patterns and regular expression are supported by the DPI system. In addition, the system is capable of dealing with a wide range of intruder patterns. [20]

5- **Scalability:** this objective should be mentioned for both hardware and software design while in a software-based system, scalability is not a major concern. In hardware-based systems, on the other hand, it is critical. As a result, hardware design must be able to handle an infinite number of signatures.

Another function that the DPI system can provide is inspecting multiple traffic sessions individually, not only inspecting intruders but also allocating them, and customizing signature subsets or full signatures to inspect.

3.2.5 DPI Challenges

- **Encrypted Data**

  the data which is encrypted cannot be inspected by DPI. However, there are some solutions to overcome this problem by plugging in the DPI component behind the decryption device

- **The Location of signature unknown**

  due to verity types of attacks on different types of applications, the pattern of intruders is not localized in specific place in the packet which means that the IDS must inspect all the payload of the packet against the attacker signatures

- **The search algorithm complexity**

  The complexity of the algorithm and the operations of comparison against the signatures of intruder decrease the throughput of the system. Thus, search algorithms are the main focus point in DPI research, whereas matching process is resource consuming. For example, the string-matching routines in SNORT [35] account for up to 70% of total execution time and 80% of instructions executed on real traces [4]

- **Slow connection**

  Deep packet inspection can **slow down network connection** by dedicating resources for the firewall to be able to handle the processing load.

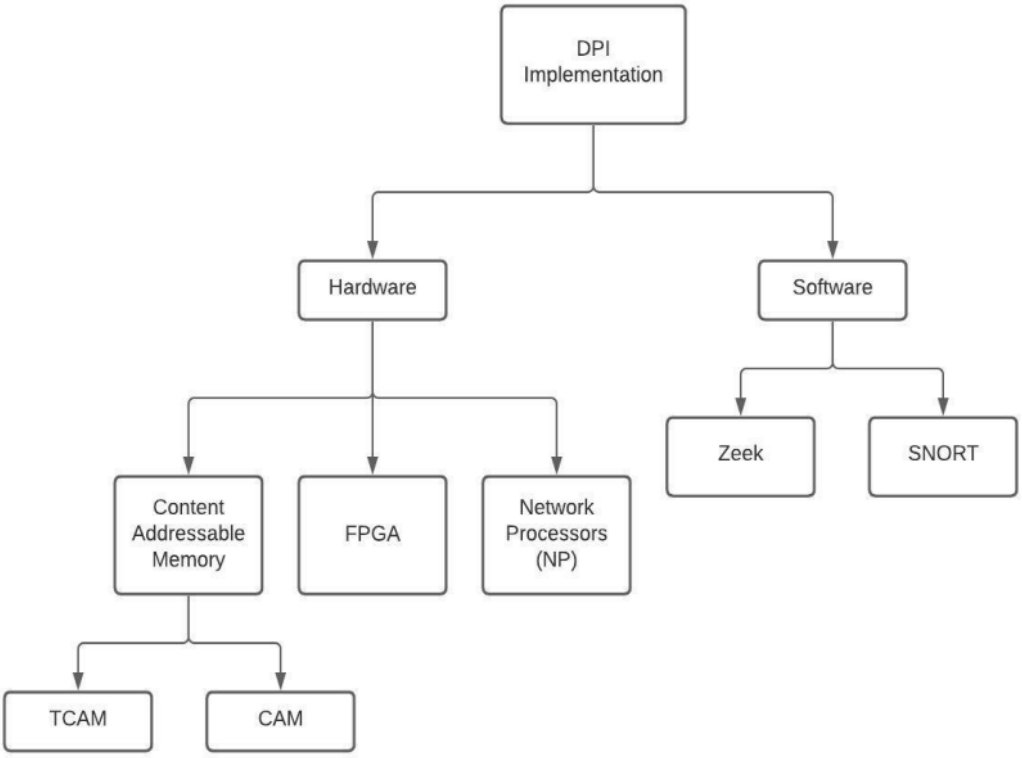

Figure 10: DPI Hardware & Software implementation

3.3 Machine Learning in Intrusion Detection System

With the increasing number of cybercrimes that have been discovered every day, especially in the last several years by using new different methods and attacks. Anti-virus software, firewalls, and intrusion detection systems are the most common cyber security tools, these methods defend networks against both internal and external threats.

3.3.1 Structure of Intrusion detection system

An Intrusion Detection Systems (IDS) is a detection system that monitors the states of software and hardware running in a network and helps to defend cyber security. IDS have taken almost obligatory lines of defense against attacks, it is a critical cyber security method, confront hurdles and keep track of the state of the network's software and hardware monitor, collect, and analyze logs, network traffic to assure its confidentiality, integrity, and availability by identifying suspicious behavior. Existing IDSs have demonstrated ineffectiveness in detecting a variety of threats, and in lowering false alarm rates (FAR). The main issue with current IDSs is that they are unable to identify unknown threats. Because network environments change rapidly, new attack types and attacks occur on a regular basis. [27]

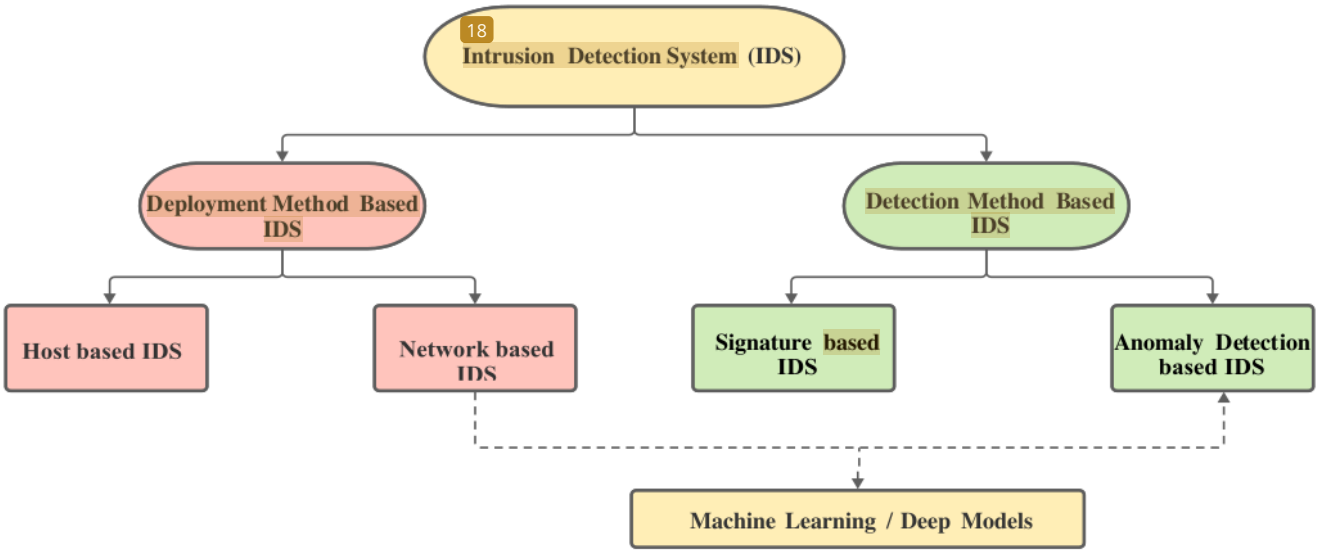

Figure 11: Intrusion detection system taxonomy

3.3.2 ML / DL based network-based intrusion detection system methodology

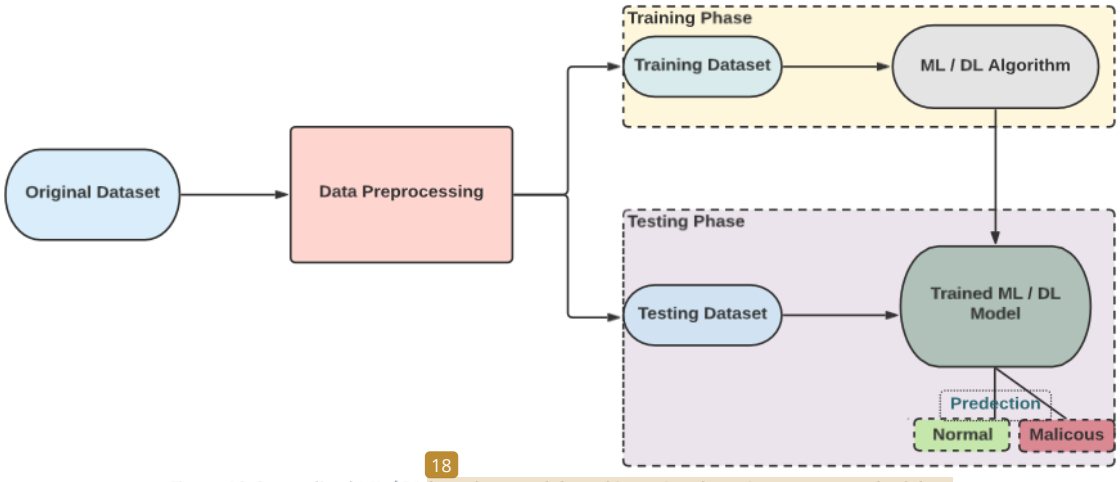

Figure 12:Generalized ML / DL based network-based intrusion detection system methodology

To solve the issues, academics have begun to concentrate on developing IDSs utilizing machine learning techniques. Machine learning and deep learning come under the big umbrella of artificial intelligence and aim at learning useful information from the big data. Both ML and DL are effective methods for extracting meaningful information from network traffic and predicting normal and abnormal actions based on the patterns learned. When enough training data is available, machine learning based IDSs can reach excellent detection levels, and machine learning models have sufficient generalizability to detect attack variants and novel attacks. With great accuracy, machine learning systems can automatically detect the main differences between normal and abnormal behaviors overhead in training and testing due to the large dataset size based on real network data. [28]

In the below table 1 we have mentioned some of the algorithms that could be applied in the supervised and unsupervised ML methods or algorithms that could be used in both.

Table 1 List of ML Algorithm Classifications.

ML ALGORITHMS	SUPERVISED AND UNSUPERVISED LEARNING METHOD
NAÏVE BAYES	Both
K-MEANS ALGORITHM	Unsupervised ML
K-NEAREST NEIGHBOUR	supervised ML
DECISION TREE AND RANDOM FOREST	supervised ML
SUPPORT VECTOR MACHINES (SVM)	supervised ML
PRINCIPAL COMPONENT ANALYSIS (PCA)	Unsupervised ML
RECURRENT NEURAL NETWORK (RNN)	supervised ML
DEEP LEARNING	Both
Q-LEARNING	None (Reinforcement ML)

3.3.3 Supervised Learning:

To train and generate predictions, supervised machine learning (SML) algorithms require prior knowledge of data. Because of its great predictive performance, SML is extensively employed in data science. However, the fact that these algorithms cannot be trained with unlabeled data, and this is a huge disadvantage. SML algorithms are divided into two categories: classification algorithms it is used to categorize the data into class or category, and it works with structured / unstructured data and regression algorithms, or Linear Regression is used for predicting dependent variable based on independent variable the benefit of this is to find the relationship between these two variables.

3.3.4 Unsupervised Learning

Unsupervised Machine Learning (UnML) has gotten a lot of press in the previous ten years. The fundamental cause for this is the vast volume of unlabeled data that is made available to the general population. It is critical to study these unlabeled data (exploratory data analysis) to uncover hidden patterns within them and minimize the amount of data for high-level activities such as labeling through dimensionality reduction to use these data effectively and efficiently. [29]

4 System Design & Requirements

4.1 System Components

4.1.1 Attack Creation tools

- **Scapy**

  Scapy is packet manipulation program [30]. A library made in Python enables the user to send, sniff and dissect and forge network packets. This capability allows construction of tools that can probe, scan or attack networks. A virtual DDOS attack has been implemented using Scapy by sending a vast number of packets and manipulating the analyzer by using the same source and destination.

- **Ostinato:**

  Ostinato is a packet crafter, network traffic generator and analyzer with a friendly GUI and powerful Python API for network test automation [31]. Craft and send packets of several streams with different protocols at different rates. It is "Wireshark in Reverse." Ostinato is a new user-level traffic generation tool that is available on a variety of platforms. Users can create multiple traffic streams using a user-friendly interface and send them to the network interface with ease. Figure 14 is a screen shot of Ostinato tool interface.

- **Nmap**

  Free open-source tool [32] for network mapping , discovery, and security auditing. It's also handy for network inventory, controlling service upgrade schedules, and monitoring host or service uptime, among other things. Nmap analyzes raw IP packets in unique ways to figure out what hosts are on the network, what services they offer (application name and version), what operating systems (and OS versions) they're running, what kind of packet filters/firewalls they're using, and thousands of other details.

- **hbing3** hping3 [33]is a network tool that can send bespoke ICMP/UDP/TCP packets and display target responses in the same way that ping displays ICMP responses.

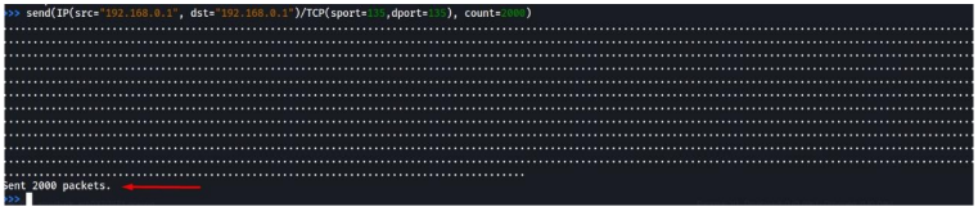

Figure 13:ScreenShots from Scapy program

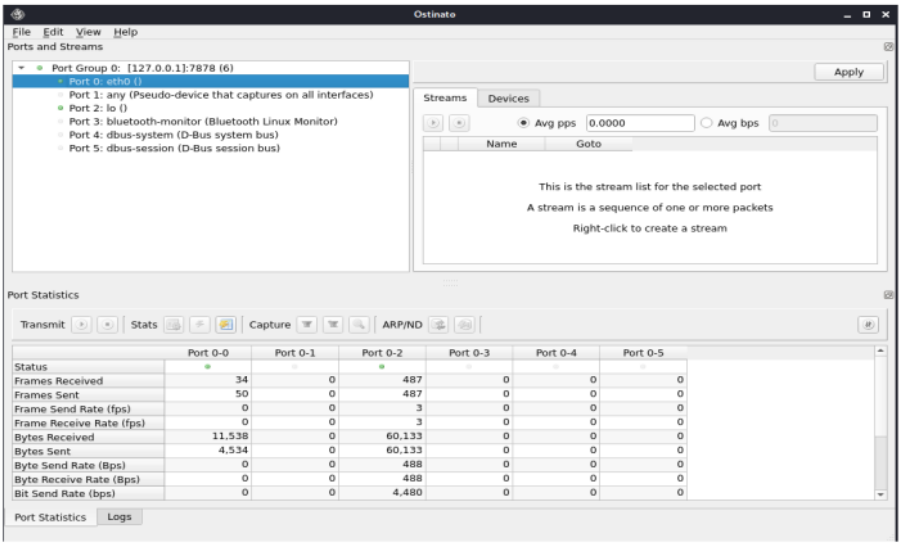

Figure 14:Ostinato GUI interface

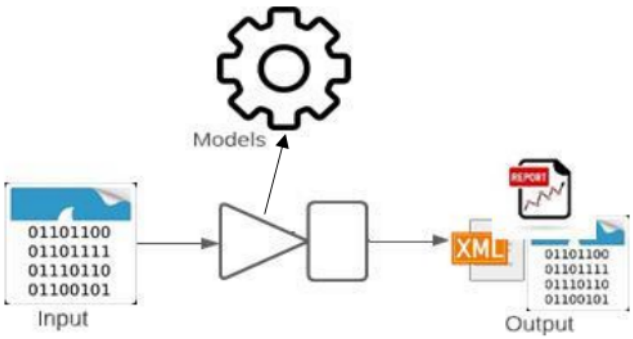

Figure 15:architecture of traffic injection tools

4.1.2 Dataset Generation tools

1. **ID2T**

   The user provides background traffic, and ID2T[34] adds attacks to the background traffic according to the user's parameters. For intrusion detection technologies to use, the injected attacks are labeled and readily identified, for example, for evaluation purposes. Figure 15 depicts the primary architectural components that work together to build datasets with synthetic attacks that mimic the features of background traffic. ID2T accepts a PCAP file containing background data and user-supplied parameters as input[35].

2. **FLAME**

   Researchers in [36] have created a tool that can inject anomalies into network flows. The Flow-Level Anomaly Modeling Engine (FLAME) is a tool that injects anomalies in one of three modes. Flows are injected without changing the background traffic in an additive mode (e.g., port scans). Flows are separated from background traffic using the subtractive mode (e.g., ingress shifts). Both modes are integrated into an interactive mode that allows you to add and remove flows (for example, to simulate congestion caused by a DoS).

3. **ISCX-UNB**

   In [37] They developed two sorts of profiling models, one for normal traffic and the other for malicious activities. The statistics of real network traffic are replicated to represent legitimate traffic. A custom description language is used to model malicious behavior. Network packets are generated from profiles using a test bed[38]. Although a dataset example is available, the tool is not easily available to the general public.

4. **INSecS-DCS**

   A framework[39] for collecting packet features from network traffic has been presented by [40] The framework either captures traffic directly from a network or accepts a PCAP file as input. The utility generates feature-value pairs in CSV format, which can be used by machine learning algorithms. The output characteristics can be customized by choosing the tracked protocols (e.g., TCP, UDP), packet attributes (e.g., IP, port), and a statistical properties time window. There is no network traffic generated by this tool. Rather, it extracts characteristics.

4.1.3 Machine Learning Frameworks

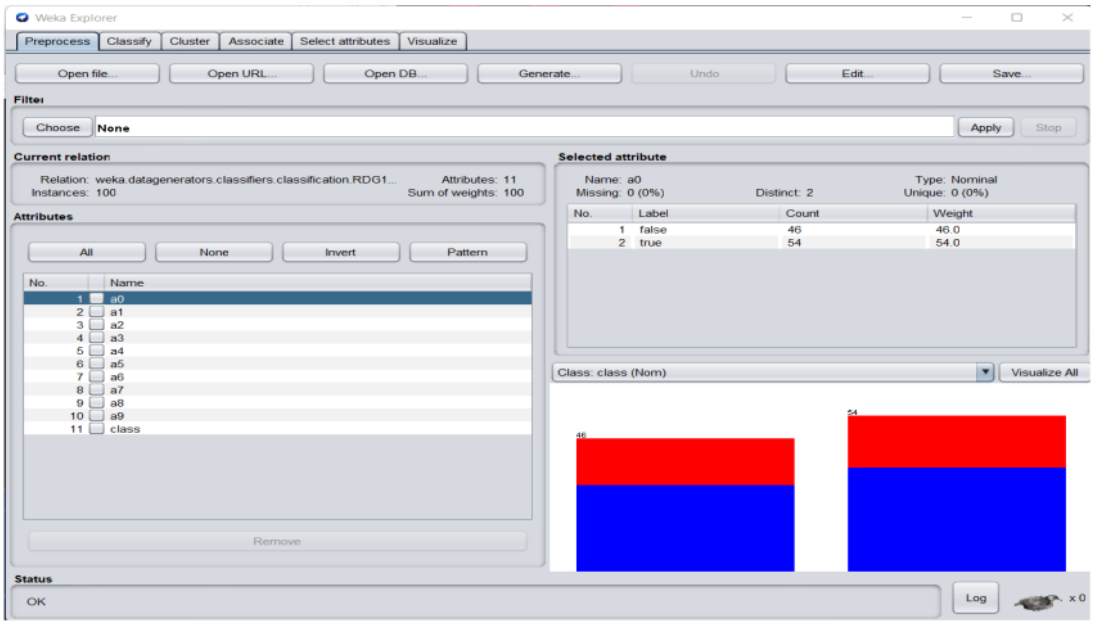

Figure 16: WEKA generated labels Of CSV file

- **Weka**

WEKA is a Java-based machine learning program that is open source. A Graphical User Interface (GUI), normal terminal apps, or a Java Application Programming Interface (API) [41] can all be used to access weka , above figure demonstrate how weka can generate labels from csv dataset , It includes supervised, semi-supervised, and unsupervised machine learning methods like Random Forest Support Vector Machine, Forest, Linear Regression, and KNN (SVM) and other ML algorithms that could be severely useful.

- **TensorFlow**

TensorFlow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages.

- **Scikit-learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python Some popular groups of models provided by scikit-learn include **Clustering**: for grouping unlabeled data such as KMeans. **Cross Validation**: for estimating the performance of supervised models on unseen data, and **Datasets**: for test datasets and for generating datasets with specific properties for investigating model behavior.

4.1.4 Feature Extraction & Network Analysis Tools

- **CIC-flowmeter**

It can be used to generate bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence more than 80 statistical network traffic features such as Duration, Number of packets, Number of bytes, Length of packets, etc. can be calculated separately in the forward and backward directions[42].

- **NF-Stream**

NFStream [43] is an open-source framework that allows high throughput network traffic flow analysis to be run on commodity hardware. The flows are being formed by aggregating packets that share a common key. A seven-tuple currently defines this flow key: the source and destination IP addresses and port numbers, and the protocol, VLAN, and tunnel identifiers.

- **Wireshark**

  let us see what is happening on your network at a microscopic level and is the standard across many commercial and non-profit enterprises, government agencies, and educational institutions.it is network administrators use it to troubleshoot network problems [44], Network security engineers use it to examine security problems. For Command line Tshark it is used for dumping and analyzing network traffic [45].

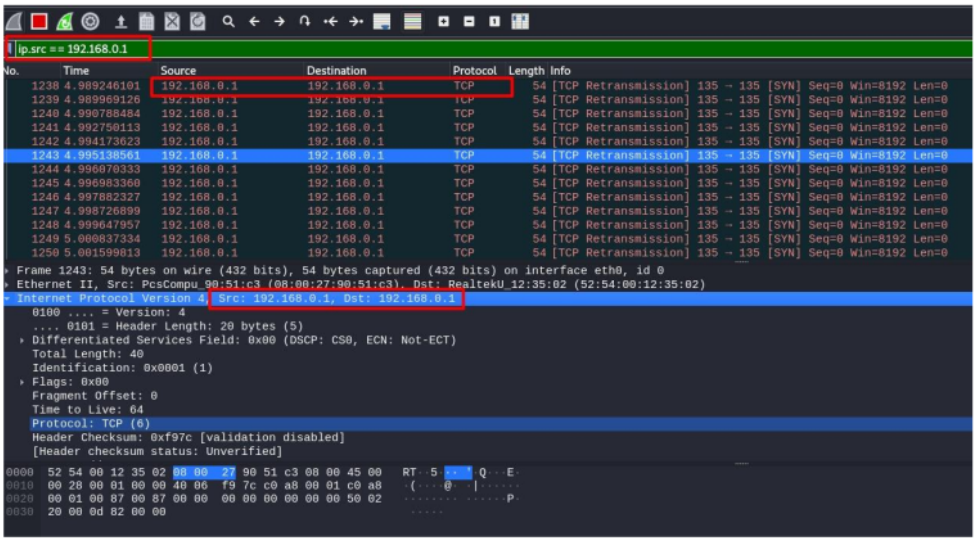

Figure 17: using the filter ip.src to show traffic related to specific IP in Wireshark

4.2 Experimental Environment

4.2.1 GNS3

Graphical Network Simulator-3 (shortened to GNS3) [46] is network software. It enables the employment of both virtual and physical devices to model complicated networks. Using Cisco IOS which is simulated using Dynamics emulation software.

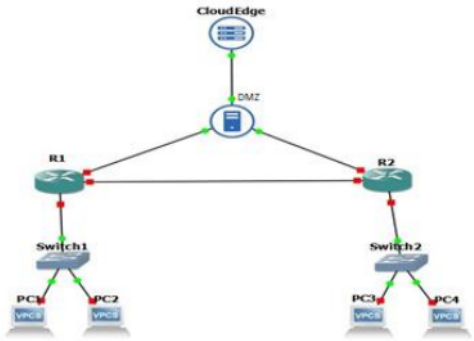

Figure 18: simulated environment created using GNS3

GNS3 consists of two software components:

- The GNS3-all-in-one software (GUI)
- The GNS3 virtual machine (VM)

You have a few options for the server part of the software:

1- Local GNS3 server
2- Local GNS3 VM
3- Remote GNS3 VM

Following figure 18 demonstrate the local GNS3 server runs locally on the same PC where we installed the GNS3 all-in-one software. We are using a Windows PC as an experimental device, both the GNS3 GUI and the local GNS3 server are running as processes in Windows.

Using the GNS3 VM is recommended, so we can run the GNS3 VM locally on our PC using virtualization software such as VMware Workstation Pro.

4.2.2 Hypervisor

A hypervisor, often known as a virtual machine monitor or VMM, is a piece of software that allows you to create and run virtual machines (VMs) [47] . A hypervisor allows a single host computer to handle numerous virtual machines (VMs) by sharing resources like memory and processing.

Types of hypervisors:

There are two types of hypervisors: "Type 1" (also known as "bare metal") and "Type 2" (also known as "hosted"). A type 1 hypervisor functions as a light operating system that runs directly on the host's hardware, but a type 2 hypervisor functions as a software overlay on top of an operating system, similar to other computer programs.

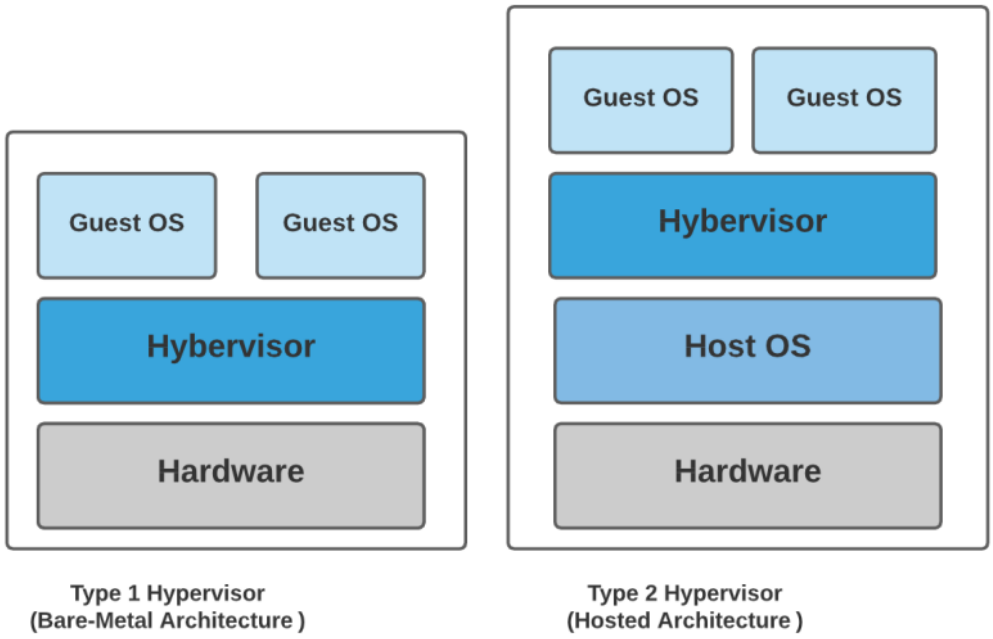

Figure 19: Types of hypervisors.

4.2.3 Cloud service models

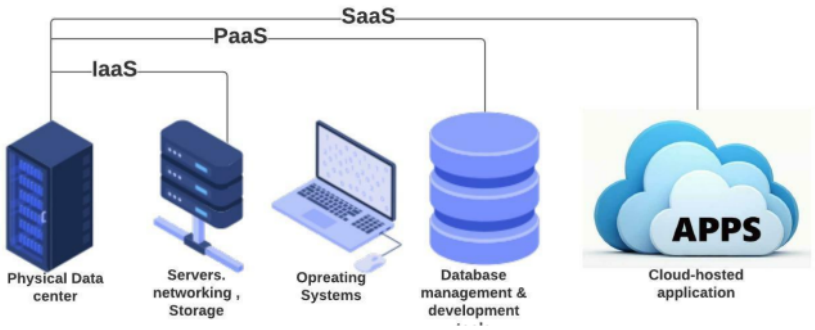

Figure 20: Cloud service models.

- **Software-as-a-Service (SaaS):**

SaaS applications are hosted on cloud servers and accessed via the Internet rather than being installed on the user's device. SaaS is similar to renting a home: the landlord takes care of the property, while the renter gets to utilize it as if it were their own. Salesforce, MailChimp, and Slack are examples of SaaS apps.

- **Platform-as-a-Service (PaaS):**

PaaS companies provide everything needed to construct an application via the Internet, including development tools, infrastructure, and operating systems. PaaS is comparable to renting all of the tools and equipment needed to build a house rather than the house itself. Heroku and Microsoft Azure are two instances of PaaS.

- **Infrastructure-as-a-Service (IaaS):**

A corporation rents the servers and storage it requires from a cloud provider in this approach. They then construct their applications on top of that cloud architecture. IaaS is similar to a firm renting a piece of land on which they can construct whatever they want, but they must provide their own construction equipment and supplies. Amazon EC2, Google Compute Engine, and Azure Virtual Machines are examples of IaaS providers.

4.3 Existed datasets

4.3.1 intrusion detection systems (IDS) Datasets

Machine learning based IDSs rely on available datasets for training, however finding a trustworthy dataset for comparison is problematic. A lack of sufficient documentation of the methodologies, a lack of comparison methodology, and a lack of crucial elements, such as ground-truth labels and publicly available and real-world environment traffic, are among the most well-known difficulties to compare between datasets. Furthermore, network traffic is mostly encrypted these days for communication security and privacy, and only a few databases represent this.

Figure 21 below gives different representation of generating and reframing the dataset one was using an online freely accessible dataset that include the desired main categories which will represent the quality of this dataset, another way of doing this is going through the road of simulating the whole process and providing a way to collect a synthetic network traffic.

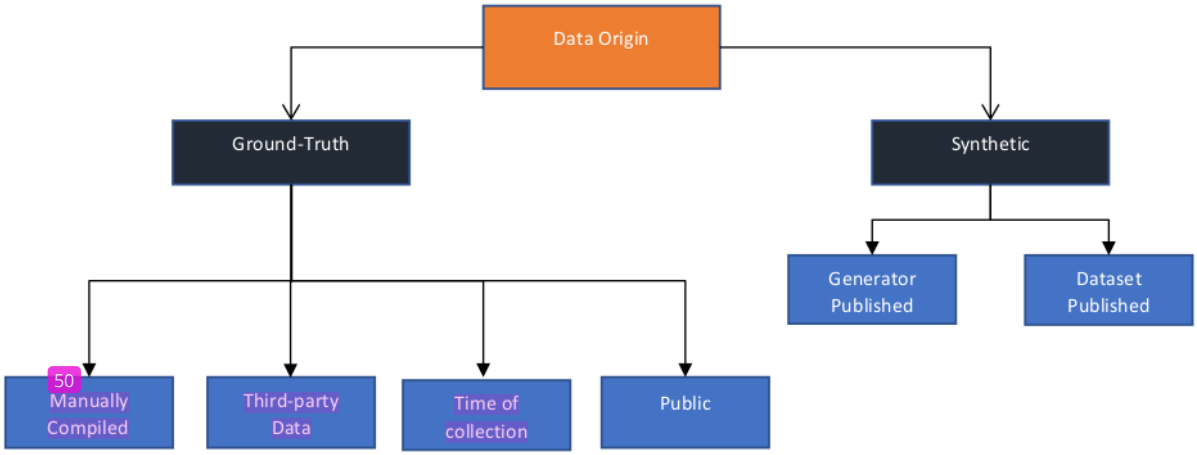

Figure 21: Dataset creation methodology

The dataset is crucial for developing machine learning-based IDS models. The procedure begins by capturing traffic from the internet, either as a packet or as a flow.

The collected traffic is then compiled into a specific type of data with network-related properties, such as tagging. The procedure of labeling is critical for the dataset. Handling ground-truth is difficult, especially when specialists are unable to determine whether the traffic is malicious or not.

The dataset is crucial for developing machine learning-based IDS models. The procedure begins by capturing traffic from the internet, either as a packet or as a flow. The collected traffic is then compiled into a specific type of data with network-related properties, such as tagging. The procedure of labeling is critical for the dataset. Handling ground-truth is difficult, especially when specialists are unable to determine whether the traffic is malicious or not.

This is one of the reasons why scientists employ synthetic traffic. However, this suggests that the generated traffic does not reflect the real-world situation. Concisely, the process of creating a dataset begins with traffic capture and finishes with preprocessing. A labeled dataset is the product of the preprocessing procedure. Each data point is assigned to one of two categories: harmful or benign. The file contains tabular data in a human-readable or binary format, such as a CSV file or an IDX file. The number of detected malicious or false alarms can be used to benchmark the dataset. Existing datasets lack the necessary encrypted traces and practicality to serve as the foundation for developing a complete model for detecting new threats. Most of the present encrypted traffic research focuses on several scopes, such as traffic classification and analysis. Even though such a study exists, the dataset is not publicly available due to the data's sensitivity. The following list of dataset represents some of the most practical real datasets that we have found during our research where they have met most of the dataset requirements that will be identified in the next sections.

4.3.2 HIKARI-2021

Publicly available up-to-date datasets to benchmark and compare among IDS are important, especially as the network traffic is changing over time. There are two main contributions of this paper. First, we made a new requirement for building new datasets which are lacking in the existing datasets, such as anonymization, payload, ground-truth, encryption, and a practical method to implement it. Anonymizing certain data will prevent privacy issues, while capturing with the payload will enrich the information that we can collect for detecting malicious traffic within encrypted traffic. Providing the ground-truth data is crucial, so no unlabeled attack is recorded in the dataset. The lack of Existing datasets with encrypted traffic, despite the fact that most modern traffic uses it to deliver assaults, has become a source of concern for us.

Second, they created a new IDS dataset called HIKARI-2021, which includes encrypted traces of network traffic. The datasets were created using a combination of ground-truth data that was not included in the existing IDS databases.

The datasets are freely available [48], Researchers have taken over 80 features from CICIDS-2017 and added more as a reference, including a source IP address (originh), source port (originp), destination IP address, and destination port. Each flow was classified as benign or attack, with benign having two categories (Benign or Background) and attack having four (Brute force, Brute force-XML, Probing, and XMRIGCC Crypto Miner). such as comprehensive capture, in which all traces (e.g., background traffic, benign, and attack) have been provided with pcap files. The payload is provided with the exception that we anonymize the background traffic, as anonymity is a requirement to safeguard privacy.

Based on the source IP address, source port, destination IP address, destination port, and protocol, the ground-truth and labeled are manually evaluated. This procedure ensures that the ground-truth is free of unlabeled attacks. Encryption is the final prerequisite. This is one of the most critical needs, as we know that unknown hostile traffic employs these attack vectors to carry out attacks. The second criterion is one of process. Its purpose is to ensure that researchers may construct their dataset by following the standards. It should be possible to obtain knowledge on how to create synthetic attacks as well as network configuration.

4.3.3 CICIDS-2017

2017 is the year of Runwick. CICIDS-2017 was an intrusion detection exercise that included multiple assault scenarios. The attack profiles in this dataset were created with publicly available tools and programs. Brute force, Heartbleed, botnet, DoS, DDoS, web attack, and infiltration attack were among the six attack profiles deployed. A B-Profile system [49] was used to produce realistic background traffic. The B-Profile system analyzed 25 user behaviors using a variety of protocols, including HTTP, HTTPS, FTP, SSH, and SMTP. CICFlowMeter [42] which retrieved 80 features from the pcap file, was used to record network traffic features. SourceIP, SourcePort, DestinationIP, DestinationPort, and Protocol were the flow labels. CICIDS-2017 is used to detect assaults such DoS attacks using feature reduction, web-attack detection and abnormal web traffic utilizing ensemble architecture and feature reduction. Others are working to improve the AdaBoost-based method by combining feature selection and information gain to locate relevant and meaningful features while also improving accuracy and execution speed.

4.3.4 KDD'99

UC Irvine's KDD'99 [50] (University of California, Irvine, 1998, 1999): The KDD Cup 1999 dataset was constructed by processing the tcpdump section of the 1998-DARPA dataset, which suffers from the same flaws as the 1998-DARPA dataset. Neptune-dos, pod-dos, smurf-dos, buffer-overflow, rootkit, satan, and teardrop are only a handful of the attacks included in KDD99. In a simulated scenario, normal and attack network traffic records are mixed, resulting in a dataset with a significant number of redundant records studded with data corruption, resulting in skewed testing results.

4.3.5 ISCX2021

ISCX2012 [37] (University of New Brunswick – 2012): The authors give a reasonable guideline for developing realistic and useful IDS evaluation datasets, and this dataset was developed using a dynamic approach. Their strategy is divided into two parts: an Alpha profile and a Beta profile. To stream the anomalous part of the dataset, the Alpha profile uses several multistage attack scenarios. The Beta traffic generator, on the other hand, simulates genuine network traffic with background noise. For the HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols, genuine traces are evaluated to produce profiles that generate real traffic.

4.4 Dataset Requirements

4.4.1 Content requirements

All the requirements are similar. However, none of the studies emphasized the significance of encrypted traffic in the dataset, despite the fact that this is one of our requirements [51]. We derived our dataset criteria based on the above works as well as a review of current datasets, which revealed that the dataset quality has the greatest impact on the NIDS system's outcome. The criteria were divided into two categories: content requirements and process requirements which focus on what is required to build a dataset, which focuses on comprehensive network traces and realistic network traffic capture. While this is insufficient, there is a lack of knowledge on how to create a new dataset that is both realistic and easy to execute.

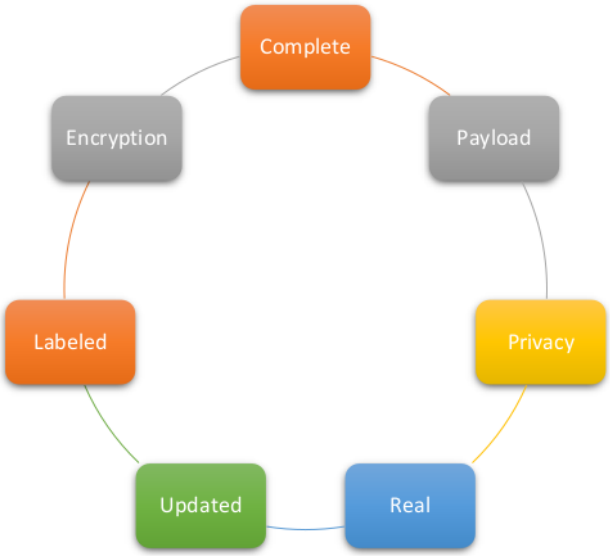

Figure 22: Content requirement.

The content requirements focus on the dataset's assets in order to accomplish a practical approach to create a dataset, whereas the process requirement defines the information on how the dataset is formed so that a new dataset can be created using the same process in the future. These prerequisites, as well as brief descriptions of each item, are listed below:

(1) **Complete capture:** captures all network traffic, including host-to-host communication, broadcast messages, domain lookup queries, and the protocol in use. The most significant aspect of comprehensive capture is having access to both flow data and pcap.

(2) **Payload:** A flow-based technique does not require payload. It is, nevertheless, critical to have comprehensive information and to get the most out of the data. HIKARI2021 is the only dataset that includes labeled encrypted traffic, whereas the other well-known datasets do not. It is possible that capturing the entire payload will be beneficial in the future.

(3) **Anonymity:** synthetic traffic should allow for full packet capture, whereas real traffic must anonymize particular packets in order to maintain privacy.

(4) **Ground-truth:** In comparison to synthetic traffic, the datasets should give actual traffic from a real production network, ensuring that there is no unlabeled attack in the ground-truth

(5) **Up to date:** by repeating the network traffic collecting operation, both packet traces from flow data and PCAP should always be accessible. Because data changes over time, repeating the methods ensures that the dataset always has the most up-to-date information.

(6) **Labeled dataset:** For accurate and trustworthy analysis, data must be accurately labeled as dangerous or benign. The flow with a combination of the source IP address, source port, destination IP address, destination port, and protocol determines the labeling procedure, which is a human effort.

(7) **Encryption Information:** information on how to determine if traffic is benign or malicious must be provided. We are concentrating on application layer attacks like brute force and probing, which leverage HTTPS and TLS version 1.2 to deliver the attacks.

4.4.2 Process Requirements

1- **Replicability** or public availability. Datasets should be open to the public or, if that is not possible, easily replicable. Many datasets, which were once freely available, are now impossible to access due to the creator's lack of upkeep. As a result, public release of a dataset is desired.

2- **Interoperability** is the second point to consider. Datasets should be shared in a standard, widely accepted format. The PCAP file format is the most often used for exchanging network packet captures.

3- **Quality** Issues and faults must be deliberately avoided or minimized in datasets. When examining the capabilities of NIDSs, high-quality datasets help to eliminate bias. We distinguish between challenges that occur from the fabrication of synthetic traffic and those that arise from the manufacture of real traffic.

5 Proposed Methodology

5.1 Network configuration for generating the dataset

Referring to the following figure23 one approach we can take in order to do the data collection and build our dataset is to gather all data from a built on simulated environment that will consist of two main nodes one for the attacker side where the attacks will be lunched and the other from the victim perspective where we will simulate the whole process and actions of normal users that may be connected in the network , the data collection process will be performed on different timetables which will allow us to run the attack on specific day to focus on one single category from the attacker profile such as running Slowloris and then for another day switch the focus on lunching different attack like brute forcing where we will implement the attack using a high loaded list of credentials. Another phase where we just focus on the normal benign side of the network where the traffic will just consist of the normal usual behavior of users. The final dimension would be a full collection of whatever type of traffic that may go through the network. Below is a full detailed explanation of each phase or profile that will be implemented.

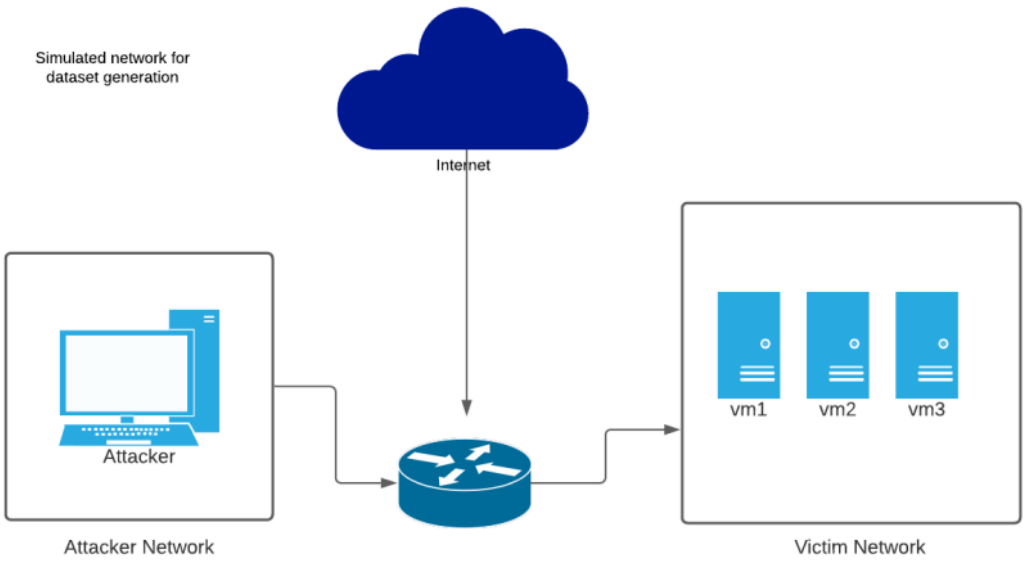

Figure 23: Simulated network for dataset generation

5.1.1 Background phase

It is critical to collect data that is realistic. In the victim network, we will capture all background traffic without using any filters or firewalls. As a result, there is a chance that background traffic contains malicious traffic or assaults. We will anonymize several pieces of information, such as the IP address and the payload, to protect privacy without compromising the analysis results.

5.1.2 Attacker phase

We will launch the attack on different virtual machines in different time periods. Our methodology will start by creating a customized python script that will mimic a brute force attack where we guarantee the attack is delivered successfully to the desired web page in the victim machine. Also, one of the most effective tools that could be used for brute forcing is Hydra which will run using an available list of dictionaries. Another dimension that could be established to launch another attack in the system is jumping to one of the devastating network attacks which is a Denial-of-services attack that will crush the machines by flooding the network with high volume of network traffic where the tenant will not be able to handle it. In our methodology we have chosen multiple unique and famous types of DOS attacks: Slowloris, syn flood, land attack, and UDP flood.

5.1.3 Benign Phase

We investigated employing a benign profile that was like human behavior to construct the benign profile. Where two headless browsers run: Google Chrome and Mozilla Firefox, to do this. These two browsers imitate people by visiting random links from a variety of websites, registering as a user, logging in, posting an article to the target victim's server, and then logging out. For each sequence of actions, we employed numerous variables, such as user-agent and random delay, to behave like a human and avoid being detected as a bot or web spider. A Python script was used to create a benign profile, which simulates benign traffic. All benign traffic is going to be recorded without the payload being anonymized. HTTPS traffic is the sole type of traffic generated.

Finally, all PCAP files should be collected and then resume with the various stages of machine learning pre-processing steps to create fully labeled processed Dataset, accordingly, extract the features to csv format, elects the best normalization method, binarize the whole string values so the dataset could be used to train the machine learning model.

5.2 Dataset Pre-processing

Because ground truth data is frequently imperfect and incompatible in certain aspects, the data preprocessing phase is critical for transforming raw data into a usable format. It reduces the amount of uncertainty in the dataset utilized and reveals the detection system's accurate statistics. The data preprocessing phase varies from research to study and data to data. Data preprocessing [52] is described in this section through sub-sections such as feature selection, duplicate removal, class imbalance, normalizing, and label encoding. Steps of pre-processing phase are displayed in

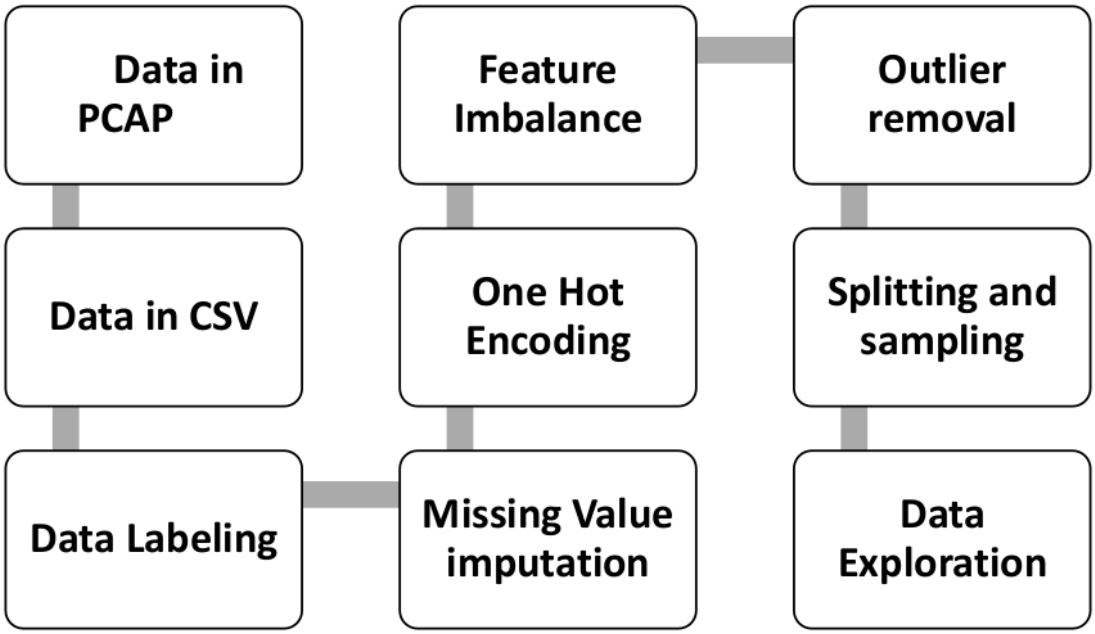

Figure 24: Data pre-processing steps

5.2.1 Labeling process

Labeling follows the principle of attaching information to some object. In data-centered disciplines labeling is often associated with querying knowledge of users about data objects. As such, the labeling process represents an essential prerequisite for algorithmic support in data mining, machine learning, and visual analytics. Two goals of almost any labeling process are being accurate and fast, i.e., effective, and efficient. In the machine learning community, labeling traditionally represents the basis for the creation of large ground truth data sets. Ground truth is necessary to enable autonomous supervised learning. The most recent and powerful supervised machine learning approaches, such as deep neural networks require large amounts of such labeled data to learn successfully [53]

5.2.2 Features Extraction

Our proposed way is using Nfstreamer [43] where it will be used to extract characteristics. The following Table represents some of the characteristics, whereas Figure 25 depicts a statistical summary of the characteristics

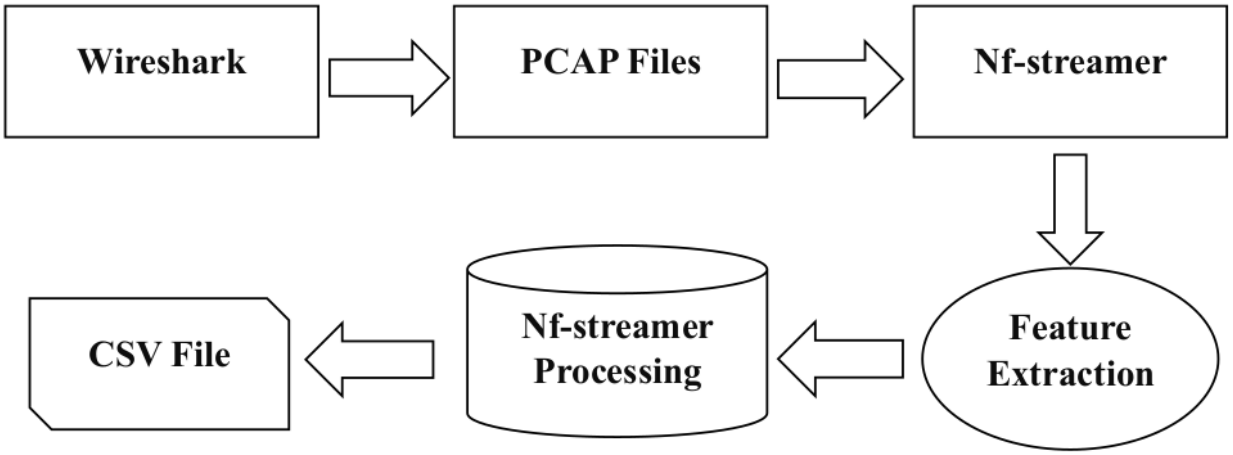

Figure 25: Feature extraction process.

5.2.3 Normalization

Normalization [54] is a procedure modifying the values in the feature vector so that they may be measured on a common scale. Many distinct types of normalization are used in machine learning. The goal of some of the most popular types of normalization is to have the values add up to one. L1 normalization, or Least Absolute Deviations, ensures that the sum of absolute values in each row is equal to one. L2 normalization, also known as least squares, ensures that the sum of squares is one. The L1 normalization approach is more robust than the L2 normalizing technique in general. Because it is resistant to outliers in the data, the L1 normalization procedure is robust.

5.2.4 Label encoding

Labeling follows the principle of attaching information to some object. In data-centered disciplines labeling is often associated with querying knowledge of users about data objects. As such, the labeling process represents an essential prerequisite for algorithmic support in data mining, machine learning, and visual analytics. Two goals of almost any labeling process are being accurate and fast, i.e., effective, and efficient. In the machine learning community, labeling traditionally represents the basis for the creation of large ground truth data sets. Ground truth is necessary to enable autonomous supervised learning. The most recent and powerful supervised machine learning approaches, such as deep neural networks require enormous amounts of such labeled data to learn successfully [53]

5.2.5 Class Imbalance

A classifier is only as good as the data that is used for training. One of the most common problems we face in the real world is the quality of data. For a classifier to perform well, it needs to see an equal number of points for each class [55]. But when we collect data in the real world, it's not always possible to ensure that each class has the exact same number of data points. If one class has 10 times the number of data points of the other class, then the classifier tends to get biased towards the first class.

5.3 ML / DNN Classification Models & Algorithms

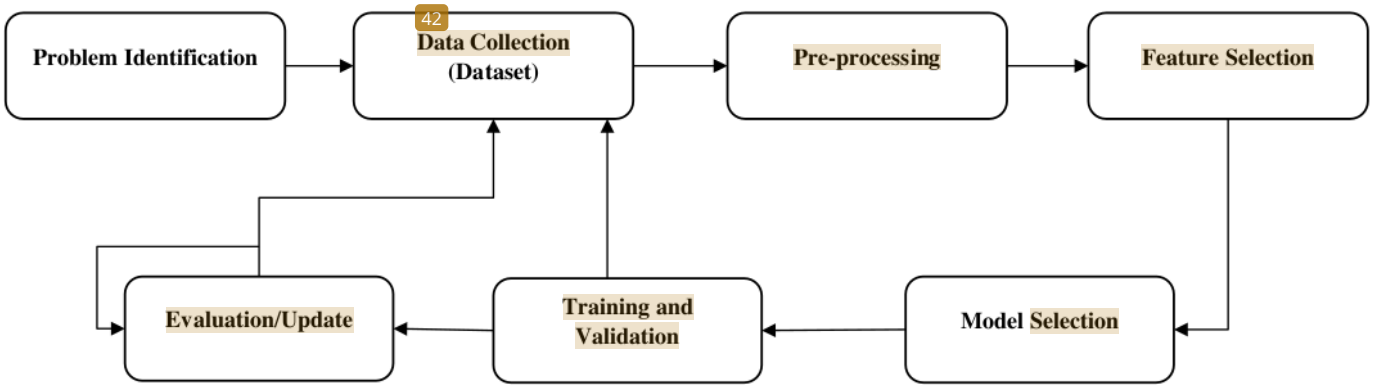

Figure 26: ML Model creation process.

Above figure 26 demonstrate the whole process of creating a highly effective and practical model that provide the best rate of accuracy as possible starting with define the overall problem and data collection and dataset traffic generating will be conducted according to that specification of the problem then preparing the dataset with different of actions starting from the pre-processing phase to the feature selection and extraction and finally training and evaluating the model with periodic evaluation methodology to ensure effectiveness and highest level of accuracy, below is a following list of some unique classifiers that have been used in a similar network traffic classification problems where they give the highest accuracy during the both training and test phase of the training and validating the dataset :

5.3.1 K-nearest Neighbors (KNN)

The KNN algorithm is based on the concept of near proximity, which is defined as the distance between two points on a space. The Euclidean distance is used to determine proximity in Euclidean space.

A parameter K determines the boundaries of each class, and as K values increase, the boundaries become smoother. Because the training and validation error rates would be significantly different at different K-values, the experiments must be conducted for a variety of K-values. The training and validation datasets must be separated from the starting dataset in order to find the optimal value of K. The K-value must be increased and iterated until the projected class does not change.

Chebyshev, cosine similarity, and Euclidean Distance are some of the various distance metrics that can be used with this approach.

5.3.2 Support Vector Machine (SVM)

Data points are represented as vertices in an n-dimensional space in the SVM, with 'n' denoting the total number of extracted features. Each feature's magnitude is given by the position of a certain coordinate [20]. Estimating the ideal hyperplane, which effectively differentiates the two groups, is used to group features into different categories or classes. In a high-dimensional space, a hyperplane is a border that divides data points into their corresponding or specialized classes. There are data points on both sides of the hyperplane that can be assigned to different classes.

5.3.3 Decision Tree (DT)

DT is an excellent strategy for defining a path to scan a dataset and also for defining a tree-like path to the expected outcomes [22]. A feature selection process is a method for determining categorization criteria that is based on empirical evidence. The data is partitioned using an attribute selection measure, which divides the data into different classes. Because they indicate the split criterion on a DT, these are considered splitting rules.

The root node for the assumed datasets is the feature with the highest score in relation to the provided selection criteria.

Information Gain, Gain Ratio, and Gini Index are some of the different attribute selection measures. The splitting attribute Information Gain is defined in terms of the volume of information necessary to define the tree and reduces the information required to classify the data points. These criteria project the partitions with the least amount of arbitrariness.

5.3.4 Adaboost

AdaBoost (Adaptive Boosting) is a popular improvement strategy in which numerous weak classifiers are combined to create a single strong classifier [25]. A single classifier cannot accurately determine the class to which an object belongs in the actual world. As a result, numerous weak classifiers are grouped together, and each classifier learns from the incorrectly categorized instances in order to build a successful and strong classifier.

5.3.5 Random Forest

Random forest (RF) is a learning technique based on ensemble trees. It consists of a set of decision trees derived from a randomly selected portion of the training dataset. It combines the votes from several decision trees to arrive at a consensus on the test data's ultimate classification. It creates decision trees for each random data sample and then uses them to make predictions. Following that, the best answer is determined by a vote. It combines a large number of DTs with a unique set of observations, and the splitting nodes are determined by a variety of factors. In addition, the RF's ultimate predicting is done by combining the projections of each individual tree.

5.3.6 Naïve bayes

The supervised classification algorithms built from Bayes' Theorem [26] are known as naive Bayes classifiers. It calculates the chance of an event occurring when it is compared to the probability of another event occurring previously. All these classifiers are assessed using assessment metrics detailed in the "Results and Discussions" section. For training and evaluating the data (i.e., classify the attacks), the best classifier with good evaluation metrics is picked. The trained model is saved for classifying fresh data into the indicated attacks and non-attacks, hence preventing infiltration, and ensuring system security. In this way, the properties of both Deep Packet inspection and Machine Learning are used effectively to eliminate network attacks.

5.4 Classification using Machine learning models

5.4.1 Binary Classification

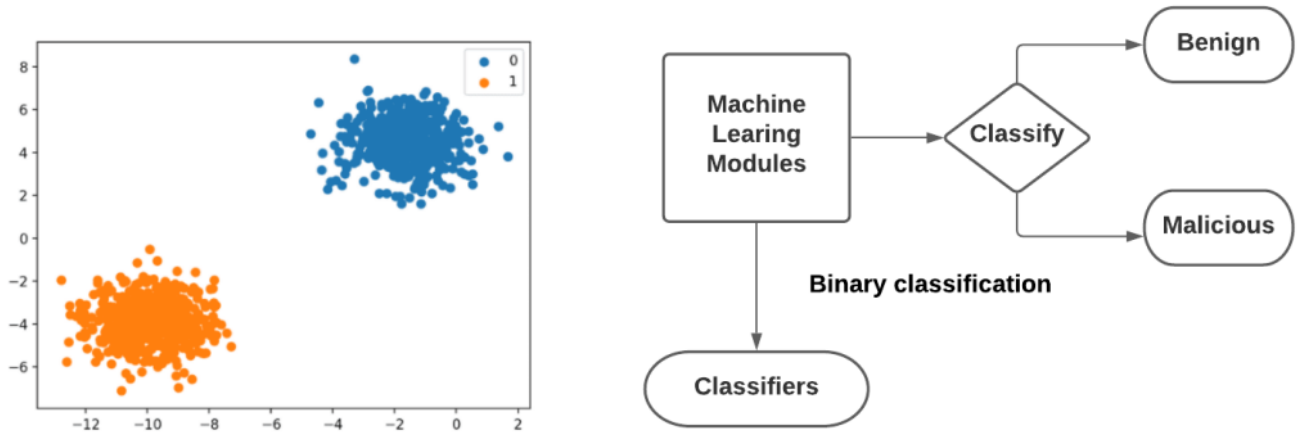

Figure 27: Binary classification.

In contacts of classifying Network traffic using machine learning our approach is going to be implemented in two different levels as shown in the figure 27 , our system will start by capturing the network packets and do a first level classification where the traffic will labeled into benign and malicious traffic and then if the network packets has been identified as malicious the classifier will go to the next level where the packet will be classified into different categories as shown in figure 27.

Classification problems with two class labels are referred to as binary classification.

In most binary classification problems, one class represents the normal state and the other represents the aberrant state.

The normal condition is "not spam," while the abnormal state is "spam". For Example, in medical perspective "Cancer not detected" is the normal condition of a task involving a cancer medical test, whereas "cancer identified" is the abnormal state. The normal state class is assigned the class label 0, while the abnormal state class is awarded the class label 1.

The Bernoulli distribution is a discrete probability distribution that describes a situation in which an event has a binary result of 0 or 1. This signifies that the model forecasts the likelihood of an example falling into class 0, or the abnormal state.

Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machines, and Naive Bayes are examples of popular binary classification techniques.

Some algorithms, such as Logistic Regression and Accept Vector Machines, are built primarily for binary classification and do not support more than two classes by default.

5.4.2 Proposed ML/DL Classification Pipeline

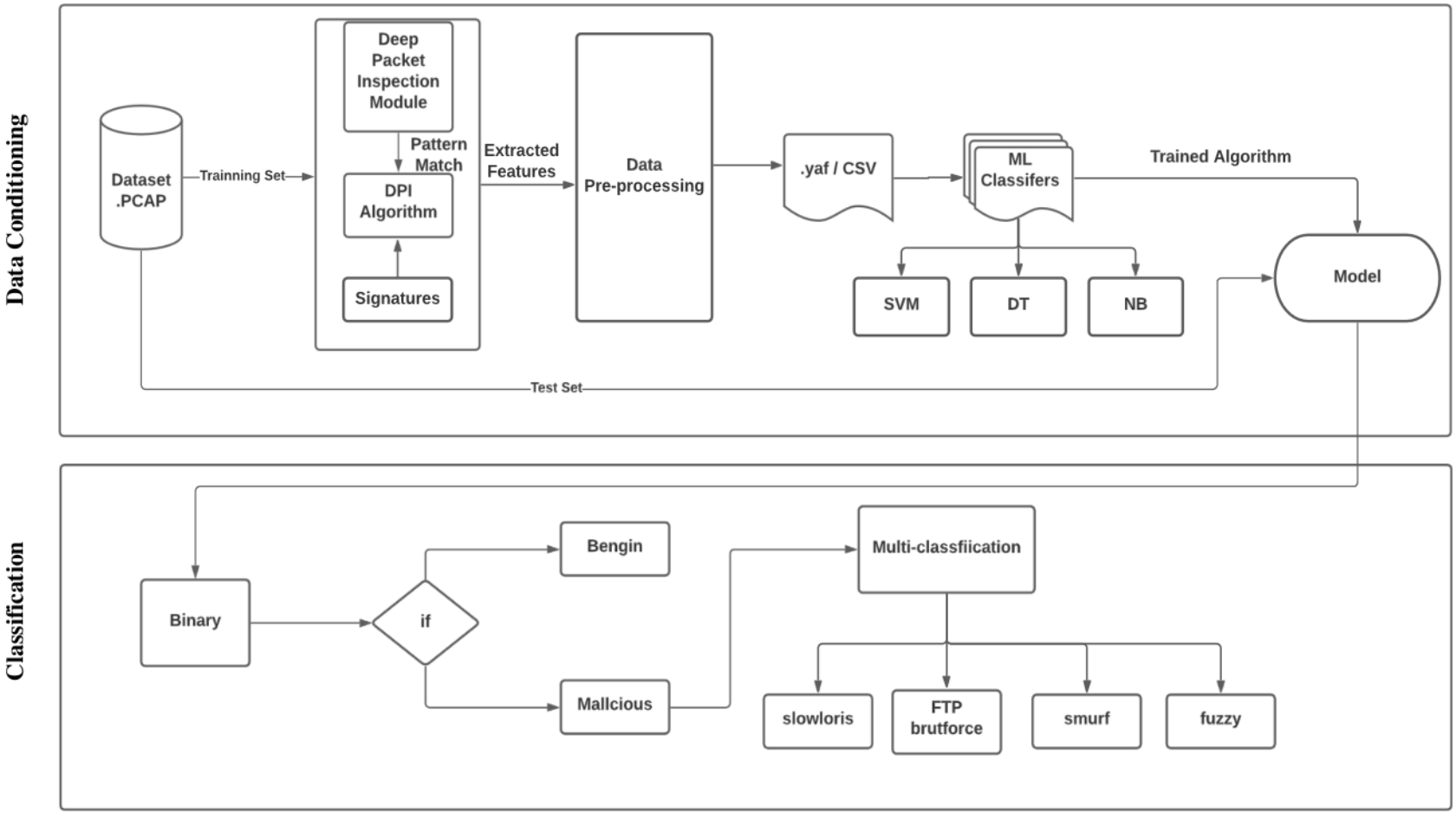

Figure 28 : Full project system Design

5.4.3 Multi-classification Methodology

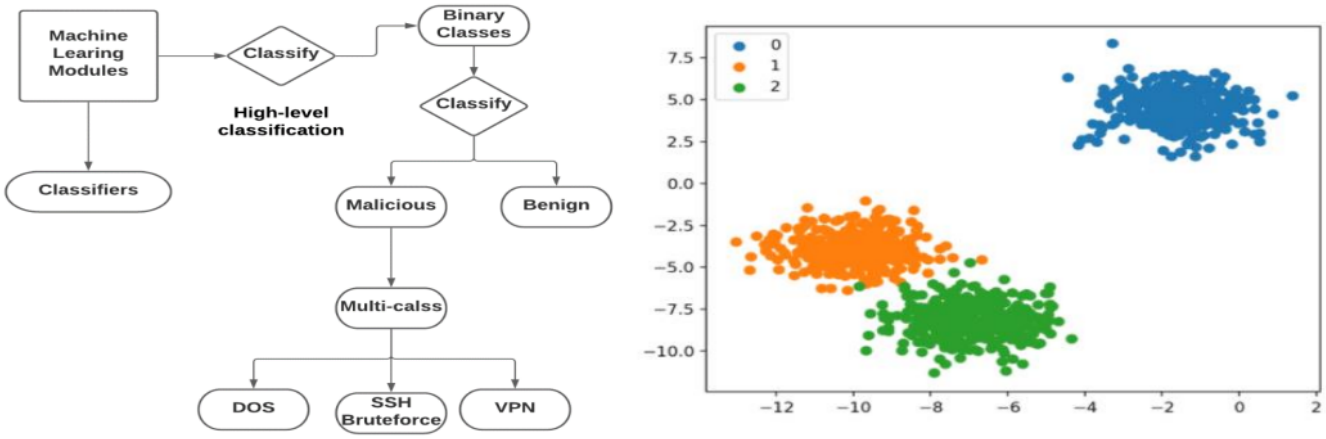

Figure 29: Multiclass classification

Classification jobs with more than two class labels are referred to as multi-class classification. Multi-class classification, unlike binary classification, does not distinguish between normal and abnormal results. Instead, examples are assigned to one of a number of pre-defined classes.

On some situations, the number of class labels can be quite big. In a facial recognition system, for example, a model might predict that a shot belongs to one of thousands or tens of thousands of faces. Text translation models, for example, are a sort of multi-class classification problem that involves predicting a series of words.

Each word in the sequence of words to be predicted is classified into many classes, with the size of the vocabulary determining the number of classes that can be predicted, which might be tens or hundreds of thousands of words. Some examples of popular multi-class classification methods are the flowing list of classifiers:

- k-Nearest Neighbors
- Decision Trees
- Naive Bayes
- Random Forest
- Gradient Boosting

Some Algorithms that are designed for binary classification can be adapted for use for multi-class problems as will.

5.4.4 Using DNN model to Train an Existed Dataset

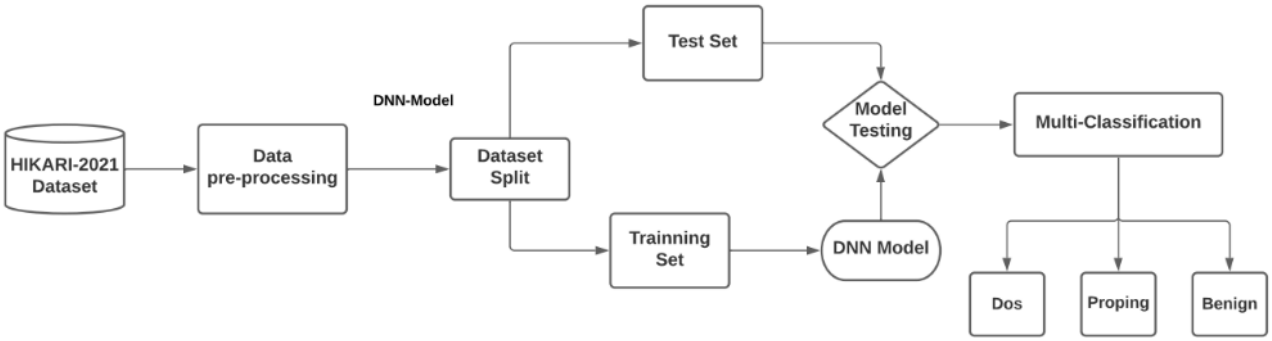

Figure 30: DNN-Model.

DNN models:

Deep learning (DL) is a subtype of ML that comprises multiple hidden layers to obtain the deep network's properties. Due to their deep structure and capacity to learn the key features from the dataset on their own and provide an output, these algorithms are more efficient than ML.

Jupyter notebook, Python programming environment via TensorFlow, and Keras library were used in the implementation process. All datasets are divided into two files for binary and multi-class classification after data preprocessing. This study's Deep learning approach is depicted in Figure 30. The dataset is first preprocessed so that the model may be applied extensively to it.

5.4.5 Packet Drop

In order to create a secure patching schema for the proposed system; after the proposed classification method finishes the system will provide a final report to the user with all of the available security flows and cyber network attacks that are targeting the production environment. where the guardian would be placed in a specific node in order to cover all the ingress and egress points in the network; whenever an accurate report is generated the final stage will conduct a rule-based approach or following the pattern of the classified instances the system will be able to drop the malicious packets and prevent any misuse or any cyber-attack that the ML classification algorithms has been trained on from entering the target environment

6 CONCLUSION

In conclusion, this research described a new methodology to detect malicious traffic in real time using machine learning classification methodology. We have shown some related studies and proposed online datasets that we could use in order to train our ML module. The research also presented our plan to generate our own dataset which includes malicious and normal traffic generated by using different tools that have been illustrated. Also, both DPI and firewall or IDS solutions would be developed in parallel to improve the proposed system's security and computing efficiency.

7 References

[1] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," *Appl. Sci.*, vol. 11, no. 17, Sep. 2021, doi: 10.3390/app11177868.

[2] Zeek, "ZEEK IDS." .

[3] S. Abt and H. Baier, "Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research," in *Proceedings - 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2014*, Apr. 2016, pp. 40–55, doi: 10.1109/BADGERS.2014.11.

[4] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *J. Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00475-1.

[5] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Comput. Secur.*, vol. 88, 2020, doi: 10.1016/j.cose.2019.101646.

[6] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," Oct. 2018, doi: 10.1109/MCOM.2019.1800819.

[7] I. D. Khalil and A. Dayton, "APPLICATION-BASED NETWORK TRAFFIC GENERATOR FOR NETWORKING AI MODEL DEVELOPMENT," 2021.

[8] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On Generating Network Traffic Datasets with Synthetic Attacks for Intrusion Detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 2, Feb. 2021, doi: 10.1145/3424155.

[9] X. Fu, C. Zhang, J. Chen, L. Zhang, and L. Qiao, "Network traffic based virtual machine migration in cloud computing environment," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 818–821, 2019, doi: 10.1109/ITNEC.2019.8729184.

[10] T. Bonald, "The erlang model with non-poisson call arrivals," *Perform. Eval. Rev.*, vol. 34, no. 1, pp. 276–286, 2006, doi: 10.1145/1140103.1140309.

[11] J. Sommers, H. Kim, and P. Barford, "Harpoon: A flow-level traffic generator for router and

network tests," *Perform. Eval. Rev.*, vol. 32, no. 1, pp. 392–393, 2004, doi: 10.1145/1012888.1005733.

[12] Iperf, "The iperf TCP/UDP Bandwidth Measurement Tool [EB/OL].," 2005. http://dast.nlanr.net/Projects/Iperf.

[13] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith, "Tmix: A tool for generating realistic TCP application workloads in ns-2," *Comput. Commun. Rev.*, vol. 36, no. 3, pp. 65–76, 2006, doi: 10.1145/1140086.1140094.

[14] S. Flood, "TCP SYN Flood."

[15] "LAND Attack."

[16] imperva, "LAND attack."

[17] J. V Bibalbenifa, S. Krishnann, H. Long, R. Kumar, D. Taniar, and H. V. Long, "Performance Analysis of Machine Learning and Pattern Matching Techniques for Deep Packet Inspection in Firewalls," 2021, doi: 10.21203/rs.3.rs-260788/v1.

[18] C. Parsons, "Deep Packet Inspection in Perspective: Tracing its lineage and surveillance Potentials," *New Transpar. Surveill. Soc. Sorting*, vol. 1, pp. 1–16, 2009, [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Deep+Packet+Inspection+in+Perspective+:+Tracing+its+lineage+and+surveillance+potentials#0.

[19] L. Deri and D. Sartiano, "Using DPI and Statistical Analysis in Encrypted Network Traffic Monitoring," *Int. J. Inf. Secur. Res.*, vol. 10, no. 1, pp. 932–943, 2020, doi: 10.20533/ijisr.2042.4639.2020.0107.

[20] nDPI, "nDPI tool." https://www.ntop.org/products/deep-packet-inspection/ndpi/.

[21] N. Github, "nDPI." https://github.com/ntop/nDPI.

[22] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "nDPI : Open-Source High-Speed Deep Packet Inspection," pp. 617–622, 2014.

[23] N. V. Verde, G. Ateniese, E. Gabrielli, L. V. Mancini, and A. Spognardi, "No NAT'd user left behind: Fingerprinting users behind NAT from NetFlow records alone," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 218–227, 2014, doi: 10.1109/ICDCS.2014.30.

[24] P. D. Smith, "On tuning the boyer-moore-horspool string searching algorithm," *Softw. Pract. Exp.*, vol. 24, no. 4, pp. 435–436, 1994, doi: 10.1002/spe.4380240408.

[25] D. Pao, W. Lin, and B. Liu, "A memory-efficient pipelined implementation of the aho-corasick string-matching algorithm," *Trans. Archit. Code Optim.*, vol. 7, no. 2, 2010, doi: 10.1145/1839667.1839672.

[26] T. AbuHmed, A. Mohaisen, and D. Nyang, "A Survey on Deep Packet Inspection for Intrusion Detection Systems," no. April 2013, 2008, [Online]. Available: http://arxiv.org/abs/0803.0037.

[27] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[28] R. Velea, C. Ciobanu, F. Gurzau, and V. V. Patriciu, "Feature Extraction and Visualization for Network PcapNg Traces," in *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017*, Jul. 2017, pp. 311–316, doi: 10.1109/CSCS.2017.49.

[29] C. S. Wickramasinghe, K. Amarasinghe, D. L. Marino, C. Rieger, and M. Manic, "Explainable Unsupervised Machine Learning for Cyber-Physical Systems," 2021. doi: 10.1109/ACCESS.2021.3112397.

[30] P. Biondi, "Scapy Packet Crafting Tool for python 2 and python 3." https://scapy.net/ (accessed Mar. 01, 2022).

[31] S. P., "Ostinato is versatile packet crafter, pcap player and traffic generator with an intuitive GUI." https://ostinato.org/.

[32] Nmap, "Nmap." https://nmap.org/.

[33] Hping3, "Hping3." https://www.kali.org/tools/hping3/.

[34] ID2T, "ID2T." https://github.com/tklab-tud/ID2T.

[35] C. G. Cordero, E. Vasilomanolakis, N. Milanov, C. Koch, D. Hausheer, and M. Muhlhauser, "ID2T: A DIY dataset creation toolkit for Intrusion Detection Systems," *2015 IEEE Conf. Commun. NetworkSecurity, CNS 2015*, pp. 739–740, 2015, doi: 10.1109/CNS.2015.7346912.

[36] D. Brauckhoff and M. May, "Brauckhoff, Wagner, May - 2008 - FLAME A Flow-Level Anomaly Modeling Engine."

[37] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012, doi: 10.1016/j.cose.2011.12.012.

[38] ISCXIDS2012, “ISCXIDS2012.” https://www.unb.ca/cic/datasets/ids.html.

[39] INSecS-DCS:, “INSecS-DCS:” https://github.com/nrajasin/Network-intrusion-dataset-creator.

[40] N. Rajasinghe, J. Samarabandu, and X. Wang, “INSECS-DCS: A Highly Customizable Network Intrusion Dataset Creation Framework,” *Can. Conf. Electr. Comput. Eng.*, vol. 2018-May, pp. 1–4, 2018, doi: 10.1109/CCECE.2018.8447661.

[41] weka tool, “WEKA.” https://www.cs.waikato.ac.nz/ml/weka/.

[42] CIC-Flowmeter, “Feature Extraction Tool.” https://www.unb.ca/cic/research/applications.html.

[43] NFStream, “Flexible Network Data Analysis Framework for feature extractrion.” https://www.nfstream.org/.

[44] Wireshark, “Network capturing and analyzer tool.” https://www.wireshark.org/ (accessed Mar. 01, 2022).

[45] Tshark, “Command line tool for wireshark.” https://www.wireshark.org/docs/man-pages/tshark.html.

[46] Gns3, “graphical network simulator,” [Online]. Available: https://www.gns3.com/.

[47] A. Desai, R. Oza, P. Sharma, and B. Patel, “Hypervisor: A survey on concepts and taxonomy,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 2, no. 3, pp. 222–225, 2013, [Online]. Available: http://msdn.microsoft.com/en-us/library/cc768520.

[48] J. Ferriyan, Andrey; Thamrin, Achmad Husni; Takeda, Keiji; Murai, “Available datasets from the paper Generating Encrypted Network Traffic for Intrusion Detection Datasets-HIKARI-2021.” https://zenodo.org/record/5199540#.YdMUk2BBy3A.

[49] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, vol. 2018-Janua, pp. 108–116, doi: 10.5220/0006639801080116.

[50] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set in Computational Intelligence for Security and Defense Applications,” *Comput. Intell. Secur. Def. Appl.*, no. Cisda, pp. 1–6, 2009.

[51] E. Vasilomanolakis, C. G. Cordero, N. Milanov, and M. Mühlhäuser, “Towards the creation of synthetic, yet realistic, intrusion detection datasets,” *Proc. NOMS 2016 - 2016 IEEE/IFIP Netw.*

Oper. Manag. Symp., pp. 1209–1214, 2016, doi: 10.1109/NOMS.2016.7502989.

[52] B. Alothman, "Raw network traffic data preprocessing and preparation for automatic analysis," *2019 Int. Conf. Cyber Secur. Prot. Digit. Serv. Cyber Secur. 2019*, pp. 1–5, 2019, doi: 10.1109/CyberSecPODS.2019.8885333.

[53] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair, "Comparing Visual-Interactive Labeling with Active Learning: An Experimental Study," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 298–308, 2018, doi: 10.1109/TVCG.2017.2744818.

[54] P. Joshi, "Normalization," in *Artificial Intelligence with Python*, 2nd ed., 2017.

[55] Class Imbalance, "Class imbalance," in *Artificial Intelligence with Python*, 2017.

ORIGINALITY REPORT

40%	31%	17%	17%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	7%
2	assets.researchsquare.com Internet Source	2%
3	eprints.cs.univie.ac.at Internet Source	2%
4	mafiadoc.com Internet Source	1%
5	Carlos Garcia Cordero, Emmanouil Vasilomanolakis, Aidmar Wainakh, Max Mühlhäuser, Simin Nadjm-Tehrani. "On Generating Network Traffic Datasets with Synthetic Attacks for Intrusion Detection", ACM Transactions on Privacy and Security, 2021 Publication	1%
6	docplayer.net Internet Source	1%
7	www.freecodecamp.org Internet Source	1%

8	timmccloud.net Internet Source	1 %
9	Submitted to CSU, San Jose State University Student Paper	1 %
10	Submitted to The Robert Gordon University Student Paper	1 %
11	Submitted to University of Wales, Lampeter Student Paper	1 %
12	Wen Ouyang, , Xu Zhang, Dongbin Wang, Junhui Zhang, and Jiqiang Tang. "A survey of network traffic generation", Third International Conference on Cyberspace Technology (CCT 2015), 2015. Publication	1 %
13	Nour Alqudah, Qussai Yaseen. "Machine Learning for Traffic Analysis: A Review", Procedia Computer Science, 2020 Publication	1 %
14	deepchecks.com Internet Source	1 %
15	www.riverpublishers.com Internet Source	1 %
16	www.ijstr.org Internet Source	1 %
17	Submitted to Intercollege Student Paper	

		1 %
18	doi.org Internet Source	1 %
19	www.varonis.com Internet Source	<1 %
20	Submitted to Melbourne Institute of Technology Student Paper	<1 %
21	Submitted to University of the West Indies Student Paper	<1 %
22	Submitted to Griffith College Dublin Student Paper	<1 %
23	Zied Aouini, Adrian Pekar. "NFStream", Computer Networks, 2022 Publication	<1 %
24	www.imperva.com Internet Source	<1 %
25	www.unb.ca Internet Source	<1 %
26	vbn.aau.dk Internet Source	<1 %
27	Shi Dong, Mudar Sarem. "DDoS Attack Detection Method Based on Improved KNN	<1 %

With the Degree of DDoS Attack in Software-Defined Networks", IEEE Access, 2020
Publication

28	Kaniz Farhana, Maqsudur Rahman, Md. Tofael Ahmed. "An intrusion detection system for packet and flow based networks using deep neural network approach", International Journal of Electrical and Computer Engineering (IJECE), 2020 Publication	<1 %
29	Submitted to University of Leicester Student Paper	<1 %
30	Chathurika S. Wickramasinghe, Kasun Amarasinghe, Daniel L. Marino, Craig Rieger, Milos Manic. "Explainable Unsupervised Machine Learning for Cyber-Physical Systems", IEEE Access, 2021 Publication	<1 %
31	Submitted to Liverpool John Moores University Student Paper	<1 %
32	linuxhint.com Internet Source	<1 %
33	Dionisios Pnevmatikatos. "A Memory-Efficient Reconfigurable Aho-Corasick FSM Implementation for Intrusion Detection Systems", 2007 International Conference on	<1 %

	Embedded Computer Systems Architectures Modeling and Simulation, 07/2007 Publication	
34	ostinato.org Internet Source	<1 %
35	Submitted to Bowie State University Student Paper	<1 %
36	Kushal Rashmikant Dalal. "Analysing the Role of Supervised and Unsupervised Machine Learning in IoT", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020 Publication	<1 %
37	Submitted to University of Bolton Student Paper	<1 %
38	Submitted to University of Greenwich Student Paper	<1 %
39	machinelearningmastery.com Internet Source	<1 %
40	Arun Kumar Sangaiah, Jaya Subalakshmi Ramamoorthi, Joel J. P. C. Rodrigues, Md. Abdur Rahman et al. "LACCVoV: Linear Adaptive Congestion Control With Optimization of Data Dissemination Model in Vehicle-to-Vehicle Communication", IEEE	<1 %

	Transactions on Intelligent Transportation Systems, 2020 Publication	
41	Submitted to National College of Ireland Student Paper	<1 %
42	arxiv.org Internet Source	<1 %
43	curve.carleton.ca Internet Source	<1 %
44	Submitted to Kaplan College Student Paper	<1 %
45	explore.openaire.eu Internet Source	<1 %
46	Submitted to Emirates Aviation College, Aerospace & Academic Studies Student Paper	<1 %
47	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1 %
48	www.marcuscurtismusic.com Internet Source	<1 %
49	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %

50	Sebastian Abt, Harald Baier. "Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research", 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2014 Publication	<1 %
51	Submitted to Somerset College of Arts and Technology, Somerset Student Paper	<1 %
52	Submitted to University of Leeds Student Paper	<1 %
53	community.nxp.com Internet Source	<1 %
54	dspace.univ-ouargla.dz Internet Source	<1 %
55	Submitted to Birkbeck College Student Paper	<1 %
56	journalofbigdata.springeropen.com Internet Source	<1 %
57	Submitted to Aston University Student Paper	<1 %
58	Submitted to Bradford College, West Yorkshire Student Paper	<1 %

59	artificialneuralnetworks.org Internet Source	<1 %
60	www.220-801.net Internet Source	<1 %
61	Submitted to Sri Lanka Technological Campus Student Paper	<1 %
62	Submitted to The NorthCap University, Gurugram Student Paper	<1 %
63	Submitted to Free University of Bolzano Student Paper	<1 %
64	Submitted to Southampton Solent University Student Paper	<1 %
65	deepai.org Internet Source	<1 %
66	library.samdu.uz Internet Source	<1 %
67	Submitted to NCC Education Student Paper	<1 %
68	www.radware.com Internet Source	<1 %
69	Submitted to Loughborough University Student Paper	<1 %
70	www.computingonline.net	

	Internet Source	<1%
71	Amirhossein Gharib, Iman Sharafaldin, Arash Habibi Lashkari, Ali A. Ghorbani. "An Evaluation Framework for Intrusion Detection Dataset", 2016 International Conference on Information Science and Security (ICISS), 2016 Publication	<1%
72	Nguyen Huy Trung, Le Hai Viet, Tran Duc Thang. "Research and development automatically generate detection rules for IDS based on machine learning technology", Journal of Science and Technology on Information security, 2022 Publication	<1%
73	citeseerx.ist.psu.edu Internet Source	<1%
74	docer.tips Internet Source	<1%
75	docs.gns3.com Internet Source	<1%
76	Submitted to Ghana Technology University College Student Paper	<1%
77	hdl.handle.net Internet Source	<1%

78	www.researchgate.net Internet Source	<1 %
79	Rasheed Ahmad, Izzat Alsmadi. "Machine learning approaches to IoT security: A systematic literature review", Internet of Things, 2021 Publication	<1 %
80	repositorio.ufmg.br Internet Source	<1 %
81	www.ijrte.org Internet Source	<1 %
82	"Web Technologies and Applications", Springer Nature, 2014 Publication	<1 %
83	Gabriel Pimenta Rodrigues, Robson de Oliveira Albuquerque, Flávio Gomes de Deus, Rafael de Sousa Jr. et al. "Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection", Applied Sciences, 2017 Publication	<1 %
84	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
85	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1 %

86	scholarworks.bridgeport.edu Internet Source	<1%
87	ti.budiluhur.ac.id Internet Source	<1%
88	www.hindawi.com Internet Source	<1%
89	www.ippari.unict.it Internet Source	<1%
90	"Information and Communications Security", Springer Science and Business Media LLC, 2020 Publication	<1%
91	Submitted to University of Essex Student Paper	<1%
92	Zhao, David, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, and Dan Garant. "Botnet detection based on traffic behavior analysis and flow intervals", Computers & Security, 2013. Publication	<1%
93	core.ac.uk Internet Source	<1%
94	docshare.tips Internet Source	<1%
95	dokumen.pub Internet Source	

		<1 %
96	dspace.bracu.ac.bd:8080 Internet Source	<1 %
97	dspace.library.uvic.ca Internet Source	<1 %
98	iasir.net Internet Source	<1 %
99	link.springer.com Internet Source	<1 %
100	www.journaltocs.ac.uk Internet Source	<1 %
101	www.softwaretestinghelp.com Internet Source	<1 %
102	www.theseus.fi Internet Source	<1 %
103	yorkspace.library.yorku.ca Internet Source	<1 %
104	Amirah Alshammari, Abdulaziz Aldribi. "Apply machine learning techniques to detect malicious network traffic in cloud computing", Journal of Big Data, 2021 Publication	<1 %
105	Vladimir Sobeslav, Ladislav Balik, Ondrej Hornig, Josef Horalek, Ondrej Krejcar.	<1 %

"Endpoint firewall for local security hardening in academic research environment", Journal of Intelligent & Fuzzy Systems, 2017

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61