Intrusion Detection Using Machine Learning

by

Adnan Athar Janwari
B.S., University of Sindh, Jamshoro, Pakistan, 2013
A Report Submitted in Partial Fulfillment of the Requirements for the Degree of
MASTER OF ENGINEERING
in the Department of Electrical and Computer Engineering

Intrusion Detection Using Machine Learning

by

Adnan Athar Janwari
B.S., University of Sindh, Jamshoro, Pakistan, 2013

Supervisory Committee

Dr. Fayez Gebali, Supervisor

(Department of Electrical and Computer Engineering)

Dr. M. Watheq El-Kharashi, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

Since the phenomenal growth in the usage of computer networks, concerns such as service availability, data integrity, and data confidentiality are becoming increasingly important. As a result, the network administrator must employ a variety of intrusion detection systems (IDS) to analyze traffic on the internet for unauthorized and hostile activity. The term "intrusion" refers to a malicious breach of security policy. As a result, an intrusion detection system analyzes the traffic passing via computer systems on a network to look for malicious activity and recognized threats and sends out warnings when it detects them. Machine learning algorithms are currently being widely used to develop efficient intrusion detection systems. Building an efficient intrusion detection system necessitates research into optimal ensemble methods. In this report, the CSE-CIC-IDS2018 dataset is utilized with eight network attacks and benign data. To identify the ten most important features, we have used Linear Discriminant Analysis (LDA). Naive Bayes, Random Forest (RF), and Decision Tree (DT) are Machine Learning (ML) techniques examined. The experiments are conducted with five-fold cross-validation utilizing the open-source software WEKA. The performance measures were used including execution time, accuracy, precision, F-measure, and recall. In terms of accuracy, precision, recall, and F-measure, the findings reveal that the Decision Tree surpasses the other methods.

Contents

List of Tables

List of Figures

List of Graphs

Abbreviation

AI .... Artificial Intelligence
AIDS ..... Anomaly-based Intrusion Detection System
API ....Application Programming Interface
CIC ..... .Canadian Institute for Cybersecurity
CPU ..... .Central Processing Unit
CSE ..... .Communications Security Establishment
DDoS ..... .Distributed Denial-of-Service
DoS ..... . Denial of Service
GNL ....General Public License
GUI ....Graphical User Interface
HIDS ....Host-based Intrusion Detection System
HTTP ..... .Hypertext Transfer Protocol
IDS ....Intrusion Detection System
IP ....Internet Protocol
IT .... Information Technology
KNN ..... .K-Nearest Neighbor
LDA ....Liner Decrement Analysis
ML ..... Machine Learning
NB ....Naïve Basyes
NIDS ..... .Network Intrusion Detection System
RAM ..... .Random Access Memory
RF..... .....Random Forest
SIDS ....Signature-based Intrusion Detection System
SIEM ..... .Security Information and Event Management
SMOTE .... Synthetic Minority Oversampling Technique
SVM .... Support Vector Machine
TD ....Decision Tree
UDP .... User Datagram Protocol
WEKA ....Waikato Environment for Knowledge Analysis

Acknowledgment

I am thankful to Almighty Allah, the Most Merciful, as well as my parents, spouse, and friends for their unwavering support, wishes, prayers, and encouragement. I'd want to thank my supervisor, Dr. Fayez Gebali, for all of his support, guidance, mentorship, flexibility, and encouragement throughout this program. I'm also grateful to the University of Victoria for proving an envoriment to learn.

Dedication

I dedicate this work to my daughter Irha Adnan Janwari

Chapter 1

Introduction

The information era has had a substantial impact on almost every industry. As a result, Information Technology (IT) industry has grown into a multibillion-dollar industry [1]. Data is increasingly seen as a more valuable resource than oil [2]. It is increasingly profitable to analyze data such as customer buying patterns. As a result, the public and private sectors have made significant investments in IT infrastructure and facilities, creating fears about security and privacy. Furthermore, the adoption of online data sources for enterprises has resulted in many cybercrimes.

In recent years, computer networks, facilities, and devices have faced a growing number of attacks. As a result, cybersecurity has emerged as a key concern. Detecting harmful behaviors, particularly those which have not happened, is a significant issue that requires immediate attention. One of the techniques to deal with cyberattacks is an Intrusion detection system (IDS). Through constant monitoring, these systems identify abnormal behavior or intrusions in a network. A traditional IDS, on the other hand, is incapable of processing massive amounts of data and may be slow to respond to new threats. Furthermore, as new threats arise and communication protocols evolve, old intrusion detection approaches that rely on detecting and comparing IP addresses and port numbers are becoming less useful. As a result, intrusion detection systems based on Machine Learning (ML) are now being designed. ML can find previously hidden trends or patterns in data and make accurate predictions. In this work, Linear Discriminant Analysis (LDA) is applied to discover the top ten most important attributes in the CSE-CIC-IDS2018 dataset, and three classifiers (Naive Bayes (NB), Random Forest (RF), and Decision Tree (DT) are assessed on the dataset to predict eight attacks and valid (benign) to find the best effective machine learning technique

1.1 Motivation

Cybercriminals have devised methods for detecting system flaws and exploiting network flaws [3]. According to IDC Data Services for Hybrid Cloud Survey, the rising frequency of cyberattacks and complexity in maintaining and administering security solutions are major problems for IT security specialists, data management experts, and company owners [3]. In March 2020, a hacker gained access to a portion of Sina Weibo's database, which contained the personal data of 538 million users, including email addresses, phone numbers, gender, and other social media records. As a result of this type of occurrence, attackers can use the information to conduct social engineering attacks. In another incident, A hostile third party gained access to the financial information, including names, email addresses, encrypted passwords, and user data connected to Quora. Financial loss, brand damage, productivity loss, and strained client relationships are all possible effects of these attacks. The purpose of this study is to analyze machine learning algorithms' accuracy, precision, recall, F-measure, and execution time to determine their suitability for IDS.

1.2 Related Work

Previous research on IDSs is presented in this section. Anderson et al. [4] were the first to notice the concept of intrusion detection. The author created a model of security surveillance for detecting anomalies in the actions of the user. Lee & Stolfo [5] established a systematic methodology for detecting intrusions that leverages data mining methods. Lippmann et al. [6] published comparative research of different intrusion detection categorization systems. Schultz et al. [7] built a system that uses various classification techniques to use a dataset of normal and malicious executable code to build classifiers so that they can classify new executables.

Hwang et al. [8] suggested a 3 layer IDS design that includes three lists: black list, white list, and multi-class. The white list contains regular traffic, the black list contains recognized traffic attacks, and the multi-class contains abnormalities discovered in regular traffic. Srinivasulu et al. [9] examined the performance of various classification techniques such as CART, Naive Bayes, and Artificial Neural Networks using a confusion matrix.

As a hybrid of Naive Bayes and Principal Component Analysis, Neetu et al. presented an IDS framework [10]. The study found that this approach can improve execution speed. Revathi et al. [11] examined the performance and accuracy of several classifiers. The greatest classifier is the algorithm with the maximum accuracy.

On the NSL-KDD dataset, Dhanabal et al. [12] compare the accuracy of J48, SVM, and Naive Bayes classification algorithms. Murthy et al [13] compare the effectiveness of Naive Bayes, J48, OneR, and RandomTree utilizing False Positive, True Positive, Accurately, and Inaccurately Classified Instances. On the above-mentioned parameters, the results showed that RandomTree outperformed the other three methods.

The CICIDS2017 IDS dataset was first introduced in [14]. Seven real-time attacks with 80 network traffic characteristics make up this dataset. The optimal feature set for identifying each attack was determined using the Random Forest Regressor feature reduction technique. The performance of these characteristics was then evaluated using seven ML classifiers. The Random Forest algorithm has the highest accuracy and the shortest execution time, according to the data. Furthermore, using the criteria for an ideal dataset in [15], a comparison was done between the CICIDS2017 dataset and 11 publicly available datasets. Except for CICIDS2017, it was determined that none of the datasets met the requirements [16]

As can be seen from the preceding discussion, writers in the literature choose the best classifier based on prediction accuracy when classifying instances. However, in our project, the following machine learning classifiers Naive Bayes, Decision Tree, and Random Forest were implemented, tested, and evaluated on the most recent publicly available dataset CSE-CIC-IDS2018 to find the best classifier that provides the maximum accuracy while requiring the least amount of time to build the model.

1.3 Report Organization

The report's structure is as follows.

Chapter 1 presented the problem and offered a high-level summary of the project. The project's associated work and motivation were described, as well as the report's format.

Chapter 2 details the IDS and categories, presents an overview of machine learning, and WEKA workbench

Chapter 3 presents the suggested framework, explains the dataset , data preprocessing steps, data splitting strategies, and ML classifiers for predicting benign and malicious traffic.

Chapter 4 gives details on the software, hardware, and test environment settings. The algorithms Naive Bayes (NB), Random Forest (RF), and Decision Tree (DT) are evaluated in terms of performance.

Chapter 5 provides conclusions and recommendations for further work on IDSs.

Chapter 2

2 Background

2.1 Intrusion Detection Systems

An Intrusion Detection System (IDS) is system that identifies, attack or policy violation and notify the network administrator or security experts using a system called Security information and event management (SIEM). SIEM receives information from a multitude of sources, resulting in hundreds of warnings to avoid false alerts it uses filtering algorithms to separates malicious and false alerts, which helps security experts to focus only on actual threats. Fig 2.1 presents the operation of an IDS system. An Intrusion Detection System is separated into two groups a network intrusion detection system (NIDS) and a host-based intrusion detection system (HIDS) are two types of intrusion detection systems

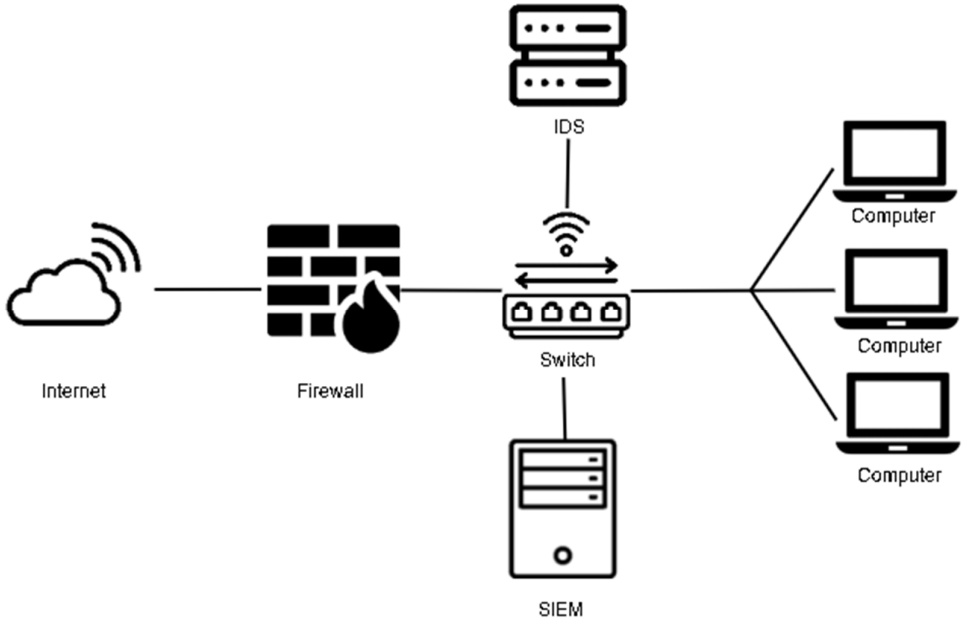


Figure 2.1: Presents the operation of an IDS system.

There are two main IDS types:

- Network IDS
- Host IDS

2.1.1 Network IDS

To defend traffic from exploitation, a Network IDS is often deployed at key locations in a network. NIDS compares the detected traffic to a database of previously identified attacks. NIDS is easy to install and intruders have a hard time detecting it. As a result, an attacker may be uninformed of the existence of a NIDS. Since it is unable to examine the payload of packets, a NIDS often scans a large quantity of network data and may overlook encrypted traffic. As a result, an administrator must keep a close eye on everything.

2.1.2 Host IDS

A Host Intrusion Detection System (HIDS) is a type of intrusion detection system (IDS) that is installed on systems and network devices that are connected to the internet. A HIDS may do a more thorough analysis of internal traffic than a NIDS and serves as a second line of protection against network threats. To detect changes, a HIDS evaluates system processes, memory areas, and properties such as modification dates, and encrypted content. It also examines the system registry, looks for unsuccessful login attempts, and looks for malicious software installations. There are two techniques for identifying invasions:

- Signature-based Intrusion Detection System (SIDS)
- Anomaly-based Intrusion Detection System (AIDS)

2.1.2.1 Signature-based IDS

Signature-based IDS scans network data for certain characteristics, like byte sequences or familiar malicious processes to identify potential threats. These identified patterns are referred to as signatures in antivirus software. While signature-based IDS may better recognize common threats, new attacks for which no signature exists are hard to detect.

2.1.2.2 Anomaly-based IDS

SIDS' shortcomings can be solved with an Anomaly-based Intrusion Detection System (AIDS). AIDS generates the standard model for computer system behavior using machine learning, knowledge-based, or statistical-based methodologies. A significant difference between models and observed behavior that can be viewed as an intrusion is defined as an anomaly. Malicious behavior, it is assumed, differs from typical user behavior. The two AIDS steps are training and testing. The model is trained using network traffic data that have been extracted and then evaluated against unseen analysis to find accuracy. Since it does not depend on signatures or patterns to detect threats, AIDS is particularly good at detecting zero-day threats [17]. AIDS raises a warning when observed behavior differs from usual activities.

2.2 Machine Learning

Machine learning (ML) is a subset of Artificial Intelligence (AI) that allows computers to learn without having to be directly programmed. It has roots in statistics, data mining, and computational mathematics. ML algorithms are trained on a dataset to learn the data and predict accordingly. ML is widely used in IDSs. The types of ML algorithms are described below.

Types of Machine Learning Algorithms

ML algorithms can be split into three main categories.

- Supervised learning
- Unsupervised learning
- Reinforcement learning

2.2.1 Supervised Learning

Supervised learning is by far the most extensively used ML approach. It's the most straightforward approach to use. A supervised machine learning algorithm is trained on labeled data to produce the intended output [18] and receives feedback on whether the anticipated output is correct or incorrect. When the algorithm has been trained properly, it will observe fresh cases and make label predictions. Task-oriented learning is another term for supervised

learning. Several implementations for this sort of learning exist, including spam detection, facial recognition, and advertisement popularity. Supervised learning is divided into two categories.

• Regression: methods for predicting continuous quantities such as earnings, age, and cost.

• Classification: algorithms that predict discrete values like yes or not, as well as true or false. Binary and multi-class classifications are also possible.

Naive Bayes, Decision Trees, KNN, Neural Networks, SVM, and Linear Regression are among some of the supervised learning algorithms.

2.2.2 Unsupervised Learning

Unsupervised learning uses an unlabeled dataset to find unseen patterns in the data [19]. As an outcome, no labeling is required to train the model, but the output is less accurate than supervised learning. Unsupervised learning is a good option, when data is not available for intended results, such as when a corporation wishes to define a target market for a newly introduced product [19]. Clustering and dimensionality reduction challenges are solved via unsupervised learning. Hidden Markov models, K-means clustering, and Gaussian mixture models are some unsupervised learning approaches.

2.2.3 Reinforcement Learning

In reinforcement learning model learns from feedback it gets as a result of its events and experiences. It works in the same way that humans think and learns from their surroundings. Computer games, industrial simulation, and resource management all involve reinforcement learning. The temporal difference, Q-learning, and Deep adversarial networks are some of the most prominent reinforcement learning techniques.

2.3 The WEKA Workbench for Machine Learning

WEKA (Waikato Environment for Knowledge Analysis) is a free program created at the University of Waikato in New Zealand and distributed under the GNU General Public License [20]. It includes data preparation, classification, regression, clustering, mining of association rules, and visualization tools [21]. WEKA was created using the Java programming language and is accessible via both a graphical user interface (GUI) and a command-line interface (CLI) Fig 2.2 shows the WEKA GUI interface. The explorer to play around with the data and test the things out, the Experimenter for controlled experiments, and the KnowledgeFlow for graphically modeling ways to work on problem in the WEKA workbench.

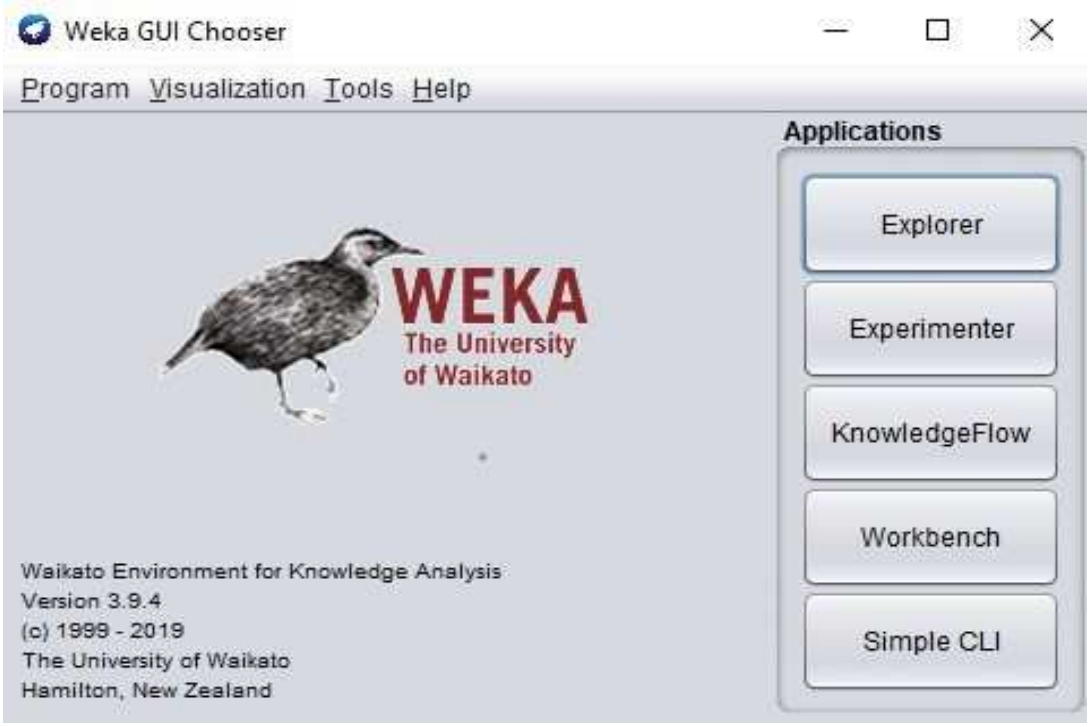


Figure 2.2: The WEKA GUI Interface.

The WEKA GUI explorer application is used in this project which contains six tabs Preprocess, Classify, Cluster, Associate, Select Attributes and Visualize. WEKA preprocessing tab supports filters such as removal of duplicate value, data normalization, under-sampling, handling with missing value, and much more to process the data on the instance and attribute level, Table 2.1 shows the list of WEKA filters we have used in the data pre-processing phase.

Table 2.1: List of WEKA filters used in this project

RemoveDuplicates
SMOTE
Randomize
Normalize
MultiClassFLDA

WEKA supports multiple file formats, however, *.arff* is the native file format for WEKA. Fig 2.3 shows the list of file formats supported by WEKA.

Figure 2.3: File formats supported in WEKA.

The preprocess tab shows the dataset's summary, including the number of instances, attributes, missing values, and labels. ML activities such as removing duplicate values and numerous undersampling and oversampling approaches are supported through a filter option [22]. Figure 2.4 illustrates the dataset's information in the WEKA preprocess panel.

ML methods can be selected from the classifying panel after data preparation. WEKA supports KNN, C4.5 (decision trees), MLP, RF, SVM, and Simple Logistic classifiers for categorical and numeric predictions.

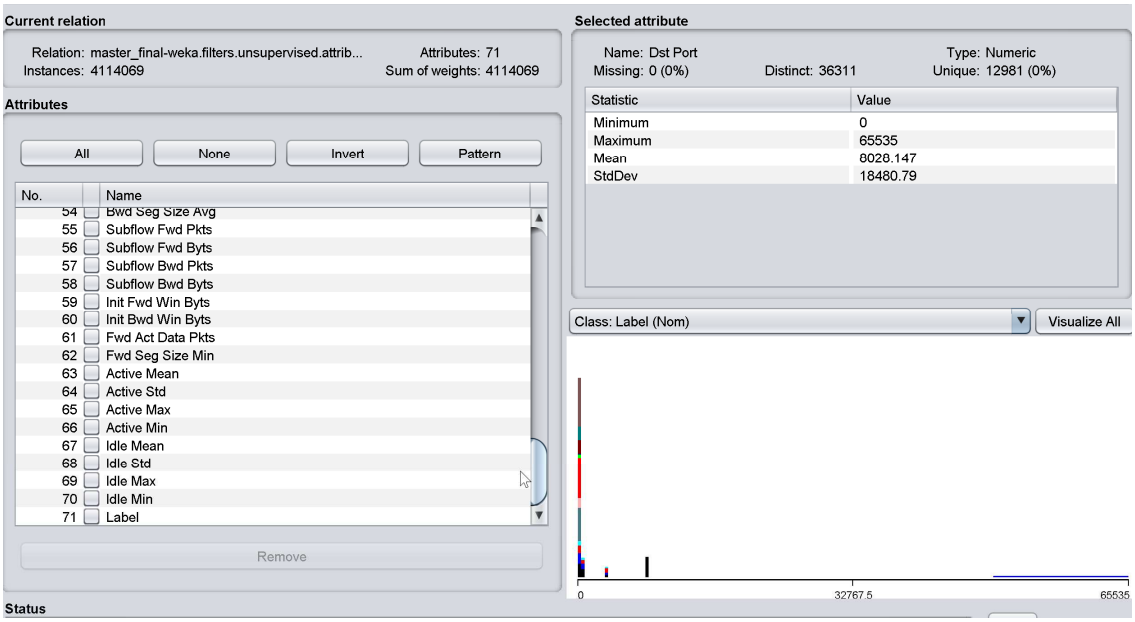


Figure 2.4: The dataset overview in WEKA.

WEKA offers two options to train and test a model, cross-validation and percentage split. Cross-validation divides the dataset using a number of folds whereas percentage split divides in explicit training and testing sets. Class labels must be specified before building and evaluating the model. In this project, nine labels are used to identify malicious and benign data. WEKA classifier output window where we can find the model output result. The output results show the performance and confusion matrix of each class. The output can be saved in .txt, .csv or .xml file format

Chapter 3

3 Proposed Framework

Figure 3.1 illustrates the suggested framework, which specifies the stages for the ML model is being trained and tested.

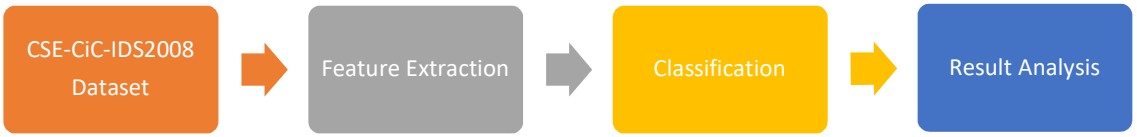


Figure 3.1: The proposed framework

3.1 CSE-CIC-IDS2018 Dataset

The CSE-CIC-IDS2018 dataset was created as part of a collaboration between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) (CSE)

This dataset includes 14 different attacks, namely Bot, Infiltration, SSH-BruteForce, FTP-BruteForce, Brute Force-Web, Brute Force-XSS, SQL-Injection, DoS attacks-Hulk, DoS attacks-GoldenEye, DoS attacks-Slowloris, DoS attacks-SlowHTTPTest, DDoS attack-LOIC-UDP, DDoS attack-HOIC, DDoS attacks-LOIC-HTTP and Benign [23].

This research considers eight attacks and benign data. The specifics of these attacks are listed below. Graph 3.1 shows attack names and benign traffic with number of instances used in our project.

3.1.1 Brute Force Attack

By guessing login details, a brute force attack attempts to get access to a computer system. Brute force attacks were estimated to be responsible for 5% of all reported cybersecurity events. Certain attackers utilize scripts and tools as brute force approaches. Other attackers have obtained access to databases of commonly used passwords as well as valid usernames and passwords via data breaches, or the dark web.

3.1.2 Botnet

A botnet is a group of infected computers linked to a malicious host computer known as a bot. By providing instructions to the system, an attacker may instruct this bot to conduct any malicious actions. Zeus and Ares, two commonly known bots, were used to build the dataset. Ares is a DDoS-initiating malware, whereas Ares is a DDoS-initiating malware that reproduces by accessing files on an affected computer or via phishing methods.

3.1.3 Denial of Service (DoS)

An attack known as a denial of service (DoS) occurs when an attacker overwhelms a server with traffic, forcing resources inaccessible to normal users. DoS attacks are launched using Slowloris, Gold-enEye, and Hulk. Slowloris attempts to take down a server by sending incomplete Hypertext Transfer Protocol (HTTP) requests to establish connections between the attacker's computer and the targeted system while bypassing other ports and services. It attempts to keep a connection for as long as possible to overload and slow down the target server.

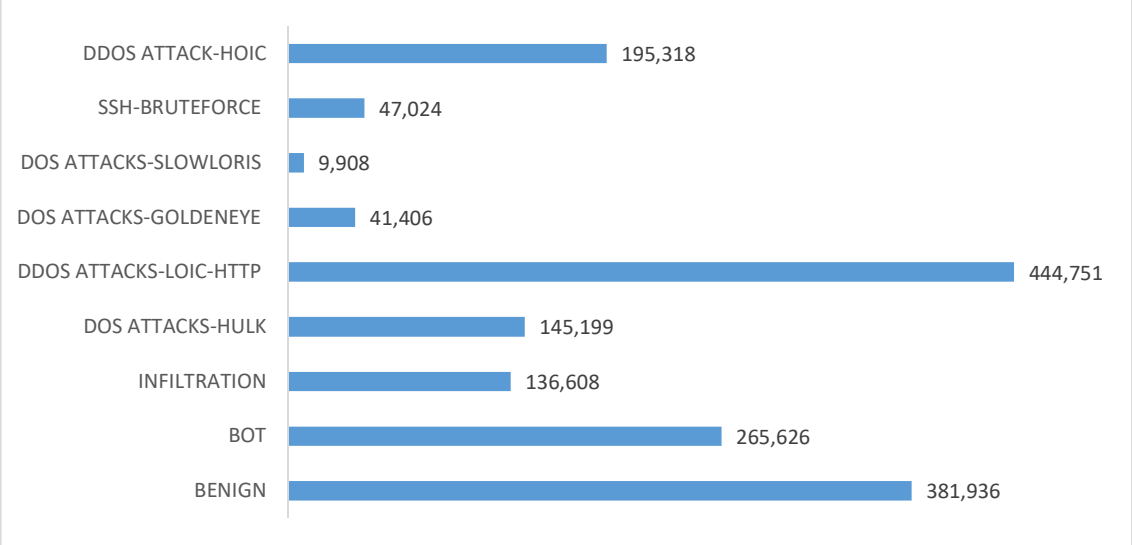
3.1.4 Distributed Denial of Service (DDoS)

DDoS attacks operate in the same way as DoS attacks, but they are conducted via a network of linked devices to drain the resources of a target system. DDoS attacks may be carried out using a variety of methods, including the High Orbit Ion Cannon (HOIC) and Low Orbit Ion Cannon

(LOIC). The open-source flooding programs HOIC and LOIC were created for network stress testing [23]. These tools might produce a lot of network traffic to utilize a lot of network capacity. Flooding HTTP and User Datagram Protocol (UDP) packets against numerous URLs at the same time might start a DDoS attack.

3.1.5 Infiltration

When a person downloads malicious software or a questionable document, this results in infiltration. On the target computer, the Metasploit program is used to install a backdoor. This enables remote monitoring, data breaches, and server takeover without the need for consent or knowledge.



Graph 3.1: shows attack names and benign traffic with number of instances used in our project

3.2 Feature Extraction

Real-world data is generally messy, incomplete with missing values, and inappropriate for creating machine learning models or solving complicated data-related issues directly. There could be incorrect data, or the data could be in an unorganized, unstructured, and unformatted state.

Data that has been preprocessed is far more important than the most complex algorithms. Clean

data produced a considerably more accurate model than unprocessed data, according to research.

Below is the list of the steps I follow to preprocess the CSE-CIC-IDS2018 dataset.

3.2.1 Data Cleaning

Data cleaning is the act of eliminating or changing data that is inaccurate, missing, unnecessary, duplicated, or incorrectly structured to prepare it for analysis.

Remove duplicate or irrelevant observations

Zero Value columns: The dataset has zero value columns Bwd URG, Fwd Byts, Fwd Pkts, Fwd Blk Rate, Bwd Byts, Bwd Pkts, and Bwd BlK which was removed. After removal, the number of attributes was reduced to 71. Fig 3.2 shows zero-value colums

Figure 3.2: Zero-value columns

Duplicated value: We used the 'RemovedDuplicate" filter to remove the duplicate value and it reduced the numbers of instances (rows) from "4114069" to "2547151".

Removed the class with a few instances: Classes like Dos attacks-SlowHTTPTest, Brute Force-Web, Brute Force –XSS, SQL Injection, FTP-BruteForce, and DDOS attack-LOIC-UDP

had less than 100 instances so we removed them to produce a more accurate model.

3.2.2 Data Transformation

Data Normalization

Data normalization is a method of rescaling one or more features to a range from zero and one. It means that every feature has the highest value of 1 and a minimum value of 0 for each value.

Randomization

By shuffling the order in which the instances are sent through it, we can ensure that each data point has an independent effect on the model, without being influenced by the points before them.

Dimensionality reduction using Linear Discriminant Analysis LDA

The linear discriminant analysis (LDA) method is a traditional statistical approach to supervised dimensionality reduction and classification. LDA achieves optimum class discrimination by computing an ideal transformation by simultaneously decreasing the internal class distance and increasing the distance among classes [25]. LDA reducsed dimension of 71 attributes to 11. Fig 3.3 shows the LDA feature reduction in WEKA

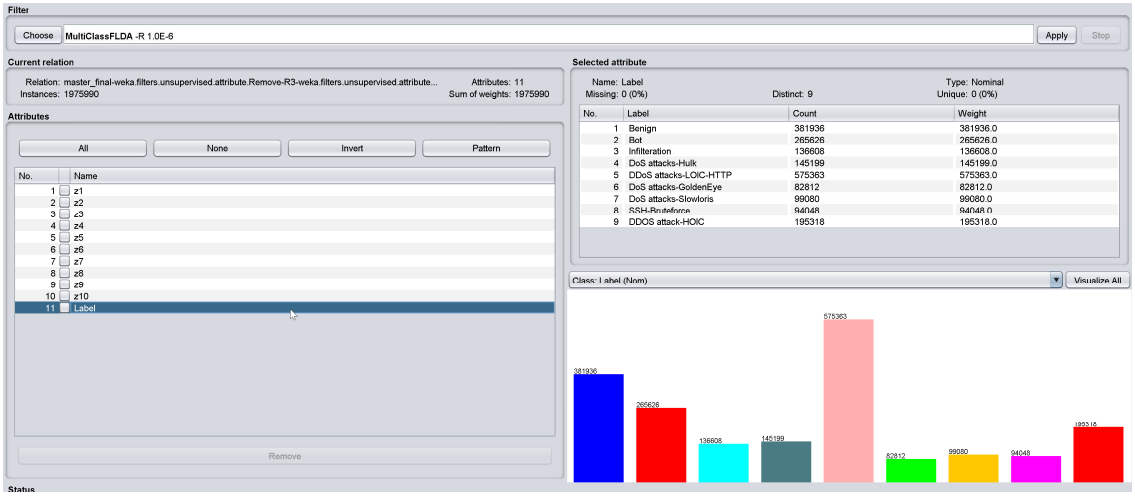


Figure 3.3: LDA feature reduction in WEKA.

3.3 Data Splitting

How to use the data that is accessible is a crucial decision. K-fold cross-validation and percentage split are the most prevalent approaches for separating data for training and testing. Percentage split is a quick and easy way for dividing data into training and test groups. The 80:20 and 70:30 division ratios are the most often utilized split ratios. Using a training set a model is created, whereas the model is tested on a test set. It is critical not to utilize the test set before the testing stage since this might skew the findings and impact model assessment.

The data in this project is subjected to k-fold cross-validation with k = 5, where k is the number of randomized and equivalent data partitions. It is a commonly used data resampling approach in machine learning. In comparison to percentage split, it generates less biased outcomes [26]. As illustrated in Figure 3.1, the model is trained using k-1 partitions, while the other is utilized as the test set [27]. The cross-validation accuracy is the mean of the k outcomes after this process is performed k times.

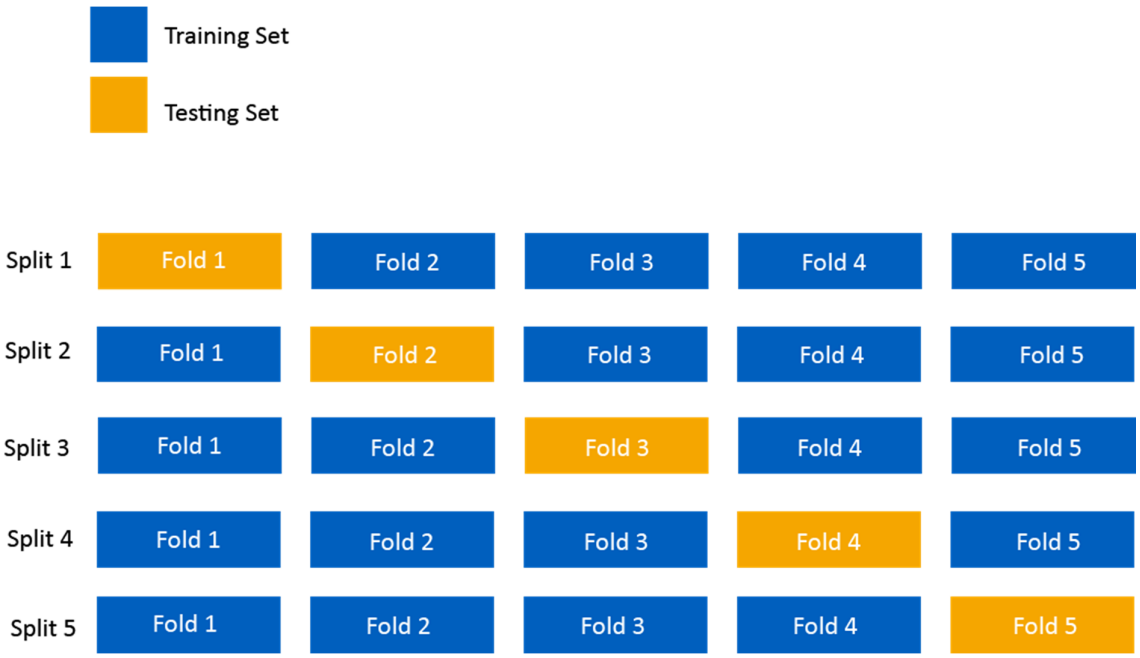


Figure 3.4: Five-fold cross-validation process.

3.4 Machine Learning Classifiers

The machine learning classifiers that were used to classify are listed below.

3.4.1 Naive Bayes (NB)

The Naive Bayes classifier is an simple version of the Bayesian probability model. The NB is based on the premise of solid independence [28]. Naive Bayes employs a basic form of the Bayes Theorem wherein the prior possibility for every class is calculated from the training set and is expected to be unrelated to one another. Although this approach simplifies the computation of probabilities, it is impractical because we expect the variables to interact and be dependent. Even under this unrealistic assumption, Naive Bayes is a highly efficient classification approach.

Naive Bayes is used to determining the posterior probability for each class, and the class with the highest probability is predicted. As a consequence, it can solve issues involving both binary and multi-class categorization.

3.4.2 Random Forest (RF)

Random Forest (RF) is a machine learning classifier that averages the results of many decision trees applied to diverse subsets of a dataset to enhance prediction accuracy. A condition is compared to one or more aspects of the incoming data at each tree node. Instead of relying on a single decision tree, RF obtains results by combining predictions from many trees [29]. The predicted class is the one with the most votes, as determined by each tree's vote. On unbalanced datasets, RF works admirably, with a low number of classification mistakes.

3.4.3 Decision Tree (DT)

A decision tree is a flowchart-like tree in which each internal node represents a test on an attribute, each branch represents a test result, and each leaf node represents a class. [29]. Because of their simplicity [30], intelligibility [31], ease of implementation, decision tree construction is quicker and produces greater accuracy than other classification algorithms, it is one of the most extensively used techniques [32] and is extremely popular among researchers [33]. Reduced Error Pruning Tree ("REPTree") is the fastest decision tree learning algorithm that builds a decision tree based on information gain or variance reduction. REP with back

overfitting is the basic pruning approach used by this algorithm. It neatly organizes numerical attribute values once and resolves missing values in fractional instances with C4.5's embedded technique. This algorithm uses the approach from C4.5, and the fundamental REP is also counted in its process [34].

3.5 Model Building and Testing

In this Step, the models are trained and evaluated on the CSE-CIC-IDS2018 dataset and the features chosen during the dimensionality reduction stage. To evaluate the classifiers' efficiency, five-fold cross-validation is employed for training and testing.

Chapter 4

4 Performance Evaluation

Experimental findings for three supervised machine learning models, namely Naive Bayes (NB), Random Forest (RF), and Decision Tree (DT) are presented in this chapter. The tests were carried out on a personal computer using the hardware and software characteristics listed in Table 4.1

Table 4.1: The hardware and software parameters.

Item	**Specification**
Manufacturer	Dell
Model	XPS 15 (7590)
Operating System	Windows 11 Professional
System Type	64-bit Operating System, x64-based processor
Processor Type	Intel(R) Core(TM) i7-9750H
Installed Memory (RAM)	32 GB
Processor Speed	2.60 GHz
Number of Cores	6
Number of Threads	12
Machine Learning Tool	WEKA version 3.9.5

4.1 Evaluation Metrics

Constructive feedback is at the heart of machine learning models. Models are created, assessed using metrics, and then tweaked to boost efficiency. For classification, a variety of metrics are used. The performance metrics used to assess the ML classifiers for intrusion detection in this study are listed below.

Accuracy is the number of correct classifications either as an attack or benign out of all instances in a dataset

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \qquad Eq.\ i$$

where true negative (TN) is the number of correct classifications of benign as benign.

Precision is the ratio of true positives to the total of false and true positives

$$Precision = \frac{TP}{TP+FP} \qquad Eq.\ ii$$

where true positive (TP) is the number of attacks classified correctly and false positive (FP) is the number of incorrect classification of benign as an attack.

Recall is the ratio of true positive to the sum of false-negative and true positive

$$Recall = \frac{TP}{TP+FN} \qquad Eq.\ iii$$

where false negative (FN) is the number of incorrect classification of an attack as benign.

F-Measure is the weighted harmonic mean of the recall and precision

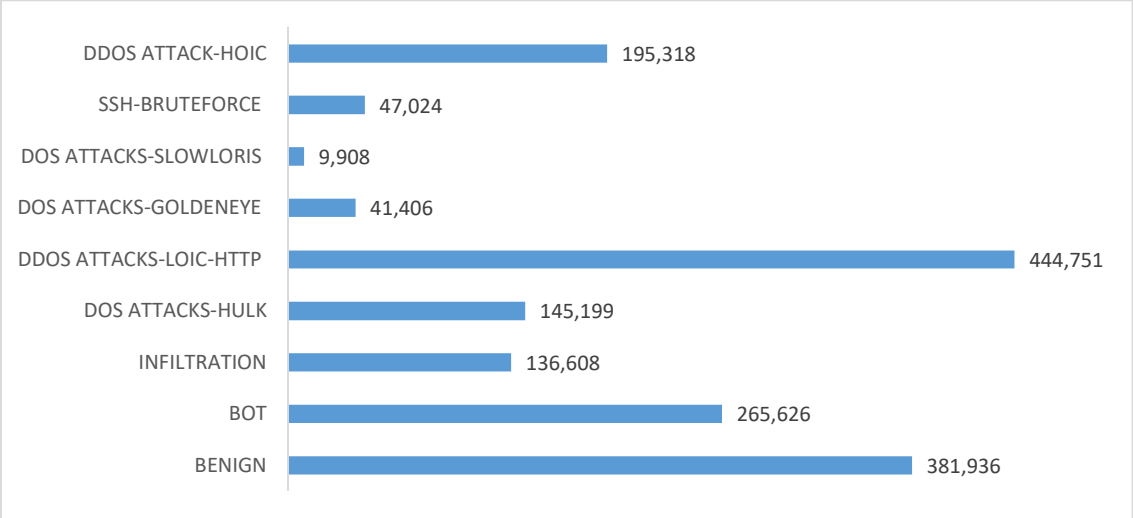
$$F\text{-}Measure = \frac{2TP}{2TP+FP+FN} \qquad Eq.\ iv$$

Execution Time is the amount of time spent on training and testing the classification model.

The accuracy, precision, recall, and F-measure are all presented as percentages in this chapter

4.2 Performance of the Classifiers

The ML models are evaluated using data from the CSE-CIC-IDS2018 dataset for eight attacks and benign traffic. In model training, the distribution of data is critical. In classification challenges, a balanced dataset can yield superior results, but an imbalanced dataset might lead to biased predictions. As a result, before model training, it is critical to balance the dataset. The strategies of the combination of undersampling and oversampling are employed to generate a balanced dataset, and they are discussed here.



Graph 4.1: The dataset without applying under and oversampling

4.2.1 Performance without Oversampling or under-sampling using 5-fold Cross-validation

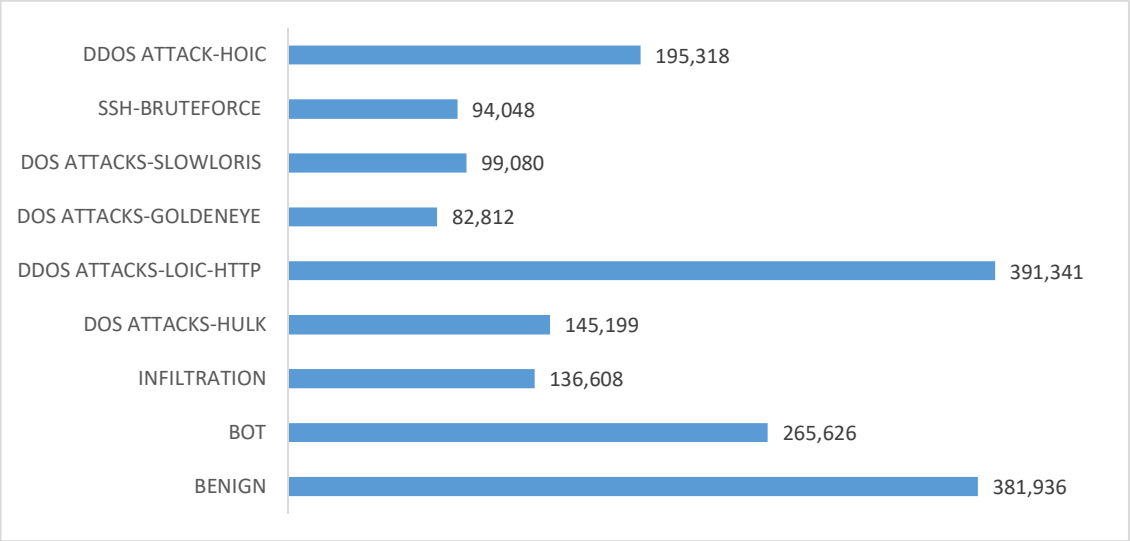
The dataset without oversampling or undersampling is addressed in this section. The dataset distribution is represented in Graph 4.1, with a total of 1,714,800 occurrences (rows). The performance evaluation results of the ML classifiers are shown in Table 4.2. The Decision tree classifier has the greatest accuracy, precision, recall, and F-measure at 93.50, 93.2, 93.5, and 92.6, followed by Random forest at 92.99, 92.5, 93.0, 92.7, and Naive Bayes at 76.10, 82.8, 76.1, and 75.0. However concerning execution time, Naive Bayes had the shortest duration of 14.17 seconds, followed by DT at 351.4 seconds, and RF at 4165.05.

Table 4.2: Performance of classifiers without under and oversampling.

Result	Naive Bayes	Random Forest	Decision tree
Accuracy	76.10	92.99	93.50
Precision	82.8	92.5	93.2
Recall	76.1	93.0	93.5
F-Measure	75.0	92.7	92.6
Execution Time (s)	14.17	4165.05	351.64

4.2.2 Performance with Oversampling and Undersampling using 5- fold Cross-validation

The ML algorithms in this section are evaluated using both oversampling and undersampling. Oversampling is a strategy for balancing a dataset by replicating minority class samples. Although random oversampling is a common approach for repeating examples, it can lead to overfitting in some classification models. The Synthetic Minority Oversampling Technique (SMOTE) was established in [35] to overcome this overfitting problem. This strategy has been demonstrated to operate in a broad variety of conditions. It creates new synthetic instances by comparing feature similarity spaces between different instances [36]. The K-nearest neighbors of minority data points are chosen, and linear interpolation is used to produce a new minority instance between these data points to create a synthetic instance. In this study, SMOTE was applied individually on the DoS attacks-GoldenEye, DoS attacks-Slowloris, and SSH Bruteforce classes, whereas under-sampling was used to balance the data by removing instances of the majority classes. Resample is a WEKA undersampling filter that randomly removes instances from the majority classes. Undersampling has the disadvantage of removing valuable information. To lower the class size, DDOS ATTACK-LOC-HTTP uses undersampling. Using resample with a percentage value of 12% from the majority classes were removed. New synthetic instances were produced repeatedly for the minority classes using SMOTE with a percentage parameter between 100 and 200. Each class's number of instances is displayed in Graph 4.2, while the performance of the ML classifiers is shown in Table 4.3. DT has the best accuracy with 93.39, followed by RF 93.28 and NB 84.14. In terms of execution time, NB has the fastest time of 4.19s, followed by DT 94.66s and RT 2152.78s.



Graph 4.2: The numbers of dataset instances after oversampling and undersampling.

Table 4.3: Result of all the classifiers after under and oversampling

Result	Naive Bayes	Random Forest	Decision tree
Accuracy	84.14	93.28	93.39
Precision	82.9	92.8	93.2
Recall	84.1	93.3	93.4
F-Measure	82.7	92.8	92.5
Execution Time (s)	4.19	2152.78	94.66

4.3 Discussion

DT and RF performed better than NB in terms of accuracy, precision, recall, and F-measure without oversampling or undersampling using 5-fold cross-validation. However, when it comes to execution time, NB was the fastest at 14.19 s. (Table 4.2).

With oversampling and undersampling, DT had the highest accuracy at 93.39% whereas, as per execution time, NB was the fastest at 4.19 s. RF had almost similar results in terms of accuracy at 93.28% (Table 4.3) however it was the slowest classifier with the execution time was slowest at 2152.78 s

Overall, in terms of accuracy with under and oversampling NB has a major difference with an increase of 8.4% followed by an approximately 1% rise in the RF however we can see a slight dip of .11% in the performance of DT. The major difference can be noticed in the execution time where DT perform 73% faster followed by 70.43% NB and 48.31% improvement can be seen in with the RF (Table:4.3)

Chapter 5

Conclusion and Future Work

Machine Learning (ML) approaches were employed in this research to counter eight network attacks: Infiltration, Bot, Dos Attack-Hulk, DDos attack-LOIC-HTTP, Dos attack GoldenEye, Dos attack Slowloris, SSH Bruteforce, and Benign class. The CSE-CIC-IDS2018 dataset from the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) was used. To balance the dataset, under and over sampling is used, where SMOTE was used to oversample and Resample filter was used to undersample. To classify the attacks and benign class, three supervised ML classifiers were used DT, RF, and NB. The classifiers were trained and tested using five-fold cross-validation. For the evaluation, the performance measures accuracy, precision, recall, and F-measure were employed. According to the data, DT had the best performance and an average execution time. The execution time for RF was the longest. The execution time for NB was the shortest, and the performance results were somewhat lower than for DT and RF. Overall, it's been determined that DT offers the optimal balance of execution time and performance.

5.1 Future Work

For future work, attack envoriment can be created to samuate real attack and test the model on it also other classifiers can be considered. Further, unsupervised ML algorithms can be employed with the CSE-CIC-IDS2018 dataset. For data splitting, techniques other than *k*-fold cross-validation such as percentage split can be employed. WEKA supports supervised and unsupervised classifiers written in Python (ScikitLearnClassifier), so it can be used to implement other algorithms and the results compared to those in this report.

Bibliography

[1] Kushairi, A., Singh, R., & Ong-Abdullah, M. (2017). *The oil palm industry in Malaysia: Thriving with transformative technologies.* J Oil Palm Res, 29(4), 431-439.

[2] *The world's most valuable resource is no longer oil, but data.* The Economist, 2017/05/06. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data/

[3] Phil Goodwin, Andrew Smith, and Robert Westervelt. *Addressing cyber protection and data protection holistically.* 2019/06. https://dl.acronis.com/u/rc/WP_IDC_Acronis_Cyber_Protection_EN-US_190626.pdf/

[4] J. P. Anderson, "*Computer security threat monitoring and surveillance,*" Technical Report, Fort Washington, Pennsylvania, USA, 1980.

[5] W. Lee and S. J. Stolfo, "*Data mining approaches for intrusion detection*" in Proceedings of the 7th conference on USENIX Security Symposium, vol. 7, San Antonio, TX, 1998.

[6] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "*The 1999 DARPA off-line intrusion detection evaluation*" in Computer Networks, vol. 34, no. 4, 2000, pp. 579-595.

[7] M. G. Schultz, E. Eskin, E. Zadok, S. J. Stolfo, "*Data Mining Methods for detection of New Malicious Executables*", in IEEE Symposium on Security and Privacy, Columbia University, 14-16 May 2000, pp.38-49.

[8] T. Hwang, T.Lee, and Y. Lee, "*A Three-tier IDS via Data Mining Approach*" in Proceedings of the 3rd annual ACM workshop on Mining network data, 2007, pp. 1-6.

[9] P. Srinivasulu, D. Nagaraju, P. R. Kumar, and K. N. Rao, "*Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison*" in IJCSNS International Journal of Computer Science and Network Security, vol. 9, no.6, 2009, pp. 11-18.

[10] B. Neethu, "*Classification of Intrusion Detection Dataset using machine learning Approaches*" in International Journal of Electronics and Computer Science Engineering, vol. 1, 2012, pp. 1044-51.

[11] S. Revathi, Dr. A. Malathi, "*A Detailed Analysis on NSL-KDD Dataset Using Various Machine*

Learning Techniques for Intrusion Detection", in International Journal of Engineering Research & Technology (IJERT), vol. 2 no. 12, 2013, pp. 1848-1853

[12] L. Dhanabal, Dr. S.P. Shantharajah, "*A Study on NSL-KDD Dataset for IntrusionDetection System Based on Classification Algorithms*" in International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 6, 2015, pp. 446-452.

[13] P. C. Murthy, Dr. A. S. Manjunatha, A. Jaiswal, B. R. Madhu, "*Building Efficient Classifiers For Intrusion Detection With Reduction of Features*" in International Journal of Applied Engineering Research, vol. 11, no. 6, 2016, pp. 4590-4596

[14] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. *Toward generating a new intrusion detection dataset and intrusion traffic characterization.* International Conference on Information Science and Security. vol. 1, pp. 108-116, Funchal, Madeira, Portugal, Jan. 2018.

[15] Amirhossein Gharib, Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. *An evaluation framework for intrusion detection dataset*. IEEE International Conference on Information Science and Security. pp. 1-6, Pattaya, Thailand, Dec. 2016.

[16] Almseidin Mohammad, Maen Alzubi, Szilveszter Kovacs, and Mouhammad Alka- sassbeh. *Evaluation of machine learning algorithms for intrusion detection system.* IEEE International Symposium on Intelligent Systems and Informatics. pp. 277- 282, Subotica, Serbia, Sep. 2017.

[17] Ammar Alazab, Michael Hobbs, Jemal Abawajy, Moutaz Alazab. *Using feature selection for intrusion detection system.* IEEE International Symposium on Communications and Information Technologies. pp. 296-301, Gold Coast, QLD, Australia, Oct. 2012.

[18] *Supervised Learning By: IBM Cloud Education.* Aug. 2020.
https://www.ibm.com/cloud/learn/supervised-learning/

[19] *DataRobot.Unsupervised machine learning*.
https://www.datarobot.com/wiki/unsupervised-machine-learning/

[20] Witten, Ian H.; Frank, Eibe; Hall, Mark A.; Pal, Christopher J. (2011). "*Data Mining: Practical machine learning tools and techniques, 3rd Edition*". Morgan Kaufmann, San Francisco (CA). Retrieved 2011-01-19.

[21] *WEKA 3: Machine Learning Software in Java*. WEKA 3 - Data Mining with Open Source Machine Learning Software in Java. (n.d.). Retrieved March 6, 2022, from https://www.cs.waikato.ac.nz/ml/WEKA/

[22] Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. *WEKA manual version 3-8-1*. University of Waikato, Hamilton, New Zealand, Dec. 2016.

[23] *Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)* https://registry.opendata.aws/cse-cic-ids2018.

[24] *HOIC (High Orbit Ion Cannon).* https://www.radware.com/security/ddos-knowledge-center/ddospedia/hoic-high-orbit-ion-cannon/

[25] Datti, Rupali, and Bhupendra Verma. "*B.: Feature reduction for intrusion detection using linear discriminant analysis.*" International Journal on Engineering Science and Technology. 2010.

[26] Prashant Gupta. *Cross-validation in machine learning.* Jun. 2017. https://towardsdatascience.com/cross-validation-in-machine-learning- 72924a69872f/

[27] Sourav Sen Gupta. *Decision trees. Towards data science.* 2017. https://towardsdatascience.com/decision-trees-in-machinelearning/

[28] Wafa' S.Al-Sharafat, and Reyadh Naoum "*Development of Genetic-based Machine Learning for Network Intrusion Detection*" World Academy of Science, Engineering and Technology 55, 2009

[29] D. Lavanya, and K. U. Rani, "*Performance Evaluation of Decision Tree Classifiers on Medical Datasets*," International Journal of Computer Applications (0975-8887), Vol 26, No.4, 2011.

[30] J. Su, and H. Zhang, "*A Fast Decision Tree Learning Algorithm*," America Association for Artificial Intelligence (www.aaai.org), 2006.

[31] D. Lavanya, and K. U. Rani, "*Performance Evaluation of Decision Tree Classifiers on Medical Datasets,*" International Journal of Computer Applications (0975-8887), Vol 26, No.4, 2011.

[32] X. Wu, and V. Kumar, "*The Top Ten Algorithms in Data Mining,"* Data Mining and Knowledge Discovery Series, CRC Press, United States of America, 2009.

[33] J. Marques de Sa, R. Sebastiao, and J. Gama, "*Tree Classifiers Based on Minimum Error Entropy Decisions,*" Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognation, Vol. 2, No. 3, 2011.

[34] Mohamed, W. N. H. W., Salleh, M. N. M., & Omar, A. H. (2012, November). *A comparative study of reduced error pruning method in decision tree algorithms*. In 2012 IEEE International conference on control system, computing and engineering (pp. 392-397). IEEE.

[36] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. *SMOTE: Synthetic minority over-sampling technique.* Journal of Artificial Intelligence Research. vol. 16, no. 1, pp. 321-357, Jan. 2002