A Study on Cyber-Attack Detection and Classification Using Machine Learning Techniques

by

Sakib Shahriar SHAFIN (160021102)
Sakir Adnan PROTTOY (160021112)
Saif ABBAS (160021105)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

February, 2021

A Study on Cyber-Attack Detection and Classification Using Machine Learning Techniques

Approved by:

____

Safayat Bin Hakim
Supervisor and Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)

Date: ....

Declaration of Authorship

We, Sakib Shahriar Shafin (160021102)
Sakir Adnan Prottoy (160021112)
Saif Abbas (160021105) declare that
this thesis titled, “A Study on Cyber-Attack
Detection and Classification Using
Machine Learning Techniques” and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, We have made clear exactly what was done by others and what we have contributed ourselves.

Signed:

Date:

ISLAMIC UNIVERSITY OF TECHNOLOGY

Abstract

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

A Study on Cyber-Attack Detection and Classification Using Machine Learning Techniques

by Sakib Shahriar SHAFIN (160021102)
Sakir Adnan PROTTOY (160021112)
Saif ABBAS (160021105)

The growth of Information Technology has seen the rise of Cyber-attacks like never before. It has prompted study on detection of the attacks with faster and more accurate techniques. Machines have been rising as a front-runner, as network traffic across all sectors is increasing and big data needs processing within a short amount of time and ML models are the tool. This study covers a diverse range of network traffic with attacks seen in recent times. A total of three datasets, UNSW-NB15, CICIDS-17 and CICDDoS-2019.

In this work, we cover over 20 attack types and 49, 79 and 78 features respectively for the above datasets. The three datasets were modified to create six datasets both signature based multiclass Classification anomaly based binary-class classification. At the pre-processing step, for feature selection Random Forest Regression method was used. The Machine Learning detection models were built using Logistic Regression, Support Vector Machines, Decision Tree, Random Forest, Artificial Neural Network & k-Nearest Neighbor techniques. The standard metrics of evaluation, accuracy, precision, recall, f1-score and roc are used for insights. The results obtained shows that ML trained show higher detection accuracy when the attack dataset for training is bigger having fewer attack types. Another observation is that Random Forest shows the best performance among all six ML Techniques.

Acknowledgements

First of all, we would like to bow to ALLAH Almighty, the most Omnipotent, the Most Merciful, the Most Beneficial, Who bestowed us with blessings so that we may want to endeavor our services in the direction of this manuscript.

We have our sincerest appreciation for the help, assistance and advice that many people have given us on countless occasions during the course of our undergraduate work. We would like to express our profound gratitude to our advisor, Safayat Bin Hakim for his guidance, support and encouragement. He has been a wonderful advisor, who has continuously inspired motivated us to complete many challenging research assignments. Acknowledgement is due to EEE department of Islamic University of Technology for supporting our BSc thesis/project work. We are very thankful to our family members for supporting us throughout our graduate study....

Contents

List of Figures

List of Tables

List of Abbreviations

IDS	Intrusion Detection System
LR	Logistic Regression
SVM	Support Vector Machines
DT	Decision Tree
RND	Random Forest
ANN	Artificial Neural Network
KNN	k-Nearest Neighbors
UNSW	University of New South Wales
CIC	Canadian Institute for Cybersecurity
DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
ROC	Receiver Operating Characteristics
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
CSV	Comma Separated Value

List of Symbols

b_0	intercept term for single input value
b_1	coefficient for the single input value
ε_i	slack variables
C	regularization parameter
S	set of samples
c_i^S	total number of samples in S
f	predicted activity
B	number of bootstrapping instances
f_{pred}	predicted activity value
k	number of sub-samples
t_{kp}	predicted outcome from neuron ′k′
o_{kp}	existing outcome from neuron ′k′
ω_{kj}	weight between neurons
θ_k	bias unit
$f(.)$	activation function of the neurons
η	learning rate
α	momemtum factor
o	output of neurons
$f_s(net)$	total input fed to the neuron
k	number of nearest neighbors

Chapter 1

Introduction

The next generation computing environment will create a platform where computing devices, including servers, desktop, smartphones, and other handheld mobile devices interact seemly with each other building Internet of Things (IoT) of truly massive size, and thereby making online transactions, businesses and sharing of personal information easy and convenient and in real-time. Industry 4.0 (Liao, 2017) will make the automation of industries and asset management highly cost effective and reliable through the use of intelligent sensors, IoT, and computer systems. Moreover, driverless vehicles equipped with numerous intelligent sensors are expected to hit the road in near future and revolutionize our transportation system.

While the advances in modern computer systems have found their applications in every aspect of businesses, education, healthcare, industry and everyday life, scientists and professionals did not think of the security of the computer software and hardware from the very beginning. As a result, hackers and malicious users exploit the flaws and weaknesses in the OS (operation Systems) and software modules which are extremely complex nowadays in order to support complex services, and a completely bug-free OS or software module can never be guaranteed.

The ubiquitous use of computing devices has made them an inevitable target of cyber-crime through the dissemination of malware. That is why software and OS vendor regularly release updates to mitigate discovered flaws. Such flaw and vulnerabilities are regularly listed in security related websites, such as Security Focus (https://www.securityfocus.com/) and Vulnerability database (https://nvd.nist/). Cyberattack can come in many different forms, the major attack types are listed below:
Trojans and viruses: Trojans are attacks which are stages as normal programs. These attack vulnerable OS & frames. A virus is a malevolent piece

of code intended to spread from device to device.
Spyware and adware: Spywares are used to sneak information from the target devices. Adwares pops advertisement on client's device in order to bring monetary gain for the attacker by clicks.
Phishing: Phishing uses covered email as a weapon. The goal is to trick the recipient into taking that the message is something they need.
DoS & DDoS: A denial of-service (DoS) attack is a kind of digital attack wherein a malignant actor means to deliver a PC or other device inaccessible to its expected clients by intruding on the device's ordinary functioning. A distributed denial of-service (DDoS) attack is a malevolent attempt to upset the normal traffic of a targeted server, service or network by overpowering the objective.
Ransomware: Ransomware enters unto a device and encodes all files so that the user can't have access. Then demands money in exchange of access.

One of the most publicized worldwide cyberattack in recent times in the WannaCry attack in 2017, first launched in the UK. The attack targeted systems running on Windows OS by encrypting the data stored in the computer using a key and then demanding payment in return for the key without which the stored data cannot be retrieved. In the first phase of the attack, it infected the computer system of UK hospitals and healthcare system. It has been estimated that the attack infected over 200,000 computers across 150 countries, and caused financial damage in the range of hundreds of millions to billions of dollars (WannaCry, 2017). In August 2018, a WannaCry variant infected about 10,000 computers in the Taiwan Semiconductor Manufacturing Company (TSMC) forcing to temporarily halt the operation of several of its chip-fabrication factories.

Another cyberattack that gained huge publicity is the Stuxnet attack in 2010. Stuxnet targeted PLCs used in the SCADA. It is widely believed that Stuxnet triggered substantiate damage to Iran's nuclear program by compromising their PLCs which eventually caused the crucial centrifuges used to enrich uranium to fail. Because of its potential for huge physical damage leading to shutdown of a nuclear plant, Stuxnet is also dubbed as the world's first digital weapon (Zetter, 2014).

As discussed above, cyberattack can create massive disruption to businesses and industrial operations, which in turn can result in huge financial losses. In 2019, Cyber Emergency Response Team (CERT) Australia reported responding to more than 13,672 cyber-attacks that amounted to an estimated annual loss of 328 million dollars (CERT, 2019). The emergence of the smart city means services like smart transports, smart buildings, and introduction of sustainable energies & they are are highly dependent on IoT and related cyber-physical systems (Jararweh, 2020). Any cyber-attack on the IoT infrastructure and other technologies of smart cities has the potential to bring down those services, questioning the sustainability of the smart cities. Figure 1.1 show the financial impact of major security breaches as surveyed by Cisco in 2019.

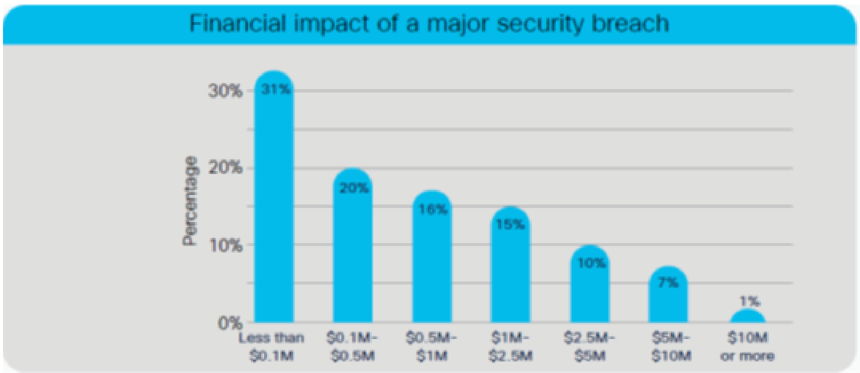

FIGURE 1.1: Impact of Security Breaches in 2019

1.1 Problem Statement and Motivation

If we try formally defining, a cyberattack is an assault launched by cybercriminals using one or more computers against a single or multiple devices or networks. These attacks can severely harm computers & servers. Because of the ever-growing number of devices and evolving networks all around the globe, we have also seen increased number of security incidents like never before. The varying complexity and sophistication of the attacks made on networks demands faster advancements and innovations in the sector of cyber security.

A widely used method of ensuring cybersecurity is called Intrusion Detection System. Intrusion detection system (IDS) is a system that monitors data to detect if there are any sort of intrusion instances in the system or network (Othman, 2018). The Network IDS (NIDS) monitors networks and the constant stream of signals to identify intrusions. It has two types. Signature based NIDS, which depends on the pre-existing classification for test uses, but aren't so much effective against classifying unknown and newer attack types, hence coming short on adaptability. The other type is Anomaly based NIDS, which creates a feature set for a normal type signal and distinguishes all other signals as attack types. Anomaly based IDS utilizes heuristic instruments to locate the obscure harmful exercises. In IDS field, the mix of both the signature based and Anomaly based is used widely.

The constant rising of new threats and processing problem relating to the said threats are inspiring Machine Learning mechanisms to gain more and more ground into the field. In recent times machine learning techniques have been used extensively in diverse domains, ranging from engineering applications to agriculture to healthcare. In particular, a number of studies have attempted to use machine learning algorithms to detect cyberattacks, such as malware detection in desktop and mobile platform (Khoda, 2019; Bae, 2020), intrusion detection in computer networks and smart city applications (Vinayakumar, 2019), distributed denial of service attack detection (Zekri, 2017) and detection of phishing attacks (Oña, 2019).

But like every new radical solution of a pre-existing condition, Machine Learning is not without its problems. The main problem is while handling a set of data with a diverse range of attacks, the models give too high FPR (False Positive Rate) too often (Staudemeyer, 2015), while also a problem is their need to be tuned to exactly what a dataset needs, and hence not generalizable to any extent, and finally, as applied to the traditional techniques, they have also not been introduced to fast paced dynamic data and complicated structure of networks (Vinayakumar, 2017).

With the world ever so driven by data, the quest for data security needs a firm push & training and testing of various types of Machine Learning method is exactly what is needed. The aforementioned challenges are the prime inspiration for this work with exploring around assessing the viability of different ML methods and apllying them to NIDS.

The prime motivation of this work is to evaluate the methods on the datasets. Hence, the need for open-sourced datasets, which have to have been recorded and vetted accordingly and has as big of a variety of attacks and features as possible is of prime consideration. There have been numerous datasets to test out the security measures since 1990s. The most famous in this case, is of course the KDD cup '99 dataset. While widely used for a long time, the KDD dataset has many attacks which are obsolete in view of the present (Wang, 2014), and that decreases the usefulness of the set as the prime focus in on testing security methods for the future, the methods who can stand the test of time. After screening and considering many sets, three datasets were selected for this study.

- UNSW-15, the dataset created by University of New South Wales. The set was created in a simulated environment, using tcpmap (unsw.adfa).
- CICIDS-17, was recorded for at a stretch of five days, using CICFlowmeter to capture the network traffic. It contains over eighty types of features, and hence is an emerging benchmark in IDS research field unb.ca, 2017).

- CICIDS-17, was recorded for at a stretch of five days, using CICFlowmeter to capture the network traffic. It contains over eighty types of features, and hence is an emerging benchmark in IDS research field unb.ca, 2017).

1.2 Research Objectives

The primary aim of this research is to design a Cyberattack detection system using the reputed datasets made publicly available for this type of research. The followings are the major research objectives of this study:

- Assessment of Cyberattack detection capability of widely used machine learning algorithms, six in total and their relative performance.
- Investigation of feature selection technique to identify the most important feature set and its impact on the performance of machine learning algorithms.
- While assessing performance, this study is particularly interested to see which learning technique produces the low rate of false negative (i.e., mis-detecting an attack as normal), even at the sacrifice of false positive (i.e., detecting a normal case as an attack).
- Staying not solely based on the features of the information, but also evaluating the performances against each other, as in examining performances of measures applied on CICIDDos-19 dataset in cases of DNS,

Portmap, SSDP, SNMP attacks separately and trying to gain meaningful insights.

1.3 Related Work

In recent times, there has been a wide array of researches done on protecting network systems from threats of cyber-attacks. Be that as it may, statistical methods can't precisely decide typical network packet organization while ML strategies require features for detection predicting . Researchers have been searching for standard & balanced datasets for a while now. For some time, that benchmark was KDD cup '99. But as time went, it has seen some criticisms as the world of cyber-attack continued to grow. As a result there has been constant analyzing of many datasets of different field of focus, with different feature selection methods & different methods of evaluation.

Like this study, researchers have compared one method against other, like kNN against SVM, using PSO to compare methods and using Neural Networks for clustering (Haykin, 1999). Researchers have also tried specifying attacks on mobile devices and their corresponding networks. The evaluation metrics used in this study are a standard in this field to understand the efficiency of a model as was seen in recent researches (Vinayakumar, 2019).

1.4 Chapter Outline

The study is structured as follows

Chapter 1 gives a brief overview of the world of Cyber-attacks. The different forms & real world examples are given. The datasets & Machine Learning were introduced. Problem statement & prime motivation for the study, along with Research Objectives are discussed. The Chapter ends with related works and outline of the whole study.

Chapter 2 provides the Background Studies needed. The Six Machine Learning methods are analyzed in-depth. Also discussed are the three datasets, their features & Evaluation Metrics.

Chapter 3 discusses the Methodology for the study, starting from choice of attack types & subsequent choice of datasets to Feature Selection, ML model training & testing to evaluating the performance of ML models.

Chapter 4 presents the performance of Logistic Regression (LR), Support Vector Machines (SVM), Decision Tree (DT), Randorm Forest (RND), Artificial Neural Network (ANN) & k- Nearest Neighbors Algorithm (kNN) for multiclass detetction and binary-class detection of UNSW-NB15, CICIDS-17 and CICDDoS-2019 dataset. the assessment & the insights found are also discussed.

Chapter 5 concludes the report and summarizes the result. It also provides directions for future work.

Chapter 2

Background Study

2.1 Machine Learning Techniques in Cyber-attacks: An Overview

In this research works, we consider a total of six machine learning techniques. Depending on the ML model used, the learning process and the time required for learning are different and therefore, the resultant trained classification model that ultimately differentiates a Cyber-attack from a normal computer application also becomes different. Since the models are different, their capability to identify Cyber-attacks is also different. One of the aims of this research is to assess the recognition capability of the learning algorithm in Cyber-attack detection applications.

2.1.1 Logistic Regression

Logistic regression is a linear binary classification algorithm often used for classification problems. Classification means a supervised machine learning approach to categorize data into distinct number of classes where we can assign label to each class. Logistic regression is a linear classification calculation that diagrams a bunch of indicators to their relating all out reaction variables. This technique will plan the chance of a result dependent on individual qualities. It is also called the sigmoid function. It depicts properties of populace growth (Ezukwoke, 2019). It's an S-shaped curve that can take any number and guide it into a value somewhere in the range of 0 and 1, yet never precisely at those limits. The equation for the sigmoid function is:

$$Y = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Where e is the base of the natural logarithms and value is the actual numerical value than can be transformed. Input values (x) are joined directly to predict the desired output value (y). A vital distinction from linear regression is that the output value being modeled is a binary value (0 or 1) as opposed to a numeric value. The equation for this is

$$y = \frac{e^{(b_0+b_1x)}}{1+e^{(b_0+b_1x)}} \tag{2.2}$$

Here y is the predicted output, b_0 is the bias or intercept term and b1 is the coefficient value of (x). Every segment in info information has a related b coefficient that should be gained by training data.

There can be a wide variety of logistic regression, like binary logistic regression, multinomial logistic regression and ordinal logistic regression. In binary logistic regression the objective variable takes one of two potential explicit values. In the event that the objective variable needs to take one of at least three potential categorical value than it is multinomial logistic regression. Ordinal logistic regression is like the multiple one, except from the target objective variables are sorted. For training the calculated model we need to follow the same process (Sperandei, 2014).

The normalizing funtion i.e., the sigmoid function can be observed below

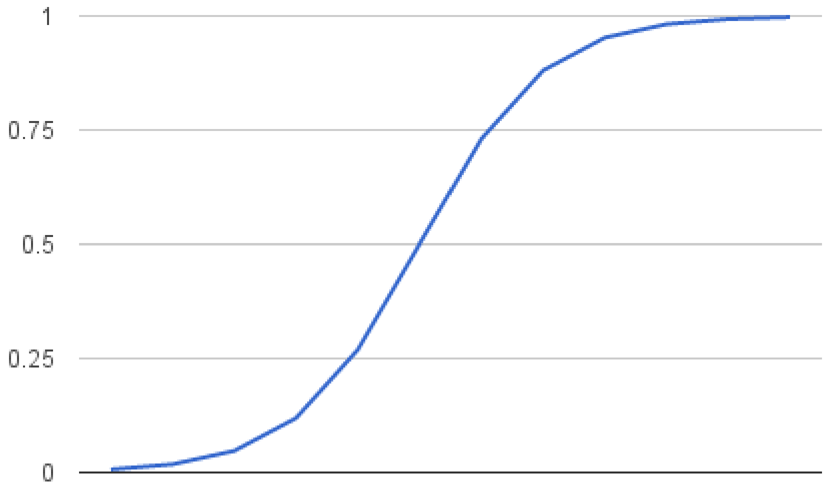

FIGURE 2.1: Logistic Regression by Sigmoid Function

The point of preparing the logistic regression model is to sort out the best weights for our linear model inside the logistic regression (ML Mastery). Important factors in logistic regression include

- Cost Function, J(θ)
- Gradient Descent/Gradient Rise
- Decision boundary is the region of a problem space

2.1.2 Support Vector Machines (SVM)

Support Vector Machinesk (Vapnik, 1995) is an effective tool in classifying data and regression tasks of all sectors. In a 2d plane, SVM aims to find an optimal separating hyperplane (OSH). In Fig. 2.2 OSH is shown that results in the most extreme edge between the two indexes. SVM also transforms data into a higher dimensional space for the construction of OSH through the use of a kernel function and then constructs a linear OSH between the two classes in the transformed feature space. Support Vectors of SVM are closest rectangles in the hyperplane space.This method makes SVM more effective than LR as we will observe in Chapter 4 (Haykin, 1999). The rule of thumb here is limiting a higher bound on the normalization as opposed to limiting the rate of error is required to perform better. The gist of the theory of SVM is as follows (Vapnik, 1995).

Consider a training set

$$D = (X_i, Y_i)_{i=1}^{L} \tag{2.3}$$

$\mathbf{x_i} \in \mathbb{R}^n$ here is the input and the output is $\mathbf{y_i} \in -1, +1$. In SVM, each input x is first mapped into a higher dimension feature space. This is where the features begin creating an impact on prediction. So, the hypplane being **w** . z + b = 0,

$$y_i(w, zi + b) \geq 1 - \varepsilon_i, \bigtriangledown i \tag{2.4}$$

where $\varepsilon_i(>0)$ are often called slack variables. This equation shows how the optimum solution is achieved and the OSH separating training & testing data **F** is the one that

$$\frac{1}{2} w, w + C \sum_{i=1}^{L} \xi_i \tag{2.5}$$

here C is the regularization parameter comparing the trade-off between highest difference of gradient & the least amount of error that can be mustered. The main goal is to minimize the second component to gain control of the hyperplane.

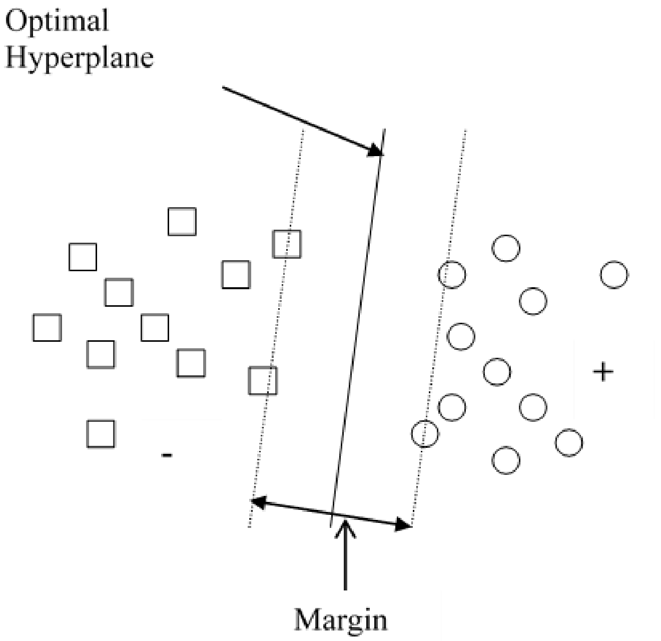

FIGURE 2.2: SVM Binary Classification with highest margin

Now the OSH of course, deals with assigning weights for the support vectors. And equation for the the vector of that, w is

$$w = \sum_{i \epsilon SVs} \alpha_i y_i z_i \tag{2.6}$$

where α1, α2 ... ,αL are the non-negative Lagrangian multipliers. For any experimental sample $\mathbf{x} \in \mathbb{R}^n$, the method gives following output

$$y = f(x) = sign(w, z + b) = sign(\sum_{i \epsilon SVs} \alpha_i y_i K(x_i, x) + b) \tag{2.7}$$

For a perfectly built classifier, regularization constant & kernel function is needed with it's values depending on trial and error. So far, no certain rule has been discovered.

2.1.3 Decision Tree

A decision tree has *nodes* and *leaves* which are generated during learning, i.e., tree building. Each node of the tree operates on an attribute and the branch-out leaves from this attribute analyze the class label depending on the value of the attribute. The sequence continues until the final class label traversing through all leaves is calculated. Once the tree is built, it acts as a classifier. Though various types of trees are proposed in literature over the years based on how nodes are created, C4.5 proposed by Quinlam (Quinlam, 1993) is a widely used one and it uses the information gain to generate nodes.

Let S be the set of samples and c_i^S be the sample size in S that is labeled as class c_i. Then the entropy of S becomes:

$$Entropy(s) = -\sum_{i=1}^{k} \frac{C_i^s}{|S|} \log_2 \frac{C_i^s}{|S|} \tag{2.8}$$

Now, it is possible that j-th attribute A_j, applying a threshold value of θ, divides S into two disjoint subsets S_1 and S_2. Considering these two subsets, the total entropy is then estimated as

$$Entropy(A_j) = Entropy(S_1) + Entropy(S_2) \tag{2.9}$$

By varying the threshold θ, the difference in entropy due to the presence of the attribute A_j is calculated, which is known as information gain of A_j,

$$Infogain(A_j) = Entropy(S) - Entropy(A_j) \tag{2.10}$$

The attribute that produces the highest gain is selected as the first node and the two branches create two subsets of the data. The tree generation algorithm repeats this process until nodes can not be divided further. This occurs when the information gain reduces to zero. These nodes with no more branches are known as leaf nodes. Once a sample data traverses through a constructed tree, on the type the node stops is recognized as the final prediction..

2.1.4 Random Forest

The main operation of this particular method is fitting a number of classifying trees into a dataset. The trees make step by step decisions, and then each tree projects their decision vote for the popular class to respond to an input vector (Breiman, 1996). The method can use a particular index method as an attribute selection measure. The algorithm starts with random bootstrap samples, where almost two thirds of the original samples repeat in one or more subsets or in one of the trees of the forest (Biner & Schumacher, 2008).

Random Forest uses the method of bootstrap aggregating, or bagging to train the tree learners. Bagging is a strategy to produce a training dataset by arbitrarily chosen substitution of N examples, where N is the size of the main training set (Breiman, 1996). Suppose, a sample set (**X= X1,X2,...,Xn**) with output (**Y= Y1,Y2,. . . ,Yn**) aggregates for B times and selects random replacements of the samples in the training set.

The algorithm then fits the classification trees into the samples. So, for the number of bootstrapping instances (**b= 1,2,. . . ,B** the algorithm at first takes the N number of samples with replacements (X_b, Y_b) then trains a classification tree on X_b, Y_b. The process is repeated B times to build B number of classifiers.

After training, prediction for the test set X′ is done by averaging the prediction from all individual classification trees on X′.

$$f = \frac{1}{B} \sum_{b=1}^{B} f_{pred} \tag{2.11}$$

where f is the predicted activity value of the k-th compound, B is the number of bootstrapping instances, f_{pred} is the predicted activity value of the k-th compound by the b-th instance.

Training a sizeable number of trees on a single dataset creates correlation, which after some samples starts showing bias which leads to greater false positive rate. The bagging operation gets rid of this circumstance, reducing the correlation between nodes which leads to variance. However, increased variance and lower correlation may lead to overfitting. Averaging the individual predictions solves that problem. This is where the Random Forest model surges ahead of Decision Tree system. Random Forest with its multiple set of trees can avoid high bias and can create very low variance, which leads to better performance and accurate prediction.

2.1.5 Artificial Neural Networks (ANN)

An ANN is an iterative learning technique where the relationship between a set of input-output data is learnt by presenting an input and then learn from the error it produces in mapping the relationship. It is done in a way that the network is likely to produce less error when the same input is presented again. An ANN consists of many computational units called neurons, imitated after the neurons in the human brain. These neurons are arranged in layers. The first layer is the input layer when the feature values from a sample are presented. Each neuron in the input layer is connected to each neuron in the next layer called the hidden layer. From every node,there is connection to every node of the next layer. At the last step, class type decides the number of output nodes. Note that, an ANN can have more than one hidden layer as well. The layers are interconnected by connection weights which are modified during training.

Backpropagation (Rumelhart, 1986) is the widely used algorithm to train ANN.What backpropagation does for a set of samples suppose, (**X1,X2,. . . ,Xp**) is whenever the data and their weights journey from one layer to another and completes the passage, the values (**Y1,Y2,. . . ,Yp**) return in the opposite direction with an updated list of errors which is then adjusted into the weights. A Backpropagation network is shown in Fig.1. The weights chosen at each noder are selected arbitrarily, without any distinction. The input, i.e., feature values are forwarded to the system which are then multiplied by the corresponding connection weights. After the iterations & the backpropagation steps for each iteration, the final output looks like.

$$o_{kp} = f(net_{kp}) = f(\sum_{j} \omega_{kj} o_{jp} + \theta_k) \tag{2.12}$$

where o_{kp} is the output of neuron 'k' , o_{jp} is the output of neuron 'j' at the preliminary layer,ω_{kj} is the weight between the neurons 'k', θ_k is the bias for unit 'k' and $f(.)$ is the function that activates the nodes after the output of the last layer enters. The input neurons are linear and its outputs are the same as inputs.

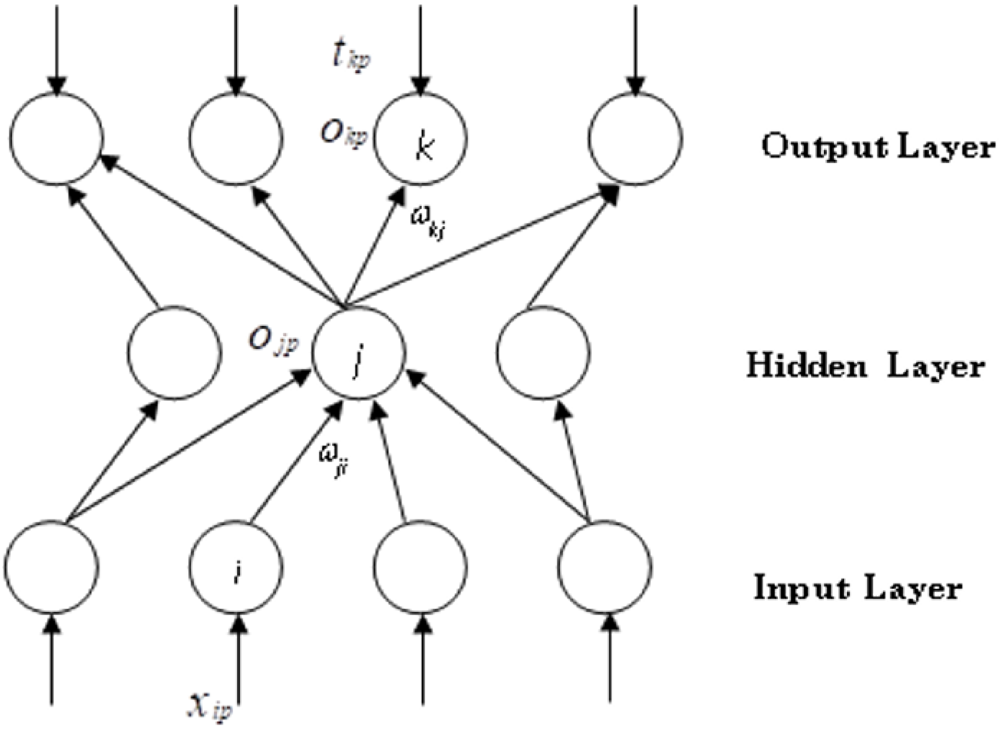

FIGURE 2.3: A Back-Propagation Network For ANN

Cost function measured after each iteration which is to be neutralized is done by following equation

$$E = \frac{1}{2} \sum_{p} \sum_{k} (t_{kp} - o_{kp})^2 \tag{2.13}$$

here, *t* and *o* are the outputs of 'k' that is to be compared for pattern 'p'. To minimize the value of cost function, the method uses steepest descent. The ML model applied in this study has provided the values of the cost function for all the iterations performed. Updating the weights after an iteration when the whole network moves backwar, is done by the following equation.

$$\Delta\omega(t) = -\eta \frac{\partial E}{\partial \omega(t)} + \alpha \Delta\omega(t-1) \tag{2.14}$$

here, η is the rate of which the weights will start to be updated and α is the momentum factor. the amount of change in weights are reliable on the cost function and activation function f(.). The standard Back=propagation formulation uses sigmoid activation function like:

$$o = f_s(net) = \frac{1}{1 + e^{-net}} \tag{2.15}$$

where net is the total input fed to the neuron and o is the output of the neuron.

2.1.6 K- Nearest Neighbors (KNN)

KNN is a famous Machine Learning Technique known for its effortlessness and accurate classification. KNN is a non-parametric and languid calculation in that, it doesn't make any estimation about the set of information rather the whole process along with the algorithm is designed based on the actual dataset, and furthermore, KNN applies no normalization from the training set (Peterson, 2009).

The classification system of KNN works on a majority vote system based on the number of k. If there is an unclassified sample, and the value of K is assumed to be 10, then the identity of that sample will be dependent on the identity of it's K=10 nearest neighbors. The sample will be identified the same as the majority in 10 votes of the neighborhood.

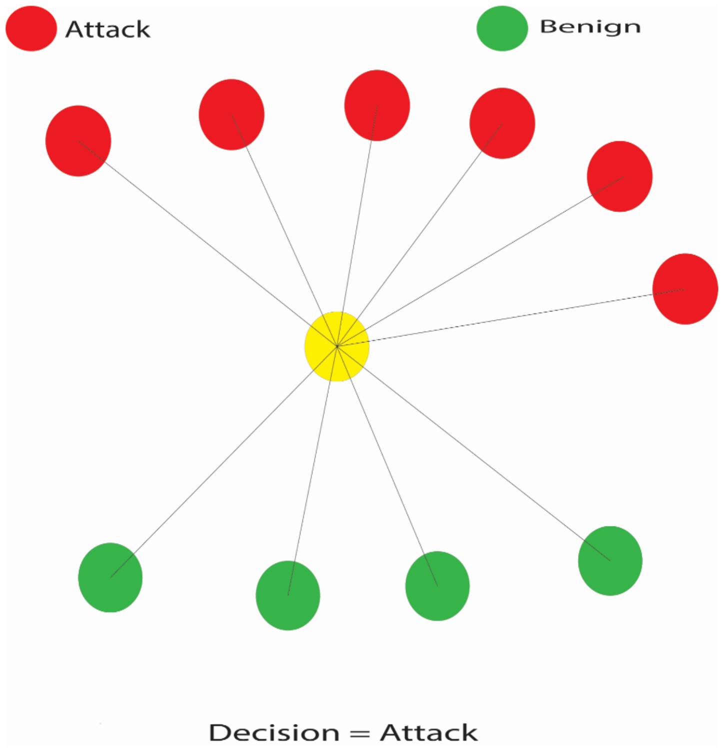

FIGURE 2.4: KNN Classifier. Yellow dot represents an unknown sample.

Fig 2.4 shows that in a distinct boundary, an unknown sample is surrounded by ten neighbor samples. Each sample casts vote to classify the unknown one. Here, out of K=10 samples, six of them are Attack type and four are Benign. Hence, the majority is Attack Type and the unknown sample will be classified as an Attack Sample.

Performance of KNN depends on two factors, choice of the number K & selection or scaling of features to improve classification. Large value of K makes the boundaries vague. There is no specific rule of thumb to determine K, but many methods are being constantly used & evaluated, such as K-Fold Cross Validation, Bayesian methods (Heller, 2007) & other heuristic methods. However, in two class/Binary classification problems, K should be chosen an odd number as this avoids tied votes.

Optimizing the feature voting weights of KNN has been a matter of constant study, as accurate feature selection is important in getting the algorithm accurate. Unrelated features can result in classifying an attack signal as a benign one, compromising the whole detection system. Modern techniques include Evolutionary Optimization of imbalanced dataset, Mutual Information-based Method etc.

2.2 Datasets

In order to gain in-depth understanding of the performance of machine learning algorithm, choosing appropriate datasets are of highest concern. Relevant & potentially dangerous attacks, which are not easy to detect when present in a huge amount of data stream, should be given utmost priority to study. Recent datasets created for intrusion detection gives us a clearer picture of what the field needs in these times. This study focuses on three datasets of the very recent past to pit the models against the very real threats that are present against networks across the world.

2.2.1 UNSW-NB15

The UNSW-NB15 dataset was created in 2015 for research purposes in intrusion detection using IXIA Perfect Storm tool in ACCS of University of New South Wales to create a hybrid of the up to date benign signals and artificial attack signals from network traffics(unsw.adfa.edu.au). Tcpdump was the network captur tool. The training and testing sets are made up of 82,332 and 175,341 records respectively(unsw.adfa.edu.au). The dataset contains nine attacks, specifically Analysis, Exploits, Fuzzers, DoS, Generic, Reconnaissance, Shellcode, Worms and Backdrop. The Argus, Bro-IDS tools are utilized. The nine attack types can be classified as three attack groups, namely Seizure attacks, Penetration attacks and Scanning attacks.

After feature selection, 25 of the 49 features were deemed relevant enough to work with. Table 2.1 lists some of the prominent features with higher selection values of the dataset.

Feature	Description
dur	Record total duration
sbytes	src to dst transaction bytes
sttl	src to dst time to live value
smean	Average of the size sent by the src
dbytes	dst to src transaction bytes
sload	Source bits per second
sjit	Source jitter (mSec)

TABLE 2.1: Feature Set used in UNSW-NB15

2.2.2 CICIDS-17

CICIDS-17, or simply IDS-17 was provided by Canadian Institute of Cybersecurity. It includes the result of the network traffic analysis using CICFlowMeter with labeled flows (unb.ca).

The data capturing period started at 9 a.m., Monday, July 3,2017 and ended at 5 p.m., Friday July 7,2017 for a total of 5 days(unb.ca). Monday was the normal day which only included the benign traffic. The attacks were executed both morning and afternoon on Tuesday, Wednesday, Thursday and Friday. The attacks contain mainy types such as DoS, PortScan, DDoS etc. The importance was given in creating a dataset with relevant features, & as we will see in Chapter 3, this particular dataset works very well in Feature Selection.

Feature	Description
PSH Flag Count	Size of packets with PUSH
Length of Fwd Packets	Total length of the forward packets
Fwd Packet Length Max	Max length of packet in fwd direction
Flow IAT Mean	Avg time between two packets sent in flow
Fwd Header Length	Bytes used for headers in the fwd direction
Destination Port	Address to receive TCP or UDP packets
Flow IAT Max	Max time between two packets sent in flow

TABLE 2.2: Feature Set used in CICIDS-2017

2.2.3 CICDDoS-2019

CICDDoS2019 dataset was also created by Canadian Institute of Cybersecurity. The difference between CICIDS2017 and CICDDoS2019 is mainly in the detection of DDoS attack types. CICDD0s2019 focuses mainly on detecting DDoS attacks. It contains benign and most recent DDoS attacks. It also includes the results of the network traffic analysis using CLCFlowMeter-V3 with labeled flows (unb.ca).

In this dataset there are different modern reflective exploratory DDoS attacks. FOr a total of two days, these were recorded. In features extraction process from the raw data, the CICFlowMeter were used. It extracted 78 features and were archived in the research center company(unb.ca).

After feature selection, 24 of the 78 features were deemed relevant enough to work with. Table 2.2 lists some of the prominent features with higher selection values of the dataset.

Feature	Description
Max Packet Length	Maximum length of a packet
ACK Flag Count	Number of packets with ACK
Packet Length Mean	Mean deviation of the packet length
Source Port	Address to send TCP or UDP packets
Flow IAT Std	Std time of two packets in the flow
Fwd Packet Length Mean	Mean deviation of packet in forward direction
Destination Port	Address to receive TCP or UDP packets

TABLE 2.3: Feature Set used in CICDDoS-2019

2.3 Evaluation Metrics

To assess the recognition ability of classifiers, we assess how a model perform in correctly recognizing the samples that the model has not seen before, i.e., on the test dataset. To evaluate the performance, we will apply the following performance metrics. Let,

True Positive= This is when an ML method successfully predicts an attack sample as an attack.

False Positive= The occurrence when a benign sample is classified as an attack sample wrongly. This adds to the error.

True Negative= It is when the algorithm correctly identifies a normal sample as a normal sample.

False Negative= When the machine predicts a sample to be a normal sample, but it's actually an attack type, then it is called a False Negative. These are to be avoided by any means because they are the biggest challenge in IDS.

N= It is the number of all the samples in total, the summation of all four parts above.

The four types of classification mentioned above creates a Confusion Matrix shown in Table 2.4.

	Predicted Class	
Actual Class	True Positive	False Negative
	False Positive	True Negative

TABLE 2.4: Confusion Matrix

The main task to evaluate the validity of the techniques against the dataset is to measure the accuracy of the set. This is evaluated with Classification Accuracy (ACC). It simply states the rate of correctly predicting type versus all predicted type.

$$ACC = \frac{TP + TN}{N} \tag{2.16}$$

The overall Classification Report of every dataset with Binary Classification or Multiclass Classification had 3 metrics.

- *Precision:* It shows the models the accuracy of detecting attack signal. The higher the precision, the better.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{2.17}$$

- *Recall:* It indicates the percentage of attack signals that were correctly classified. It is also known as sensitivity.

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{2.18}$$

- *F1-Score:* : It is also called or F-measure. It is the the measure of a model's accuracy on a dataset as a whole. The model is better and more accurate when it is higher.

$$F1 - Score = \frac{2 * Sensitivity * Precision}{Sensitivity + Precision} \tag{2.19}$$

2.3.1 Receiver Operating Characteristics

ROC is a widely used tool for visualizing Machine Learning model performances. ROC has another sub-function called AUC, or Area Under Curve. From one graph, we can get insights into performance metrics using ROC.

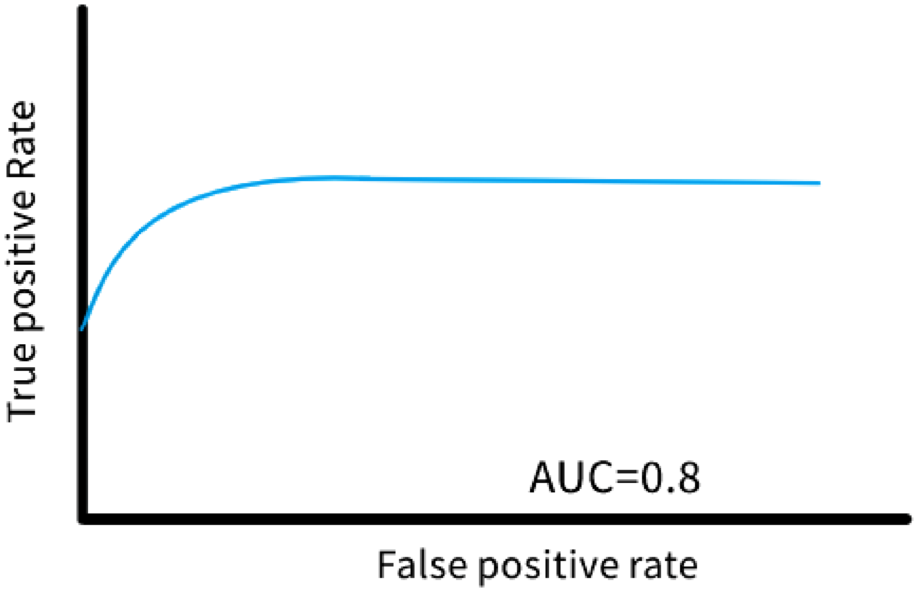

FIGURE 2.5: ROC Curve

From Fig 2-5, the ROC curve represents the ML model's ability to correctly classify individual attack types from several types. It gives an understanding of how the model is performing at predicting as well as if the feature selection process was successful.

AUC is a probability measure that shows the area under curve.

For example, from the curve, it can be seen that there is AUC= 0.8. So, the area under curve is 0.8, and there is a 80% chance of the model to correctly classify a sample from Benign to Attack type. ROC & AUC are directly linked to the accuracy of the models, hence the ROC and AUC of Decision Tree, Random Forest will be higher than that of Logistic Regression.

2.4 Specifications

This study uses Python 3 language to implement the Intrusion Detection Method. For dataset manipulation & analysis, Pandas was used as the analyzing tool atop the python language. Jupyter notebook was used as the editing software. Jupyter Notebook is a server client-based application that uses a Local Host system that can be accessed via a web browser, such as Chrome, Opera Mini etc. It also has a control panel that can be used to control the kernels.

The device used to implement the algorithms is an HP Pavillion Laptop Computer with Intel(R) Core i7 processor with 8.00 GB installed RAM with Windows Version 10 operating system.

Chapter 3

Methodology

This chapter describes the methodology used in this study to evaluate UNSW-15, CICIDS-17 & CICDDoS-2019 datasets using six Machine Learning methods. The following article provides detailed explanation of the algorithm and elaborate the process of applying the ML algorithms in each dataset to detect cyber-attacks. Below is the general flow chart of the process in this study.

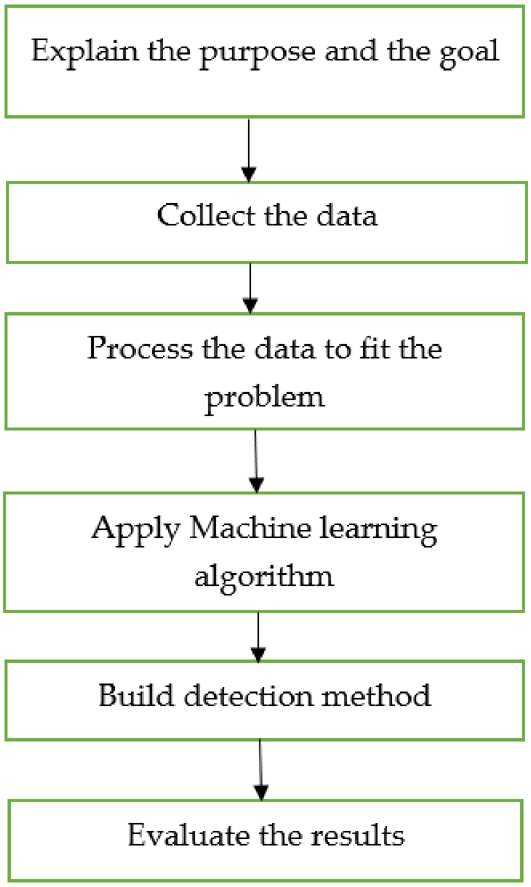
Explain the purpose and the goal
Collect the data
Process the data to fit the problem
Apply Machine learning algorithm
Build detection method
Evaluate the results

FIGURE 3.1: General Flow-Chart of the Study

The first step is to define the purpose and the goal of this study. Understanding the purpose is integral to our method because it will guide us along the necessary path in applying the tools. Machine Learning is a vast world of knowledge, and the lack of certainty of the goal may lead us to vague goals and lower standards of accomplishment. The general problem throughout this study is to detect cyberattacks in network systems. When we delve into the details, the detection of attack type becomes specific for singular types and its behavior. Each dataset covers a certain area of network attacks, and each attack has their individual prediction system. The definition of an attack needs to answer the following questions:

- What is the attacker's ultimate goal?
- What is the function of a specific attack type?
- How is the attack launched?
- What methods can be used to correctly classify the attack?

First, we need to understand the goal of the attacker and why one would choose a specific type of attack signal to invade into a network. The related works mentioned and the referenced articles in this study, blog posts and network security reports such as Cisco Internet Report 2019-2023 helps a great deal into understanding the motive for such actions. Full comprehension of the examined attack indicates its conduct and its effect on the networking data, for example, network traffic and log files. Extensive examination of the purpose is also necessary to choose the type of datasets with the selected slate of features we need for the study, as well as to choose the methods of Machine Learning Techniques.

It is understandable that in light of the constant growth of technology, tools and mechanisms of cyberattack will also develop and become more complex. As the old attack types become easier to detect, newer attacks will also spawn. Hence, for advancement of this sector we need to emphasize the newer, more complex attack types that need to be examined at the same time excluding the older and redundant attacks. For example, KDD cup'99 is considered to be a benchmark in this field.

However, many of the attributes in here, particularly, remote client address, TTL are very fewer in number. But they rae growing as the years go by. Hence, continuous and singular use of the dataset is not desired for advancements (Sabhnani, 2004). Also, DARPA / KDDCup 88 failed to evaluate the classical IDS, which also resulted in facing harsher criticisms (Vinayakumar, 2019). A real-life example can perhaps prove the need even more. Since 2014, the National Information Security Vulnerability Sharing Platform (CNVD) in China has seen a growth of 15 percent for security concerns. Among them, security concerns recorded in 2018 was 14,201 (China Internet Network Security Report, 2019).

3.1 Data Collection

Analyzing the attack types and their purpose, the next step is procuring the data. The collected information straightforwardly influences the characteristics and the highlights that can be understood from it, and consequently the detection mechanism. So, the data collection step cannot be considered totally free of another step, which includes the feature extraction and implementing the ML algorithms. We have used three types of dataset, namely, UNSW-15, CICIDS-17 & CICDDoS-2019.

UNSW-15: This dataset has nine types of attacks. Reconnaissance and Worms are classified as scanning type attacks. Scanning attacks are information gathering network attack in quest of the status. These attacks are also unsafe for hosts and the network. Analysis, Backdoors, Exploits, Generic and Shellcode are classified as penetration type attacks. Penetration attacks utilize imperfection in the software design and development and use it to change the state of the system. Lastly, DoS is classified as seizure type attacks. Seizure attacks grasp a system resource and refuses to release it for other users to use it, therefore it results to a seizure of computer resource.
Fuzzers, Exploits and Generic are most frequent type of attacks. Fuzzers scan to locate faults and security loopholes. Exploits are a grouping of informational that takes advantage of a glitch, bug, or defenselessness to be caused by an inadvertent or unsuspected behavior on a host or network. Generic attacks are a system works against all block-ciphers without consideration about the structure of the block-cipher.

CICIDS-17: CICIDS-17 contains benign traffic and most recent common attacks. This dataset contains 4 attacks with large sample, and 3 with very small samples. CICIDS2017 dataset includes benign and the most recent common attacks. It also contains the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). This dataset contains information as five days traffic data. DoS, Botnet,

web attack and DDoS were most frequent types of attack.
Web applications are effectively open to programmers. They are likewise a rewarding assault target since they additionally store important information, for example, charge card numbers and monetary information.

CICDDoS-2019: CICDDoS2019 contains benign and widespread DDoS attacks. This Dataset also uses CICFlowmeter to detect real time network data in a controlled environment and focuses on different type of DDoS attacks. As DDoS attacks are a growing phenomenon, this dataset consists of a total 12 types of attacks. There are two major type of attack. Reflection attack, a response based authentication system utilizing exact protocol in all directions. It consists DNS, TFTP, NetBIOS types. Another is exploitation type, in which attacker tries to exploit a vulnerable system for self-inflicting harm. It consists of SYN, UDP, UDP-Lag.
The main reason for selecting this dataset is that it is one of the latest in the IDS field for detecting DDoS attacks, and can be used greatly to get an insight into the future.

All three datasets were collected from their parent research websites.

3.2 Resizing & Transforming The Datasets

After selecting the datasets, comes the factor of resizing the data. For logistical restrictions, it is not feasible to use every sample of datasets, as CICIDS-17 has almost 3 million samples, and CICIDDoS-2019 has over 50 million samples. UNSW-15 dataset has 0.25 million samples; hence it needed no resizing.

Resizing Process of the two datasets mentioned has the common steps listed below:

- The datasets simply could not be fractioned, as it deters the ration of classes in the original dataset. Hence, at first the Normal Class and all the attack types of the original dataset had to be isolated into individual data frames.
- Next the data frames of the singular classes were fractioned. For CICIDS-17, 10% of the original samples were taken. For CICIDDoS-2019, 2% of the original samples were taken. The difference was done purely based on logistical reasons, as CICIDDoS-2019 has nearly 10 times the sample size of CICIDS-17.

- The fractioned data frames of individual classes were appended together, and shuffled multiple times using the shuffle function of Scikit Learn to avoid Machine Bias.
- The fractioned data frame contain randomized serial of all the classes were transformed into a csv file to be fitted into the models.

3.3 Feature Selection

It is important for labeled data to properly represent the real-world network scenario. Representative data should include different networking and communication scenarios seen in real communications over computer networks. The samples should not include less important features, which might lead the model to bias more toward a feature which has very less indication of the signal being an attack or benign type. To ensure optimal performance, the features with the most added value to the detection should be kept, and less important features should be excluded. So, there is high need for a feature selection system to rectify the dataset.

Random Forest Regression method was used for feature selection in all three datasets in this study. As seen from Chapter 2, Random Forest uses cross-validation technique which greatly helps in reducing noise, i.e., unnecessary features. When using a dataset with a sizeable number of samples, after some iteration, the features start correlating with each other, which creates machine bias, and leads to greater False Positive rate which we try to avoid as much as possible. Running the features through Random Forest Regression, it creates adequate number of trees, adjusts for the missing values in any feature and creates an iterative tree method, where it adjusts and compares one sample with another and gradually creates a forest where the top tree will show the most effective features. Through regression, it normalizes their values, adding propagating weight system that gives us a list of each feature's Information Gain with respect to the dataset. By setting a threshold, the features with the most contributing factor can be accessed, and rest of them are discarded to create a balanced dataset.

After retaining the features and subsequently creating the most impactful dataset for the detection of a specific attack type, the next step is to use machine learning to build a method for its detection. The detection of network attacks used in this study is both anomaly detection and signature-based detection. For each dataset, after resizing and feature extraction, two datasets were created, one built for signature-based detection, another for anomaly base detection.

- The Multiclass Classification dataset consists of Benign/Normal type, and singular attack types, of respectively nine, four and twelve specific type of attack. This one is used for signature-based detection.
- Binary Class Classification was created with only two types, Benign/ Normal & Attack type. If the system gets any sample which defies the nature of the Normal states, it predicts it as Anomaly Based.

The methods used are all part of supervised & semi-supervised learning system, as the dataset has labeled data, and after the train & test split, the model required to label the test predictions as a class type. This made the choice to use unsupervised ML techniques unwise. So, LR, SVM, DT, RND, ANN kNN were used.

3.4 Training and Testing the Model

Then is the pivotal task of fitting the polished dataset into the machine learning model to first, train the machine using labeled data, and then testing the model using test set data, comparing with the pre-existing target class and evaluate its performance. Training and testing the model has the following steps:

- Reading the dataset.
- Getting rid of NaN values as they are inconceivable by the machine, and hence, unusable (If any).
- Encoding the string values of the attack types by encoding the Labels, assigning a value to each class.
- Separating the Feature class and Target class into two arrays for evaluation & comparison.
- Splitting the Train & Test dataset. As per the rule of thumb used in splitting data for ML models, 80% of all the samples were used to train the data, and 20% were used for testing the accuracy.

- Next is to modify the model to start training & testing to predict the result. This step consists of Declaring the classifier, defining the learning curve function so that it does not over iterate and increase the error rate instead of decreasing it. Here we have to determine Train & Test sizes, scores, mean and standard which are necessary in predicting the target. Sequencing the task one after another is also done in this step.
- The final step is fitting or transforming the dataset by the array requirements of the pipeline. After running the tasks, the machine learning model is finally able to predict target class for both the Train & Test sample.

Following the steps mentioned above, a Machine Learning model is created that can learn from the feature set, and by iterative method can gradually go towards predicting the target. Using 80% of the samples with ten iteration for all models and a hundred hidden layers for ANN, the machine gets prepared to predict the target.

3.5 Evaluation of the Prediction

The next step is to evaluate the result using the four evaluation metric, Accuracy, Precision, Recall, F1-Score.
The first metric accuracy score compares the model's prediction with the actual target class. All the samples are compared individually, and then a normalized average is displayed. In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in *ytrue*, i.e., the actual target class in the dataset. The models produce prediction for both the Train and Test set, to get better understanding of their accuracy capability. The variable used as input here, are

ytrue: Correct/existing Class type.

ypred: Predicted class, as returned by a classifier model.

The variable used as output are

Score: It brings the fraction of correctly predicted sample class (float), otherwise returns the number of correctly predicted samples (int). So, the accuracy is between 0 & 1.

The next three metrics, Precision, Recall & F1-Score are evaluated and put into report. Using the formulas mentioned in Chapter 2.3, we take as input the actual values in the target class ytrue, value predicted by the model ypred, target names matching the target class and sampleweight, the array shape of the total sample size. The accuracy mentioned above, the weighted average & the micro average are also produced.

After the numerical analysis, it helps to gain more understanding of the model we used by adding some visual representation of the predictive result accuracies. A heat map is produced for this purpose, where we implemented the confusion matrix as described in Chapter 2.3. The confusion matrix along with the heat map is an easy way of understanding the True Positive, False Positive, True Negative & False Negative Rates. The heat map helps to see the concentration of the samples in the confusion matrix. This method was used in binary class classification.

The last visualizing method used is plotting the ROC Curve of each model. ROC Curve plots True Positive Rate against False Positive Rate, deriving a greater understanding of how successful a model was detecting specific types of attack. AUC or the Area Under Curve shows the probability area of the model under the ROC Curve. To simply put a relation between the two, ROC is the curve, and AUC is the mathematical measurement.

Lastly, the duration of the pipelining, the training and testing process of each method were taken to get insight into the effectiveness of the models. However, as the duration was not measured under absolutely controlled environment & due to shuffling, they varied by a very small margin. So the duration time measurement might not be exact, but some broader idea of the performances could be derived and discussed in Chapter 4.

3.6 Conclusion

After evaluating the results through data analysis and data visualization, in-depth understanding of a Machine Learning model, and its performance in intrusion detection of a variety of attack and normal/benign signals is sufficient to achieve. Following the blueprint of evaluating methods in Intrusion Detection, this study delves deep into understanding the purpose, the tools and the choice of a specific type of attack. Using three different dataset which covered Seizure Attack, Penetration Attacks to DDoS and DoS type of attack, and building models of six machine learning methods, this study tries analyzing the performance of a wide range of methods that can be the tool of the future in securing networks from harmful attacks.

Chapter 4

Performance Evaluation

This chapter describes the performance of the six method and comparison between them. We took 0.25 million samples from UNSW-NB 2015 dataset. From CICIDS-2017 dataset we took 0.28 million samples. Lastly, from CICDDoS-2019 dataset we took 0.5 million samples. Table 4.1 to table 4.6 shows the performance of the different methods in the three datasets. Table 4.7 to table 4.10 shows 1v1 evaluation of DDoS-2019 dataset in repect to the four attacks with largest sample size along with normal type. Table 4.11 to table 4.16 shows each methods performance in the three datasets separately.

4.1 UNSW-NB15

Table 4.1 shows the evaluation of the six classifiers in the UNSW-NB 2015 multi class dataset. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.7376, 0.82, 0.74, 0.76, respectively, followed by KNN. The lowest accuracy (0.5661) was observed for Logistic Regression. ANN acquired the highest ROC (0.98). Logistic Regression also had the worst precision (0.61), recall (0.57) and f1-score (0.57). The evaluation value is low because the number of samples compared to the classes is less as we will see in the future .

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.5661	0.61	0.57	0.57	0.66
SVM	0.6429	0.68	0.64	0.65	0.81
Decision Tree	0.7170	0.80	0.72	0.75	0.66
Random Forrest	0.7376	0.82	0.74	0.76	0.84
ANN	0.6876	0.75	0.65	0.63	0.98
KNN	0.7289	0.80	0.73	0.76	0.76

TABLE 4.1: UNSW-NB2015 Multi-class Classification

Table 4.2 shows the evaluation of binary classification of the UNSW-NB 2015 dataset. The metrics are highest for Random Forest at 0.8742, 0.89, 0.87, 0.88, respectively, followed by Decision Tree, ANN and KNN. The lowest accuracy (0.6723) was observed for SVM. Random Forest acquired the highest ROC (0.95). SVM had the worst precision (0.61), recall (0.57) and f1-score (0.57). Logistic Regression also showed bad accuracy (0.6843), precision (0.70), recall (0.68) and f1-score (0.67) like SVM. The evaluation values are also low here, but better than Table 4.1. The lesser class number contributed to the change, as it was the only parameter that changed.

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.6843	0.70	0.68	0.67	0.76
SVM	0.6723	0.69	0.68	0.67	0.73
Decision Tree	0.8501	0.86	0.86	0.86	0.85
Random Forrest	0.8742	0.89	0.87	0.88	0.95
ANN	0.7869	0.81	0.80	0.80	0.91
KNN	0.7808	0.79	0.78	0.78	0.87

TABLE 4.2: UNSW-NB2015 Binary Classification

4.2 CICIDS-17

The results of the CICIDS-2017 dataset are shown in table 4.3. The metrics are highest for Decision Tree at 0.9988, 0.99, 1.00 & Random Forest at 0.9986 ,1.00 ,1.00 ,1.00, then KNN and ANN. The lowest accuracy (0.9386) was observed for SVM. Random Forest, ANN and KNN obtained the highest ROC (0.99). Logistic Regression acquired the worst precision (0.94), recall (0.94) and f1-score (0.94).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9482	0.95	0.95	0.95	0.93
SVM	0.9386	0.94	0.94	0.94	0.96
Decision Tree	0.9988	0.99	1.00	1.00	0.98
Random Forrest	0.9986	1.00	1.00	1.00	0.99
ANN	0.9748	0.97	0.97	0.97	0.99
KNN	0.9887	0.99	0.99	0.99	0.99

TABLE 4.3: CICIDS-2017 Multi-class Classification

Table 4.4 shows binary classification. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.9988, 1.00, 1.00, 1.00, respectively, followed by Decision Tree, KNN and ANN. The lowest accuracy (0.9210) was observed for Logistic Regression. Random Forest acquired the highest ROC (1.00). SVM had the lowest precision (0.92), recall (0.92) and f1-score (0.92). Logistic Regression also showed quite similar precision (0.92), recall (0.93) and f1-score (0.92) like SVM.

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9210	0.92	0.93	0.92	0.94
SVM	0.9214	0.92	0.92	0.92	0.97
Decision Tree	0.9986	0.99	0.98	0.99	0.99
Random Forrest	0.9988	1.00	1.00	1.00	1.00
ANN	0.9741	0.97	0.97	0.97	0.99
KNN	0.9797	0.99	0.99	0.99	0.99

TABLE 4.4: CICIDS-2017 Binary-Class Classification

The evaluation values are high in this dataset as sample size increased, and the number of classes decreased from the previous dataset (From 10 to 5) compared to UNSW-NB15.

4.3 CICDDoS-2019

Table 4.5 shows the evaluation of the six methods in the CICDDoS-2019 multi class dataset. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.8854, 0.86, 0.88, 0.86, followed by KNN, ANN and Decision Tree. The lowest accuracy (0.7536) was observed for Logistic Regression. Random Forest and ANN acquired the highest ROC (0.94). Logistic Regression also had the worst precision (0.73) and f1-score (0.72). SVM had the worst recall (0.73).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.7536	0.73	0.75	0.72	0.89
SVM	0.7863	0.77	0.73	0.75	0.85
Decision Tree	0.8494	0.85	0.85	0.85	0.86
Random Forrest	0.8854	0.86	0.88	0.86	0.94
ANN	0.8497	0.85	0.85	0.82	0.94
KNN	0.8798	0.84	0.86	0.86	0.90

TABLE 4.5: CICIDS-2019 Multi-class Classification

Table 4.6 shows binary classification of the dataset. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.9999, 1.00, 1.00, 1.00, followed by ANN, Decision Tree and KNN. The lowest accuracy (0.9934) was observed for Logistic Regression. Random Forest acquired the highest

ROC (1.00). Logistic Regression had the lowest precision (0.98) and ROC (0.97).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9934	0.98	0.99	0.99	0.97
SVM	0.9992	0.99	1.00	1.00	0.99
Decision Tree	0.9995	1.00	0.99	0.99	0.99
Random Forrest	0.9999	1.00	1.00	1.00	1.00
ANN	0.9996	1.00	1.00	1.00	0.99
KNN	0.9993	1.00	0.99	0.99	0.98

TABLE 4.6: CICIDS-2019 Binary-Class Classification

In this dataset evaluation values are average for multi class and high for anomaly class. For multi class there is not enough samples compare to classes that's why the evaluation values were not high enough.

4.3.1 CICDDOS-2019 Further Investigation

CICDDOS-2019 all attack type were fractioned. The biggest fractioned attack type was NTP, Portmap, DNS and SSDP. This subsection shows the evaluation of original sample size of these attack type with Benign samples. This evaluation proves that the CICDDoS-2019 resizing method was a succes.

Table 4.7 shows the 1v1 evaluation of benign vs NTP of CICDDoS-2019 dataset. CICDDoS-2019 has over 0.12 million NTP attack samples. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.9999, 1.00, 1.00, 1.00, followed by Decision Tree , KNN and ANN. The lowest accuracy (0.9971) was observed for Logistic Regression. All of the models acquired the highest ROC (1.00). Logistic Regression and SVM had the lowest precision (0.93), recall (0.96), f1-score (0.94).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9971	0.93	0.96	0.94	1.00
SVM	0.9979	0.93	0.96	0.94	1.00
Decision Tree	0.9988	1.00	1.00	1.00	1.00
Random Forrest	0.9999	1.00	1.00	1.00	1.00
ANN	0.9981	0.94	0.98	0.96	1.00
KNN	0.9987	0.94	0.97	0.96	1.00

TABLE 4.7: CICIDS-2019, Benign vs NTP

Table 4.8 shows the 1v1 evaluation of benign vs portmap of CICDDoS-2019 dataset. The accuracy, precision, recall and f1-score are highest for Random Forest at 0.9999, 1.00, 1.00, 1.00, followed by Decision Tree, KNN and

ANN at 0.9981, 0.94, 0.98, 0.96. The lowest accuracy (0.9971) was observed for Logistic Regression. All of the models acquired the highest ROC (1.00). Logistic Regression and SVM had the lowest precision (0.93), recall (0.96), f1-score (0.94).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9984	0.98	0.99	0.98	1.00
SVM	0.9984	0.98	0.99	0.98	1.00
Decision Tree	0.9991	1.00	1.00	1.00	1.00
Random Forrest	0.9998	1.00	1.00	1.00	1.00
ANN	0.9992	0.99	1.00	0.99	1.00
KNN	0.9993	0.98	1.00	0.98	1.00

TABLE 4.8: CICIDS-2019, Benign vs Portmap

Table 4.9 shows the 1v1 evaluation of benign vs DNS of CICDDoS-2019 dataset. There are 0.5 million samples of DNS attack in CICDDoS-2019 dataset. Here we can observe the accuracy, precision, recall and f1-score are highest for Random Forest and Decision Tree at 0.9999, 1.00, 1.00, 1.00, followed by SVM and Logistic Regression at ,0.9997 ,0.92 ,0.85, 0.88. The lowest accuracy (0.9995) was observed for ANN. All of the models acquired the highest ROC (1.00). ANN had the lowest precision (0.88), recall (0.70), f1-score (0.76).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9997	0.92	0.85	0.88	1.00
SVM	0.9997	0.92	0.85	0.88	1.00
Decision Tree	0.9999	1.00	1.00	1.00	1.00
Random Forrest	0.9999	1.00	1.00	1.00	1.00
ANN	0.9995	0.88	0.70	0.76	1.00
KNN	0.9997	0.90	0.85	0.88	1.00

TABLE 4.9: CICIDS-2019, Benign vs DNS

Table 4.10 shows the 1v1 evaluation of benign vs SSDP of CICDDoS-2019 dataset. This attack type has over 0.26 million samples. In this table the accuracy, precision, recall and f1-score are highest for Random Forest and Decision Tree at 0.9999, 0.99, 0.99, 0.99, followed by ANN, Logistic Regression and SVM. The lowest accuracy (0.9992) was observed for KNN. All of the models acquired the ROC of (1.00).

Model	Accuracy	Precision	Recall	F1-Score	ROC
Logistic Regression	0.9998	0.89	0.84	0.86	1.00
SVM	0.9998	0.89	0.84	0.86	1.00
Decision Tree	0.9999	0.99	0.99	0.99	1.00
Random Forrest	0.9999	0.99	0.99	0.99	1.00
ANN	0.9998	0.91	0.83	0.87	1.00
KNN	0.9992	0.91	0.84	0.88	1.00

TABLE 4.10: CICIDS-2019, Benign vs SSDP

In these tables we can see that the evaluation values are almost perfect. This is because of the huge number of attack samples.

4.4 Evaluation across Machine Learning Methods

4.4.1 Logistic Regression

Table 4.11 shows the evaluation of logistic regression method in different datasets. We can observe from the table that Logistic Regression has the highest accuracy (09934)., precision (0.98), recall (0.99) and f1-score (0.99) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 multi class dataset. The worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset.

Logistic regression has the best roc (0.97) in CICDDoS-2019 binary class dataset. We can observe from the table that for UNSW-NB-2015 logistic regression evaluation values are very low. This is because of the low number of samples compared the classes. Logistic Regression has the worst performance among the six methods.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Multi-class	0.5661	0.61	0.57	0.57	0.66
UNSW Binary-class	0.6843	0.70	0.68	0.67	0.76
CICIDS Multi-class	0.9482	0.95	0.95	0.95	0.93
CICIDS Binary-class	0.9210	0.92	0.93	0.92	0.94
CICDDoS Multi-class	0.7536	0.73	0.75	0.72	0.89
CICDDoS Binary-class	0.9934	0.98	0.99	0.99	0.97

TABLE 4.11: Logistic Regression

4.4.2 Support vector Machines

Table 4.12 shows the evaluation of SVM method in different datasets. Here we can perceive that SVM has the highest accuracy (09992), precision (0.99), recall (1.00) and f1-score (1.00) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 multi class dataset & the worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset. SVM has the best roc (0.99) in CICDDoS-2019 binary class dataset. SVM has similar performance like logistic regression across all three datasets. The trend as we have seen, shifts to better results when it comes to Binary Class Anomaly Detection.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Multi-class	0.6429	0.68	0.64	0.65	0.81
UNSW Binary-class	0.6723	0.69	0.68	0.67	0.73
CICIDS Multi-class	0.9386	0.94	0.94	0.94	0.96
CICIDS Binary-class	0.9214	0.92	0.92	0.92	0.97
CICDDoS Multi-class	0.7863	0.77	0.73	0.75	0.85
CICDDoS Binary-class	0.9992	0.99	1.00	1.00	0.99

TABLE 4.12: Support vector Machines

4.4.3 Decision Tree

Table 4.13 shows the evaluation of Decision Tree method in different datasets. Decision Tree has much better performance than SVM or Logistic Regression. In the table we can observe that Decision Tree has the highest accuracy (0.9995)., precision (1.00), recall (0.99) and f1-score (0.99) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 multi class dataset. The worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset. Decision Tree has the best roc (0.99) in CICDDoS-2019 binary class dataset. This method performance is the closest to random forest.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Multi-class	0.7170	0.80	0.72	0.75	0.66
UNSW Binary-class	0.8501	0.86	0.86	0.86	0.85
CICIDS Multi-class	0.9988	0.99	1.00	1.00	0.98
CICIDS Binary-class	0.9986	0.99	0.98	0.99	0.99
CICDDoS Multi-class	0.8494	0.85	0.85	0.85	0.86
CICDDoS Binary-class	0.9995	1.00	0.99	0.99	0.99

TABLE 4.13: Decision Tree

4.4.4 Random Forest

Table 4.14 shows the evaluation of Random Forest method in different datasets. Random Forest has the highest accuracy (09999)., precision (1.00), recall (1.00) and f1-score (1.00) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 binary class dataset. The worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset. Random Forest has the best roc (1.00) in CICDDoS-2019 binary class and CICIDS-2017 binary class datasets. Random Forest method has the best performance among all the other methods. From observing the other methods table and discovered that Random Forest offers the best performance. This has been a constant observation of us across all datasets and across the two methods of detection.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Multi-class	0.7376	0.82	0.74	0.76	0.84
UNSW Binary-class	0.8742	0.89	0.87	0.88	0.95
CICIDS Multi-class	0.9986	1.00	1.00	1.00	0.99
CICIDS Binary-classs	0.9988	1.00	1.00	1.00	1.00
CICDDoS Multi-class	0.8854	0.86	0.88	0.86	0.94
CICDDoS Binary-class	0.9999	1.00	1.00	1.00	1.00

TABLE 4.14: Random Forest

4.4.5 Artificial Neural Network (ANN)

Below it is shown the evaluation of ANN method in different datasets. ANN has the highest accuracy (0.9996)., precision (1.00), recall (1.00) and f1-score (1.00) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 multi class dataset. The worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset. ANN has the best roc (0.99) in CICDDS-2017 binary class and multi class datasets. This method has overall above average performance. However, this has to be noted that for logistical reasons i.e., device limitation the hidden layer was was kept at a moderate number, hence the network was not very deep. Increased number of layers will result in even better performance.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Mult-class	0.6876	0.75	0.65	0.63	0.98
UNSW Binary-class	0.7869	0.81	0.80	0.80	0.91
CICIDS Multi-class	0.9748	0.97	0.97	0.97	0.99
CICIDS Binary-class	0.9741	0.97	0.97	0.97	0.99
CICDDoS Multi-class	0.8497	0.85	0.85	0.82	0.94
CICDDoS Binary-class	0.9996	1.00	1.00	1.00	0.99

TABLE 4.15: Artificial Neural Network (ANN)

4.4.6 k-Nearest Neighbors (KNN)

Finally, Table 4.16 shows the evaluation of KNN method in different datasets. KNN has the highest accuracy (09993)., precision (1.00), recall (0.99) and f1-score (0.99) in CICDDoS-2019 binary class dataset. The second highest accuracy and other metrics are in CICIDS-2017 multi class dataset. The worst accuracy and other metrics are in UNSW-NB-2015 multi class dataset. KNN has the best roc (0.99) in CICDDS-2017 binary class and multi class datasets. This method has shown similar performance like ANN method.

Data set	Accuracy	Precision	Recall	F1-Score	ROC
UNSW Multi-class	0.7289	0.80	0.73	0.76	0.76
UNSW Binary-class	0.7808	0.79	0.78	0.78	0.87
CICIDS Multi-class	0.9887	0.99	0.99	0.99	0.99
CICIDS Binary-class	0.9797	0.99	0.99	0.99	0.99
CICDDoS Multi-class	0.8798	0.89	0.86	0.84	0.90
CICDDoS Binary-class	0.9993	1.00	0.99	0.99	0.98

TABLE 4.16: k-Nearest Neighbors (KNN)

4.5 Duration

The duration of each ML method were measured, but due to a lack of exact timing due to shuffling & randomizing the datasets to avoid bias, the results were not in previous tables. However, the duration varied generally in a range of 15-20 seconds, so broader understanding of the effectiveness of the models could be understood. Table 4.17 shows the average duration of training & testing datasets in seconds on a Laptop with 8 gigabytes of RAM.

UNSW-NB15	Duration	CICIDS-17	Duration	DDoS-19	Duration
LR	317	LR	220	LR	800
SVM	950	SVM	815	SVM	1705
DT	42	DT	95	DT	516
RND	54	RND	113	RND	570
ANN	410	ANN	178	ANN	1133
KNN	1258	KNN	1351	KNN	2396

TABLE 4.17: Duration of prediciton for ML Models (In Seconds)

From the table we can broadly understand some common traits and relate them to the tables above for better performance evaluation. We can see that Decision Tree has the shortest amount of time across all datasets, but Random Forest has almost simlilar runtime as DT. their difference is very nominal and we can say that performance with respect to time is best for both the methods. It is a trade-off for Random Forest and Decision Tree as Random Forest gives better accuracy with nominally longer time, and Decision Tree takes slightly shprter time but underperforms than Random Forest.

The next two are Logistic Regression and ANN. As we can see from Table 4.15, ANN gives good accuracy, behind only DT and RND, but the duration is significantly higher than them. Logistic Regression takes time in the range of ANN, but it's accuracy is the worst of the six, as seen in Table 4.11. If we compare Table 4.1 to 4.6, we can see KNN gives accuracy very much on par with RND & DT. But in table 4.17, it can be seen that KNN takes a very long time to accomplish the accuracy, 20/30 fold of the time for the two best methods. So this is a disadvantage of using KNN. Support Vector Machines technique gives average to above average accuracy, but it takes similar time like KNN, which is a very long time compared to the other four methods.

Chapter 5

Conclusion and Future Study

This study tried to evaluate the performance and validity of Machine Learning algorithms in Intrusion Detection in conditions of huge data, with a variety of attack types. For that purpose, a methodology was built from the ground up, understanding the goal of these attacks and the popular type of attacks on network traffic. The ML methods were applied for two types of IDS detection, Signature Based (Multiple Class) and Anomaly Bases (2 Classes, Benign and Attack).

To explore new areas, old standards must be left. Which is why KDD Cup '99 dataset was not used as it had many redundant features and obsolete attack types, and also because it has been studied ample times. We collected three recently created datasets for this study. The dataset from University of New South Wales was selected for important attacks types like Fuzzers, Shellcode and Worms, which are known to make networks particularly vulnerable. CICIDS-2017 also had many popular attacks including DoS and DDoS. This is where the need for a dataset particularly devoted to DDoS attacks was realized, as these type of attacks were growing all over the world, which resulted in analyzing CIDDoS-2019, a huge dataset of over 50 million samples focused on DDoS attacks, including DNS and UDP type attacks.

After procuring and resizing the data to work according to our constraints, the focus was on creating these datasets with as relevant features as possible. Now the common Information Gain based Feature Selection, where Decision Tree method is used, was excluded from this study as we were facing a huge number of samples, and train the Tree very deep would very likely have caused in correlation, as a result high bias and high variance. To get rid of this problem Random Forest Regression method was used. With multiple tree processing and it's Bootstrap Aggregating or Bagging method, RND Regression got rid of the problem faced by a deeply trained tree, improving upon the method. Setting a threshold, 25, 32 and 24 features were taken for UNSW-NB15, CICIDS-17 and CICDDoS-19 respectively.

Then the datasets were trained and tested to produce prediction from each of the Six Machine Learning methods. Their predictions were compared using five metrics, Accuracy, Precision, Recall, F1-Score and ROC. The average duration time was also noted. After investigating the methods for Signature and Anomaly Based detection, some assessments were made.

5.1 Assessment

Comparing Table 4.1, 4.3 and 4.5 with Table 4.2, 4.4, 4.6, we can understand that Machine Learning gives better performance and can predict the type of a sample very accurately in case of Anomaly Type Detection. The performance deteriorates in case of Signature based detection. Overall when the Class type is increased, the accuracy of all six models decrease. Also examining table 4.1 to 4.6, we can also understand that the performance also relates with sample size. Every model had their worst performance in UNSW-NB15 dataset, which has the least amount of samples and ten class types. 10 % CICIDS-17 had more sample than that, and had a lesser number of class types, so the performance improved significantly. 2% DDoS-19 had half a million samples, and although it had ten class types, the models had a very good accuracy prediction, and for binary/2 class classification, their prediction was near perfect. Which proves, Machine Learning Algorithms are perfectly suited for processing huge amounts of network data, as the bigger the volume of data, the more accurate the models get at detecting cyber-attacks.

Evaluation the models against each other, we can understand from Table 4.11 to 4.17, it is evident that the best performance was given by Random Forest across all six dataset. Decision Tree model also gave performance very closer to Random Forest, and it is slightly faster than Random Forest. So these two are the best option to consider. The trade-off is accuracy and speed, where they are very close. The next best performance was by k-Nearest Neighbors, but the method takes the highest execution time which is a major setback. The same can be said for SVM too. Logistic Regression was the worst performing algorithm of the six. ANN had similar performance like kNN, but it takes significantly less execution time than kNN. However it should be noted that for logistical reasons number of hidden layers applied were less. Introducing larger amount of hidden layers will result in better performance but longer execution time.

In short, Machine Learning is a great tool for IDS. It performs better in Anomaly based detection system and in networks producing large amount of traffic. Finally, the best methods from the six applied in this study are Random Forest, and Decision Tree in a close second.

5.2 Future Study

In the wide world of Machine Learning and Artificial Intelligence, there can be ways of improving the detection system and examining further attack types. The opportunities include

- Evaluating Man in the Middle attacks and POST Application Layer attacks
- Deep learning techniques like Convolutional Neural Network, DNN and other Deep Learning methods.
- Giving proper importance to attacks of very few samples using SMOTE based feature selection techniques.
- Access to licensed network traffic processing environment will result in testing ML methods in dynamic testing scenario.
- Unsupervised learning using unlabeled data can be evaluated for processing dynamic data.

Bibliography

1. Othman, S.M., Ba-Alwi, F.M., Alsohybe, N.T. et al. Intrusion detection model using machine learning algorithm on Big Data environment. J Big Data 5, 34 (2018).

2. R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," South Afr. Comput. J., vol. 56, no. 1, pp. 136–154, 2015.

3. Wang, Yan & Yang, Kun & Jing, Xiang & Jin, Huang. (2014). Problems of KDD Cup 99 Dataset Existed and Data Preprocessing. Applied Mechanics and Materials. 667. 218-225.

4. Yongxin Liao, Fernando Deschamps, Eduardo de Freitas Rocha Loures & Luiz Felipe Pierin Ramos (2017): Past, present and future of Industry 4.0 - a systematic literaturereview and research agenda proposal, International Journal of Production Research,

5. WannaCry, 2017 The Hacker News. Retrieved 7 August 2018.

6. Zetter, K. (2014). Countdown to Zero Day: Stuxnet and the launch of the world's first digital weapon. Broadway books.

7. CERT, 2019. Australian Cyber security report 2019. Accessed: Jun 27, 2020.

8. Jararweh, Y.; Otoum, S.; Al Ridhawi, I. Trustworthy and sustainable smart city services at the edge.

9. Khoda, M.E., Imam, T., Kamruzzaman, J., Gondal, I. and Rahman, A., 2019. Robust Malware Defense in Industrial IoT Applications using Machine Learning with Selective Adversarial Samples. IEEE Transactions on Industry Applications.

10. Bae, S.I., Lee, G.B. and Im, E.G., 2020. Ransomware detection using machine learning algorithms. Concurrency and Computation: Practice and Experience, 32(18), p.e5422.

11. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, pp.41525-41550.

12. Zekri, M., El Kafhali, S., Aboutabit, N. and Saadi, Y., 2017, October. DDoS attack detection using machine learning techniques in cloud computing environments. In 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech) (pp. 1-7). IEEE.

13. Oña, D., Zapata, L., Fuertes, W., Rodríguez, G., Benavides, E. and Toulkeridis, T., 2019, October. Phishing Attacks: Detecting and Preventing Infected E-mails Using Machine Learning Methods. In 2019 3rd Cyber Security in Networking Conference (CSNet) (pp. 161-163). IEEE.

14. Cisco Annual Internet Report (2018-2023) –https://www.cisco.com, accessed Jan 20, 2021

15. Fireeye and Mandiant (2021), A Global Reset: Cyber Security Predictions, accessed Jan 20, 2021

16. D.E Rumelhart, J.L. McClelland and the PDP research group, Parallel Distributed Processing, vol. 1, MIT Press, 1986.

17. V.N. Vapnik, The Nature of Statistical Learning Theory. Springer, NY, 1995.

18. S. Haykin, Neural Networks – A comprehensive Foundation. Upper Saddle River, NJ: Prentice Hall, 1999.

19. V. Chercassky and P. Mullier, Learning from Data, Concepts, Theory and Methods. NY: John Wiley, 1998.

20. Quinlan JR, C4.5: programs for machine learning, vol. 1, Morgan kaufmann, 1993

21. Machine Learning, 24, 123-140 (1996) © 1996 Kluwer Academic Publishers. Boston. Manufactured in The Netherlands. Bagging Predictors LEO BBEIMAN Statistics Department, University qf Cal!'lbrnia. Berkele), CA 94720

22. Leif E Peterson. "K-nearest neighbor". In: Scholarpedia 4.2 (2009), p. 1883.

23. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15- Datasets/

24. https://www.unb.ca/cic/datasets/ids-2017.html

25. https://www.unb.ca/cic/datasets/ddos-2019.html

26. Ezukwoke, Kenneth & Zareian, Samaneh. (2019). LOGISTIC REGRESSION AND KERNEL LOGISTIC REGRESSION A comparative study of logistic regression and kernel logistic regression for binary classification.

27. Sperandei, Sandro. (2014). Understanding logistic regression analysis. Biochemia medica. 24. 12-8. 10.11613/BM.2014.003.

28. M. Sabhnani and G. Serpen, "Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set," Intell. Data Anal., vol. 8, no. 4, pp. 403–415, 2004

29. National Computer Network Emergency Technical Processing Coordination Center, The 2018 China Internet Network Security Report, People's Posts and Telecommunications Press, Beijing, China, 2019.

30. Olasehinde, Olayemi Alese, Boniface & Adetunmbi, Adebayo. (2019). Machine learning approach for information security. International Journal of Information and Computer Security. 16. 91-101.

31. Kurniabudi, Kurniabudi & Stiawan, Deris & Dr, Darmawijoyo & Idris, Mohd Bamhdi, Alwi Budiarto, Rahmat. (2020). CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection. IEEE Access. PP. 1-1.