DETECTION OF APPLICATION LAYER DDOS ATTACKS USING AUTOENCODER AND LOGISTIC REGRESSION

Karamjot Singh

(Registation Number: 2006510)

CE902 Professional Practice and Research Methodology

Supervisor:
Dr Michael Kampouridis

School of Computer Science and Electronic Engineering
University of Essex

April 2021

ABSTRACT

The role of IT services in society is to ensure that businesses can effectively and efficiently offer their services to the community. However, several hindrances are associated with the use of IT services in the business. The continuous Cyberattack over the business networks has raised questions on the safety of such avenues in delivering services to the customers. Several tools have been put in place to detect, mitigate, and track the attacks. However, some of the tools have proved to be effective, while others are associated with different drawbacks. One of the common attacks is a Distributed Denial of Service attack, which has been considered challenging while detecting because it sometimes seems like the normal network traffic. Several studies have been conducted in the past, which proved that detecting them using the traditional methods is a challenge. However, the development of technology has been accompanying by different modern tools that can be used in the detection process. There is an issue with the time lag of detecting an attack and a high rate of classifying a normal event as an attack. The project is meant to investigate how autoencoders can be used in the extraction process to reduce the rate of false-positive rate. The study will use the logistic regression model to improve the attack detection time, considering that it uses few computational resources.

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

Providing IT services with high availability is the primary goal of businesses which creates a lasting impact on users and as well as on the service bottom line. However, some bad actors try to hinder these services' availability on the internet with various possible techniques. Therefore, the need for Cybersecurity is prominent to ensure the network's safety, which provides tools to detect, mitigate, and trace back the attacks.

The foundational security principles of Cybersecurity is based on confidentiality, integrity and availability, generally known as a CIA triad. The Distributed Denial of Service (DDoS) is the most disruptive Cyberattack that affects network availability. Since its first attack in 1999, the DDoS are becoming prevalent and caused damage worth billions. According to the BBC, Amazon was exposed to the largest ever recorded DDoS attack having bandwidth of 2.3 Terabyte per Second bandwidth.

DDoS attacks are growing exponentially not only in terms of frequency but also in magnitude and sophistication, making them one of the most potent tools to disrupt any network. Therefore, real-time stratification of network traffic into DDoS and legitimate communications is significantly complex because of the upgrading techniques attackers use to tackle the firewall of cyber-defence. Considering the nature of attacks, several defensive tools were created to detect and mitigate the attacks to prevent damage.

The tools' primary motive is to distinguish between normal and malicious traffic to reduce false-positive signal. In other words, they block malicious traffic while keeping legitimate users secure and connected to their network. However, the difference between both the requests of normal and attacker is complicated to trace as attackers apply new methods to imitate normal requests. For instance, they compromise bots and computers to attack a network from all over the world. Moreover, sometimes attackers also apply a divergent approach to mislead detection tools such as attacking the web with different attack vectors, known as a multi-vector attack. As a result, many DDoS attacks successfully penetrate the network and consume all the bandwidth of the server, leaving connection error for legitimate requests.

DDoS exists for quite a long but got a breakthrough after the rise of the internet of things (IoT) devices in the network. These devices 24/7 connectivity to the internet creates a significant loophole, for the cybersecurity and opportunity, for the intruders. IoT devices are used as bots in DDoS attacks, also known as botnets, just like zombies with a hacker as their commander, used to send several request packets to a particular server. Collectively these requests from botnets generate a lot of traffic which becomes exhausting to manage for any network.

Cyber-security cannot rely on basic strategies to detect these modern attacks as the attack vectors are constantly upgrading. The problem is far from being solved. As a result, designing effective DDoS attack detection strategies remains an active and relevant field of study [1]. Therefore, this work will focus on building a machine learning model to detect multi-vector attacks in a short duration while maintaining a low false-positive rate.

This paper proposes an effective way to detect DDoS attacks by using a benchmark dataset (CIC-DDoS). Two effective strategies are combined in our system: feature extraction and machine learning. First, we will train an autoencoder to learn essential feature characteristics. Then we will extract the trained encoder's output, which represents the features in the lower dimensions. Finally, we will build a machine learning model to distinguish between normal and DDoS traffic.

This project is meant to explore and describe the basic terminologies related to DDoS detection, an overview of the tools and techniques used in detecting the attacks, a link to the previous literature, and the earlier studies' limitations. Data analyses, approaches of putting the research into action, a description of the methods used in validating the performance of the system, and a discussion on future work have also been discussed.

2. BACKGROUND

The section presents a brief description of the network, Distributed Denial of Service attack, and machine learning approaches used in the study.

2.1 Network Terminologies

2.1.1 Network Protocol

A network protocol is a term used to refer to the rules and regulations governing the information exchange securely and reliably within or outside a network. They are also known as the language that two devices use to communicate with each other. These rules are the guidelines that regulate the different characteristics of a network, such as a machine configuration, physical topologies, and type of cabling or speed of data transfer. The majority of network protocols used today are fundamentally based on the OSI model. The OSI (Open System Interconnection) model is one of the critical aspects that define the path that will be followed in this project.

The Open Systems Interconnection helps in outlining the manner in which the applications and layers in a network communicate with each other. The Open Systems Interconnection is made up of seven layers. They include the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer and the application layer. Each network packet follows these 7-layer OSI model to transmit data. This study aims to detect application-layer attacks, and thus we only include the protocols related to it.

• **Transmission Control Protocol** (TCP) –The TCP refers to a transport layer protocol that helps applications requiring secure transmission. It provides secure communications between endpoints through various processes to ensure that all endpoints are connected. The TCP is commonly used in most application layer services, including email, web, and file transfer. TCP allows retransmissions of packets and provides feedback after every transmission makes it vulnerable to DDoS attack.

• **User Datagram Protocol** (UDP) – The UDP is one of the transport layer protocols commonly used in applications that do not require feedback after transmission, making it unsecured. These connections can afford data loss instead of waiting for reconnection and does not require TCP services like feedback and two-way connection. It is used for most application layer services such as Domain Name Services (DNS) and video conferencing applications.UDP connections bypass the vulnerabilities of TCP but could be attacked due to flaws in the applications themselves.

2.1.2 Network Packets

To transmit data remotely, it first has to be divided into smaller segments. These segments are known as packets. A network packet is a fixed unit of data that contains the information that is being sent across the network. They are used for transmitting segments of data from one location to another. Following is the structure of a packet which is divided into three parts:

Network Packet	
Headers	Sender's IP address Receiver's IP address Protocol Packet number
Payload	Actual data to be sent
Trailers	Footers which denote the end of the packer

2.2 DDoS Attack

To prepare the detection tool, there is a need to know how a DDoS attack works. The DDoS attack happens when several devices flood a targeted system's bandwidth or services with numerous web servers. These attacks are effective because they use many compromised computer systems as attack traffic sources.. They mainly attack system resources and network bandwidth, ranging from the Network layer to the Application layer [2] of the OSI model. They can be divided into three types:

- **Volumetric attacks** are the most common form of DDoS; they use methods to produce large amounts of traffic to saturate the victim's bandwidth thoroughly.
- **Protocol attacks** are intended to drain the computing power of network infrastructure services such as routers, firewalls, and load balancers by sending malicious access requests.
- **Application attacks** are meant to target particular applications, the most common of which are database servers, but they can target any application.

2.2.1 Attack Principle

A DDoS architecture is made up of three major components:

1. **Attacker**-a hacker or an AI program that aims to damage a targeted service.

2. **Botnets**- is a group of malware-infected computers controlled by an attacker.

3. **Target** - an online service, server or a website

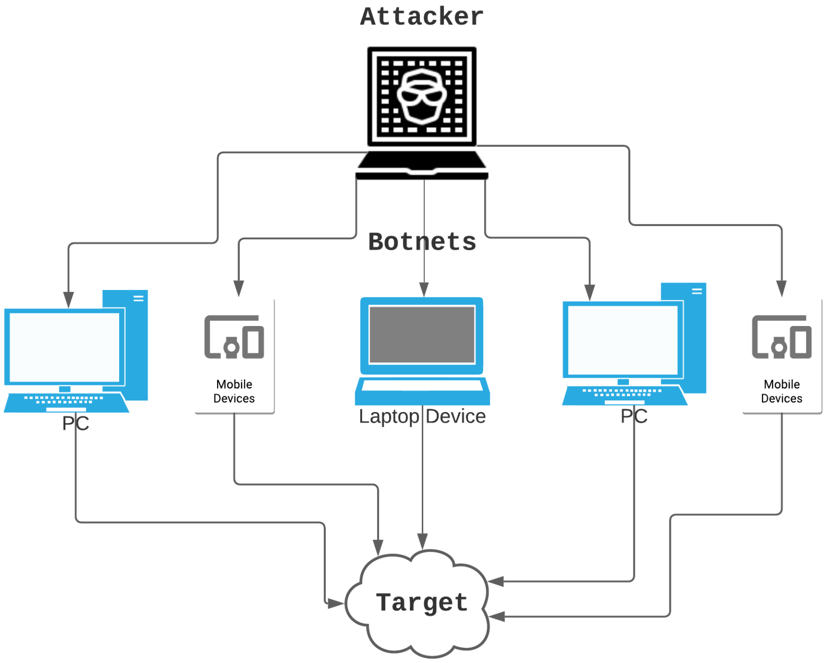


Figure 2.1: DDoS Architecture

These botnets can be composed of computing machines such as personal computers, servers, mobile phones, IoT devices, IP cameras, etc. Attackers first find vulnerabilities on a user's device, install bugs or malware to build a botnet, and then send the instructions to all of their botnet nodes. After that, the botnets will send the pre-programmed request to the target server. The legitimate requests from all these botnets made it hard for the defence system to distinguish between botnets and normal users, creating a traffic jam that results in denial of service to the normal traffic.

2.2.2 DDoS Attacks at Application Layer

Application DDoS attacks, also known as App-DDoS, have become very influential in cyber attackers. In App-DDoS attacks, attackers overload the victim server with valid requests where every zombie machine (botnets), in this case, needs to create a TCP connection to the server of the victim. This means that a valid IP address must be used for the connection to be effective; otherwise, it will not be created [3]. The attack tended to be genuine at first glance due to the valid IP address and focused on highly complex attack sequences rather than volume occurring at layers 5 to 7 of the OSI stack. These attacks are getting more robust with rising computer complexity and network bandwidth and mainly hinder popular online services such as commerce, social media and digital marketing.

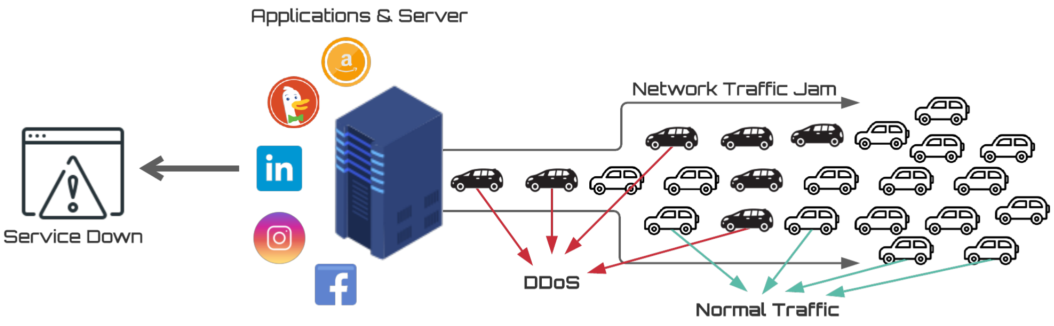


Figure 2.2: Structure of App-DDoS Attack

The Figure 2.2 describes the App-DDoS attack, where the attack traffic is blocking the legitimate traffic to reaching the server with valid requests. The intention of such attacks is to block the application server by spamming it with high load of unsolicited network traffic, resulting in server down for legitimate users.

2.3 Machine Learning

2.3.1 Auto-Encoders

learns to reproduce input. The main objective of an autoencoder is to train the network to learn a compact latent representation of the input by avoiding noise in the data. It is composed of two models, i.e., Encoder and a Decoder, in a sequence. The encoder is used to compress the input, and the decoder aims to reconstruct the input from the same compressed input. An encoder can perform a variety of applications, such as Data Compression, Image Denoising, Dimensionality Reduction, Feature Extraction and Image Generation.

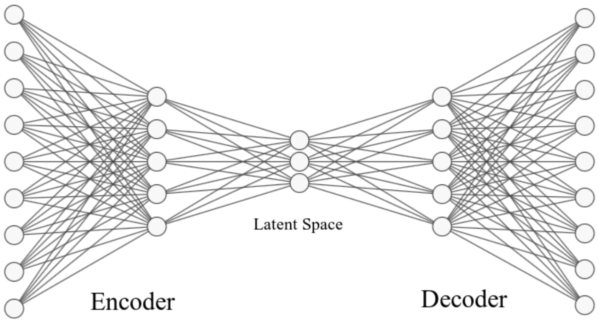


Figure 2.3: Autoencoder Structure

The aim is to minimise the reconstruction error, which is the difference between the original input and decoder's output. The encoder receives an input (x) and converts it to a latent vector (z), and then the decoder converts(z) to (x'), which is an estimate of the input (x). The reconstruction error here is $(x - x')^2$. We can obtain a subset of observed features that can retain the original data's intrinsic knowledge by simultaneously minimising the reconstruction error [4]. Thus, auto-encoder can be used as a tool to transform representation. They reduce dimensionality and detect repetitive structures, which make them different from other dimensionality reduction methods like LDA and PCA [5]. Autoencoder and its extensions have been used in a number of areas, including image processing, natural language processing, smart grids, fault diagnosis, and so on [6].

2.3.2 Logistic Regression

Logistic regression models are a widely used machine learning algorithm used to solve different machine learning problems. This algorithm is commonly used in classifying and solving problems associated with neural networks. It is based on the principle of Logistic function, also known as sigmoid function, and has many benefits over other complex classifiers; Training and using this model for forecasting is extremely straightforward, It does not require high computational power, It is easily computed and can be used in an online setting (while running in real-time) [7].

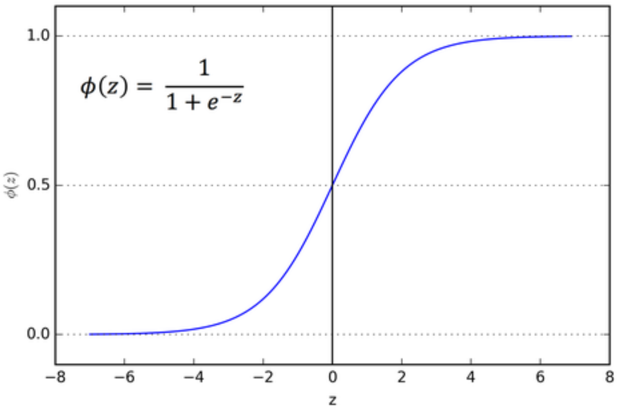


Figure 2.4: Sigmoid Function

The Figure 8.1 depicts the sigmoid function $\phi(z) = \frac{1}{1+e^{-z}}$ which takes a real value (z) and maps it to the range [0,1] $f(z)$ [7]. It is used to map the linear model in logistic regression to map the linear predictions to outcome probabilities (bounded between 0 and 1), which are easier to interpret for class membership [8]. We will used Multinomial Logistic Regression that extends to classification with more than two distinct outcomes.

3. LITERATURE REVIEW

Since the first attack, many diverse methods have been suggested for detecting DDoS attacks. The most widely applied methods in the literature are signature-based and anomaly-based detection. Anomaly-based detection is used for variations in behaviour, while signature-based detection is used for known attacks.

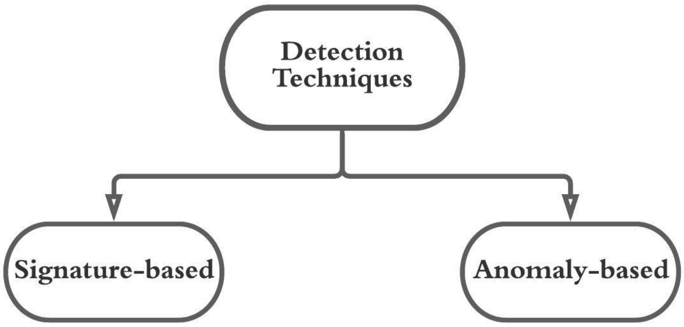


Figure 3.1: Types of DDoS Detection

3.1 Signature-based

A signature-based detection (SIDS) technique identifies potential attacks or events by matching threat characteristics (signatures) with the known network packets. In other words, a warning signal is activated when an attack signature matches with the signature in the database. It is the most simple, efficient and industry-tested system but requires regular maintenance of data of known patterns.

The network traffic profile was treated as a baseline [9] where an assumption was made because the network traffic is taken to be different from the normal traffic. The baseline, in this case, is used in detecting whenever an attack happens since the network traffic will deviate from the baseline set. A model tree was created, which indicated the abnormal flow of traffic-related to four main types of attacks: SYN, TCP, ICMP, and UDP. The false-positive rate, in this case, was between $1.2\%-2.4\%$.

This paper [10] attempts to show the detection based on packet transmission pattern using correlation coefficient. They analyse that when the attack sources initiate a DDoS attack on the victim, their propagation rate tends to be predictable in a short period of time. On the other hand, normal traffic has a time limit for receiving responses to their requests. They used a statistical model to detect the deviation between normal and attack traffic flow after realising the packet transmission pattern.

The core concept of this research [11] by Shuyuan and Daniel (2004) represents that the correlations could describe the network traffic data's characteristics among its features. In this way, identifying differences in the association between various features helped them determining the frequency of abnormalities. Their simulation results demonstrated that the system was extremely effective at detecting SYN flooding attacks in DDoS by plottingcovariance matrix under different attack rates. Another paper [12] contributed to outlining the patterns that can be followed in network flow while extracting information.

The signature detection is associated with a series of disadvantages. The main one is that it is not able to detect unknown attacks. In most cases, this approach's use is ineffective in detecting all types of attacks because some of the attacks do not have the corresponding signatures [13]. On the other hand, the rate of false positives is too high. This means that the rate of evasion will remain high during attack detection [14]. There is also a problem that is associated with the modification of signatures [15]. They are prone to errors and then cannot be able to detect novel attacks [16].

3.2 Anomaly-based

Anomaly-based detection identifies unusual events or observations that vary significantly from the standard, also known as outlier detection. Anomaly detection is essential in the sense that the anomalies that are detected in data can be useful in giving actionable and critical information used in a variety of applications [17].

A statistical model was built by Laura and Dan (2003) to capture various points of a specific field from the packet headers. The captured points of each request collectively form a distribution of the source address, which was used to measure the addresses' randomness or uniformity with respect to the measurements [18]. A Chi-Square detector was made to detect the distribution of discrete values, which triggered by an exceptional value—anomalies in the network determined by the largest value in the chi-square term.

The entropy-based anomaly was tested in 2007 [3] with a combination of two approaches, CUSUM Algorithm, which detect the mean-variance of the network flow, Time-Based Entropy Detection, which helps in setting out the threshold value; hence the system can receive an alarm when the statistics at some point exceeds the threshold value. These threshold-based methods are vulnerable to misclassify new evolutions of attacks. The issue with anomaly-based statistical identification is that they can only determine uniform network packet distribution [11].

3.2.1 Detection using Machine Learning

Benign and malicious traffic data was created with three IoT devices and Kali Linux virtual machine with spoofed IP and MAC address. Additionally, statistical feature extraction of captured traffic was carried out to represent each device's features like a simple time series of the 10-second time window. In this case, the limited-feature was helpful in restricting the computation overhead of real-time classification [19]. They used five machine learning models and got an F1 score higher than 0.99. Different feature extraction suggested by the authors in the paper can help distinguish between the traffic easily.

A semi-supervised detection technique was proposed [20] obtain the least false positive rate with high accuracy on attack classification. In order to reduce false alarm and save runtime resources, only suspected time windows of sessions were considered for training to filter out DDoS traffic data. The aim associated with using the method in question was to ensure that the irrelevant traffic and noise associated with the data are rejected before the pre-processing and classification is done.

In this case, the co-clustering algorithm was used in (2020). A Benchmark dataset (CIC-DDoS [21]) was used in this paper [22] to evaluate the efficiency of the proposed detection method. They used a multi-layer perceptron network (MLP) with SoftMax function to identify significant weights for each input feature set to extract valuable features for the classification. In order to handle the imbalance dataset, they used the weighted loss to make the classifier heavily weigh the few examples, maintaining the sum of all the weights of the classes the same. They used recall and precision for the performance evaluation.

3.3 Feature Extraction with Autoencoder

In a study conducted paper [23], the authors came up with an essential novel unsupervised feature method of selection that can help learn the importance of weights and a self-representation auto-encoder model associated with each of the features. The auto-encoder, in this case, is useful in reflecting each of the functions in a non-linear manner using the different weights.

In the above case, the group sparsity regulations and the reconstruction errors are minimised. This helps in retaining the intrinsic knowledge of the data. This method is said to be somehow superior as compared to others. The superiority of the technique has been confirmed by studies conducted by [24] [25] meant to show that the autoencoder helps capture the intrinsic information of the input data. However, these studies show that the irrelevant units that are captured do not equal importance when it comes to task classification. Another study conducted helped in demonstrating that the concrete autoencoder is essential in data reconstruction and selection as opposed to the other methods used[26]. Finally, a study conducted [27] revealed that the best method that can be used in distinguishing between the regular networks and Botnet is the use of the USML. This method is essential in creating a distinction in terms of the False Alarm Rate (FAR), specificity, accuracy, the False Positive Rate (FPR) and in terms of sensitivity [27].

3.4 Evaluation of Detection System

The study [28] conducted by Al Mehedi et al. presents an intrusion method using the Random Forest and the SVM. This approach was necessary since the detection rate, false-negative rate, accuracy, and overall performance was commendable. Further, a study by Qin et al. [29] came up with a suggestion of detection time of 0, 1-seconds. The model used a three level approach in detecting intrusion. The detection systems, in this case, were aimed at ensuring that the false negative and false positive rates are reduced. The proposed approach, in this case, was meant to use a technique based on the malicious TCP. Finally, the two studies proposed that a combination between machine learning and feature engineering can help detect and handle the detection of DDoS [30].

3.5 Logistic Regression for DDoS Detection

Logistic regression was used to characterise the DDoS attack, demonstrating its ability for real-time flow categorisation at a lower computational cost with the fastest evaluation time compared to other classifiers [31]. This research concentrated solely on TCP and UDP protocol attacks because they are the most commonly used protocols for launching an attack and used basic supervised machine learning models such as Logistic Regression and Naïve Bayes [32]. In this paper [33], authors used the Dataset (CICIDS2017) from the Canadian Institute for Cybersecurity to identify irregularities in network traffic and compared many supervised classification algorithms such as Logistic Regression and Random Forest using metrics such as maximal detection precision, lowest false negatives estimation, and training and running time. They demonstrated the significance of a simple machine learning method in reducing the latency period of real-time detection.

4. MOTIVATION

Application layer DDoS attack is clever because it tends to look like normal traffic, which makes it hard to detect by traditional methods [34]. To detect these attacks, advanced technology is required to be able to detect a behavioural element of a legitimate connection robustly and other attack vectors as fast as possible. It is also significant to distinguish between various attack vectors as different mitigation techniques are required for different attacks. There are two main concerns in current detection methods that we will try to address in our work which are:

- **High false-positive rate** - the rate of normal events that are incorrectly classified as attacks
- **Latency in detection time**–is the time lag in the detection of the attack.

The high false-positive rate occurs when the model cannot learn essential characteristics of the attack and hence fails to distinguish them. In contrast, detection latency occurs when the service is overloaded with network traffic, making it difficult for the model to process. In order to reduce the false-positive rate, we are using Autoencoders as a feature extraction technique to learn the latent representation of the incoming flow, and logistic regression will be used to improve the detection time as it can run on few computational resources.

5. METHODOLOGY

To accomplish our objectives, we created a conceptual framework (prototype) which we would use to develop and validate the final model. In this section, we'll go through all the tools used to implement the prototype, as well as the steps involved in putting the research into action.

The data pre-processing was done exclusively with Python3.9, using the NumPy and Pandas packages. For the analysis, we used a combination of yellow brick, matplotlib, and seaborn libraries to visualise the patterns and characteristics of different attacks. Furthermore, we use the SciPy and Stats libraries to conduct various statistical methods to understand the data, discover associations between variables, and use Keras for Autoencoder and scikit-learn for Logistic Regression feature extraction and classification, respectively. Figure 5.1 illustrates the steps of the methodology in sequential order.

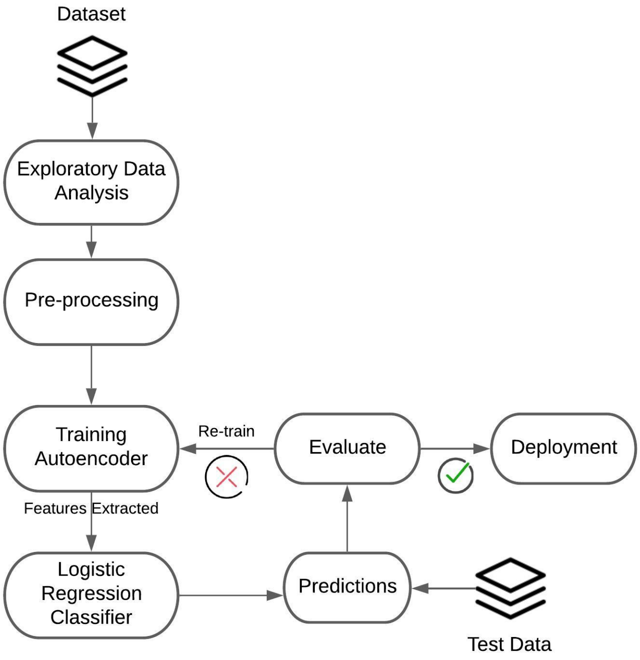


Figure 5.1: FlowChart

5.1 Dataset

Canadian Institute for Cybersecurity in 2019 shared (CIC-DDoS) [21] a labelled dataset that contains benign and common DDoS attacks. The abstract activity of 25 users was created for this dataset using the “HTTP, HTTPS, FTP, SSH, and email protocols.” It comprises numerous reflective DDoS attacks, such as “PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP.” They also used the CICFlowMeter-V3 [35] [36] for feature extraction from raw data, extracted more than 80 traffic features, and publicly shared them as a CSV file. However, this is a true reflection of real-world data, with most attack traffic corresponding to legitimate traffic.

5.2 Exploratory Data Analysis (EDA)

The size of data is the biggest challenge during the pre-processing. The data contains 80 attributes along with approximately 20 lacs rows of each attack file. The total size of data is 22 GB containing 11 files. Processing the data on a limited RAM computer was difficult. Hence, we compressed the data in a way that the distribution of all the classes remains the same as in the original data. Following is a graph representing the distribution of all attacks in the data. The graph 5.5 demonstrates that the classes are imbalanced, which means that the number of instances varies. When training and testing machine learning models, this can lead to overfitting.

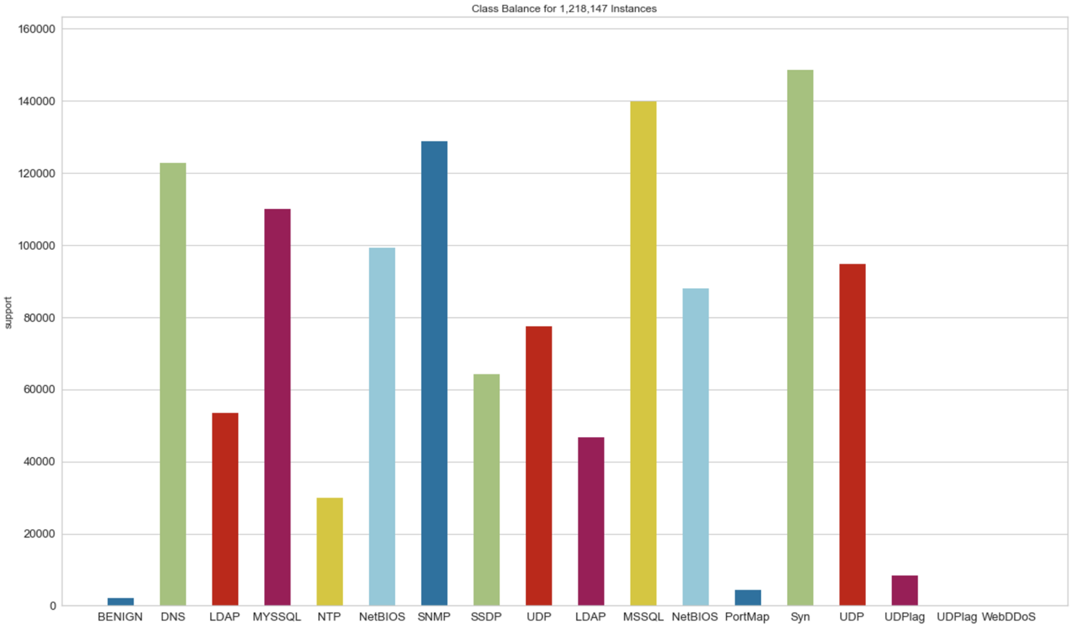


Figure 5.2: Distribution of classes in Dataset

The rate of flow packets per second in each attack is shown in the graph below. The plot clearly shows a rise in packet flow during attacks relative to benign (normal traffic) as attacks overload the target with packets. Other data attributes, such as Flow Bytes/sec, Flow Mean Time Interval, and Packet Length, showed a similar trend. There are some exceptions, like WebDDoS, which is a unique kind of attack with normal traffic features. For instance, if we create a simple prediction model based on this pattern, the model will end up distinguishing WebDDoS as normal traffic, which will increase the false-positive rate.

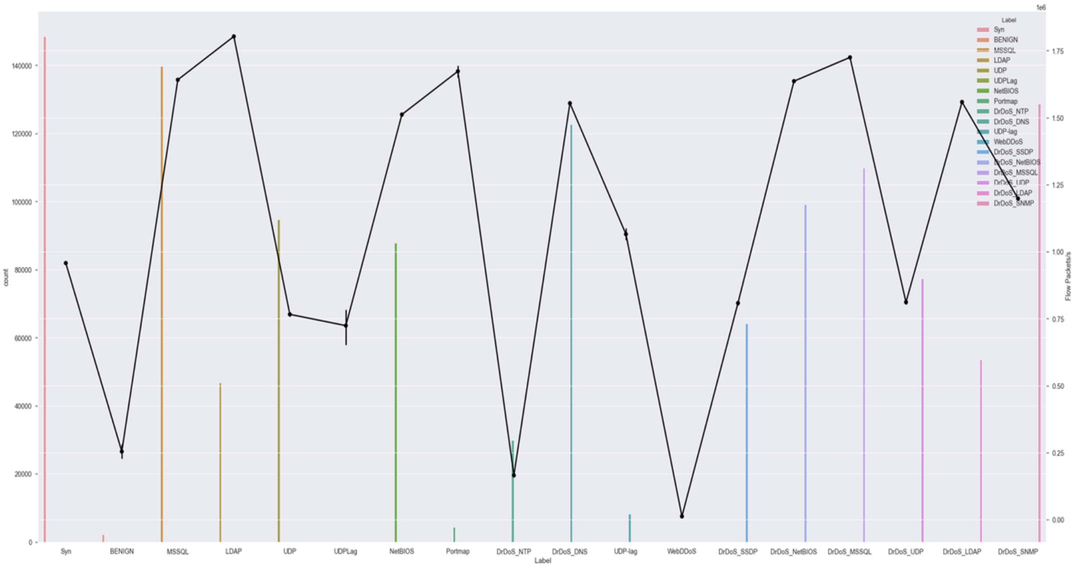


Figure 5.3: Flow Packets in each attack

The graph below shows a count plot between the protocol and the attacks, which consist of three types of protocols. 0 indicates hop-to-hop, 6 indicates TCP, and 17 indicates UDP. The graph clearly shows that most attacks occur in UDP, which is a connectionless protocol and therefore vulnerable to attacks.

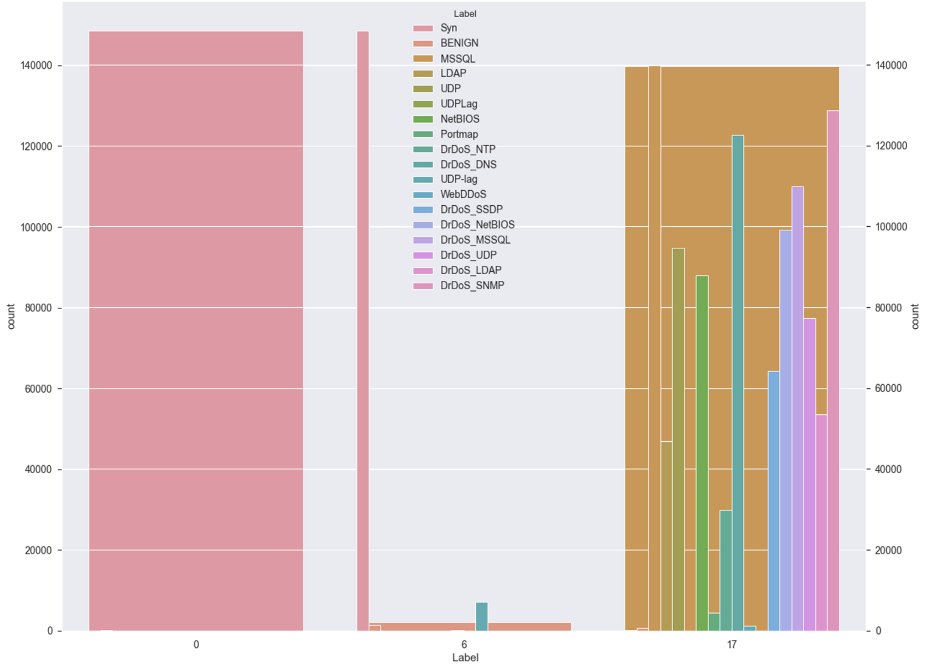


Figure 5.4: CountPlot of Protocols and Attacks

The dataset contains 77x77 pairs of continuous attributes. To determine the statistical relationship between the functions, we used the Pearson correlation test. We discovered there were 174 pairs of features with a Pearson value greater than 0.9. As a result, feature extraction could be complicated and requires an automated procedure to retrieve as much information as possible from the data. We, therefore, use autoencoder to extract valuable information from data as a feature extractor in our final model. The following are the pair plots of some attributes from the dataset which clearly depicts the linear relationship (Pearson value near to 1) between some of them.

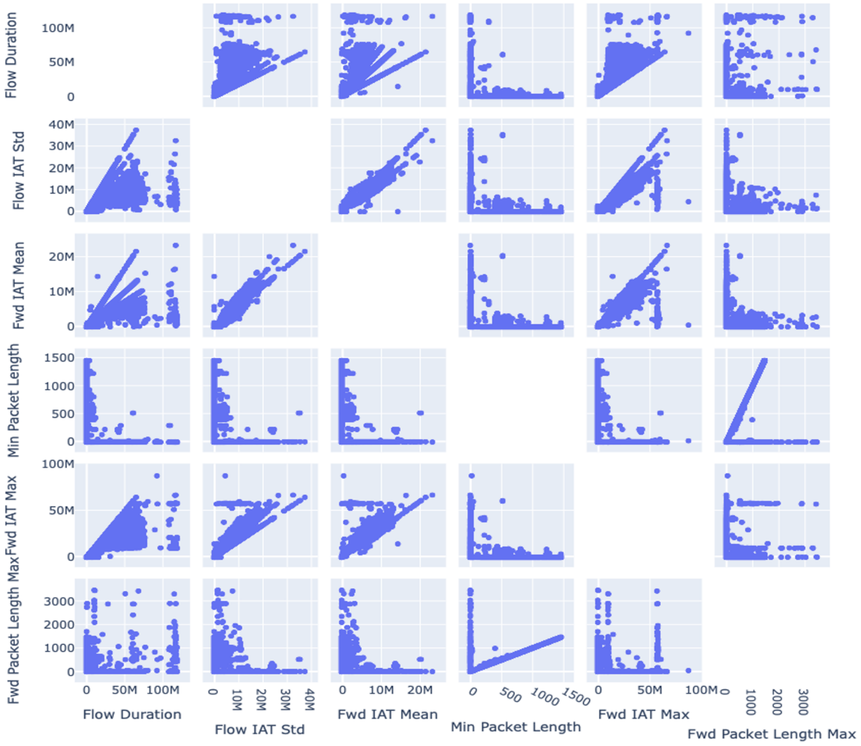


Figure 5.5: Pairplots of attributes

5.3 Data Pre-processing

- **Imbalanced Dataset**–Overfitting can arise due to the unequal distribution of classes, as seen in the data analysis. The dataset was oversampled using the Synthetic Minority Oversampling Technique (SMOTE), which produces new instances from current minority classes to address the problem.

- **Missing data** - Since certain machine learning algorithms do not accept missing or infinity values, handling them during the pre-processing of the dataset is critical. As there were few rows of missing data, we built a new Pandas DataFrame and removed the rows of missing values.

- **Datatypes** - The dataset includes socket data that must be changed to vector format so the paradigm of machine learning can be satisfied such as IP address. Using a single hot encoder system, the labelled string is converted to numerical values.

- **Feature Scaling** - there were numerical features in the data which range to millions which could create a bias and can lead to poor performance of the model. In order to neutralize the negative effect of this, we scaled the features to a range 0 to 1.

- **Data Spitting** – also known as train-test split is a procedure which is used to evaluate the performance of the machine learning model. We will split the data into two subsets: train and test set. Train set will be used to fit or train the model and test set will be used for predictions.

5.4 Feature Extraction and Classification

Feature detection is an essential aspect. It helps to increase the performance in the prediction of the classification models, and Artificial neural networks (ANN) are a strong tool for approximating complex features. Therefore, we will use Autoencoders as a feature extraction technique on the attributes as it contains more than 80 features. There are two steps involved in the process:

- **Step 1**: We train the autoencoder with the data features to reduce the reconstruction loss as much as possible. This is an unsupervised learning model, so we are not providing any target values.

- **Step 2**: After the training of the model, we discard the Decoder part and model up to the point of bottleneck, i.e., the only encoder can be used. The model's output at the encoder can be used as a feature vector in a supervised learning model.

When the autoencoder training is complete, we will construct a Multinomial Logistics Regression Classifier with the encoder's output as features and map it to the target from the initial data. In the end, the classifier will be used to measure accuracy for the training data using 10-fold cross-validation. We aim to determine the type of attack from the incoming network.

6. PERFORMANCE EVALUATION

The proposed combination of Autoencoder and Logistic Regression will be evaluated to determine the accuracy of the detection system using the following metrics, which are widely used in the literature:

- **False Positive rate** $\frac{FP}{FP+TN}$ is the rate of normal events that are incorrectly classified as attacks
- **Recall** $\frac{TP}{FN+TP}$ is the percentage of correctly identified attacks out of all attacks
- **Precision** $\frac{TP}{FP+TP}$ is the percentage of events that the model correctly identifies as attack out of all events.
- **F1 score** $2 * \frac{precision*recall}{precision+recall}$ is the harmonic mean of the model's precision and recall

Whereby

- **True positive (TP)**: “the number of correctly classified attack.”
- **True negative (TN)**: “the number of correctly classified normal traffic.”
- **False-positive (FP)**: “the number of times misclassifies attack as normal traffic.”
- **False-negative (FN)**: “the number of times misclassifies normal traffic as an attack.”

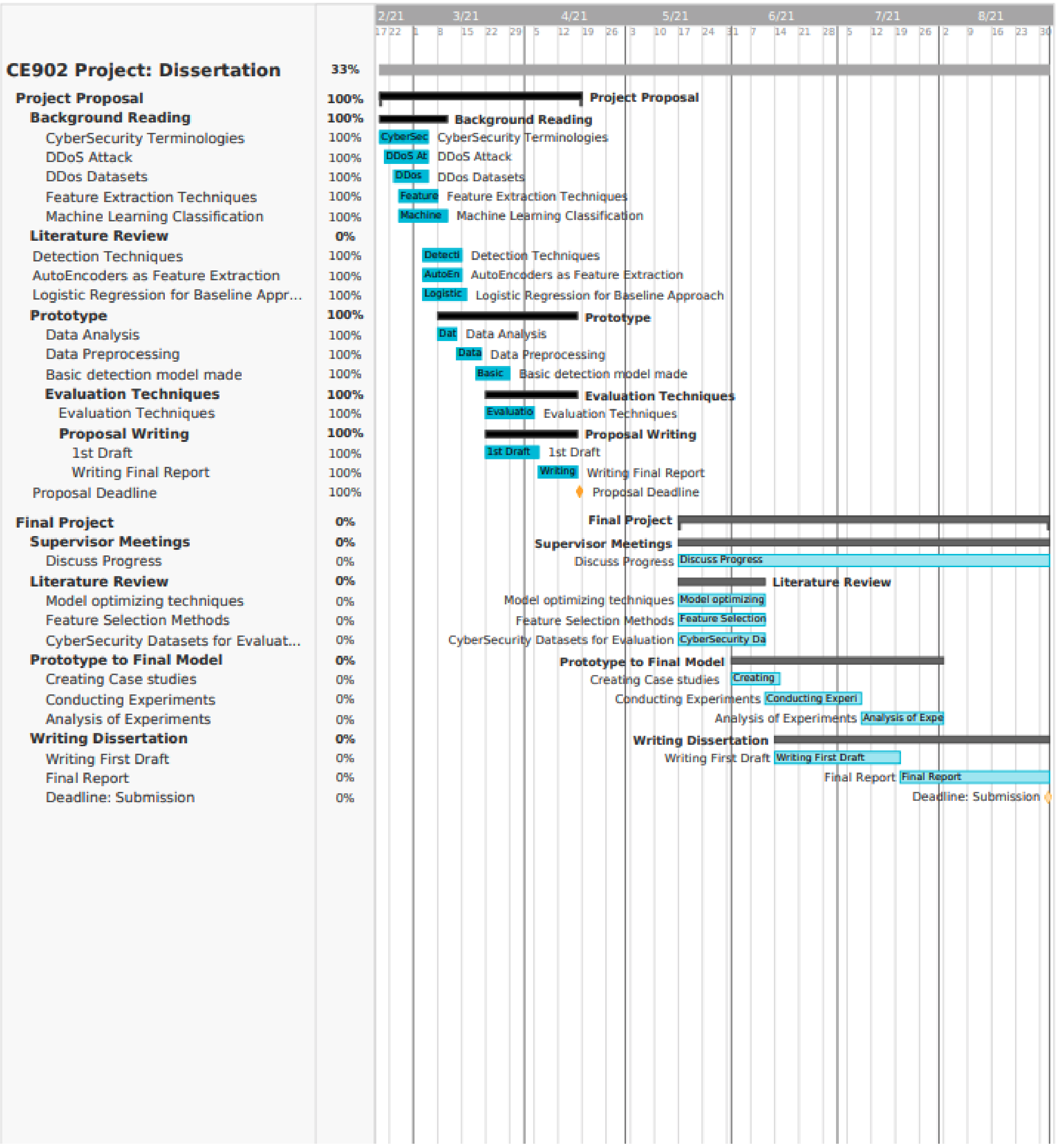
The test will be performed on the CIC-DDoS2019 dataset, which is well-known for detecting DDoS attacks. Several other data sets may be considered in the future to validate the credibility of the given approach, such as CIC-DDoS2017 [21].

7. DISCUSSIONS AND FUTURE WORK

With the help of a prototype, this preliminary work reports our ongoing effort on developing DDoS detection with Logistic Regression and Autoencoders. We have described basic terminologies and our approach to detect DDoS attack. We also have analysed the potential vulnerabilities in the literature that can be exploited for attacks. This method not only distinguish between attack and normal traffic but also classify the exact type of attack. In this proposal, we have provided a detailed description of our research methodology and the relevant experiments. Once we have obtain the results using our model, the performance of our method can be evaluated based on predefined metrics.

For future work, we would prepare a simulation environment that more closely resembles a real life setting. Moreover, it is desirable to perform DDoS detection using other supervised machine learning techniques such as SVM, random forest and decision tree. Our approach does not focus on reducing irrelevant feature space but this task is essential for the implementation of real time DDoS detection solution. This is due to traffic analysis requiring minimum number of features. Therefore, improvements of our work will include feature selection with statistical approach to reduce computational requirements and limit the feature space.

8. PROJECT PLAN



CE902 Project: Dissertation	**33%**
Project Proposal	**100%**
Background Reading	**100%**
CyberSecurity Terminologies	100%
DDoS Attack	100%
DDos Datasets	100%
Feature Extraction Techniques	100%
Machine Learning Classification	100%
Literature Review	**0%**
Detection Techniques	100%
AutoEncoders as Feature Extraction	100%
Logistic Regression for Baseline Appr...	100%
Prototype	**100%**
Data Analysis	100%
Data Preprocessing	100%
Basic detection model made	100%
Evaluation Techniques	**100%**
Evaluation Techniques	100%
Proposal Writing	**100%**
1st Draft	100%
Writing Final Report	100%
Proposal Deadline	100%
Final Project	**0%**
Supervisor Meetings	**0%**
Discuss Progress	0%
Literature Review	**0%**
Model optimizing techniques	0%
Feature Selection Methods	0%
CyberSecurity Datasets for Evaluat...	0%
Prototype to Final Model	**0%**
Creating Case studies	0%
Conducting Experiments	0%
Analysis of Experiments	0%
Writing Dissertation	**0%**
Writing First Draft	0%
Final Report	0%
Deadline: Submission	0%

Figure 8.1: Gantt Chart

BIBLIOGRAPHY

[1] Sajal Bhatia, Desmond Schmidt, and George Mohay. Ensemble-based ddos detection and mitigation model. In *Proceedings of the Fifth International Conference on Security of Information and Networks*, pages 79–86, 2012.

[2] Xiaoyong Yuan, Chuanhuang Li, and Xiaolin Li. Deepdefense: identifying ddos attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8. IEEE, 2017.

[3] Hakem Beitollahi and Geert Deconinck. Tackling application-layer ddos attacks. *Procedia Computer Science*, 10:432–441, 2012.

[4] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2941–2945. IEEE, 2018.

[5] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[6] Xiaoyi Luo, Xianmin Li, Ziyang Wang, and Jun Liang. Discriminant autoencoder for feature extraction in fault diagnosis. *Chemometrics and Intelligent Laboratory Systems*, 192:103814, 2019.

[7] Daniel Jurasky and James H Martin. Speech and language processing: An introduction to natural language processing. *Computational Linguistics and Speech Recognition. Prentice Hall, New Jersey*, 2000.

[8] The ultimate guide to logistic regression for machine learning 2021. *Keboola.com*, 2021.

[9] Yi-Chi Wu, Huei-Ru Tseng, Wuu Yang, and Rong-Hong Jan. Ddos detection and traceback with decision tree and grey relational analysis. *International Journal of Ad Hoc and Ubiquitous Computing*, 7(2):121–136, 2011.

[10] Theerasak Thapngam, Shui Yu, Wanlei Zhou, and S Kami Makki. Distributed denial of service (ddos) detection by traffic pattern analysis. *Peer-to-peer networking and applications*, 7(4):346–358, 2014.

[11] Shuyuan Jin and Daniel S Yeung. A covariance analysis model for ddos attack detection. In *2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577)*, volume 4, pages 1882–1886. IEEE, 2004.

[12] Ahmad Sanmorino and Setiadi Yazid. Ddos attack detection method and mitigation using pattern of the flow. In *2013 International conference of Information and communication technology (ICoICT)*, pages 12–16. IEEE, 2013.

[13] Abdullah J Alzahrani and Ali A Ghorbani. Real-time signature-based detection approach for sms botnet. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 157–164. IEEE, 2015.

[14] Masoud Narouei, Mansour Ahmadi, Giorgio Giacinto, Hassan Takabi, and Ashkan Sami. Dllminer: structural mining for malware detection. *Security and Communication Networks*, 8(18):3311–3322, 2015.

[15] Neminath Hubballi and Vinoth Suryanarayanan. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, 49:1–17, 2014.

[16] Md Nasimuzzaman Chowdhury, Ken Ferens, and Mike Ferens. Network intrusion detection using machine learning. In *Proceedings of the International Conference on Security and Management (SAM)*, page 30. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2016.

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Survey of anomaly detection. *ACM Computing Survey*, 41(3):1–72, 2009.

[18] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, and Darrell Kindred. Statistical approaches to ddos attack detection and response. In *Proceedings DARPA information survivability conference and exposition*, volume 1, pages 303–314. IEEE, 2003.

[19] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 29–35. IEEE, 2018.

[20] Mohd Zeeshan, Mohd Yousuf Ali, Md Raheem Sohail, and Abdul Bari. Semi-supervised machine learning approach for ddos detection.

[21] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A Ghorbani. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE, 2019.

[22] Duy Cat Can, Hoang Quynh Le, and Quang Thuy Ha. Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset. *ACIIDS 2021*, 2021.

[23] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2941–2945. IEEE, 2018.

[24] Shuyang Wang, Zhengming Ding, and Yun Fu. Feature selection guided auto-encoder. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2725–2731. AAAI Press, 2017.

[25] B Chandra and Rajesh K Sharma. Exploring autoencoders for unsupervised feature selection. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2015.

[26] Abubakar Abid, Muhammad Fatih Balin, and James Y. Zou. Concrete autoencoders for differentiable feature selection and reconstruction. *CoRR*, abs/1901.09346, 2019.

[27] Tong Anh Tuan, Hoang Viet Long, Raghvendra Kumar, Ishaani Priyadarshini, Nguyen Thi Kim Son, et al. Performance evaluation of botnet ddos attack detection using machine learning. *Evolutionary Intelligence*, pages 1–12, 2019.

[28] Md Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal, and Shamim Ahmad. Support vector machine and random forest modeling for intrusion detection system (ids). *Journal of Intelligent Learning Systems and Applications*, 2014, 2014.

[29] Jie Zhang, Zheng Qin, Lu Ou, Pei Jiang, JianRong Liu, and Alex X Liu. An advanced entropy-based ddos detection scheme. In *2010 International Conference on Information, Networking and Automation (ICINA)*, volume 2, pages V2–67. IEEE, 2010.

[30] Muhammad Aamir and Syed Mustafa Ali Zaidi. Ddos attack detection with feature engineering and machine learning: the framework and performance evaluation. *International Journal of Information Security*, 18(6):761–785, 2019.

[31] Michael Siracusano, Stavros Shiaeles, and Bogdan Ghita. Detection of lddos attacks based on tcp connection parameters. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6. IEEE, 2018.

[32] Kubra Saeedi. Machine learning for ddos detection in packet core network for iot, 2019.

[33] Alma D Lopez, Asha P Mohan, and Sukumaran Nair. Network traffic behavioral analytics for detection of ddos attacks. *SMU Data Science Review*, 2(1):14, 2019.

[34] Yi Xie and Shun-Zheng Yu. Monitoring the application-layer ddos attacks for popular websites. *IEEE/ACM Transactions on networking*, 17(1):15–25, 2008.

[35] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of tor traffic using time based features. In *ICISSp*, pages 253–262, 2017.

[36] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, pages 407–414, 2016.