REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

CYBER SECURITY AWARENESS POLICY IN SMALL BUSINESS USING MACHINE LEARNING

Mustafa Khalid Abdulateef ALGBURI

Master's Thesis

Supervisor

Asst. Prof. Dr. Hakan KOYUNCU

İstanbul, 2024

CYBER SECURITY AWARENESS POLICY IN SMALL BUSINESS USING MACHINE LEARNING

Mustafa Khalid Abdulateef ALGBURI

Electrical and Computer Engineering

Master’s Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled CYBER SECURITY AWARENESS POLICY IN SMALL BUSINESS USING MACHINE LEARNING prepared by MUSTAFA KHALID ABDULATEEF ALGBURI and submitted on 26/01/2024 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering

Asst Prof .Dr. Hakan KOYUNCU
Supervisor

Thesis Defense Committee Members:

Asst Prof .Dr. Hakan KOYUNCU	Department of Computer Engineering, Altınbaş University	_____
Asst.Prof .Dr. Abdullahi Abdu IBRAHIM	Department of Computer Enginering, Altınbaş University	_____
Asst.Prof.Dr. Warish PATEL	Department of Computer Enginering, Altınbaş University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Mustafa Khalid ABDULATEEF

Signature

DEDICATION

I dedicate this work to my great Father and my beloved Mother, Also, this thesis is dedicated to my Family members, to all Professors, academic Staff, and lecturers in my great university; Altınbaş University, in addition, I dedicate this work to my beloved Friends, And to Every person who guided and helped me in achieving this thesis, And to all of you, I dedicate this thesis.

PREFACE

I want to thank my advisor Asst. Prof. Dr. Hakan KOYUNCU, please let me express my profound feeling of appreciation and gratefulness to both of you for the information: direction and unrestricted help you have given me. I want you to enjoy all that life has to offer and further achievements and accomplishments throughout your life.

ABSTRACT

CYBER SECURITY AWARENESS POLICY IN SMALL BUSINESS USING MACHINE LEARNING

ALGBURI, Mustafa Khalid Abdulateef

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Hakan KOYUNCU

Date: 04 / 2024

Pages: 77

The process of protecting information by reducing information risk is often referred to as information security or secure information technology. It is an essential component of data risk management. Typically, this entails avoiding or minimising the likelihood of unlawful or incorrect data access, as well as the possibility of misuse, disclosure, interception, deletion, and change or distortion of information. It involves actions intended to lessen the impact of the predicament. As a result, the network is safe from any attacks, internal or external, Because of the application utilizing several machine-learning algorithms. In the context of information security, the goal of this work is to utilise various machine learning techniques to identify cyber risks. This paper first studies the CICIDS2017 dataset related to different types of cyber-attacks. Second, the datasets are ready to feed into machine learning algorithms to build various models and use nominal encoding methods to convert non-numeric features in each dataset to numeric features. Following that, Data sets are analysed using machine-learning techniques such random forests, decision trees, naive Bayes, adaptive boosting, gradient boosting, steep gradient boosting, and multi-level inference. The dataset is used to conduct two experiments aimed at detecting various forms of attacks: 1) Asymmetric categorization to differentiate between benign and malevolent strikes, and 2) Settlement of several categories to identify different types of malicious attacks. These

experiments were carried out to differentiate between distinct Denial of Service (DoS) attacks within each algorithmic dataset. The results of the experiments are as follows: Firstly, each algorithm surpasses its own performance in terms of the detection mechanism's parameters. These outcomes provide evidence that machine learning algorithms are highly effective in recognizing and distinguishing standard attacks from DoS attacks. Secondly, in the second experiment, all algorithms exhibit strong performance in the detection task across all regions. These findings demonstrate that machine-learning algorithms can efficiently detect and distinguish between the two kinds of DoS assaults.

Keywords: Information Security, Machine Leaning, CICIDS2017, Detection Process, Deep Leaning.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

GBDT	:	Gradient Boosted Decision Trees
ML	:	Machine-Learning
SVM	:	Support-Vector-Machine
LR	:	Logistic-Regression
DT	:	Decision-Tree
RF	:	Random-Forest
GB	:	Gradient-Boosting
AdaBoost	:	Adaptive-Boosting

1. INTRODUCTION

1.1 THE BACKGROUND OF THE THESIS

The process of protecting information by reducing its risks is called information security, or more commonly known as information security as shown in Figure 1.1. This is an important part of information risk management. This means preventing or limiting illegal or inappropriate access to the data, as well as misuse, disclosure, interception, deletion, corruption, modification, authentication, recording or neglect of the data. It includes measures designed to reduce the impact of such situations [1].

Figure 1.1: Information Security Term.

In either case, the protected information can be intangible (e.g., paper records) or physical (e.g. electronic data) (e.g. knowledge). The main objective of information security involves the implementation of the CIA triad, a three-part strategy often denoted as safeguarding the equilibrium among maintaining the secrecy of information, ensuring its accuracy, and enabling its accessibility, as seen in Figure 1.2.

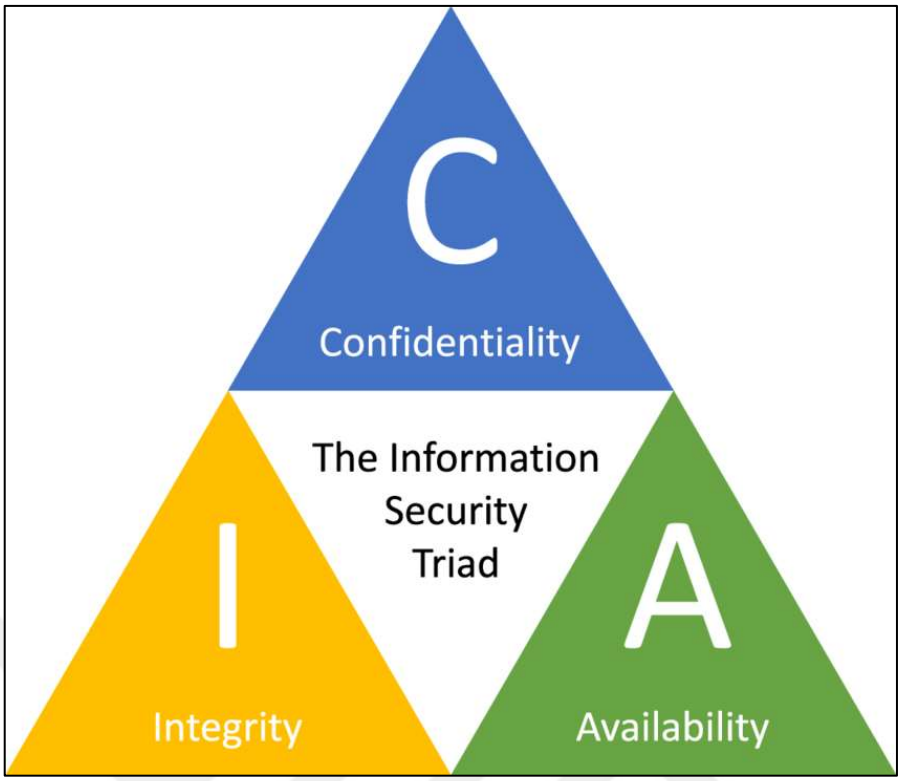

Figure 1.2: The Information Security CIA Triad.

Additionally, it keeps the focus on implementing policies effectively without sacrificing organizational productivity [2]. Information security faces risks from various attacks that aim to gain unauthorized access, modify, delete, corrupt, insert, or disclose information without proper authorization or legal permission. It affects both individuals and organizations. Attacks can be internal, external, targeted, passive, active, clickjacking, brandjacking, employing a botnet, phishing, spamming, outside, and more [3]. Active assaults seek to modify system resources or prevent their use. Active assaults include the production of misleading data or the alteration of data streams. Denial of service (DoS), message tampering, repudiation, replay, and masquerade are examples of such assaults. [4].

The resources of the system are unaffected by passive attacks, which attempt to use or obtain information from it. Passive attacks watch transmissions or maintain track of them. Information being transmitted is something the enemy wants to intercept and capture. Passive assaults can look like the following: Traffic analysis and message content release [5].

anticipates or appraisals Machine learning is the study of "education" algorithms (ML), as shown in Figure 1.3, which improve their performance on a specific set of tasks by using data. Machine learning engines use training data to create models that are then used to make decisions or predictions without the need for careful programming. Machine learning approaches are used when normal algorithm development is difficult or impossible for crucial jobs including computer vision, speech recognition, email filtering, and medicine. [6].

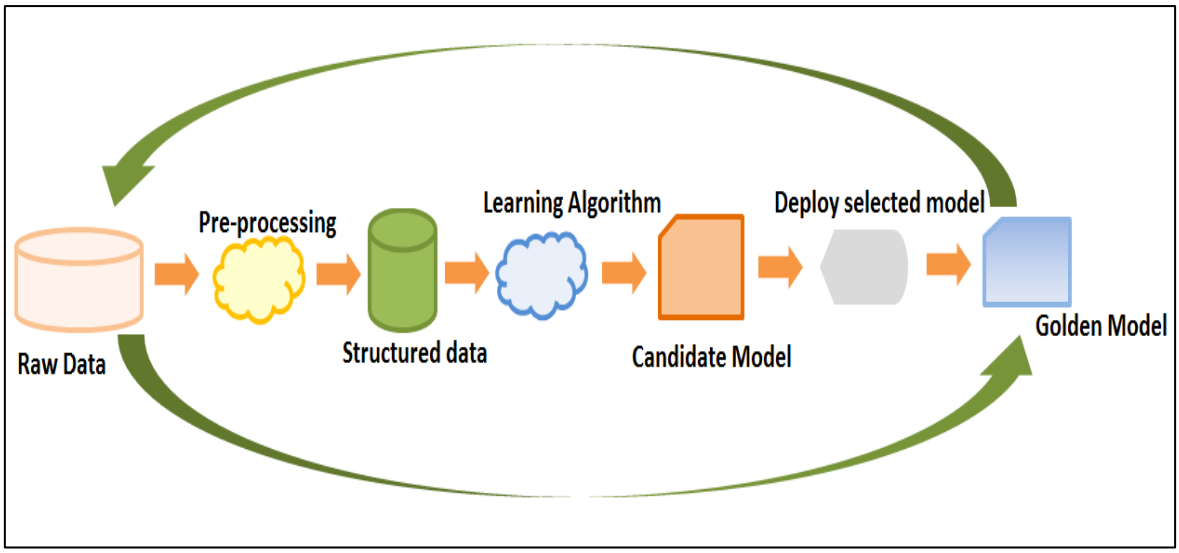

Figure 1.3: The Process of ML.

Artificially learning can be employed in systems of information security for investigation patterns and gain insights From their own. By doing so, these systems can effectively thwart repeated attacks and adjust their strategies according to evolving behaviors. Consequently, this proactive approach empowers information security teams to swiftly prevent threats and respond in real-time to any attacks that may arise [6]. By minimizing the duration dedicated to mundane tasks, machine learning can enable companies to distribute their resources more intelligently. To summarize, the speaker emphasized that machine learning holds the promise of simplifying, streamlining, enhancing effectiveness, and increasing cost-effectiveness in the realm of information security [6], [7]. However, achieving this is contingent on having precise and reliable machine learning data that effectively reflects the oceans. As the saying goes, "Eliminate the noise, eliminate the noise." [6].

Figure 1.4: The Information Security ML.

1.2 PROBLEM STATEMENT

Numerous assailants utilizing a range of strategies might aim at any network communication with the intention of purloining or gaining entry to confidential information. The attacks would primarily involve sizable entities and enterprises engaged in the unlawful procurement and retrieval of data. Any network, whether internal or external, can be the target of these attacks. Understanding the characteristics of these attacks is complex due to the unique traits of each attack, which creates difficulty in telling them apart. Additionally, people encounter notable obstacles in identifying diverse information security breaches Unless supported by a variety of machine learning algorithms produced by computers utilizing a variety of programming languages.

1.3 FINDINGS OF THE RESEARCH

The study's importance arises from its exploration and implementation of several machine - learning approaches to improve the efficacy of identifying distinct information security breaches on any given network.

1.4 PRIMARY GOAL OF THE RESEARCH AND SUBORDINATE AIMS

The major goal of this study is to detect and classify distinct information security assaults that may target both internal and external networks. By utilizing a detection model built on three machine learning (ML) approaches, this will be accomplished. The attacks of interest include Blackhole, Grayhole, TDMA, and Flooding. The proposed detection model not only identifies these attacks but also safeguards the network against such threats. By utilizing ML methods, it is anticipated that the accuracy of detection will surpass that of previous studies. Furthermore, the research encompasses several minor objectives that contribute to the overall completion of the thesis, as detailed below:

a. The goal is to develop a more profound comprehension of the diverse constraints and difficulties encountered by distinct intrusion detection systems. These concerns can impact the usability, efficacy, and efficiency of these systems in different detection assignments.
b. The objective is to investigate how machine learning algorithms could enhance the effectiveness and precision of intrusion detection systems.
c. The goal is to choose a well-known dataset and conduct several experiments to assess the ability of the chosen machine learning algorithms to accurately identify these dangers.
d. Two experiments will be conducted on the dataset mentioned above. Firstly, the samples will be separated into two groups: one for normal data and the other for malicious data. Subsequently, the malicious samples will be further categorized as Flooding, TDMA, Grayhole, or Blackhole.
e. The aim is to employ an advanced machine learning algorithm for the examination of cybersecurity attacks in information security. This analysis will be conducted using a widely recognized network dataset.

1.5 THESIS STATEMENT

"Employing Machine Learning Methods in Information Security: Detecting Network Attacks" is the title of the study. The primary goal of this study was to look at the process of recognizing various cyber-attacks in the field of information security. This goal was met by conducting two distinct tests.

1.6 STUDY QUESTION

The following questions are the focus of this thesis:

a. How can we identify different intrusion threats originating from unfriendly networks?

b. How can we utilize machine learning to discern between legitimate and malicious attacks?

c. How can we differentiate among multiple malicious attacks using machine learning?

d. What characteristics facilitated the identification of the attack?

e. What machine learning algorithms are best suited for effectively detecting attacks?

1.7 RESEARCH ORGANIZATION

The five chapters that make up this thesis address a wide range of subjects and ideas about the role that machine learning plays in enhancing detection performance. The upcoming chapters will delve into these subjects in detail:

a. The material of Chapter One, named "INTRODUCTION," provides a brief summary of the background, the issue statement, the importance of the study, the primary purpose, subsidiary objectives, and the research inquiries.
b. Chapter Two, entitled "Literature Review," showcases several research papers Talk about how machine learning methods can be used to detect network assaults.
c. Chapter Three, titled "Research Methodology," presents an overview of the approach employed in this thesis, covering aspects such as the dataset selection, preprocessing techniques, feature extraction, and a concise explanation of the machine learning algorithms utilized.
d. In Chapter Four, titled "Results and Discussions," the experimental outcomes for each algorithm are elucidated and examined.
e. Chapter Five, labeled as "SUMMARY AND SUGGESTIONS," outlines the primary discoveries of the research and connects them to the numeric outcomes produced by the Python software employed in the investigation. Additionally, this segment provides a range of suggestions and valuable perspectives meant for upcoming undertakings. These recommendations have the objective of aiding researchers and engineers in advancing

the realm of cybersecurity attack detection by applying Machine Learning algorithms within the Information Security domain.

2. LITRETURE REVIEW

This section gives a summary of previous studies related discussing the use of techniques for machine learning in identifying furthermore classifying different forms of network attacks within the realm of information security. The data is drawn from different datasets related to network interruptions, in agreement with Tables 2.1 and 2.2. A few machine learning methods have been actualized utilizing these datasets, driving to the improvement of different models designed To protect networks against threats to their security from inside and exterior.

2.1 UTILIZATION OF MACHINE LEARNING METHODS FOR HANDLING INFORMATION SECURITY DATA

[8] A machine learning model was proposed to recognize organize dangers from suspicious movement. It utilized a Decision Tree with four varieties and was tried on the UNSW-NB15 dataset containing various attack sorts. A real-time dataset, RTNITP18, was too utilized. The model's adequacy was assessed utilizing measurements like memory, exactness, and accuracy. Comes about demonstrated its predominance over other models, with an exactness of 90.74% and solid execution over diverse metrics.

Amaizu et al. [9] utilized a profound learning neural network model (DNN). The NSL-KDD, UNSW-NB15, and CSECIC-IDS2018 datasets were utilized by them. Two preprocessing steps were carried out on the three datasets earlier to bolstering them into the DNN show: For profound neural networks to work at their best, for extricating highlights, it is basic to utilize diminish in dimensionality approaches such as principal component analysis (PCA), they decide the number of layers and their positions. The Deep Neural Network (DNN) input layer is taken after by the dataset being directed to murky 1, the essential covered up layer. After that, there are three more layers, each called Dense2, which have a thick structure. Each session involves passing the output of these Dense2 layers via the activation function of the Rectified Linear Unit (ReLU). This layer is activated using the ReLU activation function, just like the other hidden layers in the model. We include a Dropout layer in each of the three levels to prevent overfitting. Ultimately, the output is supplied to the output layer as the source, which is powered by the final layer's sigmoid function and connected at connection layer 1 (connection layer). Network traffic is categorized by the model as either

1 for attack traffic or 0 for secure traffic. Once the model has been effectively trained and evaluated, the final stage involves examining fresh network traffic to identify any anomalies. For all valid network traffic, the system sets the value to 0, and when an anomaly is discovered, it changes the value to 1. Four assessment metrics—Accuracy, Precision, Recall, and F1 Score—are used to assess the performance of the model. Based on the NSL-KDD, UNSW-NB15, and CSECIC-IDS2018 datasets, the DNN model provides better results than 97.89%, 89.99%, and 76.47%, respectively, in each dataset.

Kasongo et al. [10] developed an intrusion detection system for the purpose of detecting cyber threats by utilizing machine learning. They worked with the UNSW-NB15 dataset containing various cyberattacks and performed three preprocessing steps: error correction, data normalization, and feature selection using XGBoost. The dataset covered nine attack types and conventional attacks with signatures. The study employed multiple algorithms, including SVM, XGBoost, ANN, kNN, LR, and DT. Notably, using XGBoost for feature selection led to a substantial 90.85% accuracy improvement in identifying entities.

group and others. [11] Five machine learning algorithms are applied to detect different cyberattacks within established cybersecurity frameworks (UNSW-NB15 and KDD99). UNSW-NB15 contains various cyberattacks, including terms related to computer security, and KDD99 helps identify normal and abnormal samples. Various methods, such as SVM, ANN, NB, DT, and Unsupervised Learning, are used in data processing. These algorithms' performance is measured using measures like as accuracy, FAR, sensitivity, specificity, FPR, AUC, and correlation coefficient. On both datasets, the Unsupervised Machine Learning (USML) method works admirably, achieving excellent accuracy and correlation coefficients.

Anvil et al. [12] utilized four machine learning techniques to identify deceptive web traffic. using the NSL-KDD cybersecurity dataset, which contains 148,517 samples categorized into normal and abnormal attacks. R2L, DoS, U2R, and Probe assaults are examples of anomalous attacks. For identification in the study (RF), the techniques Gradient Boosting Decision Tree (GBDT), Random Forest, and Support Vector Machines (SVM) were used. Four criteria—These algorithms were evaluated using four metrics: prediction time, training time, specificity, and efficiency. Random Forest (RF) performed the best when compared to the other algorithms, achieving the maximum accuracy of 85.34 percent.

Su et al. [13] Introduced was the BAT model, an advanced deep learning method designed to identify network threat intrusions. Its adequacy was evaluated utilizing the broadly recognized NSL-KDD dataset, commonly utilized for examining cyber-attacks. There are 148,517 tests add up to in this dataset, isolated into subsets for preparing and testing. These subsets cover five attack categories, classified into typical and irregular sorts. The BAT demonstrate illustrated a eminent execution, accomplishing an 84.25% precision in precisely recognizing these intrusion-related attacks.

Xiao et al. [14] A novel autoencoder-driven design comprising five layers has been displayed to improve the discovery of network abnormalities. This approach was defined utilizing the famous NSL KDD dataset, which holds critical ubiquity in the space of cyber danger examination. The dataset includes a add up to of 148,517 occasions, isolated into preparing and testing subsets. Eminently, it envelops five particular categories of assaults, each encourage divided into subcategories such as R2L and DoS attacks. The demonstrate shown a strong capability in distinguishing interruptions, accomplishing an amazing location exactness rate of 90.61%.

Kuisha et al. [15] presented a unused interruption location calculation based on a variation of back vector machines called One-Class SVM (OCSVM). They assessed the algorithm utilizing the NSL-KDD dataset, a well-known cybersecurity dataset with 148,517 tests categorized into ordinary and unusual attacks. The unusual attacks were advance classified into R2L, DoS, U2R, and Test categories, each containing particular attack sorts. The algorithm accomplished a striking 81.29% exactness in identifying network intrusions.

Perian et al. [16] A novel cybersecurity dataset named ALLFLOWMETER HIKARI2021 has been presented for identifying diverse cyber-attacks. Sourced from Zeek, the dataset contains 86 traits and 555,278 occasions categorized into six attack sorts. Typical occurrences are spoken to by Foundation and Kind categories, whereas noxious attacks incorporate XMRIGCC CryptoMiner, Testing, Bruteforce, and Bruteforce-XML. The study employs various machine learning models and achieves a recognition accuracy of around 0.99%, evaluated using accuracy, balanced precision, recall, and F1 score metrics.

Kadala et al. [17] they aimed to enhance device cybersecurity using multiple machine learning techniques. This was accomplished by using three datasets: "ALLFLOWMETER-

HIKARI-2021," "NSL-KDD," and "UNSW_NB15." The algorithms included random forests, decision trees, multilayer perceptron, extreme gradient boosting, K nearest neighbors, gradient boosting, and voting algorithms. The datasets encompassed various attack types, such as Background, Benign, Bruteforce, and more. Performance was measured using metrics like recall, precision, and f1 score. Notably, the XGB algorithm showed superior performance in tests on the UNSW-NB15 dataset.

Pelletier et al. [18] Using a machine learning technique, they made predictions whether a network connection is normal or malicious. Using random forests and artificial neural networks, they worked with the CIC IDS-2017 dataset. The dataset contains 80 features, spanning 15 attack categories and over 15 million instances. Benign instances include DDoS, Port Scanning, Bots, Exfiltration, Brute Force, XX, SQL Injection, and more. The researchers preprocessed the data by removing duplicates, replacing infinity or NAN values with numerical ones, and eliminating instances with missing values. The Boruta software was employed to automate the process of correlation testing, which helped in determining crucial features while eliminating less important ones. Important variables such Nit win bytes sent, Ack flag count, and Fwd packet per sec were taken into account. On the other hand, less significant elements like Bwd psh flag, Bwd urg flag, Fwd average bytes per batch, and others were disregarded. The outcomes demonstrated that Random Forest surpassed alternative techniques, achieving an impressive detection accuracy of 96.24%.

Jabbar, among others. [19] To test and run hybrid machine learning techniques designed to identify botnet assaults, the CICIDS2017 dataset was employed. The dataset contains 80 features with categories "botent" (1,996 examples) and "benign" (191,033 examples). Data cleaning, zero feature removal and network flow feature removal are the three preprocessing methods they adopt. Additionally, they rescale the dataset to a specified range using min-max normalization. The model reduces the data dimension and temporal complexity of the botnet identification process by using filter-related feature estimates and principal components, which are based on two distinct types of various filters on botnet attributes. The detection performance of the dataset was improved by reducing the dataset's 80 features to 9. To create better recognition models, six classifiers—RF, IBK, JRip, Multilayer Perceptron, Naive Bayes, and OneR—were created and assessed during the classifier's training phase. Reliability and efficacy of the suggested framework are confirmed through

the use of commonly used metrics such as F-Measure, accuracy, and recall. When utilized with the CICIDS2017 dataset, the Correlation Attribute Eval (filter) and JRip (classifier) methods can incredibly upgrade the prepare of botnet distinguishing proof. This combo yields way better results than other accessible options.

Giulianto et al. [20] CIDC-2017 stands as a famous data security dataset planned for upgrading the adequacy of interruption location frameworks utilizing machine learning strategies. Its essential objective is to recognize network attacks, especially centering on distributed denial of service (DDoS) and generous assaults. The dataset contains a comprehensive collection of data, counting 80 special traits and hundreds of thousands of occasions. In this dataset, you will discover occasions where the conveyance is imbalanced, comprising both DDoS and benign attacks. To unravel these challenges, to guarantee an rise to number of two names, they utilize a strategy called engineered minority oversampling (Smote). To unravel these troubles, they utilize a method known as manufactured minority oversampling to guarantee an break even with amount of two names (Smote). Four measurements are utilized to evaluate the approach's execution: f1-score, precision, accuracy, review, and recipient working characteristic (ROC). When combined with PCA and Destroyed, the proposed AdaBoost classifier accomplishes a 92 percent AUROC, or region beneath the recipient working characteristic bend, agreeing to the assessment comes about. Together with EFS and Destroyed, the AdaBoost classifier's exactness, review, and F1 score are, in that arrange, 81.83 percent, 100 percent, and 90.01 percent.

Goryunov and others. [21] A capably planned algorithm was created for the reason of distinguishing computer attacks utilizing machine learning strategies. The researchers opted for the notable CICIDS2017 dataset, renowned for its comprehensive public data. This dataset encompasses 80 distinct attributes and labels spanning across 15 categories. To enhance efficiency, the feature count was curtailed to 10, and the number of classes was reduced from 15 to four, specifically focusing on brute force, XSS, and SQL injection attacks. Several preprocessing procedures, including data cleansing, normalization, and the application of techniques like SMOTE for data augmentation, were employed to prepare the dataset for machine-learning algorithms. AdaBoost, k-nearest neighbors, decision trees, random forests, and logistic regression were among the methods used. To improve the model's performance, a nearly optimal set of hyperparameters was constructed in contrast to

the outcomes reported in previous published studies. In real network traffic, the built-in model was tested for attack detection. Because critical elements are reliant on both the network's physical layout and equipment configuration, The model's efficacy was proven only when it was trained on data from a particular network. Acknowledging these limitations, the possibility of using machine learning techniques for identifying computer-related vulnerabilities has been recognized.

Panwar et al [22] Utilizing diverse machine learning methods, a range of network attack detection algorithms were implemented on a shared dataset named CICIDS-2017, which focuses on information security breaches. The dataset contains a diverse range of attack types, including DDoS attacks such as heartbleed/DOS, heartbleed/brute force, heartbleed/web, heartbleed/infiltration, as well as botnet, port scanning, and distributed denial of service.

To enhance the dataset, techniques for data preparation such as data cleaning, normalization, and estimation were employed. To recognize critical highlights from this dataset, a few include determination approaches such as classification subset evaluator and correlation-based highlight choice were used.

The machine learning arms stockpile comprised three algorithms: Naive Bayes, J48, and Decision Tree. Assessment of these algorithms' viability was conducted utilizing four rating scales—accuracy, review, F1 score, and another exactness metric. Surprisingly, the comes about showcased an exactness surpassing 99%, successfully showing whether the joining of mystery had upgraded J48's performance.

Stephen et al. [23] The inquire about centered on analyzing broad organize activity to improve the accuracy and speed of recognizing unordinary activity designs. The conventional strategy for selecting highlights in Intrusion Detection System (IDS) investigate includes collecting and categorizing information. The think about tried the CICIDS-2017 dataset with different classification algorithms and preprocessing procedures. Comes about demonstrated that the precision and handling speed changes are connected to the number of important highlights. The J48 approach took the longest but gotten the most elevated exactitude of 99.87% with 52 highlights, whereas the RF algorithm accomplished 99.86% with 22 features.

Pangsuban et al. [24] Particularly utilizing the CICIDS2017 dataset given by the Canadian Cybersecurity Institute, developing the architecture, and creating a conceptual show to assess dangers in data frameworks. The accentuation was on surveying the dangers related with freely accessible network data. The dataset included more than 15 million tests and 80 characteristics. Concurring to the study's discoveries, the fundamental components of A chance lattice, organize information examination, information mining procedures, and pooled information from the CICIDS2017 dataset will all be utilized. The inquire about utilized information mining calculations and the CICIDS2017 dataset to recognize dangers in real-time inside an information system. It refined the center concepts into four categories: capturing organize information, making a prescient machine learning demonstrate utilizing CICIDS2017, foreseeing arrange interruptions, and evaluating data compromise utilizing a chance framework. The structural plan centers on arrange information collection, prescient hazard examination, and creating chance evaluation reports.

Al-Masry et al [25] created three machine-learning-driven Aids models, utilizing KNN, KNN, and local external factors (LOF) techniques. In the CICIDS2017 dataset, the three strategies were practiced, tried, and assessed. The CIDC 2017 dataset, containing millions of tests and 80 traits related to different attack sorts, was utilized to evaluate how well these three models performed. A few preprocessing assignments were conducted on the dataset, counting disposing of copy highlights and superfluous ones, disposing of copy records, substituting zero-variance highlights with the cruel, and changing over records with limitlessness or NaN values to zeros. In our examination, we compared the three ways, and us demonstrate created favorable comes about. The LOF procedure has a normal precision of 90.5 percent. In differentiate to past ponders, our inquire about found an exceptional normal location rate of 92.74 percent for new, untrained odd data.

Wankhede et al. [26] When applying machine learning and neural network methods, an effective Denial of Service (DoS) attack location strategy exists. The emphasis is mostly on identifying DoS attacks happening at the application level rather than focusing on detecting DoS assaults in the transportation and networking domains. The experiment use the CIC IDS 2017 dataset, which is current in terms of Denial of Service (DoS) assaults. The data set was divided in numerous ways during the experiment, and the finest partition for each method - RF and MLP - was discovered. Across millions of samples, this dataset has 80 characteristics

and 15 attacks. Following that, these strategies were used to identify the sample as DoS assaults or benign cases, therefore isolating a DoS attack from the dataset. Radiofrequency beat MLP in terms of precision, obtaining an accuracy greater than 99 percent across seven studies.

Table 2.1: Datasets from Earlier Security of Data Research.

Ref	Year	Attack	Dataset for information security	Algorithm	Result
[8]	2020	"Analysis Backdoor Dos Generic Normal Exploits Fuzzers Reconnaissance Shellcode Worms"	"UNSW_NB15"	"DT"	"Accuracy = 90.74%"
[9]	2020	"NSL-KDD\sR2L\sDoS\sU2R\sProbe\sUNSW-NB15"	"CSECIC_IDS2018, NSL_KDD and UNSW_NB15"	"DNN"	"The accuracy scores in three separate datasets are 97.89%, 89.99%, and 76.47%,

Table 2.1: Datasets from Earlier Security of Data Research "Table Continued".

[10]	2020	"DoS Reconnaissance Backdoor Fuzzers Analysis Worm Shellcode Generic"	"UNSW_NB15"	"SVM, XGBoost, kNN, LR, ANN and DT."	"XGBoost accuracy = 90.85%"
[11]	2019	"Examination Scouting Denial-of-Service (DoS) Utilize Universal Typical Malicious Software Covert Access Point Program Instructions"	"UNSW-NB15"	"SVM, ANN, NB, DT, and USML"	"USML accuracy = 94.78%."

Table 2.1: Datasets from Earlier Security of Data Research "Table Continued".

[12]	2021	"N2L Dos U2R Probe Normal"	"NSL_KDD"	SVM, GBDT, and RF	RF Accuracy = 85.34%
[13]	2020	"N2L Dos U2R Probe Normal"	"NSL_KDD"	"BAT_model"	"Accuracy = 84.25%"
[14]	2021	"N2L Dos U2R Probe Normal"	"NSL_KDD"	"5-layer auto encoder (AE)_based model"	"Accuracy = 90.61%"
[15]	2021	"N2L Dos U2R Probe Normal"	"NSL_KDD"	"One Class SVM"	"Accuracy = 81.29%"
[16]	2021	"Background Bening Bruteforce _XML Probing XMRIGCC Crypto Miner"	"ALLFLOWMETER_ HIKARI2021"	"KNN, SVM, RF, and MLP"	"Accuracy = 0.99"

Table 2.1: Datasets from Earlier Security of Data Research "Table Continued".

[17]	2022	"ALLFOLLOWMETE R HIKARI2021 Background Benign Bruteforce-XML Probing XMRIGCC Cryptominer NSL_KDD: R2L DoS U2R Probe UNSW_NB15 Analysis Reconnaissance DoS Exploits Fuzzers G Normal Worm Backdoor Shellcode eneric"	"ALLFLOWMETER_ HIKARI 2021, NSL_KDD, and UNSW_NB15"	"RF, DT, MLP, Extreme Gradient Boosting, KNN, GB, and Vote."	"All algorithms gave the same results"

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[18]	2021	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC 2017”	“ANN RF”	“RF accuracy is 96.24%”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[19]	2020	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack: SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“RF, IBK, JRip, Multilayer Perceptron, Naive Bayes, and OneR”	“JRip gave the higher performance”
[20]	2019	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“AdaBoost”	“Accuracy = 81.83%”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[21]	2020	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“k_nearest neighbors, decision tree, random forest, AdaBoost, logistic regression”	“Accuracy of all algorithms are almost same”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[22]	2019	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“Naïve_Bayes J48 DT”	“J48 accuracy over 99%”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[23]	2020	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“RF, Bayes Net , (RT), Naive Bayes, and J48”	“RF and J48 gave the highers results.”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[24]	2020	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“Machine learning”	“ML can detect the network intrusion very well”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[25]	2020	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“KNN improved KNN LOF”	“LOF accuracy = 90.5%”

Table 2.1: Datasets from Earlier Security of Data Research “Table Continued”.

[26]	2018	“Benign DDoS,Web Attack: Infiltration Web Attack Port scan,Bot Brut Force Slow ,xx Web Attack : SQL injection Dos Huclk Dos , Slow HTTPtest Golden eye And Healthbleed”	“CIDC_2017”	“RF MLP”	“RF accuracy is over 99%”

2.3 SUMMARY

In this section, we introduced various categories of machine learning algorithms that have been employed and put into action on multiple network intrusion datasets within the realm of information security. The aim was to construct models that safeguard networks against unauthorized access. The outcomes indicate that these algorithms perform admirably in achieving this objective.

3. METHODOLOGY

3.1 INTRODUCTION

In this part, we introduced the framework's proposed strategy for identifying various network assaults. Any network could be the target of these acts of violence, depending on the guidelines provided regarding: the dataset's description, the pre-processing phase, and the application of feature extraction techniques. methods and building eight machine learning methods. Figure illustrates the suggested approach's flow diagram 3.1.

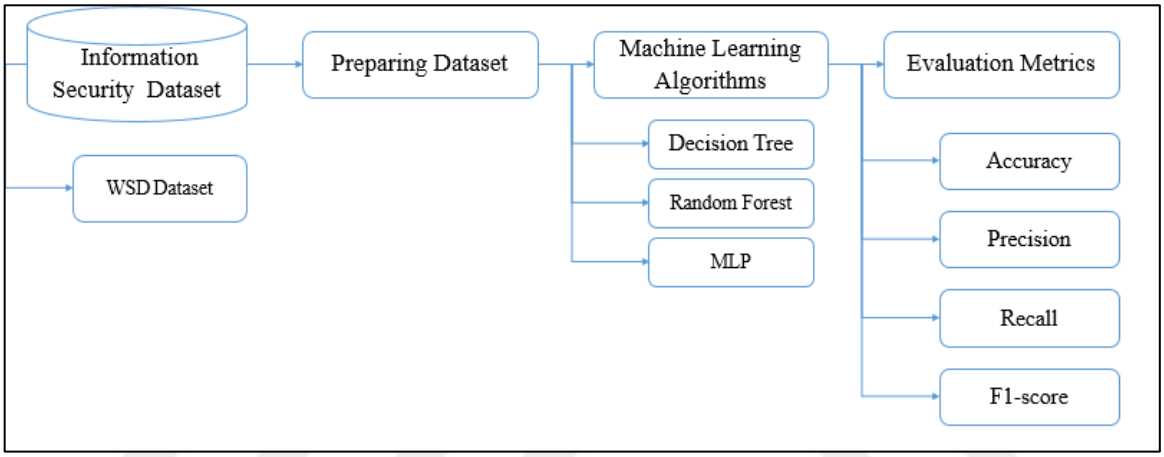

Figure 3.1: Methodology Suggestions Flow-Chart Diagram.

3.2 DATASET EVALUATION

This part of the study presents the Databases are used that was utilized in this thesis to identify different network intrusions in the area of information security. The dataset has a credible origin and contains 23 characteristics and 374,661 occurrences. Each occurrence is labeled with a classification indicating various forms of network attacks, as illustrated below.

a. Node ID: A unique identifier assigned to each sensor node in any round and stage. It consists of three sets of numbers representing the round, stage, and node number (e.g., 001 003 025).
b. Time: The existing moment in the node's simulation.
c. Is CH? A binary indicator with a value of 1 to designate a node as a Cluster Head (CH) or 0 to indicate a regular node.
d. Who CH? The identification number in the present cycle of the Cluster Head.

e. RSSI: In the current cycle, the signal strength indication between the node and its Cluster Head.
f. CH Distance: The distance in space that the node currently has from its Cluster Head in the current round.
g. Maximum distance to Cluster Head (CH): The maximum possible distance between the cluster's nodes and cluster head.
h. Average distance to Cluster Head: The typical distance that separates each cluster head from each node inside the cluster.
i. Current energy: The node's energy level at the current round's start.
j. The amount of energy utilized by the node during the previous cycle.
k. ADV CH send: How many Cluster Head broadcast messages the node sent out.
l. ADV CH receives: How many advertising messages from the Cluster Head the node got.
m. Join REQ send: Count of join request messages that nodes have submitted to the cluster head.
n. Join REQ receive: The quantity of node join request messages that the Cluster Head has received.
o. ADV SCH send: The number of TDMA (Time Division Multiple Access) schedule announcement broadcast messages transmitted by the Cluster Head.
p. ADV SCH reports the quantity of messages sent by the Cluster Head for TDMA scheduling.
q. Rank: This node's place in the TDMA schedule.
r. The quantity of data packets sent from a sensor node to its cluster head.
s. How many packets of data the cluster head transmitted.
t. How many data packets were transferred in total from the Cluster Head to the Base Station (BS).
u. From CH to BS, the distance between the Cluster Head and the Base Station is measured.
v. The cluster from which the data is transferred is represented by the send code.
w. The node's attack type, which can be one of five: blackhole, grayhole, flooding, scheduling, or regular (non-hostile).

Table -3.1 and Figure- 3.2 show that the label for the class encompasses both categories of harmful actions (Grayhole, Blackhole, TDMA, and Flooding) and the Normal category.

Table 3.1: Categories of Attacks in the CICIDS2017 Dataset.

-Attack-	**-Label-**	**-Frequency-**
Normal	Normal	340066
Grayhole	Malicious	14596
Blackhole	Malicious	10049
TDMA	Malicious	6638
Flooding	Malicious	3312

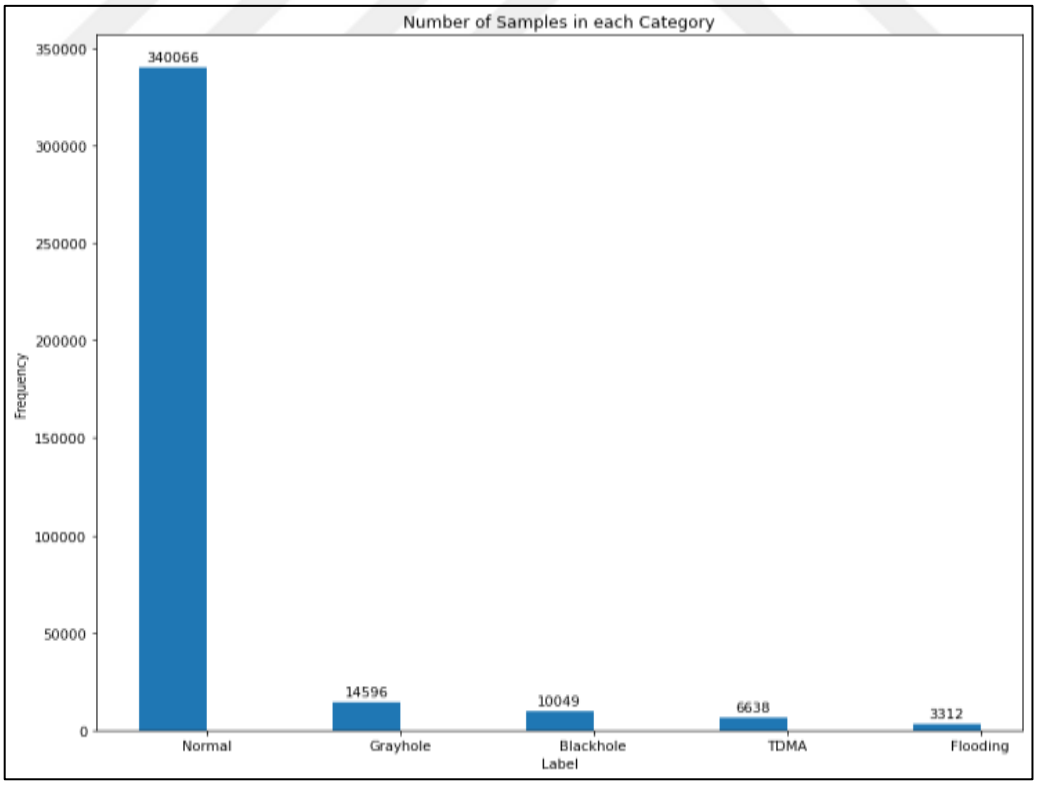

Figure 3.2: Attack Classes in WSN-DS Dataset.

3.3 BUILDING UP A DATASET

To take advantage of several machine -learning algorithms to this dataset, we first employ a commonly known encoding approach to convert the non-numerical information into numerical features [27]. This technique is commonly referred to as "Label Encoder," which facilitates the conversion of non-numerical information into a machine-readable format. It achieves this by appointing a distinct number, starting from 0, to each unique value [27]. All features, except for the Attack type (label), are originally presented as numerical values. Therefore, we employed this method to transform the Attack type into a numerical representation as well.

3.4 MACHINE- LEARNING ALGORITHMS

The data is preprocessed and then submitted in terms of eight Various machine-learning algorithms to discover the previously disclosed cybersecurity vulnerabilities inside each dataset. To demonstrate the outcomes of a 0.1 test size, we employed the hold-out strategy as observed in Figure 3.3. divides the historical data into portions for training as well as testing. Ten percent of the total data is in the test dataset, while the remainder ninety percent is in the training dataset. Using the training data, many machine learning models are built, and the test data is used to evaluate each model's performance.

Machine learning research employs computational techniques to transform empirical data into practical models [28]. This field has emerged from the fusion of traditional statistics and artificial intelligence [29], [30]. Over the past decade, machine learning has gained immense popularity, largely due to the endeavors of major companies like Amazon, Microsoft, Facebook, Google, and so on. Through their commercial activities, these firms have accumulated enormous volumes of data, and this tendency is anticipated to continue [29]. Consequently, there is hope for the reappearance of statistical and computational methods for automatically creating reliable models from this data. Many machine learning algorithms are available as publicly available versions that can be utilized without requiring a significant amount of code thanks to application programming interfaces (APIs) [30] Several popular programming languages include Orange, Python, R, Rapid Miner, and Weka [31]. Utilizing Tableau and Spotfire, two popular visual analytics tools, these algorithms' outputs may be utilized to create realistic pipelines and interactive dashboards [32].

Machine learning or learning programs are computer programs that pick up new skills by watching how they behave. Depending on what kind of learning input or response they can handle, machine learning algorithms are divided into four categories.. Figure 3.3 depicts these groups clearly [33] :

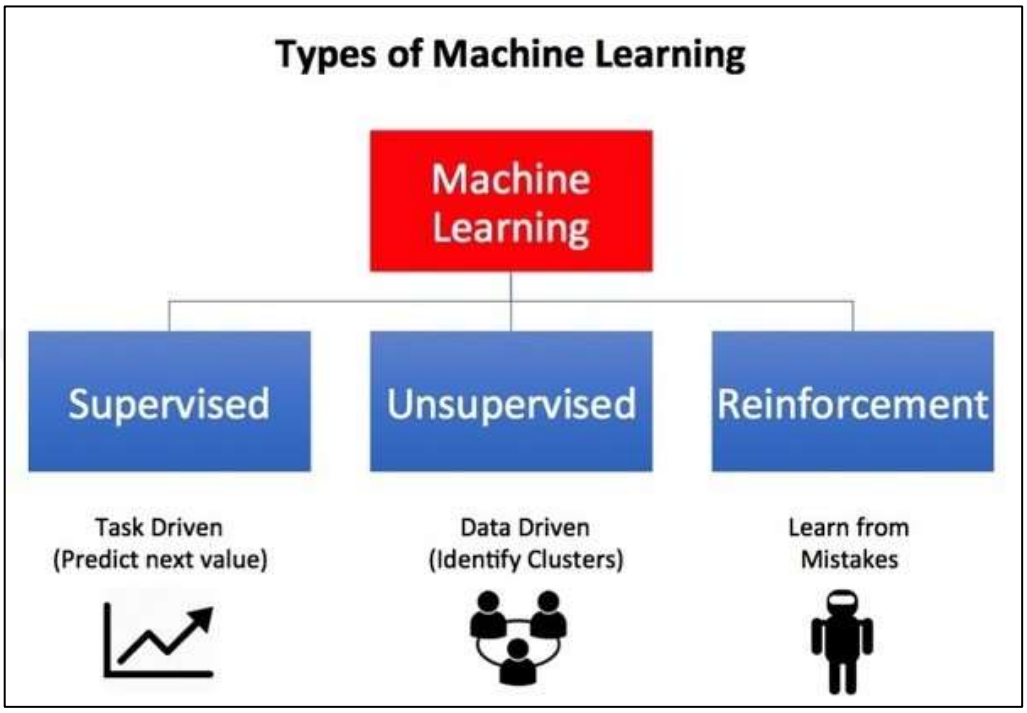

Figure 3.3: Machine Learning Types.

a. Supervised learning is a discipline of artificial intelligence and machine learning identified as "supervised machine learning", as shown in Figure 3.4. [34] is the source. Similarly, it instructs the computer to classify data correctly or to predict events based on a set of labeled data. The model's weights are modified to reflect fresh data that is input into it. [35], which occurs during the validation process. Companies can use supervised learning to find scalable solutions to many real-world problems, such as sending spam to folders other than your email [36]. Supervised learning trains a model using a dataset containing precise input-output pairs, enabling the model to refine its predictions gradually. A loss function is used to determine the model's accuracy, and the training process iteratively reduces errors until they reach an acceptable level. In the domain of data mining applied to supervised learning, challenges are categorized into two main groups: classification problems, where the aim is to assign inputs to predefined

classes, and regression problems, where the goal is to make predictions of numerical values based on input features. [37], [38]:

a. The categorization method is used to separate test data into several groups. The challenge entails finding strategies to recognize or discriminate distinct things within a labeled dataset. In the context of linear classification, several well-known classification methods may be applied. Examples include random forests, decision trees, support vector machines (SVM), and k nearest neighbors. In the context of linear classification, several well-known classification methods may be applied. Some examples are (SVM), (DT), k_NN, and (RF). [37].
b. Regression is a method used in statistics to determine the correlation between independent and dependent factors. This is commonly used to forecast a company's sales and profitability. Regression techniques that are commonly used include logistic, polynomial, and linear regression. [38].

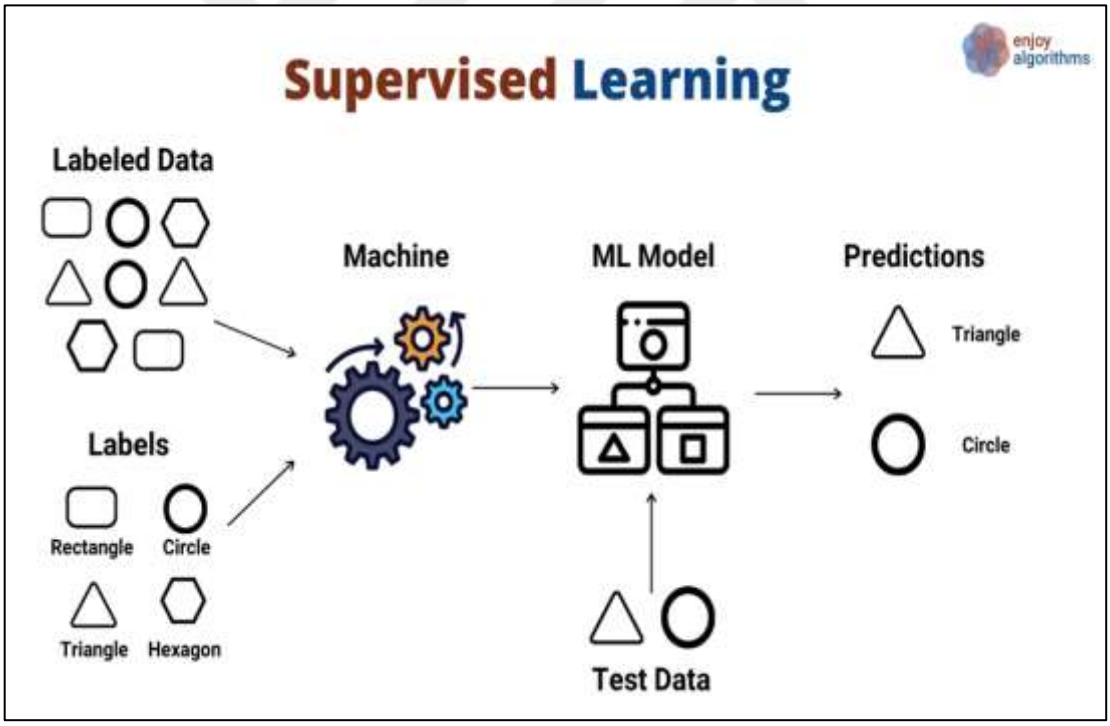

Figure 3.4: Supervised Learning Architecture.

b. Unsupervised Learning: Machine learning algorithms in unsupervised machine learning, also referred to as unsupervised learning analysis and clustering of unlabeled data sets. This is shown in Figure 3.5. [39]. Without human help, these algorithms find

clusters or hidden patterns in data. Research data analysis is an excellent choice for sales tactics, customer segmentation, and pattern recognition due to its ability to find similarities and inconsistencies in data [40]. Three primary problems are addressed using unsupervised learning models: clustering, association, and dimensionality reduction. Below, you will find explanations for each learning method, accompanied by examples of common strategies and algorithms utilized in these approaches. [41], [42].

c. In a clustering data mining approach, unlabeled data is organized based on similarities or dissimilarities. Clustering algorithms are employed to arrange unstructured and unordered data elements into groups, representing patterns or structures within the data. These clustering algorithms take various forms, including exclusive, overlapping, hierarchical, and probabilistic methods. [43].

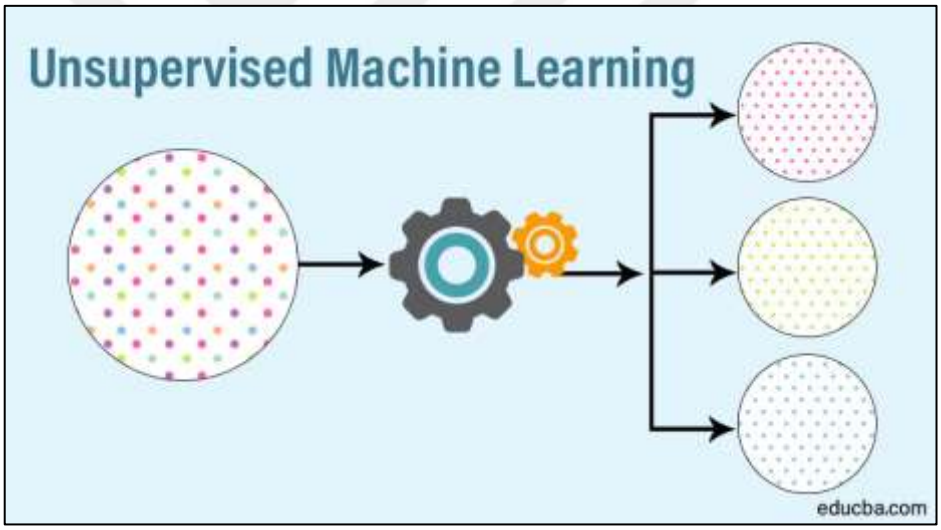

Figure 3.5: Unsupervised Learning Architecture.

d. Reinforcement learning belongs to the realm of artificial-intelligence and involves the process of teaching intelligent agents the most effective ways to act in specific scenarios in order to optimize their overall perception of rewards over time, as shown in Figure 3.6. [44]. Reinforcement learning, like supervised and unsupervised- learning, is one of the fundamental categories of machine-learning. Reinforcement learning, in contrast to supervised learning, does not need the instant rectification of wrongdoings or the presentation of clear input-output instances [45]. Currently, the primary emphasis of existing knowledge centers around attaining equilibrium between exploiting known

information and exploring new possibilities. This particular domain remains partially unexplored. Partially Supervised Reinforcement Learning algorithms effectively merge the strengths of both Supervised and Reinforcement Learning algorithms.

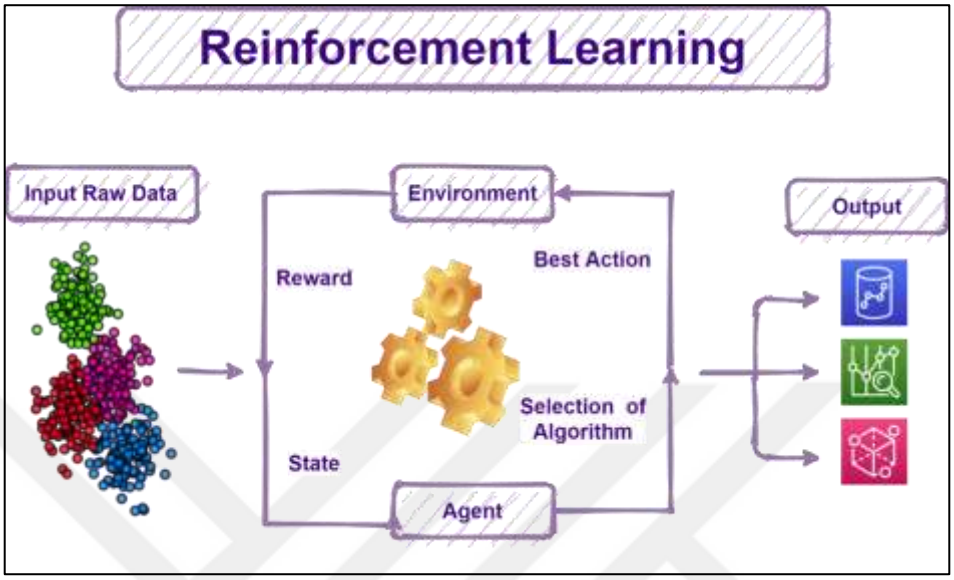

Figure 3.6: Reinforcement Learning Architecture.

e. Semi-supervised Learning is a machine-learning methodology that falls between the supervised and unsupervised learning methodologies [46]. Semi-supervised learning, as illustrated in Figure 3.7, In this process, multiple neural network models and training methods can be integrated [47]. The overall operation of semi-supervised learning is demonstrated by the example below:

a. In place of employing a substantial volume of labeled data for training purposes, this method starts with minimal labeled data and iteratively improves the model until it becomes consistently accurate.
b. These algorithms make use of unlabeled data with fabricated labels, which could introduce errors in the obtained outcomes.
c. Pseudo-labels and the actual labels from the limited labeled data are now linked together.
d. The model is trained again using a mix of pseudo-labeled and original labeled data, resulting in fewer errors and improved model accuracy.

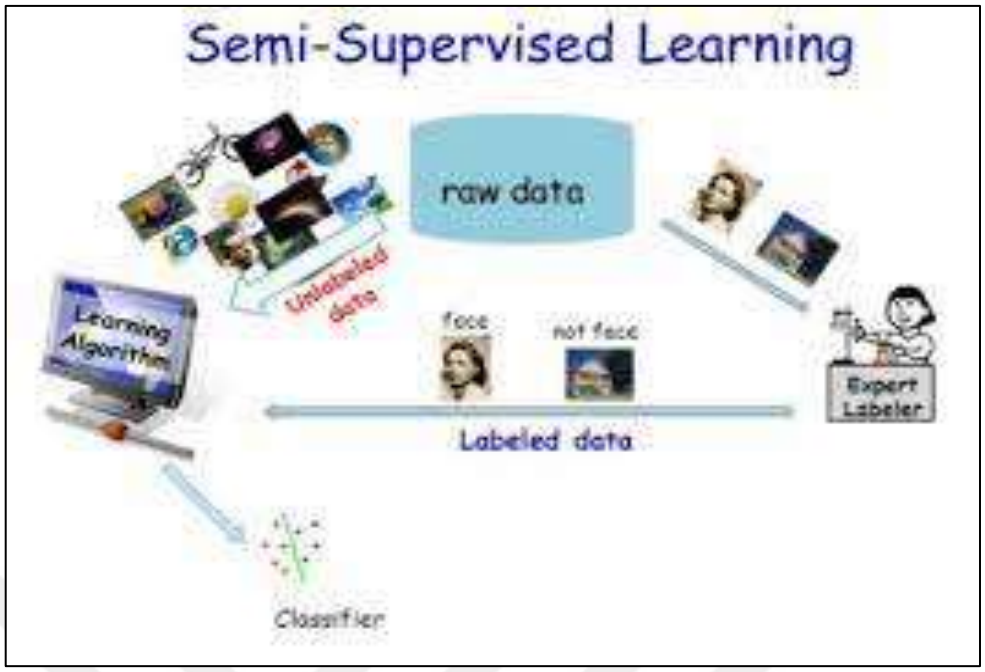

Figure 3.7: Semi-Supervised Learning Architecture.

Due to the availability of labeled data in the dataset, we used broad spectrum of supervised machine-learning algorithms in this thesis. The algorithms employed were Ridge, MP, GB, Adaptive Boosting, NB, eXtreme GB, DT, and RF.

3.4.1 Decision Tree Algorithm (DT)

A DT, a type of SML system, makes decisions in the same way as people do by following a set of rules, as illustrated in Figure 3.8.[48]. DT learning, also known as decision tree induction, is a anticipatory modeling technique used in three areas: data mining, statistics, and machine-learning. [49]. It progresses from making observations about a sample to reaching conclusions (symbolized by the branches) A decision tree is used to make assumptions about a sample's target value, with the leaves standing in for different attack strategies [50].

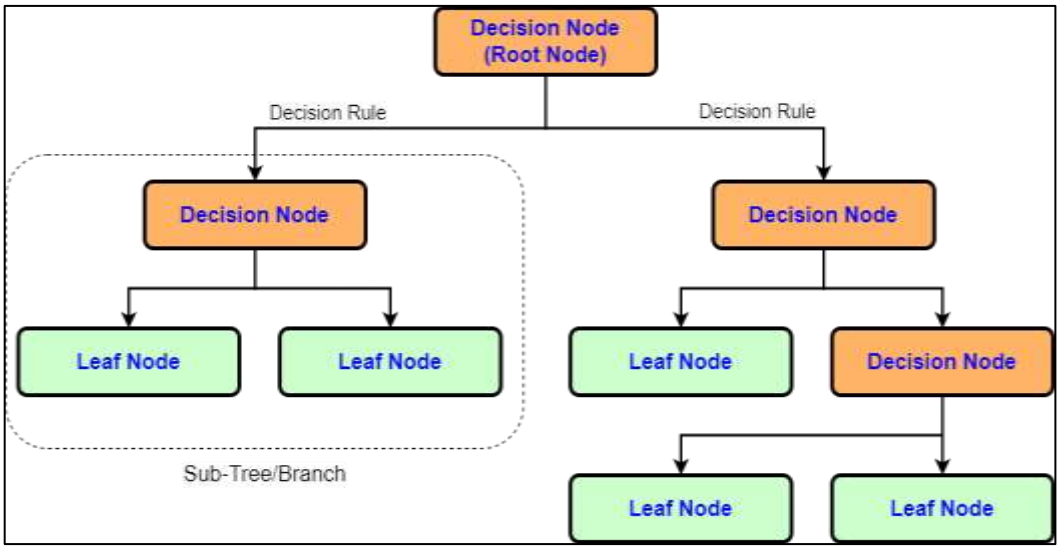

Figure 3.8: DT Architecture.

Models are classification trees that use tree-like structures to cope with discrete target variables. The branches of these trees reflect the dataset attributes used to forecast the class labels, while the leaves represent various sorts of assaults. [51], [52]. When decision trees are used to predict continuous target variables, such as real numbers, they are known as regression trees. [51]. Decision trees have grown in popularity as a machine-learning algorithm due to its simplicity and ease of usage.

There are many advantages of DT:

a. Decision trees need minimal data preprocessing effort compared to alternative algorithms.
b. Normalization and scaling of data are unnecessary for decision tree models.
c. The presence of missing data does not significantly affect the decision tree construction process.
d. Decision tree models are easily understandable and explainable to technical and non-technical audiences.

In addition, the DT has the following disadvantage:

a. Decision trees are sensitive to small data changes, leading to unstable models.
b. Decision tree computations can be more intricate than those in other algorithms.
c. Decision tree training can be time-consuming compared to alternative algorithms.

d. Decision tree training Involves considerable complexity and time overhead.
e. The DT technique is unsuitable for jobs that require regression or prediction of continuous variables.

Since our experiment's label is categorical, we chose the Decision Tree classification type. The Decision Tree was configured with the following settings: criterion=gini_ min_samples_split = 2, and random_state=42.

3.4.2 Random Forest Algorithm (Rf)

The ESL method known as random forests, which is applied to both classification and regression applications, involves training multiple decision trees. The class that the majority of trees collectively choose for categorization issues determines the expected result. In contrast, for regression tasks, the average prediction from each tree is considered [44]. Random decision forests address the tendency of decision trees to overfit their training data. As a consequence, the Random Forest classifier is employed to organize each cybersecurity dataset containing multiple attacks [47], [53].

Figure 3.9 illustrates the four essential stages of the random forest algorithm.

a. Step 1: the Random Forest algorithm initiates by randomly choosing n records from a dataset consisting of a total of k records.
b. Step 2: Following that, separate decision trees are constructed for every one of these subsets randomly chosen for sampling.
c. Step 3: Every individual decision tree generates its distinct output or prediction.
d. Step 4: entails merging the expectations of all decision trees to produce the final result. This is performed by Majority Voting for classification tasks, and Averaging for regression tasks.

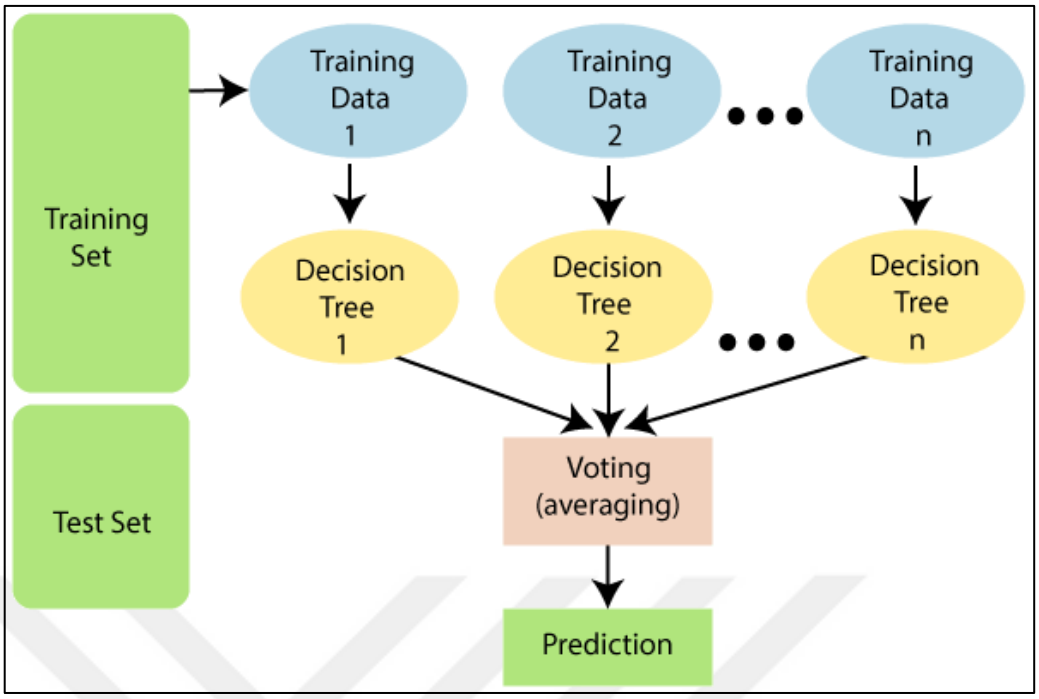

Figure 3.9: RF Architecture.

When employing RF, several crucial elements need to be taken into account.

a. Diversity plays a role in the construction of individual trees as each one is unique, leading to certain characteristics, factors, or features being omitted during the process.
b. Feature space is compressed, making it immune to the dimensionality curse. Each tree only considers a portion of the features.
c. Parallelization: Each tree is constructed separately using a variety of data and properties. This suggests that we can generate random forests by fully utilizing the CPU.
d. In a random forest, a portion equivalent to 30% of the data remains unseen by the decision tree, rendering it unnecessary to split the data into separate training and testing groups.
e. Stability is achieved by making decisions through a majority vote or average.

3.4.3 Gradient Boosting Algorithm

Regression and classification issues may both be solved using the flexible machine learning strategy known as gradient boosting. It combines a number of weak prediction models, such as decision trees, to create a strong ensemble prediction model. [51]. The gradient boosting approach, as shown in Figure 3.10, merges numerous weak learners, represented by decision

trees, into a single robust learner. It is evident that individual decision trees themselves prove to be ineffective learners [52]. Each tree in the series is related to its predecessor and works to correct the flaws of the preceding tree. Training boosting algorithms often need time due to their sequential structure, but the final accuracy is outstanding. In the realm of statistical learning, models that learn slowly tend to outperform those that learn faster [51], [52]. As the model evolves, each new student in the group of less experienced students gets integrated into the residuals of the previous stage. To create a potent learner, the final model incorporates the insights from every phase. One uses a loss function, like mean square error (MSE) for classification or logarithmic loss (log loss) for regression, to detect residuals. It is noteworthy that the model does not alter even after a new tree is included. The newly introduced decision tree aligns with the remainders of the existing model [9], [41], [51].

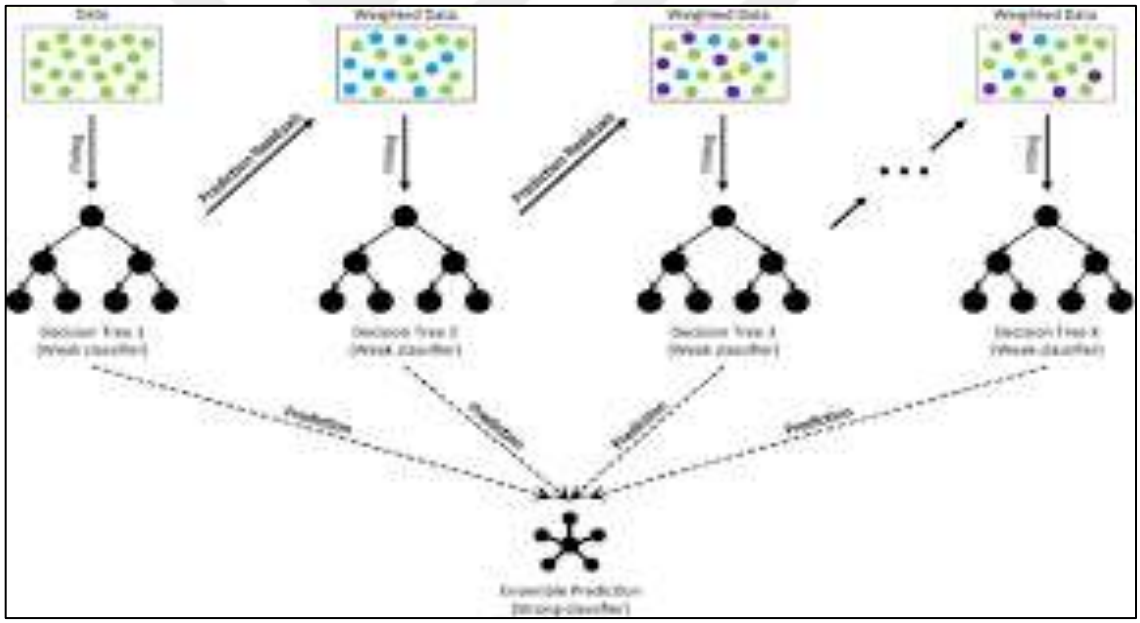

Figure 3.10: GB Architecture.

Advantages of Gradient Boosting are:

a. Frequently offers unrivaled forecasting accuracy.
b. The function fit demonstrates significant versatility, as it can be optimized for different loss functions and provides numerous options for adjusting hyperparameters.

c. No pre-processing of the data is necessary; category and numerical values frequently function well when used directly.
d. focuses on handling missing data; imputation is not required.

Isn't that pretty awesome? Now, let's also explore some drawbacks:

a. The ongoing objective of Gradient Boosting Models is to minimize all errors, but an excessive focus on outliers might result in overfitting.
b. Expensive in terms of computing since it frequently requires a large number of trees (more than 1000), which can be memory and time consuming.
c. Because of the technique's high degree of adaptability, a range of elements interact and have a significant influence on how the approach works (number of iterations, tree depth, regularization parameters, etc.). This adjustment necessitates a thorough grid search.
d. Less interpretative in nature, however this can be changed easily. with a range of tools.

Because the label in our research is categorical, we decided to use Gradient Boosting (GB) with a classification method. The particular configuration of GB we utilized included the following parameters: sub_sample=1.0, learning_rate=0.1, criterion = friedman mse, n_estimators=100, and random_state=42.

3.4.4 Adaptive Boosting (Adaboost) Algorithm

Yoav Freund and Robert Schapire created AdaBoost, a statistical classification meta-algorithm, in 1995. In 2002, they were awarded the Gödel Prize in appreciation of their contributions. By integrating various learning strategies, its effectiveness can be improved. This involves combining the results of different learning algorithms, known as "weak learners," as illustrated in Figure 3.11. The combination process is facilitated through AdaBoost, where weights are assigned to generate an aggregated score, resulting in an enhanced classifier outcome. Despite the fact that AdaBoost may be used for numerous classes or only certain ranges on the real number line, binary classification situations represent its main field of use [45], [54].

AdaBoost exhibits adaptability by rectifying misclassifications made by previous classifiers, thereby enhancing performance for future scenarios In some instances, Compared to other

learning algorithm types, it is less prone to overfitting. Particularly, even if the individual learners beat random guessing just modestly, the whole model converges to a robust learner. AdaBoost is often used to blend weak foundation learners such as decision stumps, but it has also been shown to be useful in combining powerful base learners such as deep decision trees, yielding a substantially more precise model [33], [52].

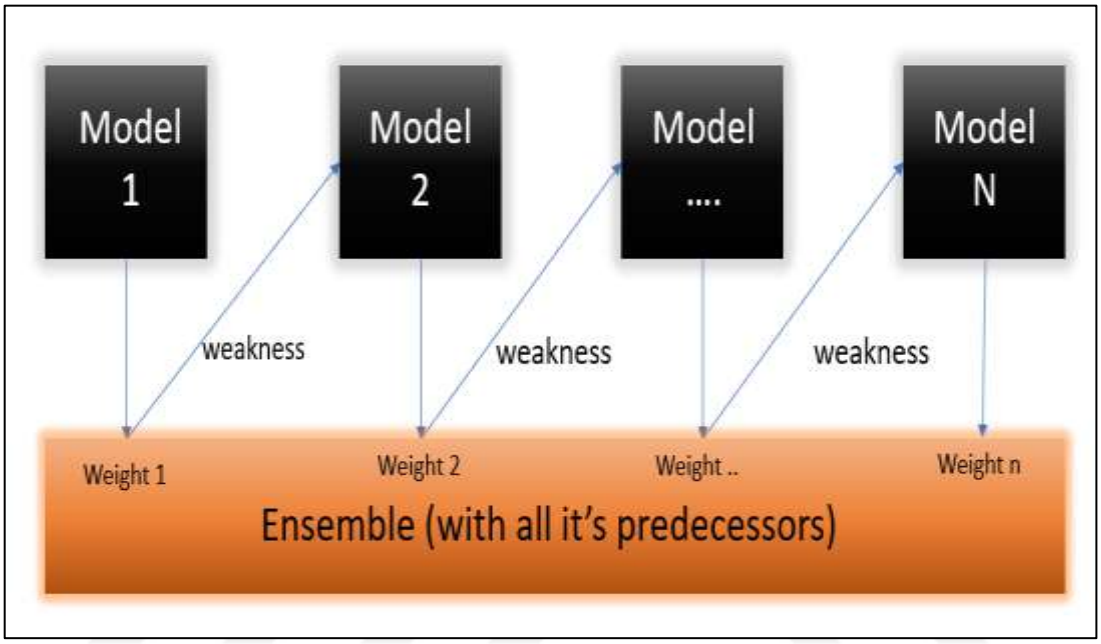

Figure 3.11: AdaBoost Architecture.

AdaBoost possesses numerous benefits, with its user-friendliness and requirement for minimal parameter adjustments standing out in comparison to algorithms such as Encouragement Vector Machine. Furthermore, the compatibility between Support Vector Machine and AdaBoost is noteworthy. AdaBoost is frequently regarded to be resistant to overfitting even in the absence of clear data. The absence of jointly optimized parameters might be attributed to the progressive estimation technique, which has the ability to stymie the learning process. AdaBoost enhances the adaptability and resilience of weak classifiers. Presently, it finds applications not only in binary classification but also in text and image classification.

Because the boosting strategy advances gradually, AdaBoost has a few constraints, one of which is that it requires high-quality data. Because of how sensitive AdaBoost is to both outliers and noisy data, it is strongly encouraged to eliminate them before using it.

We used a classification strategy since We used a discrete label in our experiment. We used particular AdaBoost configurations, such as a 1.0 learning rate.

3.4.5 Naïve Bayes (NB)

Supervised machine learning employs a method called NB to handle classification and regression tasks. The Bayes Theorem is used in this strategy, which holds that each forecast is distinct from the others. The Naive Bayes classifier categorizes data by assuming that the presence of one feature within a class is independent of the presence of other characteristics. Gaussian Because it works with continuous projected labels from the model, Naive Bayes is a beneficial version of Naive Bayes. This type of Naive Bayes is employed when assuming a Gaussian distribution and working with continuous predictor values [33], [34].

The following points show the advantages of the NB:

a. This approach can save plenty of time because it is quick and effective.
b. With naive Bayes, multi-class prediction problems can be resolved.
c. If its premise regarding feature independence is correct, this model has the potential to outperform others while requiring substantially less training data.
d. Categorical input variables are more suitable for Naive Bayes than numerical input variables.

While the following points show the disadvantages of the NB:

a. Naive Bayes makes the mistaken assumption that each predictor or characteristic is independent, which is uncommon but incorrect. This limitation reduces the algorithm's suitability for real-world situations.
b. The "zero-frequency problem" arises when the category of a categorical variable is available in the test dataset but was not encountered during the training phase. Consequently, this results in assigning a probability of zero to that specific category.
c. Employing a smoothing approach to address this matter would be beneficial. Due to occasional inaccuracies in its estimations, it is advisable not to place excessive confidence in its probability outcomes.

Because our experiment's label was distinct, we utilized a classification technique. We used XGBoost with modified parameters, setting alpha to 1, fit prior to True, and random state to 42, to achieve this.

3.4.6 Extreme Gradient Boosting

A method for addressing problems with regression and categorization is called "XGBoost." This approach, based on gradient boosting, has undergone meticulous optimization and parallelization to enhance its performance [35] Parallelizing the entire boosting process results in a notable reduction in training period. Rather than constructing the ideal model from the data, we train many models using various training dataset subsets. [35] Next, select the most effective model, similar to the process used for traditional methods. This is demonstrated in Figure 3.12, where XGBoost consistently outperforms conventional gradient boosting techniques [38]. In the Python implementation, you are granted access to various core parameters that can be modified to enhance precision and accuracy [39].

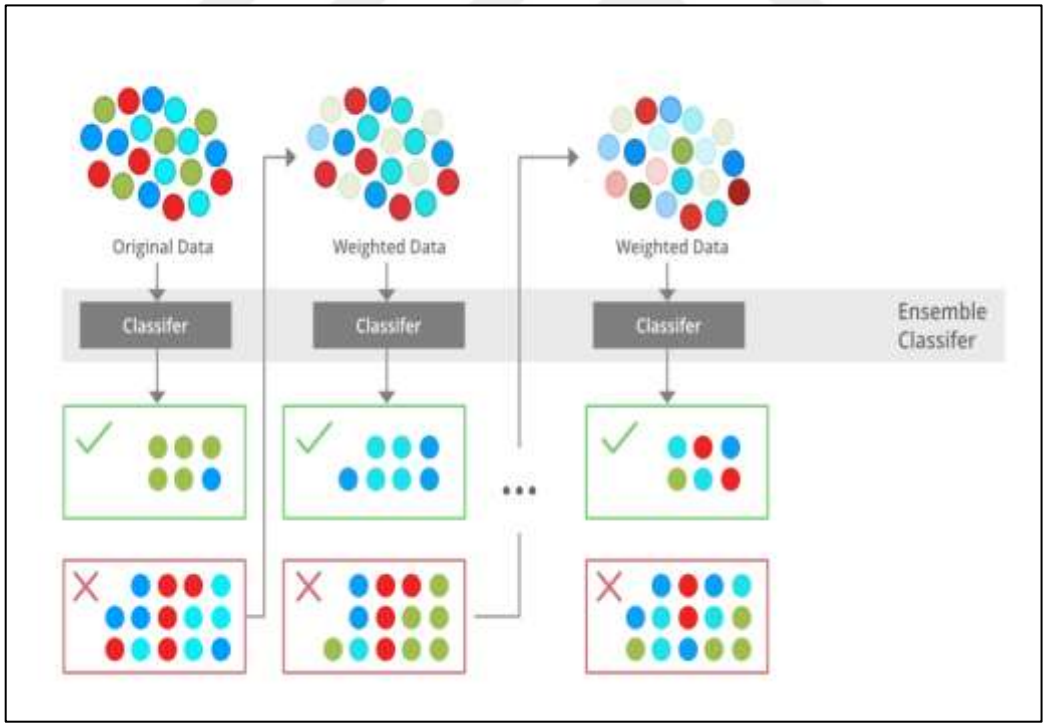

Figure 3.12: XGBoost Architecture.

The principal goal of this algorithm is to enhance the performance of DTs, which are considered weak learners. By converting them into strong learners, the algorithm generates

the final prediction label by averaging the predictions made by each individual classifier [39]. The XGBoost [35], [38], [39] The model possesses several notable attributes, such as: 1) Parallel processing capability: It is created to function concurrently with on multiple CPU cores. 2) arraying options: XGBoost offers a number of arraying penalties to prevent overfitting. These penalties facilitate efficient training, allowing the model to generalize effectively. XGBoost possesses the ability to recognize non-linear patterns in data and effectively learn from them. Additionally, the integration of cross-validation is already provided. Furthermore, XGBoost is scalable, allowing it to handle massive data volumes via distributed servers and clusters such as Hadoop and Spark. Julia, Python, C++, and Java are among the programming languages that may be used.

XGBoost has many advantages such as:

a. Requires fewer capable engineers (no data scaling, no standardization, no better missing value management).
b. Evaluable attribute values (can be used to find the value of each attribute for feature selection).
c. easy to understand.
d. Foreigners are less important.
e. Efficiently manage large databases.
f. quicker time of execution.
g. Full model app (winner of most Kaggle competitions).
h. More convenient, that is.

XGboost has several disadvantages, such as the following:

a. Difficult to show and explain.
b. If the configuration is incorrect, settings may not be required.
c. More difficult setup due to hyperparameters.

We employed a classification technique with the default XGBoost parameters for our investigation since our experiment's label was discrete.

3.4.7 Ridge Algorithm

The Ridge Algorithm, a machine learning regression model, can also serve as a classification method by converting target values to "-1, 1." In this approach, It tackles the issue as a regression task, with the multiclass situation requiring a multi-output regression [40]. The technique of ridge regression, utilized for assessing datasets with multicollinearity, employs L2 regularization. In situations where multicollinearity is a concern, ordinary least squares remains unbiased, yet notable variances in the predicted values compared to the actual values lead to substantial divergences [40], [42].

Ridge has many pros, such as the following points:

a. Serves to avoid excessive fitting of the model.
b. Objective estimators are unnecessary for them.
c. They introduce only a slight bias, allowing the estimates to reasonably approximate true population values.
d. They maintain their efficacy with extensive multivariate datasets, even when the number of predictors surpasses observations (n).
e. In cases of multicollinearity, the ridge estimator excels in improving the least-squares estimation.

Ridge has many cons, such as the following points:

a. The ultimate model includes every predictor.
b. They lack the ability to choose features.
c. They diminish coefficients until they become zero.
d. They substitute bias for variance.
e. Because our experiment involved a categorical label, we employed a classification approach, utilizing Ridge with particular configurations: alpha = 1.0, fit-intercept = True, and normalize = False.

3.4.8 Multilayer Perceptron Algorithm

Multilayer perceptrons (MLPs) are a type of feedforward artificial neural network. The term "MLP" can sometimes apply to any feedforward ANN, and it can also refer to networks composed of several layers of perceptrons that are triggered by threshold. As seen in Figure 3.13, "vanilla" neural networks are simply multilayer perceptrons with only one hidden layer [43].

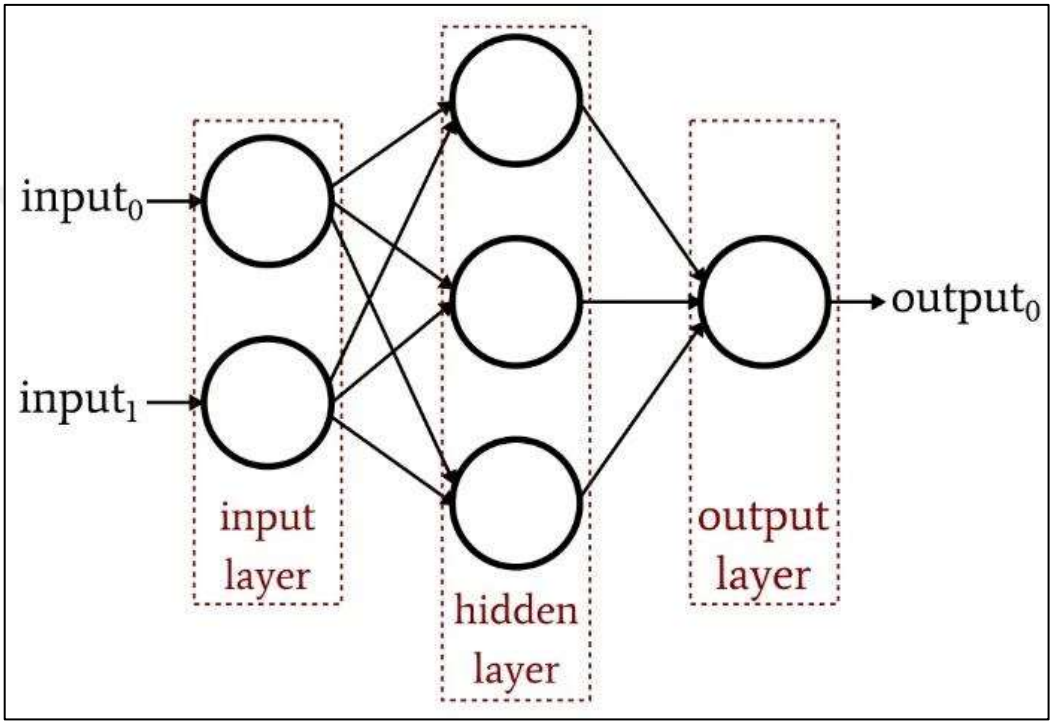

Figure 3.13: Architecture for General MLP.

As an instance of an artificial-neural network operating in a feedforward manner is the multilayer perceptron (MLP). The term "MLP" can encompass various meanings, sometimes denoting any feedforward neural network, and at other times indicating networks comprised of multiple layers of perceptrons that activate based on a threshold. The neural networks shown as standard configurations in Figure 3.13 essentially represent multilayer perceptrons with just a solitary concealed layer, as noted in references [43] and [55]. During its training process, the Multi-Layer Perceptron (MLP) employs the backpropagation technique as a form of supervised learning. Differing from a linear perceptron, MLP distinguishes itself through its incorporation of multiple layers and the utilization of non-linear activation

functions. Consequently, MLP holds the capability to differentiate between data points that cannot be segregated using a straight-line boundary, as clarified by reference [56].

Among the many advantages of the multilayer perceptron are:

a. It has the potential to tackle complex nonlinear problems.
b. It adeptly handles extensive quantities of input information.
c. Following training, it rapidly forecasts results.
d. Comparable levels of accuracy can be attained even when working with smaller sample sizes.

Because our experiment's label is characterized by its distinct nature, we utilized a Multi-Layer Perceptron (MLP) using a classification strategy. The chosen MLP parameters consisted of tanh activation, adam solver, automatic batch_size, and a random state set to 42.

SUMMARY

An overview of the methodology described in this thesis for detecting Denial of Service (DoS) assaults using a well-known dataset is given in this chapter. The methodology includes delineating the dataset, preparing it, and employing seven distinct machine learning algorithms. The subsequent chapter will delve into a comprehensive explanation and analysis of the experimental outcomes associated with each algorithm. These trials include two scenarios: one in which you must identify between a standard attack and a DoS assault, and another in which you must distinguish between two separate DoS attacks. The next chapter will provide detailed insights into these trials and their distinct outcomes.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 INTRODUCTION

Here, we explained the conclusions that were obtained, what the appraisal metric was utilized to determine the success of machine learning codes, and we compared our results with the previous work.

4.2 EVALUATION METRICS

Different evaluation metrics used for checking used ML codes, consisting correctness, sensitive, recollection, and (F1 score) [52]. There are some formulas to these scales are like the followings: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives:

Integrity, or more simply, the ratio of properly forecasted patterns out of all samples, may be thought of as the most basic performance metric. It represents the percentage of correctly predicted pattern samples out of all patterns.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision can be characterised as the ratio of correctly predicted positive samples to all correctly anticipated positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall is the proportion of accurately predicted positive samples to all of the expected positive samples.

$$Recall = \frac{TP}{TP + FP} \tag{4.3}$$

The F1-score represents the balanced mean of Precision and Recall.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.3 SIMULATION CONSEQUENCES

This area exhibits the experimental discoveries of This study was done utilizing different machine learning strategies to segregate between unmistakable sorts of attacks: normal attacks and particular malevolent attacks counting Blackhole, Flooding, Grayhole, and TDMA.

Through this explore, the dataset's tests are dissected utilizing machine learning strategies to decide if they speak to a ordinary event or a particular kind of destructive attack. such as Blackhole, Grayhole, TDMA, or Flooding. The results of this ponder, displayed in Table 4.1 and Figure 4.1, appear how well machine learning calculations perform against the four criteria of accuracy, integrity, f1-score, and recall. The XGB and RF models yielded favorable results, adjusting closely with each other in their discoveries. These comes about clearly suggest that the machine learning strategies in utilize are competent of recognizing between true blue and pernicious assault occurrences.

Table 4.1: Performance-Results of ML Algorithm.

Models	Accuracy	Precision	Recall	F1-score
DT	0.994742	0.961717	0.969694	0.965638
RF	0.996717	0.977838	0.978225	0.977659
GB	0.988443	0.88512	0.786148	0.793517
XGB	0.996717	0.98059	0.976558	0.97824
AdaBoost	0.979956	0.932569	0.879484	0.902877
Ridge	0.954573	0.823603	0.71542	0.753623
NB	0.929004	0.835332	0.569281	0.570592
MLP	0.989724	0.932804	0.937711	0.934495

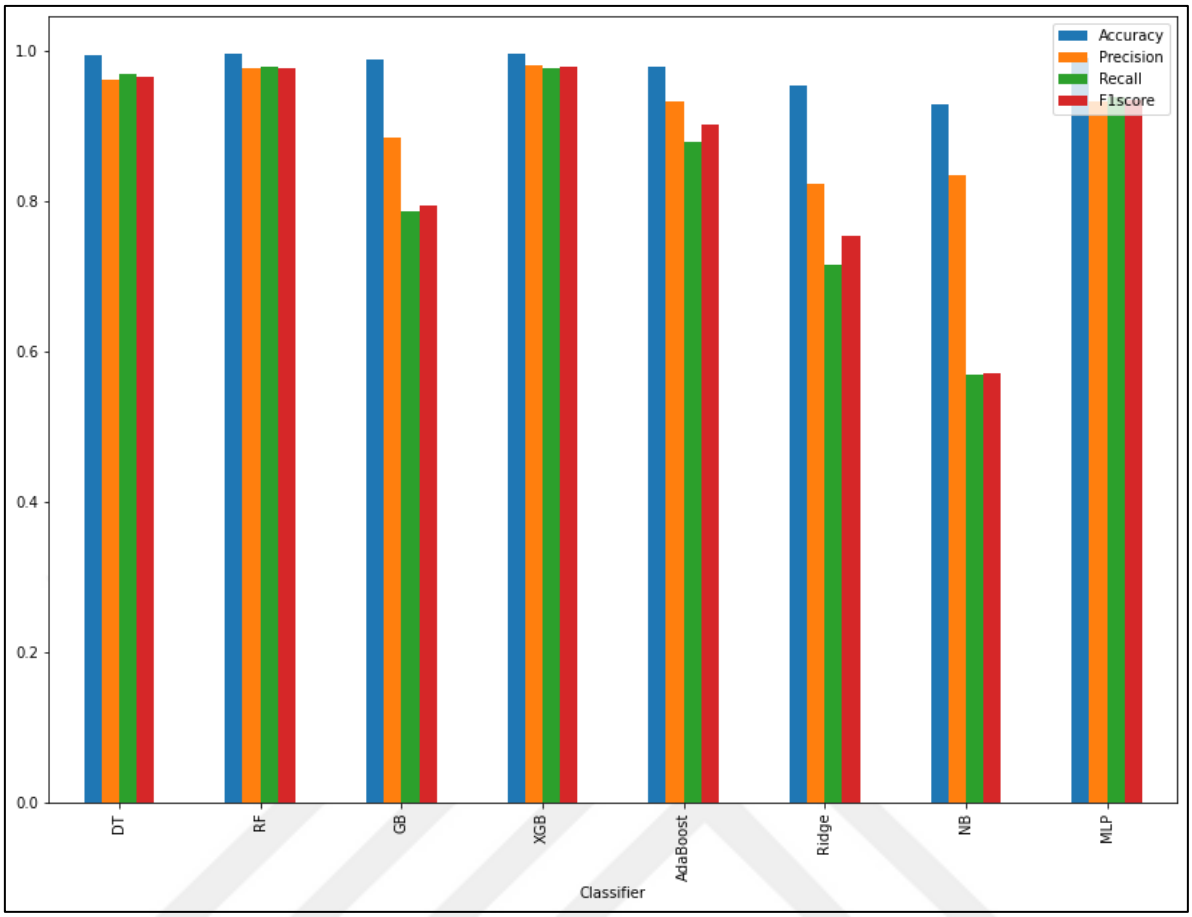

Figure 4.1: Performance Results of ML Algorithm.

In arrange to pick up a superior comprehension of the results and the handle behind deriving accuracy, precision, f1-score, and recall values, it is basic to produce the confusion matrix for each algorithm. Figures 4.2 to 4.19 show the confusion matrices for the unique experiment's machine learning methods. The complexity design is a graphical representation utilized to evaluate the usefulness of a reviewing algorithm. This complexity demonstrate captures and rearranges the execution of a graduation algorithm. This metric is characterized by four terms: "TN" means True Negative, signifying the count of precisely reviewed negative occurrences. Essentially, "TP" stands for True Positive, speaking to the count of precisely classified positive occasions. "FP" compares to False Positive, implying the count of real negative occurrences wrongly labeled as positive. Finally, "FN" shows Wrong Negative, speaking to the count of real positive instances incorrectly classified as negative.

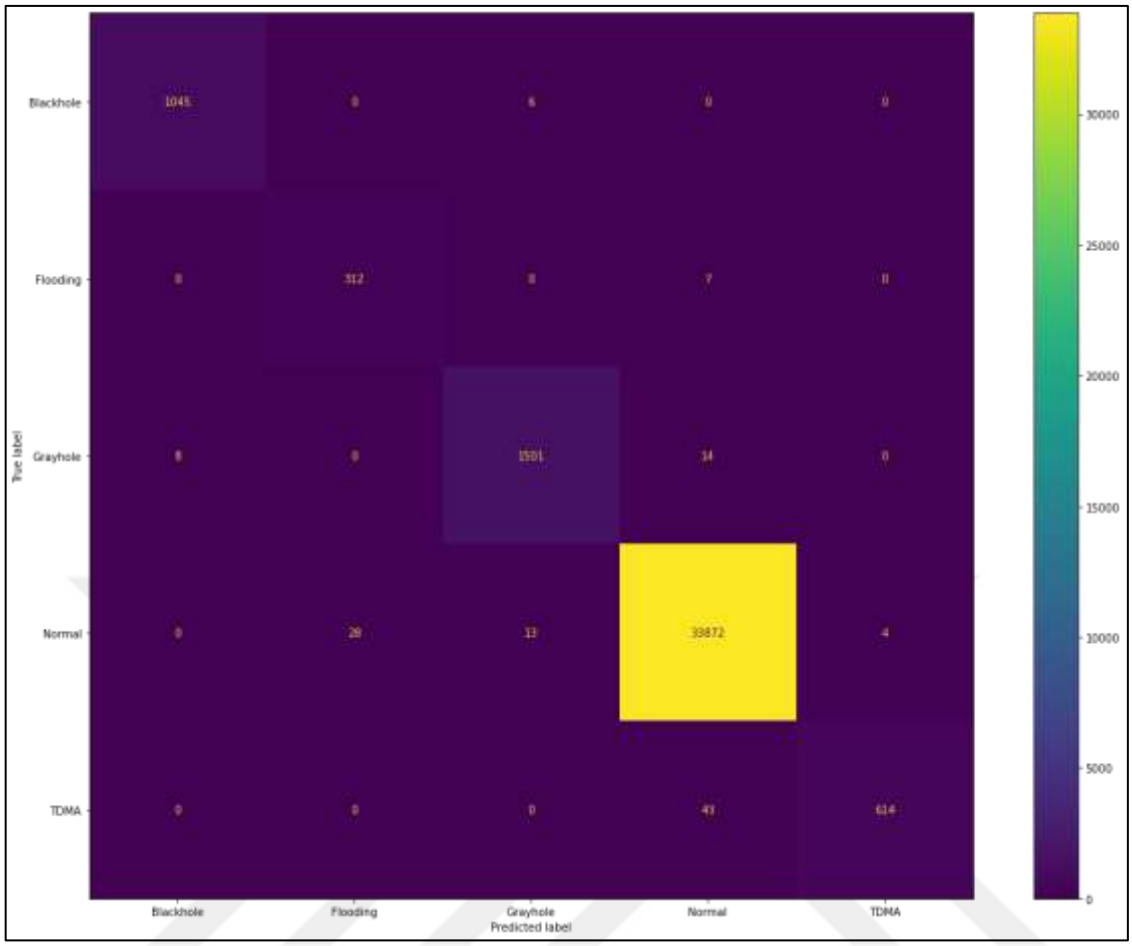

Figure 4.2: RF Confusion Matrix.

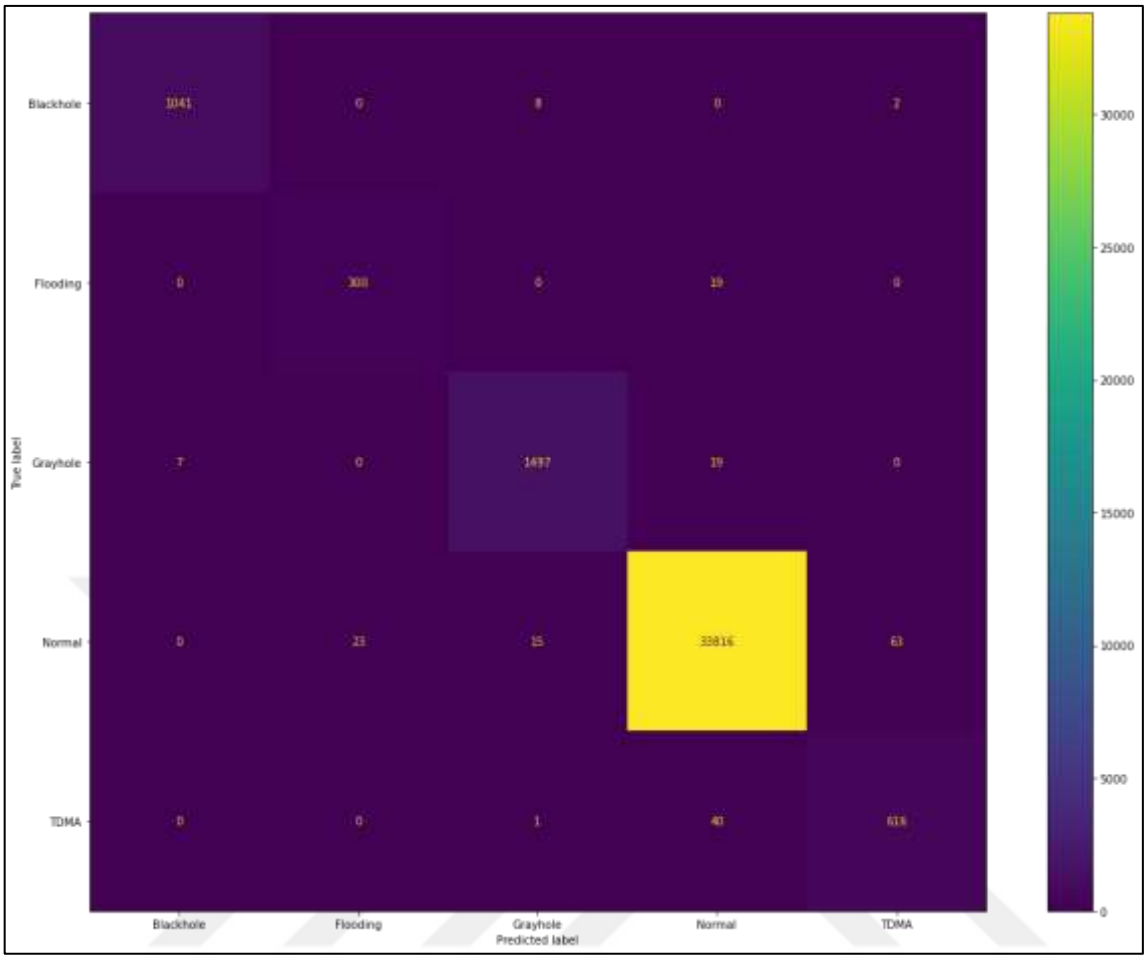

Figure 4.3: DT Confusion Matrix.

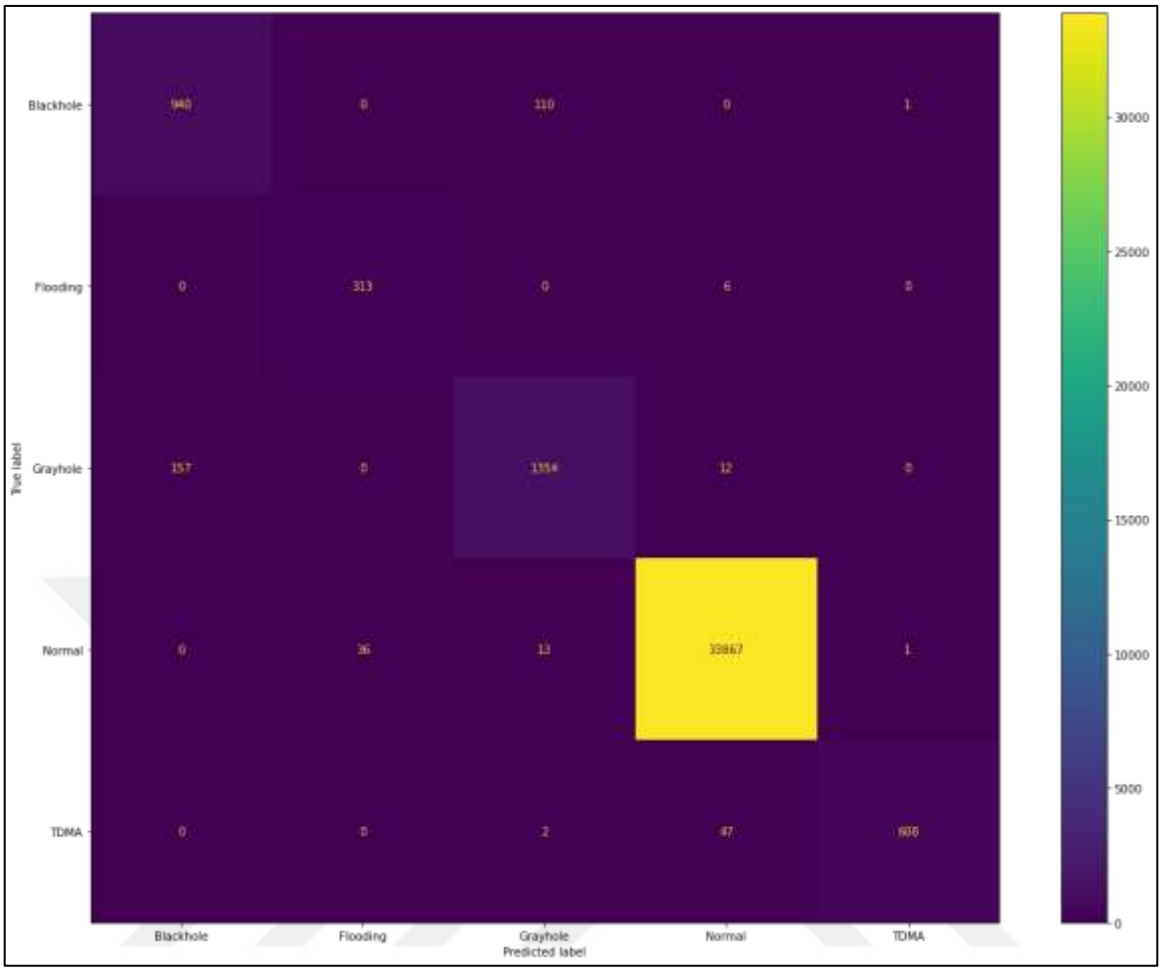

Figure 4.4: MLP Confusion Matrix.

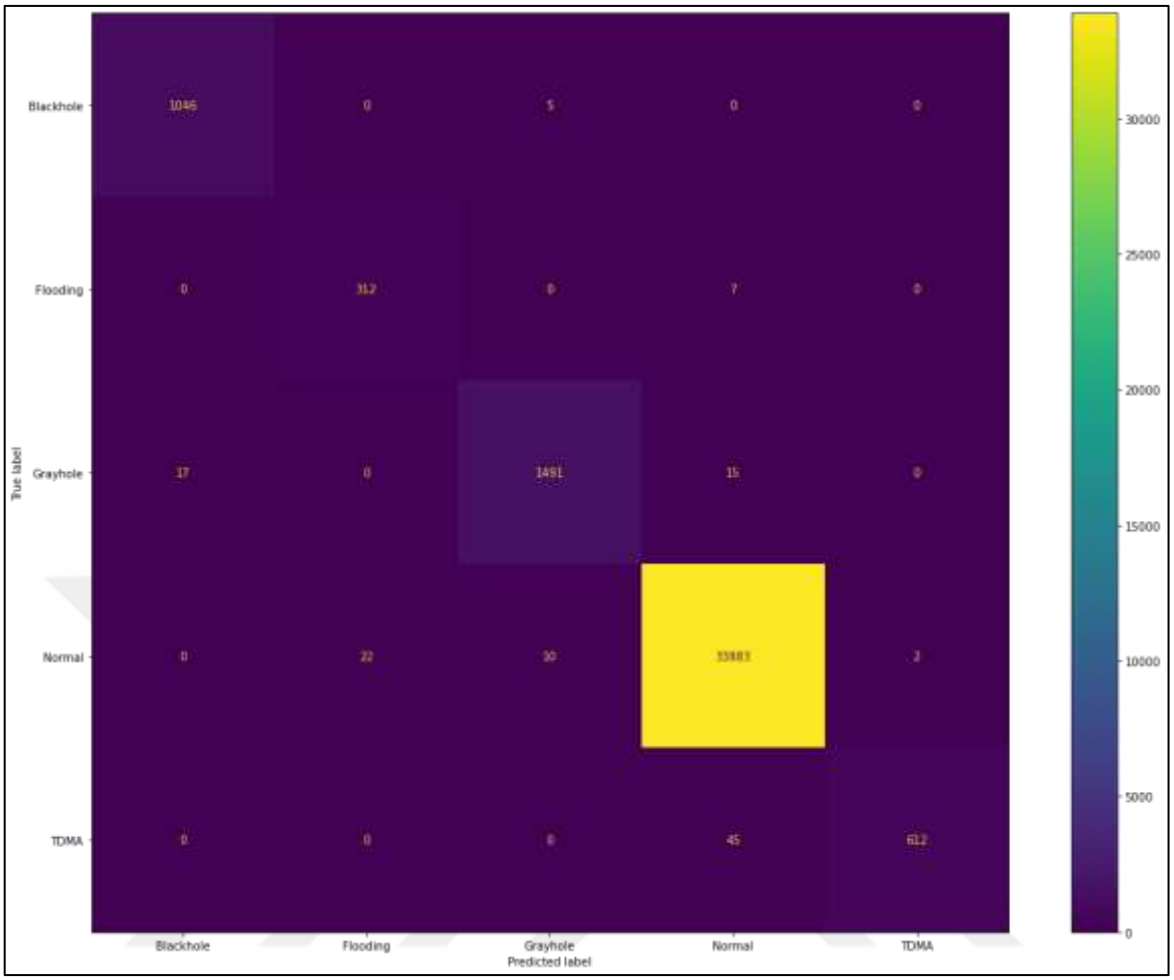

Figure 4.5: XGB Confusion Matrix.

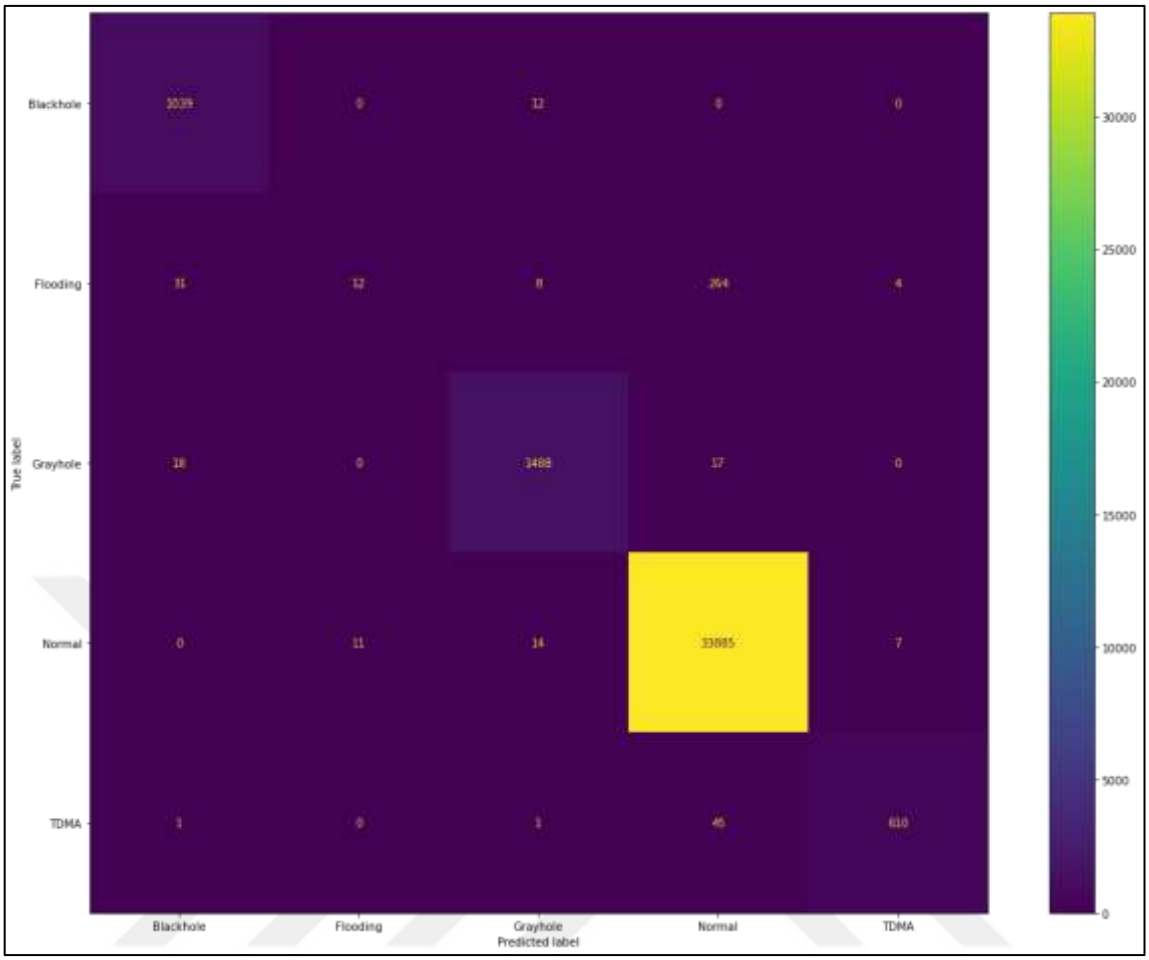

Figure 4.6: GB Confusion Matrix.

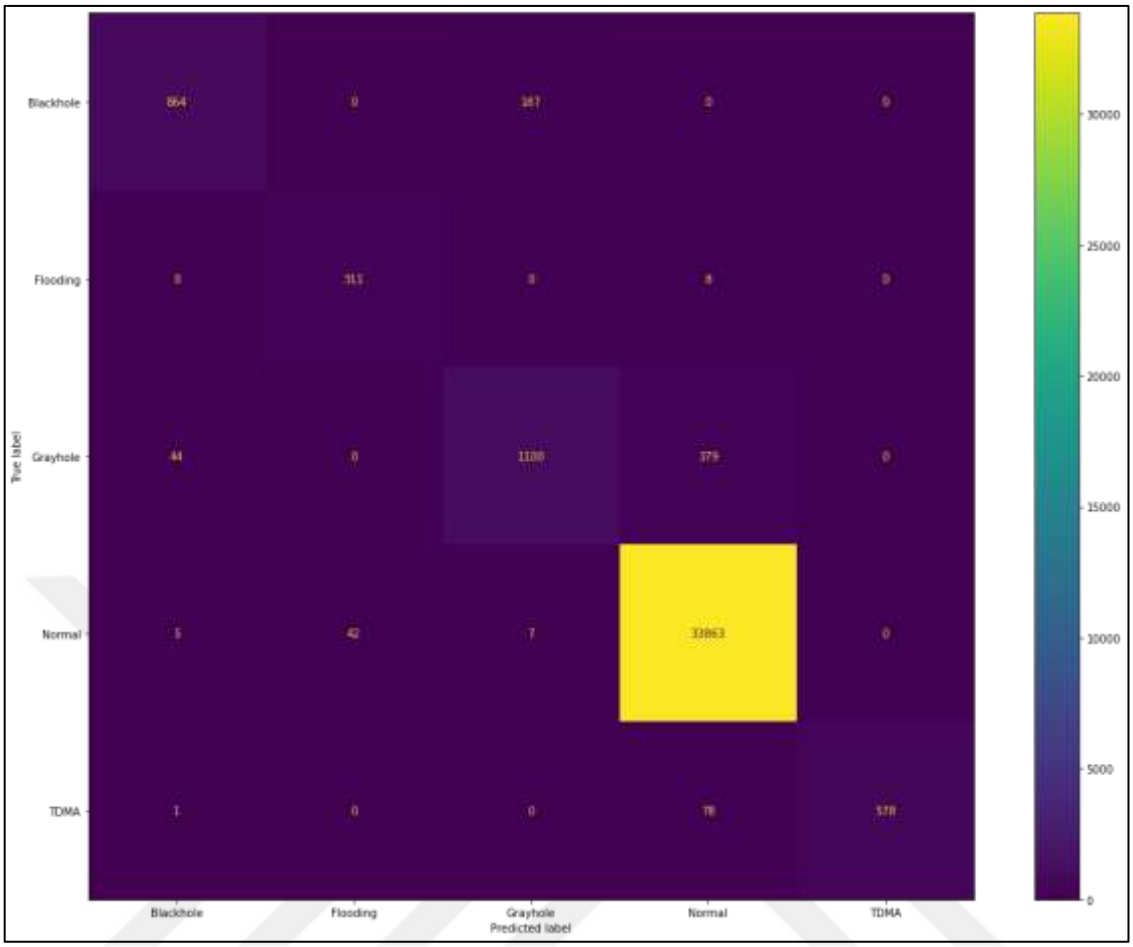

Figure 4.7: AdaBoost Confusion Matrix.

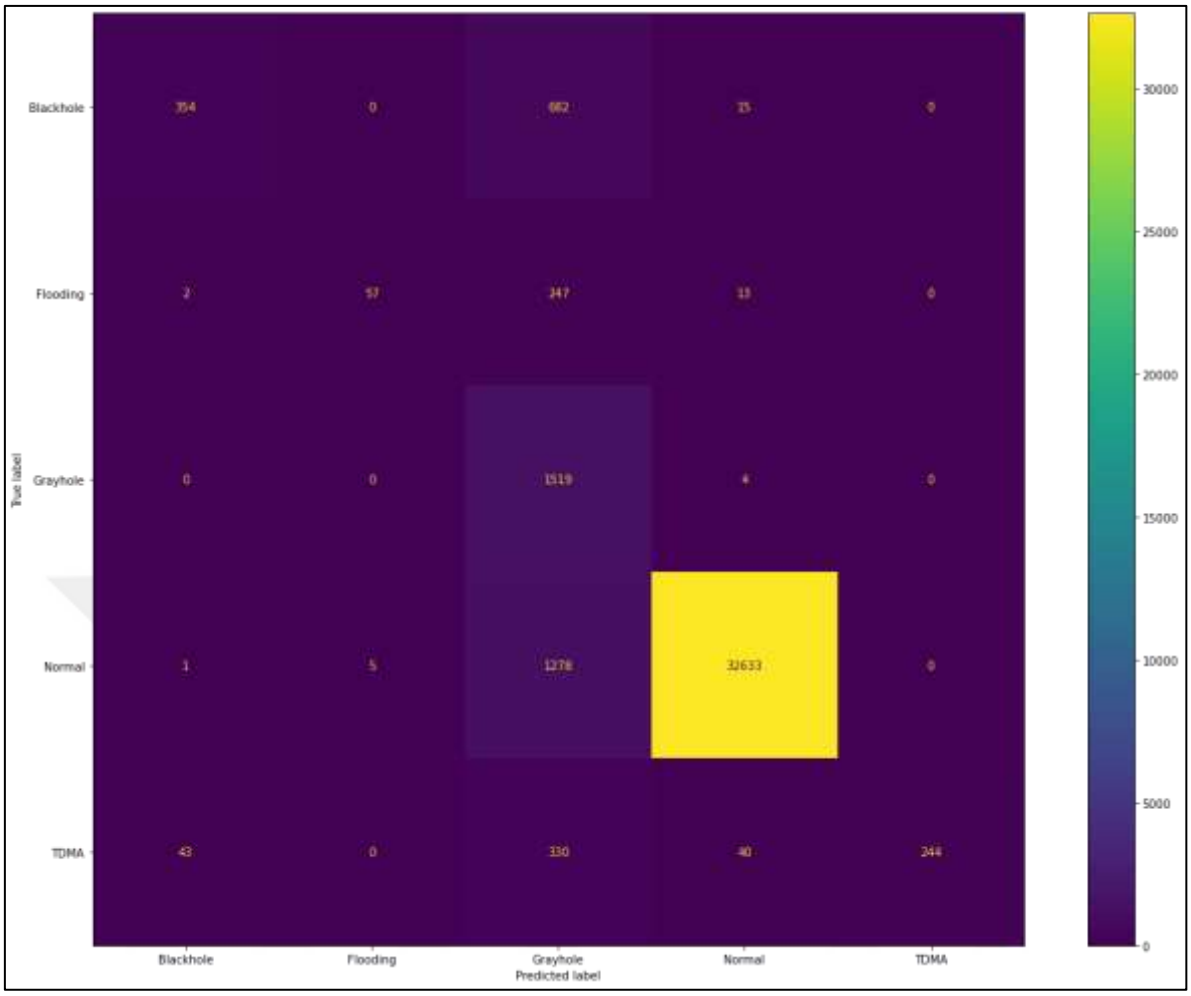

Figure 4.8: NB Confusion Matrix.

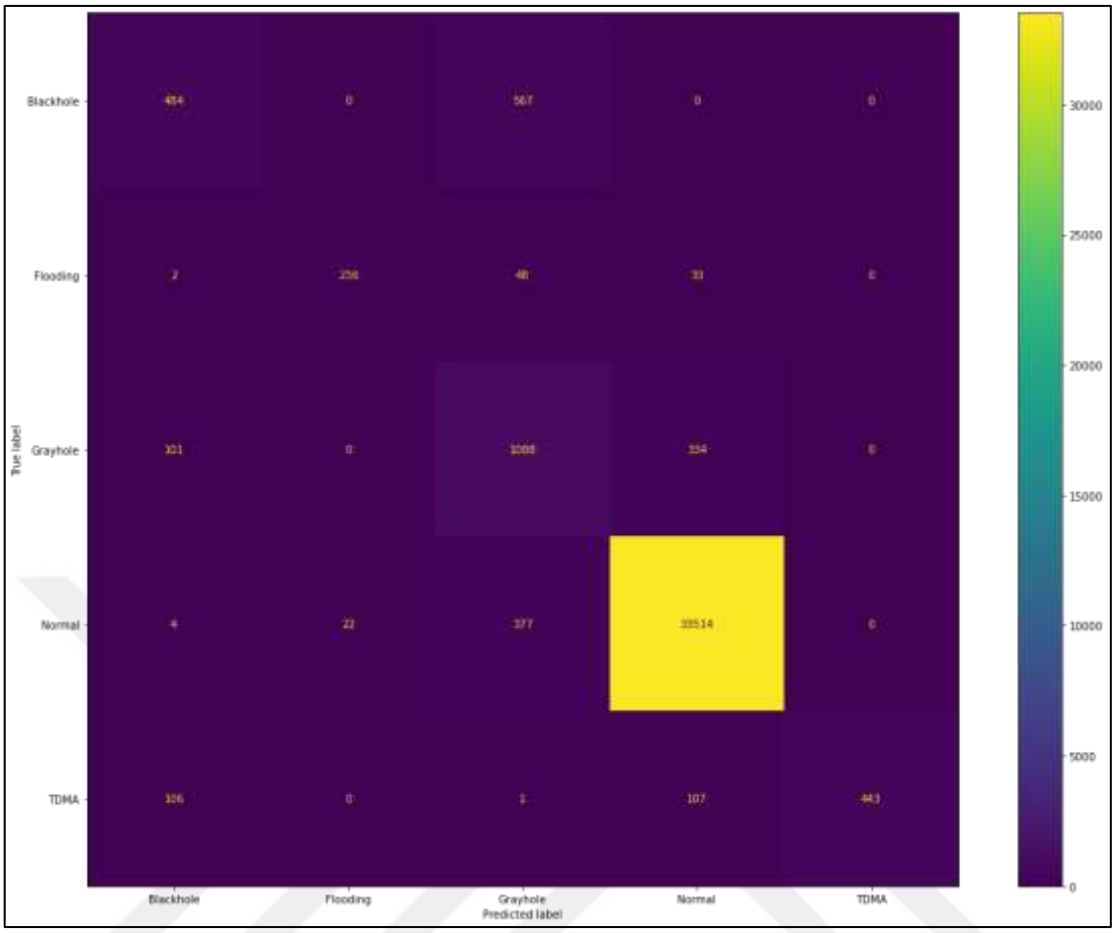

Figure 4.9: Ridge Confusion Matrix.

4.4 SUMMARY

In this part, I am describing the outcomes of our experiments concerning assessment metrics for individual machine learning algorithms. Our experimental findings reveal that all of the machine algorithms yielded the best outcomes using metrics such as precision, accuracy, f1-score, and recall. These outcomes point out that the machine-learning algorithms are adept at accurately recognising and discriminating between legitimate and malicious threats.

The Figure 4.10 compares the performance of various machine learning algorithms applied to different datasets for network intrusion detection systems (NIDS). The datasets evaluated include UNSW_NB15, NSL_KDD, CICIDS2017, and CIDC_2017. The algorithms assessed are Decision Tree (DT), Deep Neural Network (DNN), Random Forest (RF), AdaBoost, XGBoost, Gradient Boosting (GB), Ridge, Naive Bayes (NB), and Multilayer Perceptron (MLP).

The first row indicates that V. Kumar et al. (2020) applied Decision Tree (DT) to the UNSW_NB15 dataset, achieving an accuracy of 90.74%. G. C. Amaizu et al. (2020) tested Deep Neural Networks (DNN) on the NSL_KDD and UNSW_NB15 datasets, with accuracies of 89.99% and 76.47%, respectively. M. Anwero et al. (2021) used Random Forest (RF) on the NSL_KDD dataset, achieving 85.34% accuracy. A. Yulianto et al. (2021) employed AdaBoost on the CIDC_2017 dataset, resulting in an accuracy of 81.83%. S. M. Kasongo et al. (2020) applied XGBoost to the UNSW_NB15 dataset, achieving 90.85% accuracy.

In contrast, the proposed system in 2024 evaluated multiple algorithms on the CICIDS2017 dataset, demonstrating significant improvements in accuracy. Decision Tree (DT) achieved 99.4742%, Random Forest (RF) 99.6717%, Gradient Boosting (GB) 98.8443%, XGBoost 99.6717%, AdaBoost 97.9956%, Ridge 95.4573%, Naive Bayes (NB) 92.9004%, and Multilayer Perceptron (MLP) 98.9724%.

This comparison illustrates the advancements in algorithm performance over time and the effectiveness of different machine learning techniques in detecting network intrusions across various datasets.

Ref	Year	Dataset	Algorithm	Accuracy
V. Kumar et al.	2020	UNSW_NB15	DT	90.74%
G. C. Amaizu et al.	2020	NSL_KDD and UNSW_NB15	DNN	89.99%, and 76.47%
M. Anweret al.	2021	NSL_KDD	RF	85.34%
A. Yuliantoet al.	2021	CIDC_2017	AdaBoost	81.83%
S. M. Kasongo et al.	2020	UNSW_NB15	XGBoost	90.85%
Our proposed system	2024	CICIDS2017	DT	99.4742%
			RF	99.6717%
			GB	98.8443%
			XGB	99.6717%
			Adaboost	97.9956%
			Ridge	95.4573%
			NB	92.9004%
			MLP	98.9724%

Figure 4.10: Our Model Compared to Other Models.

5. CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In the cybersecurity domain, we conducted an experiment using a widely known dataset to identify network attacks. To achieve this, we employed eight different machine learning algorithms. To enable the use of various algorithms, we first converted non-numerical features in the dataset into numerical ones. This transformation was accomplished using the "Label Encoder" technique, It starts at 0 and gives each non-numerical data point a distinct numerical value. As a result, all features were transformed into numerical form, with the exception of the assault type (label), so that the algorithms could use them.

Because the dataset came with labeled information, our emphasis in this research was on utilizing supervised machine learning techniques. The specific algorithms we chose for analysis Random Forest, Decision Tree, Adaptive-Boosting (AdaBoost), Naïve-Bayes, Gradient-Boosting, Extreme-Gradient Boosting, Ridge, and Multilayer Perceptron. Once these algorithms were applied, we assessed their effectiveness using established metrics like precision, accuracy, f1-score, and recall.

Following the assessment, it was found that the XGB and RF algorithms outperformed the others in detecting and distinguishing between normal and malicious attacks. These encouraging findings demonstrate that machine-learning algorithms possess the ability to accurately identifying and distinguishing between different kinds of cyberattacks.

5.2 RECOMMENDATIONS

The proposed offered a set of crucial features and key suggestions that can improve in the proposed output of this work and additional studies, based on the numerical findings and results acquired from the Python code analysis, which include the following recommendations:

a. To use all the algorithms used in this thesis to detect the DoS attack very well in two experiments.
b. To use this dataset in the detection process, we applied Numerous machine-learning algorithms or deep learning to it.

5.2 FUTURE WORK

As part of our ongoing research, we intend to apply these algorithms to various datasets to see if they can reliably identify various assaults across a variety of network datasets. Also, the deep learning algorithms on this dataset and another dataset to know whether deep learning can detect various types of networks or not.

REFERENCES

[1] G. Breda and M. Kiss, “Overview of Information Security Standards in the Field of Special Protected Industry 4.0 Areas & Industrial Security,” *Procedia Manuf*, vol. 46, pp. 580–590, 2020, doi: 10.1016/j.promfg.2020.03.084.

[2] M. Singh, “An Overview of Automotive Vehicles and Information Security,” 2021, pp. 1–13. doi: 10.1007/978-981-16-2217-5_1.

[3] F. Alkhudhayr, S. Alfarraj, B. Aljameeli, and S. Elkhdiri, “Information Security:A Review of Information Security Issues and Techniques,” in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, May 2019, pp. 1–6. doi: 10.1109/CAIS.2019.8769504.

[4] N. Zhang, R. Wu, S. Yuan, C. Yuan, and D. Chen, “RAV: Relay Aided Vectorized Secure Transmission in Physical Layer Security for Internet of Things Under Active Attacks,” *IEEE Internet Things J*, vol. 6, no. 5, pp. 8496–8506, Oct. 2019, doi: 10.1109/JIOT.2019.2919743.

[5] J. Ning, J. Xu, K. Liang, F. Zhang, and E.-C. Chang, “Passive Attacks Against Searchable Encryption,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 789–802, Mar. 2019, doi: 10.1109/TIFS.2018.2866321.

[6] J. H. Lee, J. Shin, and M. J. Realff, “Machine learning: Overview of the recent progresses and implications for the process systems engineering field,” *Comput Chem Eng*, vol. 114, pp. 111–121, Jun. 2018, doi: 10.1016/j.compchemeng.2017.10.008.

[7] F. O. Fedin, O. v. Trubienko, and S. v. Chiskidov, “Machine Learning Model of an Intelligent Decision Support System in the Information Security Sphere,” in *2020*

International Russian Automation Conference (RusAutoCon), Sep. 2020, pp. 215–219. doi: 10.1109/RusAutoCon49822.2020.9208122.

[8] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput*, vol. 23, no. 2, pp. 1397–1418, Jun. 2020, doi: 10.1007/s10586-019-03008-x.

[9] G. C. Amaizu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Investigating Network Intrusion Detection Datasets Using Machine Learning," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2020, pp. 1325–1328. doi: 10.1109/ICTC49870.2020.9289329.

[10] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *J Big Data*, vol. 7, no. 1, p. 105, Dec. 2020, doi: 10.1186/s40537-020-00379-6.

[11] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evol Intell*, vol. 13, no. 2, pp. 283–294, Jun. 2020, doi: 10.1007/s12065-019-00310-w.

[12] M. Anwer, S. M. Khan, M. U. Farooq, and W. Waseemullah, "Attack Detection in IoT using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, Jun. 2021, doi: 10.48084/etasr.4202.

[13] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[14] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, “Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset,” *IEEE Access*, vol. 9, pp. 140136–140146, 2021, doi: 10.1109/ACCESS.2021.3116612.

[15] K. S. Dr. U. M. N. Dr.R.Venkatesh, “Network Anomaly Detection for NSL-KDD Dataset Using Deep Learning,” *INFORMATION TECHNOLOGY IN INDUSTRY*, vol. 9, no. 2, pp. 821–827, Mar. 2021, doi: 10.17762/itii.v9i2.419.

[16] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, “Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic,” *Applied Sciences*, vol. 11, no. 17, p. 7868, Aug. 2021, doi: 10.3390/app11177868.

[17] Iyas Ahmad Abdul Rahman Qaddara, “APPLYING MACHINE LEARNING TECHNIQUES ON CYBER SECURITY DATASETS: DETECTING CYBER ATTACKS,” *Editorial*, vol. 54, no. 7, pp. 95–110, 2022.

[18] Zachariah Pelletier and Munther Abualkibash, “Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R,” *International Research Journal of Advanced Engineering and Science*, vol. 5, no. 2, pp. 187–191, 2020.

[19] A. F. Jabbar and I. J. Mohammed, “Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CICIDS2017 Dataset,” *IOP Conf Ser Mater Sci Eng*, vol. 928, no. 3, p. 032027, Nov. 2020, doi: 10.1088/1757-899X/928/3/032027.

[20] A. Yulianto, P. Sukarno, and N. A. Suwastika, “Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset,” *J Phys Conf Ser*, vol. 1192, p. 012018, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012018.

[21] M. N. Goryunov, A. G. Matskevich, and D. A. Rybolovlev, “Synthesis of a Machine Learning Model for Detecting Computer Attacks Based on the CICIDS2017 Dataset,” *Proceedings of the Institute for System Programming of the RAS*, vol. 32, no. 5, pp. 81–94, 2020, doi: 10.15514/ISPRAS-2020-32(5)-6.

[22] S. S. Panwar*, P. S. Negi, and Y. P. Raiwani*, “Implementation of Machine Learning Algorithms on CICIDS-2017 Dataset for Intrusion Detection using WEKA,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 3, pp. 2195–2207, Sep. 2019, doi: 10.35940/ijrte.C4587.098319.

[23] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. bin Idris, A. M. Bamhdi, and R. Budiarto, “CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection,” *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

[24] P. Pangsuban, P. Nilsook, and P. Wannapiroon, “A Real-time Risk Assessment for Information System with CICIDS2017 Dataset Using Machine Learning,” *Int J Mach Learn Comput*, vol. 10, no. 3, pp. 465–470, May 2020, doi: 10.18178/ijmlc.2020.10.3.958.

[25] T. Elmasri, N. Samir, M. Mashaly, and Y. Atef, “Evaluation of CICIDS2017 with Qualitative Comparison of Machine Learning Algorithm,” in *2020 IEEE Cloud Summit*, Oct. 2020, pp. 46–51. doi: 10.1109/IEEECloudSummit48914.2020.00013.

[26] S. Wankhede and D. Kshirsagar, “DoS Attack Detection Using Machine Learning and Neural Network,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Aug. 2018, pp. 1–5. doi: 10.1109/ICCUBEA.2018.8697702.

[27] J. Guo, A. Nomura, R. Barton, H. Zhang, and S. Matsuoka, “Machine Learning Predictions for Underestimation of Job Runtime on HPC System,” 2018, pp. 179–198. doi: 10.1007/978-3-319-69953-0_11.

[28] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artif Intell Rev*, vol. 39, no. 4, pp. 261–283, Apr. 2013, doi: 10.1007/s10462-011-9272-4.

[29] N. ben Amor, S. Benferhat, and Z. Elouedi, “Naive Bayes vs decision trees in intrusion detection systems,” in *Proceedings of the 2004 ACM symposium on Applied computing - SAC ’04*, 2004, p. 420. doi: 10.1145/967900.967989.

[30] M. Pal, “Random forest classifier for remote sensing classification,” *Int J Remote Sens*, vol. 26, no. 1, pp. 217–222, Jan. 2005, doi: 10.1080/01431160412331269698.

[31] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” *Procedia Comput Sci*, vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

[32] M. Idhammad, K. Afdel, and M. Belouch, “Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest,” *Security and Communication Networks*, vol. 2018, pp. 1–13, Jun. 2018, doi: 10.1155/2018/1263123.

[33] J. Son, I. Jung, K. Park, and B. Han, “Tracking-by-Segmentation with Online Gradient Boosting Decision Tree,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 3056–3064. doi: 10.1109/ICCV.2015.350.

[34] Sven Peter, Ferran Diego, Fred A. Hamprecht, and Boaz Nadler, “Cost efficient gradient boosting,” *31st Conference on Neural Information Processing Systems*, 2017.

[35] R. Blagus and L. Lusa, “Gradient boosting for high-dimensional prediction of rare events,” *Comput Stat Data Anal*, vol. 113, pp. 19–37, Sep. 2017, doi: 10.1016/j.csda.2016.07.016.

[36] R. E. Schapire, “Explaining AdaBoost,” in *Empirical Inference*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 37–52. doi: 10.1007/978-3-642-41136-6_5.

[37] Peter L. Bartlett and Mikhail Traskin, “AdaBoost is Consistent,” *Journal of Machine Learning Research 8*, 2007.

[38] D. P. Solomatine and D. L. Shrestha, “AdaBoost.RT: a boosting algorithm for regression problems,” in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, pp. 1163–1168. doi: 10.1109/IJCNN.2004.1380102.

[39] A. Vezhnevets and V. Vezhnevets, “Modest AdaBoost-teaching AdaBoost to generalize better,” *Graphicon* , vol. 12, no. 5, pp. 987–997, 2005.

[40] G. I. Webb, E. Keogh, R. Miikkulainen, R. Miikkulainen, and M. Sebag, “Naïve Bayes,” in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 713–714. doi: 10.1007/978-0-387-30164-8_576.

[41] I. Rish, “An empirical study of the naive Bayes classifier,” *IJCAI* , vol. 3, no. 22, pp. 41–46, 2001.

[42] R. Mitchell and E. Frank, “Accelerating the XGBoost algorithm using GPU computing,” *PeerJ Comput Sci*, vol. 3, p. e127, Jul. 2017, doi: 10.7717/peerj-cs.127.

[43] B. Pan, “Application of XGBoost algorithm in hourly PM2.5 concentration prediction,” *IOP Conf Ser Earth Environ Sci*, vol. 113, p. 012127, Feb. 2018, doi: 10.1088/1755-1315/113/1/012127.

[44] W. Dong, Y. Huang, B. Lehane, and G. Ma, “XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring,” *Autom Constr*, vol. 114, p. 103155, Jun. 2020, doi: 10.1016/j.autcon.2020.103155.

[45] G. C. McDonald, “Ridge regression,” *Wiley Interdiscip Rev Comput Stat*, vol. 1, no. 1, pp. 93–100, Jul. 2009, doi: 10.1002/wics.14.

[46] L. Noriega, “Multilayer Perceptron Tutorial,” 2005.

[47] J. Tang, C. Deng, and G.-B. Huang, “Extreme Learning Machine for Multilayer Perceptron,” *IEEE Trans Neural Netw Learn Syst*, vol. 27, no. 4, pp. 809–821, Apr. 2016, doi: 10.1109/TNNLS.2015.2424995.

[48] C. Kingsford and S. L. Salzberg, “What are decision trees?,” *Nat Biotechnol*, vol. 26, no. 9, pp. 1011–1013, Sep. 2008, doi: 10.1038/nbt0908-1011.

[49] J. R. Quinlan, “Induction of decision trees,” *Mach Learn*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[50] B. de Ville, “Decision trees,” *Wiley Interdiscip Rev Comput Stat*, vol. 5, no. 6, pp. 448–455, Nov. 2013, doi: 10.1002/wics.1278.

[51] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, “Multilayer Perceptron: Architecture Optimization and Training,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 1, p. 26, 2016, doi: 10.9781/ijimai.2016.415.

[52] H. Dalianis, “Evaluation Metrics and Evaluation,” in *Clinical Text Mining*, Cham: Springer International Publishing, 2018, pp. 45–53. doi: 10.1007/978-3-319-78503-5_6.

[53] D. Hu, Y. Xie, X. Li, L. Li, and Z. Lan, “Inclusion of Machine Learning Kernel Ridge Regression Potential Energy Surfaces in On-the-Fly Nonadiabatic Molecular Dynamics Simulation,” *J Phys Chem Lett*, vol. 9, no. 11, pp. 2725–2732, Jun. 2018, doi: 10.1021/acs.jpclett.8b00684.

[54] A. Mahabub, “A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers,” *SN Appl Sci*, vol. 2, no. 4, p. 525, Apr. 2020, doi: 10.1007/s42452-020-2326-y.

[55] E.-S. M. El-Kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, “Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images,” *IEEE Access*, vol. 8, pp. 179317–179335, 2020, doi: 10.1109/ACCESS.2020.3028012.

[56] M. A. Khan *et al.*, “Voting Classifier-Based Intrusion Detection for IoT Networks,” 2022, pp. 313–328. doi: 10.1007/978-981-16-5559-3_26.