



REPUBLIC OF TÜRKİYE
ALTINBAŞ UNIVERSITY
Institute of Graduate Studies
Electrical and Computer Engineering

**INTRUSION DETECTION SYSTEM FOR IOT
APPLICATION BASED ON SUPERVISED LEARNING**

Sharafaldeen Abdulkadhum Abbas OBAID

Doctor of Philosophy

Supervisor

Asst. Prof. Dr. Abdullahi Abdu İBRAHİM

İstanbul, 2025

INTRUSION DETECTION SYSTEM FOR IOT APPLICATION BASED ON SUPERVISED LEARNING

Sharafaldeen Abdulkadhum Abbas OBAID

Electrical and Computer Engineering

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2025

The thesis titled INTRUSION DETECTION SYSTEM FOR IOT APPLICATION BASED ON SUPERVISED LEARNING prepared by SHARAFALDEEN ABDULKADHUM ABBAS OBAID and submitted on 31/05/2025 has been **accepted unanimously** for the degree of PHD of Science in Electrical and Computer Engineering.

Assist. Prof. Dr. Abdullahi Abdu İBRAHİM

Supervisor

Thesis Defense Committee Members:

Assist. Prof. Dr. Abdullahi Abdu İBRAHİM	Department of Computer Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Yaser ALAIWI	Department of Mechanical Engineering, Altınbaş University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical, İstanbul Arel University	_____
Asst. Prof. Dr. Tariq Abed MOHAMMED	Department of Computer Science and Information, Kirkuk University	

I hereby declare that this dissertation meets all format and submission requirements for a PhD dissertation.

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.

Sharafaldeen Abdulkadhum Abbas OBAID

Signature

ABSTRACT

INTRUSION DETECTION SYSTEM FOR IOT APPLICATION BASED ON SUPERVISED LEARNING

OBAID, Sharafaldeen Abdulkadhum Abbas

PhD., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu İBRAHİM

Date: 05/ 2025

Pages: 89

The Internet of Things (IoT) has evolved dramatically of recent years, and development of commercial and personal applications that can track a person's daily routine, attracting hackers to exploit the vulnerability to steal or modify collected data or disrupt system functions. To prevent hackers from compromising IoT devices, it is important to develop a method that captures and examines network traffic to identify and categorize malicious behavior that attackers may use. Therefore, this work uses a new IoT intrusion detection system designed to determine whether IoT network traffic is normal or potentially anomalous. The algorithm also determines the kind of anomaly if network traffic is deemed to be possibly abnormal. The performance of the proposed system was tested after selecting a dataset called IoTID20, which consists of 625,783 IoT network traffic packets, with 83 distinct features for each packet. The dataset is classified into three classes. The first class (Class I) classifies network traffic into normal or potential anomaly packets. The second class (Class II) classifies each anomalous network traffic into four main classes: 1. Mirai attacks. 2. Denial of service attacks (DoS), 3. Scanning. 4. Man-in-the-middle (MITM). The third class (Class III) classifies the anomalous network traffic into subclasses of the main classes. To identify the class of an IoT network packet, this system goes through four main stages: 1. Feature Preprocessing. 2.Feature selection. 3.Hyperparameter optimization. 4.Classification. In the feature preprocessing stage, the proposed system automatically

removes the feature ID column and empty row values and handles useless feature distributions. Therefore, to find salient characteristics to represent each IoT network traffic, the feature extraction step uses correlation coefficient, particle swarm optimization (PSO), and grey wolf optimization (GWO). In this step, features 17, 16, and 22 are selected to represent network traffic for Category 1, Category 2, and Category 3, respectively. In addition, in this stage, four features are selected because they are the common features chosen from all the feature selection algorithms that make our system consume limited computing resources. In the penultimate stage, the Decision Tree (DT) hyperparameter is determined utilizing a Coronavirus Herd Immunity Optimizer (CHIO) to boost the effectiveness of the suggested system. Finally, IoT network traffic is classified using a decision tree algorithm through three phases. In the first phase, it is classified as normal or anomalous. In the case of an anomaly, the system identified the main and subcategories of the anomaly in the second and third phases. Many machine-learning techniques are used to train and test our proposed system; however, the decision tree outperforms other machine-learning algorithms, achieving 99.96%, 99.56%, and 77.6% accuracy for Phases 1, 2, and 3.

Keywords: Internet of Things, Decision Tree, Coronavirus Herd Immunity Optimizer, Intrusion Detection System, GWO.

TABLE OF CONTENTS

	<u>Pages</u>
ABSTRACT	v
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
ABBREVIATIONS.....	xiii
1. INTRODUCTION	1
1.1 RESEARCH OVERVIEW	1
1.2 IMPORTANT TERMINOLOGIES.....	3
1.2.1 Internet of Things (IoT)	4
1.2.2 Kinds of Attacks	4
1.2.3 Intrusion Detection System (IDS).....	5
1.2.4 Decision Tree	6
1.2.5 Optimization Techniques with Coronavirus Herd Immunity Optimization (CHIO)	7
1.3 RESEARCH MOTIVATIONS.....	7
1.4 RESEARCH CONTRIBUTIONS	8
1.5 RESEARCH SCOPE	8
1.6 SIGNIFICANCE OF RESEARCH.....	9
2. LITERATURE REVIEW	10
2.1 MACHINE LEARNING	10
2.1.1 Unsupervised Learning	10
2.1.2 Supervised Learning	11
2.1.2.1 Logistic regression (LR)	12
2.1.2.2 Naïve base (NB).....	12
2.1.2.3 Artificial neural networks (ANN)	12
2.1.2.4 Support Vector Machine (SVM).....	13
2.2 DECISION TREE ALGORITHM (DT)	13

2.3 CLASSIFICATION OF SUPERVISED LEARNING	14
2.4 INTERNET OF THINGS	15
2.5 IOT ATTACKS	17
2.5.1 Comparison Between IoT Attacks and IT Attacks	17
2.6 INTRUSION DETECTION SYSTEM (IDS).....	19
2.7 IOTID20 DATASET	20
2.8 FEATURE SELECTION.....	23
2.8.1 Wrapper Methods.....	24
2.8.2 Embedded Methods	25
2.8.2.1 Particle Swarm Optimization (PSO)	26
2.8.2.2 Grey Wolf Optimizer (GWO)	26
2.8.3 Filter Methods	27
2.8.3.1 Correlation coefficient	28
2.9 RELATED WORK	29
2.9.1 Summary of Comparison of Previous Works Conducted on Different Types of Datasets	29
2.9.2 Summary of Comparison of Previous Research Conducted on Dataset IoTID2033	
2.10 SUMMARY	36
3. RESEARCH METHODOLOGY	37
3.1 RESEARCH APPROACH	37
3.2 STAGES OF DESIGN RESEARCH METHODOLOGY	39
3.2.1 Research Clarification.....	39
3.2.2 Descriptive Study1	40
3.2.3 Prescriptive Study (PS)	41
3.2.3.1 Experimental design.....	43
3.2.3.2 Verification and validation.....	43
3.2.4 Descriptive Study 2 (DS2)	43
3.3 TECHNIQUES FOR IMPLEMENTATION AND EVALUATION	44
3.3.1 Options for Implementation and Evaluation	44

3.3.2 A Proposed Technique for Designing an Internet of Things Intrusion Detection System Using the IoTID20 Dataset	45
3.3.3 Evaluation Metrics	45
3.4 SUMMARY	46
4. EXPERIMENTS, RESULTS, AND EVALUATION.....	47
4.1 OUTLINES THE INTRUSION DETECTION SYSTEM IN GENERAL	47
4.2 DATA PROCESSING AND FEATURE SELECTION.....	50
4.2.1 IoTID20 Dataset.....	51
4.2.2 Data processing.....	51
4.2.3 Feature Selection.....	55
4.2.3.1 Correlation Coefficient.....	55
4.2.3.2 Particle Swarm Optimization	56
4.2.3.3 Gray Wolf Optimization	58
4.3 DEVELOPING AN IDS FOR THE IOT.....	63
4.3.1 Hyperparameter Optimization (Coronavirus Herd Immunity Optimizer)	63
4.3.2 Classification.....	65
4.4 EXPERIMENTAL RESULTS AND EVALUATION.....	65
4.5 SUMMARY	69
5. CONCLUSION AND FUTURE WORKS.....	71
5.1 CONCLUSION.....	71
5.2 FUTURE WORKS	71
REFERENCES	73

LIST OF TABLES

	<u>Pages</u>
Table 2.1: Comparison of Six Machine Learning Algorithms[27]	15
Table 2.2: Summary of Comparison of Previous Research Conducted on Different Types of Datasets.....	32
Table 2.3: Summary of Comparison of Previous Research Conducted on Dataset IoTID20	36
Table 4.1: Feature Selection Results	62
Table 4.2: Decision Tree Results.....	68
Table 4.3: Comparison With Related Work	69

LIST OF FIGURES

	<u>Pages</u>
Figure 1.1: IoT Applications [1].....	2
Figure 1.2: Intrusion Detection System Architecture	6
Figure 2.1: Depicts The Operation of Unsupervised Learning	11
Figure 2.2: Depicts The Operation Of Supervised Learning.....	12
Figure 2.3: Decision Tree	14
Figure 2.4: A Representation of Various IoT Applications [30].....	17
Figure 2.5: Distribution of Different Attack Categories Within a Data set[31].....	19
Figure 2.6: IoTID20 Dataset Testbed Environment [34]	21
Figure 2.7: IoTID20 Dataset Normal and Attacked Instances	22
Figure 2.8: The Procedure of Wrapper Selection.	25
Figure 2.9: Schematic of Filter Method	28
Figure 3.1: Research Approach	38
Figure 3.2: The Research Clarification Steps.....	39
Figure 3.3: Key Procedures for DS1	41
Figure 3.4: PS Steps	42
Figure 3.5: Key Steps in the Evaluation	44
Figure 4.1: General Architecture of the IDS	48
Figure 4.2: Proposed System Architecture	49
Figure 4.3: Data Processing and Feature Selection Steps	50
Figure 4.4: Binary, Category, and Subcategory of IoTID20 Dataset	51

Figure 4.5: :Best Feature Distribution	53
Figure 4.6: Heatmap of the Label Target's Ten Most Label.....	54
Figure 4.7: Heatmap Shows the Target Category's Ten Most Crucial Attributes	54
Figure 4.8: Heatmap Displaying the Chosen Subcategory's Top Ten Key Attributes	55
Figure 4.9: Flowchart of CHIO Algorithm[72].....	64
Figure 4.10: Results Using all 67 Features.....	66
Figure 4.11: Results Using the Full Selected Features.....	66
Figure 4.12: Decision Tree Hyperparameter Values After Optimization	67
Figure 4.13: Decision Three Results	68

ABBREVIATIONS

ANN	:	Artificial Neural Networks
ARP	:	Address Resolution Protocol
DCNN	:	Deep Convolutional Neural Network
DRM	:	Digital Rights Management
DT	:	Decision Tree Algorithm
DT	:	Decision Tree
EC	:	Evolutionary Computation
ELM	:	Extreme Learning Machine
GB	:	Gradient Boosting
GWO	:	Grey Wolf Optimizer
HTTP	:	Hyper-Text Transfer Protocol
IDS	:	Intrusion Detection System
IoT	:	Internet of Things
IP	:	Internet Protocol
LR	:	Logistic Regression
MAC	:	Media Access Control
MITM	:	Man-In-The-Middle
PS	:	Prescriptive Study
PSO	:	Particle Swarm Optimization

RF : Random Forest

RFID : Radio-Frequency Identification

SI : Swarm Intelligence

SVM : Support Vector Machine

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

WSN : Wireless Sensor Networks

1. INTRODUCTION

1.1 RESEARCH OVERVIEW

The dynamic and streamlined evolution witnessed in a wide range of technologies that are deeply interrelated with wireless sensor networks (WSN), mobile communications, as well as radio-frequency identification (RFID), and a host of other lightweight protocols, have long contributed and provided tremendous support to the innovative and revolutionary vision of what we today know and identify as the Internet of Things. At the very heart of the underlying concept that drives and fuels the ambitious vision of IoT is the dynamic and complex interconnection of billions and billions of different units or devices that not only coexist but flourish within an enormous mega-ecosystem, which is driven to perfection either by conventional wired means or through advanced wireless communication systems. Such an unprecedented degree of connectivity is made possible with the invaluable help of intelligent sensors, advanced actuators, and an enormous range of other devices that assume central and pivotal roles within this large network [1].

The Internet of Things (IoT) is the new wave of the Internet in which any object can transfer information without human intervention. Each IoT system consists of three main components are: microcontroller, sensors, and actuators. Sensors collect data from the environment and then send the collected data to a microcontroller, which processes the data and sends action to actuators. Two or many IoT systems can share the collected data through the internet. Because IoT aims to improve individuals' daily lives, it services many applications [2]. Figure 1.1 shows some IoT applications and shows how the IoT field helps these applications become smarter[1].

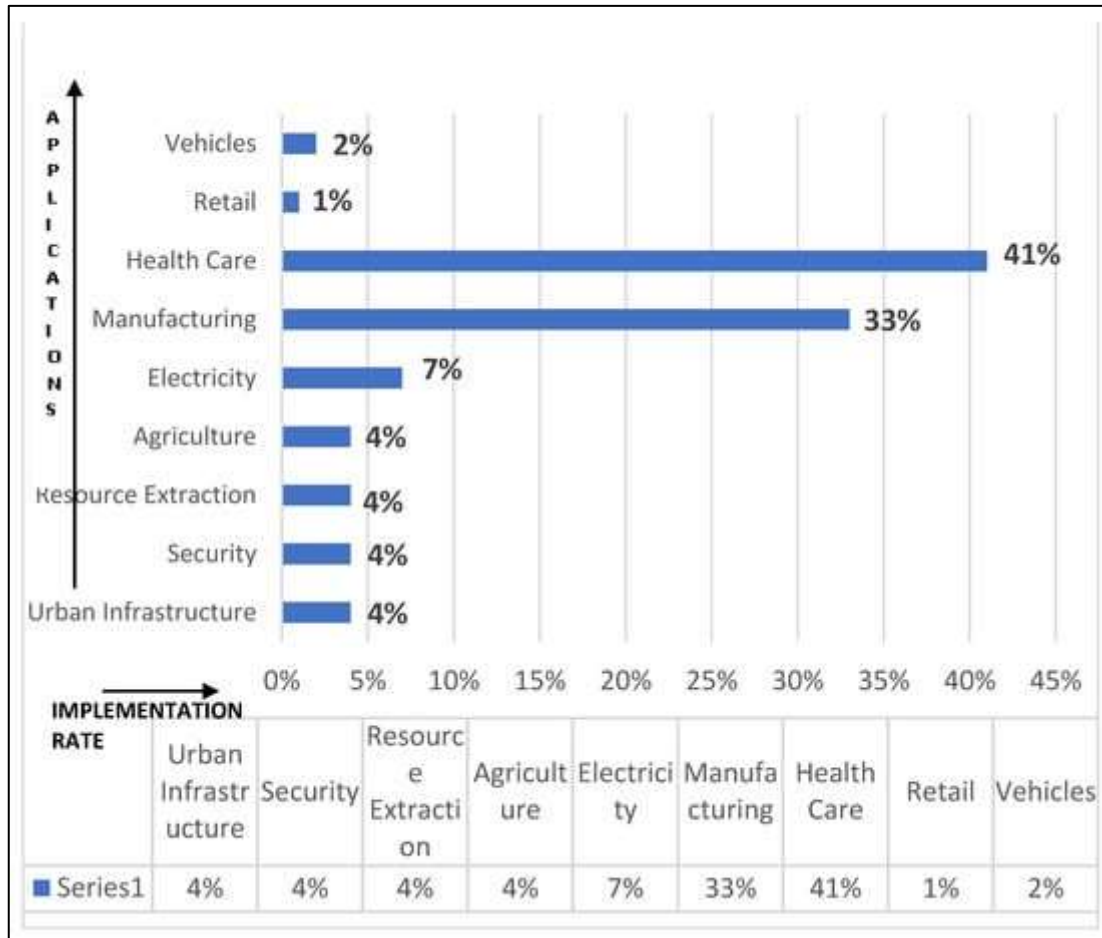


Figure 1.1: IoT Applications [1].

With so many areas served by the Internet of Things, about 27 million IoT devices were used in 2017, and scientists say there will be 125 billion IoT devices all over the globe 2030[3]. The main core of IoT devices depends on connecting these devices across the internet to share and send the collected data from sensors. This makes IoT-collected data vulnerable to attack or modification by hackers using an intrusion [4]. Intrusion in the context of IoT means that attackers access the network traffic that travels from or to the IoT system and modifies. Many types of intrusion can harm IoT network packets, such as Man-in-the-Middle (MitM) attacks, Denial of Service (DoS) attacks, and Mirai. In DoS attacks, hackers send many false network packets through the same machine to overwhelm IoT devices and then cause IoT devices to fail or down. While hackers intercept the communication between the sender and recipient to spy on the data and then steal or modify it in MitM attacks, hackers attack vulnerable devices such as routers and IP cameras to infect them with malware in Mirai attacks. IoT devices are subject to intrusion attacks every day due to the significance

of the data they gather. According to the Kaspersky Lab, 962 million IoT devices have been attacked by different types of intrusion across 187 countries [5].

By analyzing every IoT network transaction, several network-based intrusion detection systems (IDS) are utilized to spot anomalous activity and stop hackers from spreading the IoT systems. Network IDS can be divided into two ways: signature-based detection and anomaly-based detection. While signature-based detection analyzes network traffic in a network packet and compares it to pre-defined patterns to determine the type of intrusion, anomaly-based detection identifies intrusion by continually monitoring the impact of network traffic on the entire system[6]. Unfortunately, network-based IDS suffers from many limitations in the IoT field, such as (1) IDS systems consuming many resources and IoT devices having limited resources due to limited power sources, (2) the difficulty of IDS monitoring all network traffic because they come from heterogeneous networks and different protocols in IoT system, and (3) IoT is real-time that collects big amounts of data from any sensors and other IoT devices that need to detect unusual behavior of network traffic as fast as possible, which is difficult to traditional IDS [7]. This study used a large and complex dataset containing smart home application traffic. A new, machine-learning-based intrusion detection system suitable for complex datasets was developed. By identifying the characteristics of each network traffic and reducing these characteristics to shorten the time it takes to categorize each network traffic, this system employs artificial intelligence (AI) and machine learning (ML) techniques to distinguish between normal and anomalous network traffic. The decision tree algorithm was chosen in this research for its advantages: classification speed, tolerance to missing values, handling discrete, binary, continuous features, and interpretability, knowledge transparency, classifications [8].

1.2 IMPORTANT TERMINOLOGIES

The primary objective of this work is to develop a new system for detecting attacker intrusions Internet of Things (IoT) to determine whether IoT network traffic is normal or potentially anomalous. In case network traffic is classified as a potential anomaly, the system also identifies the type of anomaly. The next section will define some essential technologies related to the domain of this study.

1.2.1 Internet of Things (IoT)

Internet of Things, or simply IoT, is the complex web that comprises an expansive range of varying physical objects, diverse equipment, and myriad sensors that communicate through the massive medium that is the Internet. This sophisticated and advanced web not only allows such entities to communicate with one another in an incredibly efficient manner but also to pass on valuable and worthwhile data that can prove to be useful for numerous purposes. In healthcare, IoT is opening doors to new personal treatment plans that cater to individualized needs for every patient, while also making it possible for healthcare services to become more efficient in their delivery. Furthermore, IoT's impact goes far beyond healthcare's boundaries; it is also profoundly disrupting other fields in addition to manufacturing, transportation, and farming. This advanced technology also has immense potential to alter multiple sectors of our society in positive ways in an incredibly impactful manner. Given these stunning breakthroughs, it is now more crucial for individuals as well as organizations to remain abreast of new developments in this exciting emerging field. Lastly, there is still plenty to discover when it comes to applying IoT to bring about positive change in numerous areas in our lives.

In addition, Internet of Things (IoT) technologies increase effectiveness of operations through automation of administrative activities, leading to time savings, while simultaneously reducing the need for a huge administrative workforce, thus lowering costs [9]. One of the major drawbacks of the Internet of Things is its lack of adequate security compared to the security in place of servers and other devices. [10].

1.2.2 Kinds of Attacks

Attacks can be classified as either passive or aggressive. They also includes an attacker attempting to access the Internet of Things. Until an active assault is carried out, the hacker adds data to the system and may even alter existing data. Active attack types include distributed masquerade, session replay, and denial of service (DoS)[11].

Trojan horses, worms, and viruses are a few types of active assaults. Additionally, the passive attack seeks to understand or the organized layers of IDS in the Internet of Things utilize system data without compromising system resources. Passive attacks include encryption, scanning, and tapping[12].

1.2.3 Intrusion Detection System (IDS)

Intrusion Detection Systems, abbreviated as IDS, is designed to closely monitor and inspect network traffic that is emerging from a range of different resources in an effort to detect any possible malicious actions that are underway. This capability is an integral element of the general discipline of computer cybersecurity. In conclusion, IDS is a systematic practice done to identify malicious actions that are directed against victims using a number of different approaches [11].

Over the past decade, threat evaluation methodologies in the scope of cybersecurity have been effective ones. These methodologies have given government as well as organizational entities a mechanism to improve their protective capabilities against relevant threats. The application of available evaluation methodologies within these settings can lead to missing emerging threats that become developed. The high connectivity rates, combined with cyber-physical as well as social structures, are inherently associated with these challenges. This essay supports innovative approaches to risk assessment for this environment, one that addresses not only the IoT's dynamic nature and special knowledge, but also the comprehensiveness inherent to risk assessment practice [13]. Figure 1.2 shows Intrusion Detection System Architecture.

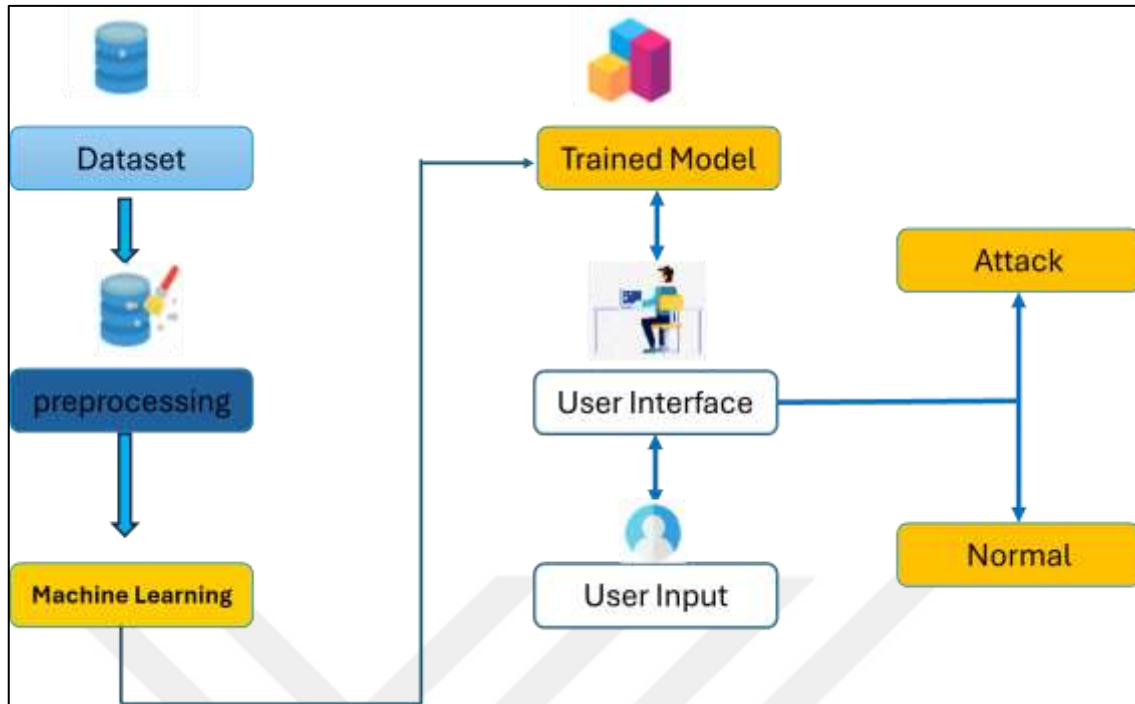


Figure 1.2: IDS Architecture.

1.2.4 Decision Tree

Applications based on machine learning are transforming many fields and businesses, like marketing, finance, and healthcare. Large datasets and technological advancements have made machine learning (ML) algorithms more potent and precise in their capacity to provide predictions and well-informed conclusions. These apps are changing the way businesses function and opening the door to a future that is more data-driven and efficient[14].

Decision tree approaches have been beneficial across a range of applications, including both classification, regression, feature selection, among others. The underlying idea behind decision tree algorithms is the recursive division of the dataset based on the values of different features to a stopping point defined by a criterion. This recursive process results ultimately in a hierarchical structure that is similar to a tree, where every node is a decision or a split on the basis of a feature [15].

1.2.5 Optimization techniques with Coronavirus Herd Immunity Optimization (CHIO)

Optimisation techniques and methods are considered influential factors in enhancing the accuracy of classification algorithms and decreasing the complexity of detection and classification problems in record time. Effective data extraction and optimal selection enable better data preparation when designing and creating classification models, especially for data containing errors or unnecessary spaces. The role of the optimiser is to make adjustments during the search and update process until the required values, deemed optimal, are achieved. The optimal solution involves finding the minimum and maximum solutions and employing them to reach the goal [15,16].

Given the multitude of optimization algorithms, we will discuss one of these algorithms, Coronavirus Herd Immunity Optimization (CHIO) is the first one: When the majority of individuals become sick and a sizable portion of the population develops immunity to the virus, this is known as herd immunity. There are three components to the herd immunity population. Those who are vulnerable are those who can get the infection from sick people. While immune people are shielded from viral infection, the diseased population can transmit the virus[18].

1.3 RESEARCH MOTIVATIONS

IoT devices manage enormous volumes of data that may be used for a variety of reasons, which makes them a desirable target for a variety of attackers, such as cybercriminals, hackers, and infrequent hackers. Potential attackers may be encouraged to steal important data, like bank account passwords, credit card numbers, and location information, by breaking into IoT devices. Additionally, they could try to target a third-party business by making use of IoT components like edge nodes. Furthermore, several risks and privacy issues have arisen as a result of the quick development of machines and technology. Within a network, smart devices exchange data and communicate with one another. If one device is hacked, the entire system is at risk.

As a result, security and privacy have become much more important in recent years. To protect against such attacks, security standards must be developed. A corrupted system, for instance, may endanger output and compromise important data. For this reason, a new detection system will be proposed.

1.4 RESEARCH CONTRIBUTIONS

The system makes several noteworthy contributions, such as

- a. The Internet of Things Intrusion Dataset 2020 (IoTID20), which includes 83 features and three classification categories, allows network traffic to be classified as normal or anomalous. Anomalous traffic is further categorized into many different kinds: man-in-the-middle (MITM) attacks, Mirai attacks, scanning, and denial-of-service (DoS) attacks. The classification also allows for a more precise distinction between specific kinds of Mirai and scanning attacks.
- b. Used a methodical approach that included crucial steps like as feature selection, classification, hyperparameter tuning, and data preparation.
- c. Efficient preparation of data includes managing null values, eliminating identification features, and dealing with noisy feature distributions. Following data preparation, this phase simplified the feature set of 67.
- d. Combining three feature selection techniques the correlation coefficient, Particle Swarm Optimization (PSO), and Gray Wolf Optimization (GWO) provide fine-tuned features for labels, categories, and subcategories. The feature set was further reduced to four essential functions using the strategic commonality technique, which made it appropriate for IoT systems with little resources.
- e. The Coronavirus Herd Immunity Optimizer (CHIO) algorithm was used to improve the system's effectiveness using a decision tree algorithm. The new system achieved good accuracy rates compared to other algorithms for classifying tags, categories, and subcategories.

1.5 RESEARCH SCOPE

The objective of the research is to design a new system that will be applied to the Internet of Things for intrusion detection. The research features the application of a hybrid system on the IoTID-20 dataset to detect intrusion and identify attack types by applying a decision tree algorithm with a hyperparametric optimization algorithm.

1.6 SIGNIFICANCE OF RESEARCH

The new hybrid implementation of the decision tree algorithm and the Corona Virus optimization algorithm is expected to identify and classify anomalies in Internet of Things (IoT) data more effectively and quickly, contributing to practical applications that mitigate attacks on IoT applications and reduce damage in terms of cost.



2. LITERATURE REVIEW

2.1 MACHINE LEARNING

Technology has seen significant advances, particularly in the field of machine learning (ML), which plays a key role in reducing manual effort. In the broader field of artificial intelligence, combining elements of computer science and statistics creates machine learning algorithms that improve performance by presenting relevant data, rather than utilizing explicit programming. These machine learning algorithms create a model population using a representative dataset, sometimes known as "training data," to enable them to make independent predictions or judgments. [19].

In every dataset applied in ML algorithms, each instance is represented using the same group of attributes. These attributes may take the form of binary, continuous, or categorical values. When examples are provided with known labels (accurate outcomes), The method is known as supervised learning. This approach involves deriving a function from labeled training data, enabling the system to evaluate unseen testing data and construct a predictive task applicable to new scenarios. Notably, every input data instance carries a predefined class label [20].

The primary goal of supervised machine learning is to construct a model that can accurately identify the given data and generalize to unpredictable data. This forms the basis of classification, where the focus is on accurately identifying the target class. Classification algorithms aim to identify the connection between input attributes and corresponding output attributes and construct a model via a structured training process. In data mining contexts, the amount of available data is typically very large. When the dataset is efficiently organized and contains a minimal number of nodes, the decision tree technique proves to be a highly successful approach [21].

2.1.1 Unsupervised Learning

In contrast, unsupervised learning focuses on identifying invisible patterns within data that hasn't been labeled. This approach does not depend on manual annotation. Instead, it typically utilizes clustering techniques. Essentially, unsupervised learning involves enabling a computer to process data without predefined labels, allowing the algorithm to function

independently [21]. The main aim is for the system to organize the raw data into categories based on recurring features, underlying patterns, or noticeable differences. as below in Figure 2.1.

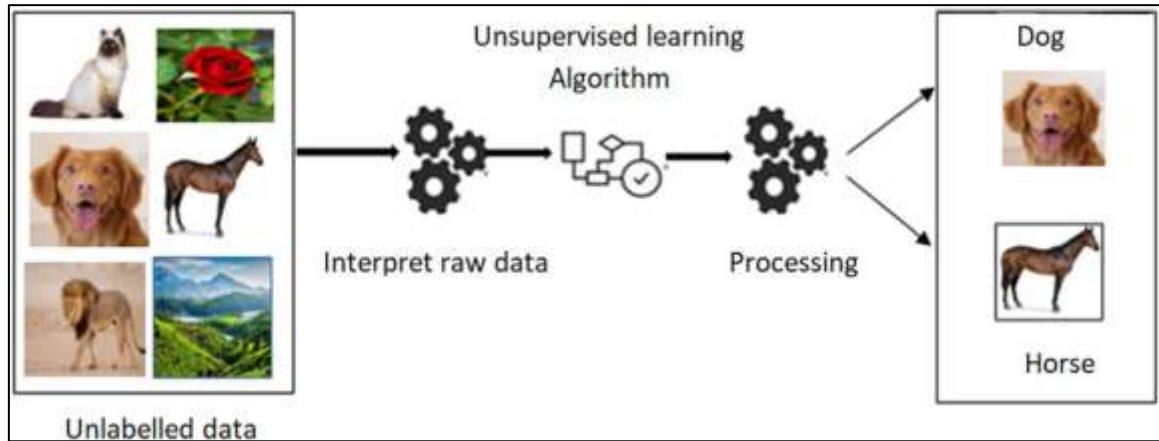


Figure 2.1: Depictions of Unsupervised Learning.

2.1.2 Supervised Learning

The primary requirement of supervised learning is manual data labeling, which can hinder the general efficiency of the system. Its main goal is to map input and output variables. This learning approach resembles a classroom environment where the teacher provides guidance to learners. Models are designed for specific tasks and rely heavily on large amounts of labeled data, which is then manually labeled. Typically, this data is randomly divided into three subsets: training, test, and validation. The effectiveness of labeled data relies heavily on computational applications, although obtaining it is expensive and time-consuming. Figure 2.2 illustrates how supervised learning works. [21][22].

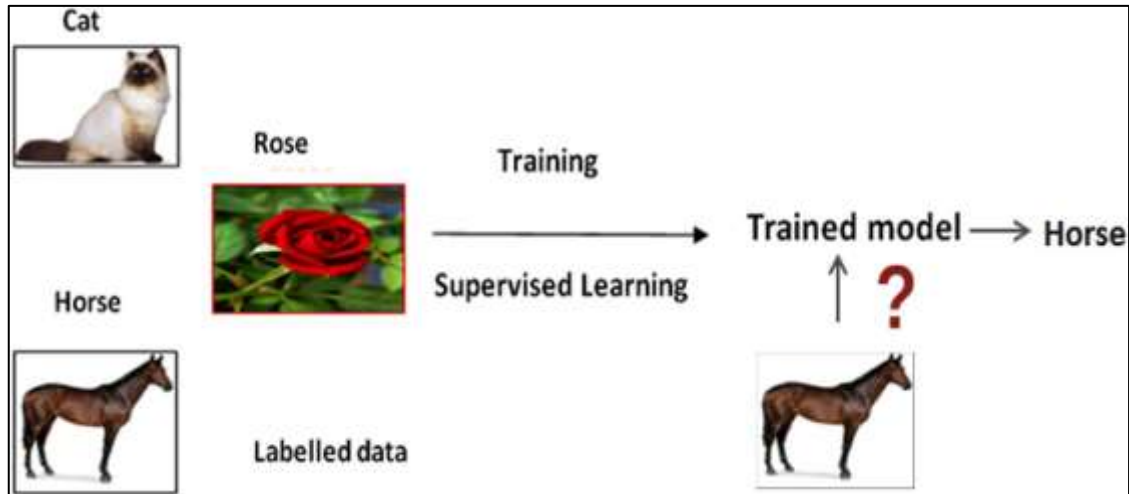


Figure 2.2: Depictions Of Supervised Learning.

2.1.2.1 Logistic regression (LR)

It is a predictive analysis technique based primarily on probability. It uses a function called a sigmoid function that converts all predicted values into probabilities that divide between 0 and 1, making it a successful method for binary and linear classification problems. This approach works well with discrete data and is relatively easy to implement [24].

2.1.2.2 Naïve base (NB)

They are several classification techniques based on Bayes' theorem. All of these algorithms are multi-method rather than a single technique, and they all follow the same guiding concept, where each pair of attributes is classified as an independent entity [25].

2.1.2.3 Artificial neural networks (ANN)

Artificial neural network technology is designed to work based on the neural network in the human brain from a biological perspective and is considered one of the machine learning techniques. The second layer receives the weight values of each neuron and serves as its output. A neural network is called feedforward when it receives input from the previous cells. One of the most common types of networks is the multilayer perceptron (MLP) feedforward network. MLP algorithms require training algorithms, the most important of which is backpropagation, which adjusts the weights between neurons to reduce error. The system can quickly adapt to new data values, but this technique exhibits slow convergence and therefore risks achieving a local optimal value [24].

2.1.2.4 Support Vector Machine (SVM)

This method primarily operates at a very high level, maximizing the distance between two classes. Classification provides the foundation for future data processing. This technique uses hyperplanes (lines) to partition groups into multiple configurations. A support vector machine (SVM) creates a learning model that partitions new samples into multiple classes. Depending on these functions, SVMs are known as binary linear classifiers or non-probabilistic classifiers. When using probabilistic classification, SVMs can use techniques such as the Platt metric [25] .

2.2 DECISION TREE ALGORITHM (DT)

AI and ML now play a vital role in contemporary information security frameworks. Their strength lies in the ability to swiftly examine extensive network activity and recognize various types of threats. As these technologies develop further, they are increasingly utilized to develop proactive incident detection systems capable of managing new and emerging forms of cyberattacks. Profiles for users, assets, and networks are generated based on historical behavior, enabling AI to identify and react to unusual or suspicious actions. A commonly applied ML algorithm for detecting harmful behavior is DT [26].

DT is a hierarchical, tree-based approach where every path for the root node into a leaf is defined by a series of data-splitting steps, ultimately resulting in a Boolean decision. It represents a structured form of knowledge, with nodes and their interconnections illustrating relationships. In classification tasks, these nodes serve specific functional roles [21]. As illustrated in Figure 2.3 DT, this method can process large volumes of intrusion detection data, helping to pinpoint critical network characteristics that signal harmful behavior. Additionally, it can uncover patterns valuable for future investigations, identifying attack traits, and supporting network monitoring operations [26].

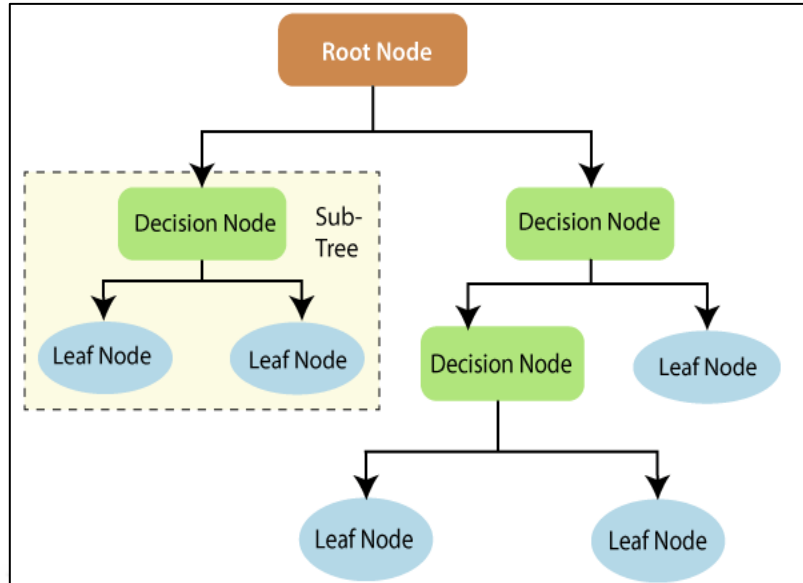


Figure 2.3: Decision Tree.

2.3 CLASSIFICATION OF SUPERVISED LEARNING

The most important machine learning algorithms that are classified as supervised and focus most of their applications on classification are: Neural networks, Bayesian networks, K-Means clustering, boosting, decision trees, random forest (RF), linear classifiers, logistic regression, naïve bayes classifier, perceptrons, support vector machines, and so on.

In general, the operating characteristics of decision trees are characterized by high accuracy, while neural networks do not, and vice versa. While the operating characteristics of rule classifiers and decision trees are largely similar, the operating characteristics of SVM and ANN algorithms are also similar. However, no learning algorithm can consistently perform better on all types of data. Features of learning techniques are compared in Table 2.1 (Based on current studies) [27].

Table 2.1: Comparison of Six Machine Learning Algorithms [27].

	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule- learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems)	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	*** (not discrete)	*** (not continuous)	*** (not directly discrete)	** (not discrete)	*** (not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

2.4 INTERNET OF THINGS

Internet of Things is a fresh and revolutionary term that has become highly relevant and significant over the past couple of years and has caught the interest of many. Its current stature can be attributed largely to the seamless and harmonious integration of a wide and diverse array of new technologies with various technologies of communication [10]. In a simple sense, Internet of Things or IoT can be defined as a complicated and sophisticated network made up of a wide and diversified array of devices, sensors, actuators, software, and various other networked entities. While all of them operate in a highly synchronized manner, they are all responsible to efficiently store, process, and transport precious data in a very efficient and streamlined manner. When we draw a more detailed glimpse at the many benefits offered by IoT in various applications and industries, we can recognize several distinct and remarkable features that distinguish them. Some of them include an

unprecedented level of process automation, better levels of seamless communication, and an increased exchange of information. Incredible enough, all this happens with much less effort and time than in many conventional practices and procedures that have been followed over time in various contexts [28]. In an IoT network, physical devices are equipped with the capability to synchronize and manage their activities autonomously, thereby becoming what we now call intelligent devices. Intelligent devices are spreading at an unprecedented rate and are integrated in an increasing manner in the fabric of everyday life, affecting a wide range of contexts that span beyond personal homes to massive industrial and institutional settings. Internet of Things has revolutionized the world to its very foundations by injecting new levels of innovation in people's everyday life, enabling indirect communication between various intelligent devices that do not necessarily interact directly with each other. However, this revolutionary development, sadly enough, also gives birth to various categories of cyber frauds and attacks, which pose a very significant threat to the security and integrity of the systems [29]. The basic principle that forms the foundation of IoT is that the ubiquitous presence of a wide array of objects, like intelligent devices, sensors, and actuators, all interact with each other towards a common aim or purpose. In this new era where the relevance and role of IoT continue to accelerate and build up, a very significant proportion of this significance can be attributed to its tremendous influence in every area of everyday life and the many activities in which people show interest and participation on a day-to-day basis, this spread led to an increase in the number of attacks on him. [10]. As below Figure 2.4.

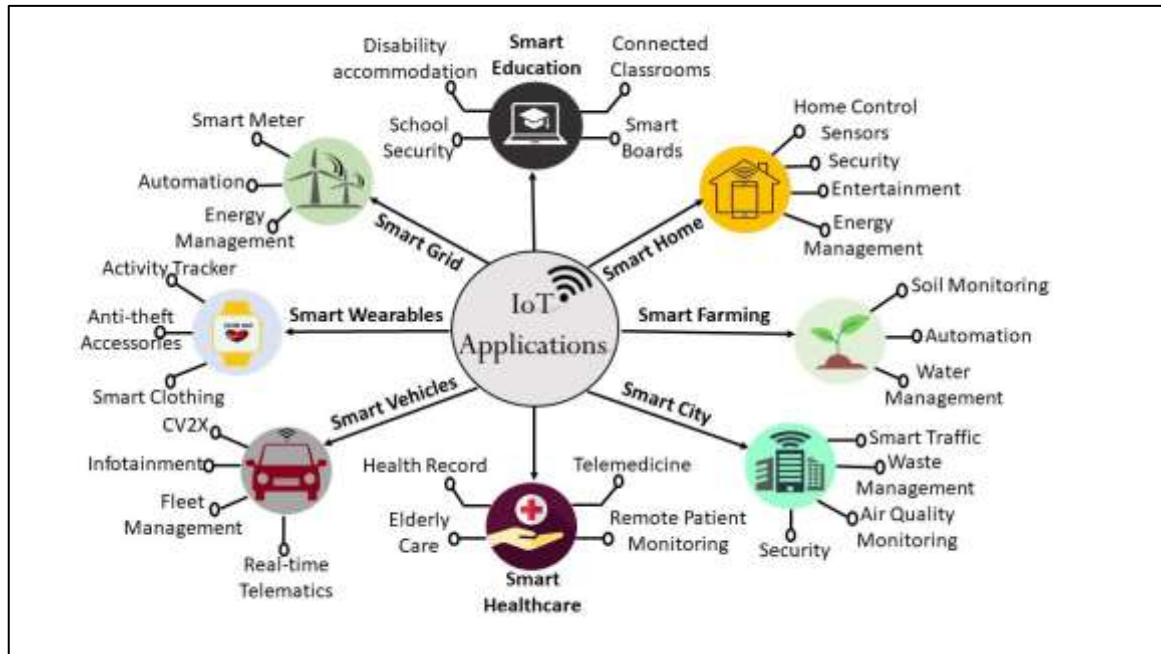


Figure 2.4: A Representation of Various IoT Applications [30].

2.5 IOT ATTACKS

IoT attacks refer to cyber intrusions that exploit connected devices to gain access to users' sensitive information. Cybercriminals often install malicious software, damage the device, or use it as a gateway to infiltrate broader organizational networks. Many IoT devices lack robust security features by design, creating vulnerable points in a company's infrastructure and posing serious security threats and being easily attacked.

Entry-level IoT devices typically do not come equipped with sufficient built-in protection to defend against such threats. Hackers can take advantage of known vulnerabilities or previously undiscovered exploits—known as zero-day attacks to compromise these devices. Once breached, attackers can repurpose them in multiple ways to launch widespread and coordinated cyber-attacks.

2.5.1 Comparison Between IoT Attacks and IT Attacks

Traditional IT threats are relatively simple compared to the significant challenges posed by IoT applications, requiring security measures specifically designed to provide comprehensive protection [10]. These challenges can be categorized in several ways:

- a. **Attack Surface:** Internet of Things devices often operate with limited processing power and memory. For these reasons, their built-in security mechanisms are inadequate, making them more vulnerable to attacks from traditional IT systems.
- b. **Device Diversity:** IoT encompasses a wide range of devices, each varying in size, software, and network protocols. This lack of standardization complicates the implementation of universal security strategies, increasing exposure to potential threats.
- c. **Physical Consequences:** IoT technology is commonly integrated into critical systems, including medical equipment and infrastructure. A cyberattack on such devices can result in severe physical harm, unlike typical IT breaches that mainly focus on data theft or service disruption [10].
- d. **Outdated Devices:** IoT products are often used far beyond their intended lifespans. As a result, many remain in operation without receiving necessary software updates or security patches, making them particularly susceptible to exploitation or compromise [10].

An examination of the dataset's attack categories uncovers a multifaceted and detailed threat environment, as shown in Figure 2.5. The distribution highlights a distinct ranking in threat occurrences, with one leading category making up 15.38% of all documented attacks. A second notable category follows, representing 11.56% of the total. The mid-level threat area is composed of several notable types, accounting for 9.66%, 8.81%, 8.74%, and 7.69% each. These are further supplemented by moderately common attack types, which include categories representing 5.70% and 4.31%.

The remaining portion of the threat distribution includes numerous less prevalent categories, each contributing under 3% to the total number of attacks. Although individually small, these may signal newly emerging or advanced attack methods that should be monitored closely [31].

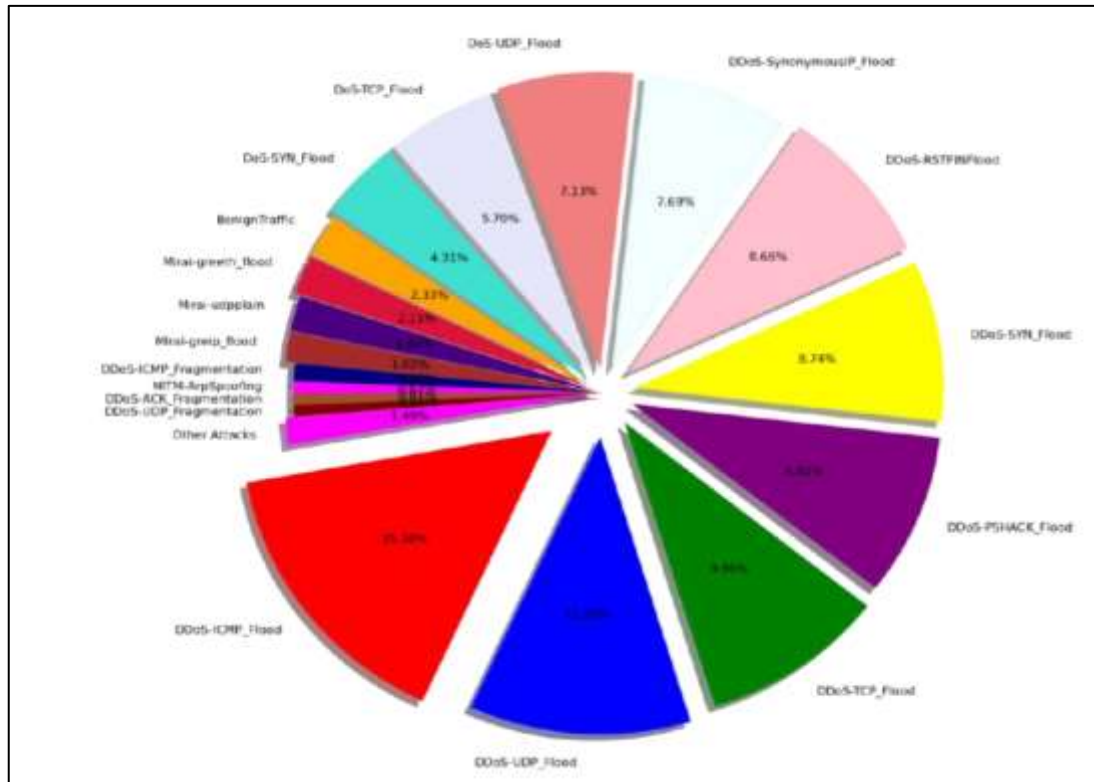


Figure 2.5: Distribution of Different Attack Categories Within a Data Set [31].

2.6 INTRUSION DETECTION SYSTEM (IDS)

Cyber threat detection systems are hardware or software solutions designed to automatically detect and respond to cyber threats. These systems work closely with intrusion response mechanisms, which continuously assess system integrity using alerts generated by IDS to quickly detect and address suspicious activities or anomalies [11]. Within the broader framework of cybersecurity, IDS play a key role by actively tracking system behavior to identify potential threats in advance.

The advancement of threat detection methods has paralleled the rise of sophisticated Machine Learning (ML) models. Early versions of IDS were quite basic, primarily depending on matching system activities to predefined patterns stored in manually maintained signature databases [32].

Intrusion response systems apply targeted countermeasures to help maintain cybersecurity. Implementing alert mechanisms is essential for ensuring these systems respond effectively. It's also important to use tools that can analyze and categorize threats based on their impact

on data confidentiality, availability, and integrity. For instance, if a cyberattack compromises the integrity of a company's database, the system must respond appropriately to preserve data accuracy. In contrast, if the threat targets the system's performance, the response should focus on enhancing resource availability and efficiency.

If an attack or intrusion is detected, the intrusion detection system issues alerts in the form of notifications, reports, etc. However, an intrusion detection system alone is not sufficient without corresponding security responses [13]. To trace the source of an attack, response mechanisms must work alongside IDS. Standard IDS models often rely on static rules, which reflect the stable behavior of the monitored systems. In contrast, IoT environments are dynamic, with nodes frequently joining or quitting the network [33]. Each IoT node can also have distinct security requirements.

Typically, a system administrator is tasked with defining and managing security rules. With virtualized infrastructures, tracking internal IoT traffic becomes more difficult, reducing visibility. Traditional physical monitoring tools are not suited for analyzing such environments. Moreover, emerging virtualization technologies may introduce new vulnerabilities, potentially increasing the risk of security breaches. These must be regularly monitored and analyzed. Traditional security tools are often not adaptable to IoT-specific conditions, which demand tailored approaches. In many cases, conventional IDS solutions fall short of meeting these unique demands. Therefore, IDS functions as a monitoring mechanism for detecting unauthorized access, activity, or file modifications within networks or computer systems [11].

2.7 IOTID20 DATASET

The dataset IoTID-20 [34] was with utmost care gathered by closely monitoring and observing the intricate network traffic that entails a vast spectrum of Internet of Things (IoT) devices, that is, the EZVIZ Wi-Fi camera and the SKT NGU, both very much seen in typical modern day smart homes. In Figure 2.6, the reader will have the advantage of seeing and intuitively gaining an understanding of just precisely how two different IoT devices are seamlessly and efficiently connected to the WiFi routers that are hosting the smart home, and this is a very critical part that allows uninterrupted monitoring of the immense IoT network traffic. Further, the router of the smart home is a hub that happens to be a center

point, and with its assistance, there are a few different types of connection made with a plethora of different electronic devices, some of them but not restricted to tablets, laptops, and smartphones. Every device that is connected plays a very critical part, since every one of them is also serving as attacking devices that were actually designed to perform attacks against the two targeted IoT devices, leading to the aggregate build-up of the immense IoTID-20 dataset. This dataset, by the way, was actually gathered by downloading Pcap files that are easily accessible all over the Internet [34]. In order to preprocess and analyze this very valuable information effectively, researchers made full use of the CIC-flowmeter program [35], a program that is especially designed to extract valuable features from the Pcap files and thereby converting the IoTID-20 dataset to a much more friendly CSV format. As soon as this convert process was done with a successful outcome, the next, very critical step was labeling each single instance that was included within the IoTID-20 dataset. This very large dataset contains a total of 625,783 instances of IoT network traffic, where each instance gets explained by 83 different features. Apart from this, there are necessary labels that belong to different areas like intrusion detection, category, and subcategory. The visual distribution of the records to the dataset, dividing the normal and anomaly classes, is graphically illustrated in Figure 2.7.

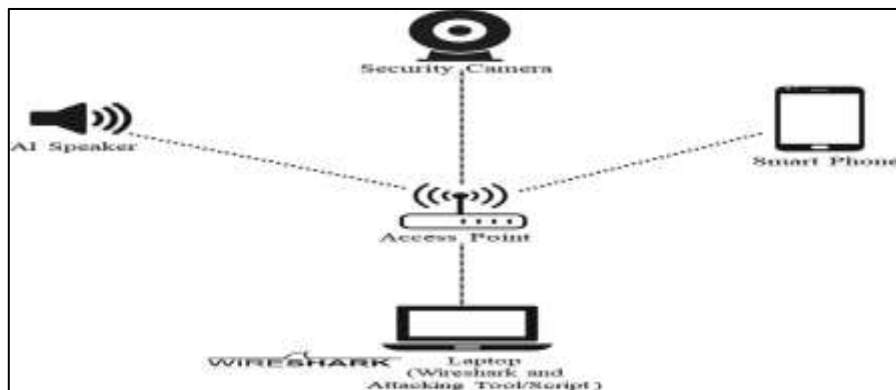


Figure 2.6: IoTID20 Testbed Environment [34].

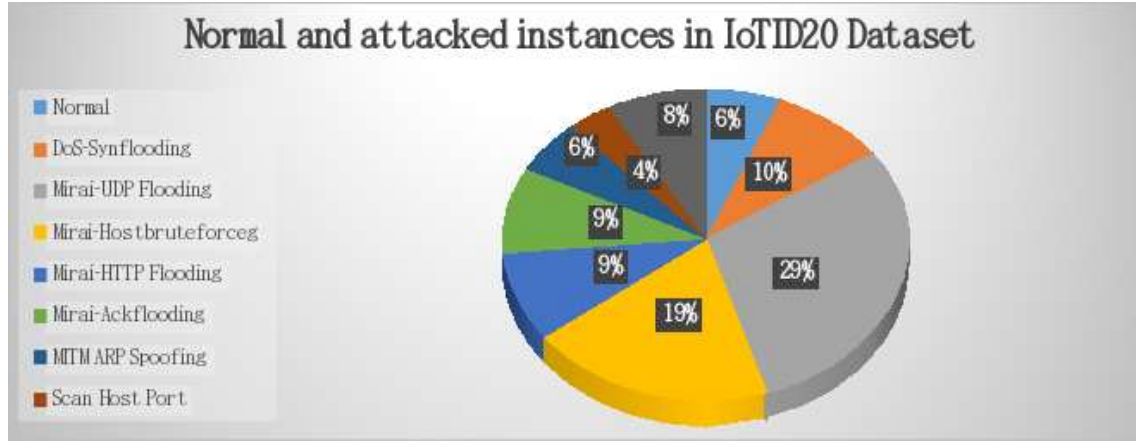


Figure 2.7: IoTID20 Normal and Attacked Instances.

The dataset includes different kinds of cyber-attacks like denial of service (DoS), man-in-the-middle (MITM), distributed denial of service (DDoS), and active scanning, so each of them has been mapped and tracked as among the most prevalent types of attacks. SYN (Synchronized) packets are commonly employed to initiate TCP connections between systems, allocating resources like ports and buffers on both ends. In DoS-synflooding attacks, this technique is exploited to disrupt access to servers or computers associated with the targeted system.

Additionally, flooding techniques using Hyper-Text Transfer Protocol (HTTP), acknowledgment packets, and User Datagram Protocol (UDP) packets are commonly utilized to emulate DDoS behavior, particularly in IoT botnets such as Mirai. The dataset also documents brute force attacks [36], which aim to decrypt protected data, thereby breaching confidentiality.

MITM attacks in this context involved altering the Address Resolution Protocol (ARP) table, mapping the attacker's Media Access Control (MAC) address to the router's Internet Protocol (IP) address. This manipulation allows the attacker to impersonate the router, enabling interference with communication between networked devices. The primary aim of such an attack is to intercept or manipulate data while it is being transmitted[35].

2.8 FEATURE SELECTION

Several elements influence how successful a machine learning model is at performing a particular task. Chief among these is how well the training data is represented and the overall quality of that data. In theory, adding more features should enhance a model's ability to distinguish between outcomes. However, real-world experience with machine learning suggests that this assumption doesn't always hold true. In many cases, learning algorithms essentially provide a (possibly biased) estimate of the likelihood that a given input corresponds to a particular class label, based on the provided features[36].

This process can be quite challenging. Often, machine learning models are trained on limited datasets, which complicates the task of accurately estimating numerous probabilistic parameters. To prevent overfitting, there's a tendency to favor simpler models that still provide a reasonable level of performance on the training data. This preference introduces a bias, leading algorithms to focus on a smaller subset of predictive features, even though a larger set—if used together—might offer better predictive power for determining class labels.

However, when the dataset contains excessive unrelated or redundant information, or when the data is loud and inconsistent, the learning process during training becomes significantly more difficult.

Feature subset selection involves identifying and eliminating as much irrelevant or duplicate data as possible. This process helps reduce the overall dimensionality of the dataset, potentially allowing learning algorithms to perform more efficiently and with improved speed. In certain scenarios, it can enhance the accuracy of future classifications, while in others, it produces a more concise and interpretable representation of the underlying concept [36].

Techniques for feature selection may be broadly divided into three kinds: filter, wrapper, and embedded methods. Filter techniques operate independently of any classification algorithm, focusing solely on selecting the most informative features. Common evaluation metrics used in this category include Relief, information gain, and the chi-square test.

In addition, Wrapper methods allow you can incorporate a classification algorithm into the selection process. These methods evaluate subsets of features by utilizing specific algorithm, with the goal of identifying the most effective feature set for that model. Particle swarm optimization, genetic algorithms, and principal component analysis are a few examples.

Embedded approaches combine feature selection directly within the model training process. The technique applied is typically chosen based on the nature of the model itself. Examples of embedded methods include random forest, singular value decomposition, and probabilistic techniques such as naive Bayes.

Regardless of the approach, the feature selection process must conduct a comprehensive investigation of every detail of the feature, evaluating different subsets to determine the most appropriate combinations. [37].

2.8.1 Wrapper Methods

Wrapper methods, first introduced by John in 1994 (as illustrated in Fig.), utilize learning algorithms to assess different feature subsets without being restricted to a specific algorithm. This flexibility allows the use of different types of algorithms including ELM, KNN, SVM, decision tree algorithm and others. The evaluation process for each subset continues until the desired level of precision is reached [38].

Due to their thorough evaluation, wrapper methods tend to produce more effective feature subsets than filter methods. However, this reaches at the cost of raised computational complexity. Despite the higher resource demands, wrapper techniques often deliver superior performance in feature selection tasks. Numerous studies have shown that, in many cases, wrapper methods outperform both filter and embedded approaches in terms of precision and significance.

Recent years have shown that the rapid expansion of data across numerous domains has led to datasets that are not only highly dimensional but also contain intricate and significant hidden patterns. Although wrapper methods often outperform both embedded and filter techniques in terms of feature selection, they still struggle to effectively manage such complex data and uncover the underlying relationships within them [38].

In particular, the formation of subsets from the search space (i.e., the set of all potential feature subsets) is controlled by a certain strategy for search, which will either be exhaustive, population-based, or sequential. Moreover, wrapper approaches typically use the selected machine learning (ML) algorithm's performance measures (for example, accuracy or AUC) for evaluating the generated feature subsets. The iterative nature of these approaches makes them computationally demanding, even for a reasonable number of features. However, their dependency on the ML algorithm ensures the selected features are optimal for the given algorithm. Moreover, using cross-validation (i.e., resampling) within the process of constructing the model ensures that overfitting the ML algorithm is less likely[39] . As below in Figure 2.8.

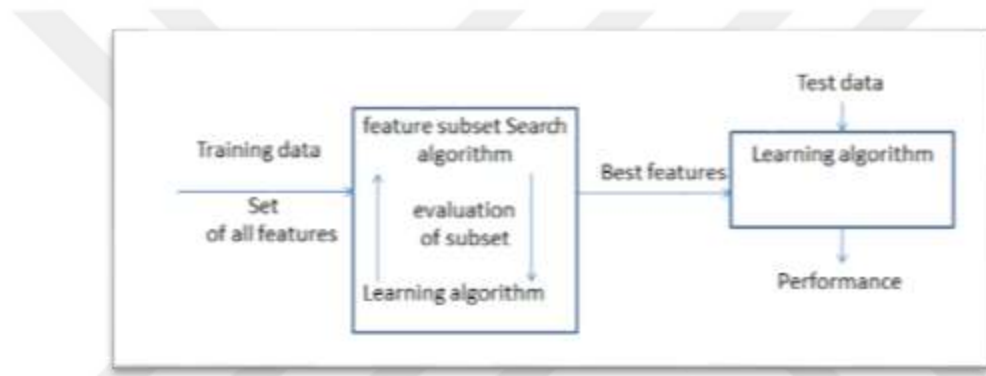


Figure 2.8: The Procedure of Wrapper Selection.

2.8.2 Embedded Methods

This method can combine the benefits of both filtration and packaging techniques. While filter methods are known for their speed, they often lack in efficiency. In contrast, wrapper methods tend to be more accurate but are computationally demanding, This is when you are dealing with a large data set. To address this trade-off, embedded approaches were introduced as a balanced solution that leverages the benefits of both methods [37].

Embedded methods work by incorporating feature selection directly during training of the selected machine learning technique, making them more specific than wrapper methods. Evolutionary computation (EC) techniques have been extensively utilized to feature selection problems and are often categorized under wrapper approaches. EC methods are well-regarded for their global search ability in identifying optimal feature subsets within a search space.

Among these techniques, particle swarm optimization (PSO) is especially notable for ease of implementation and rapid convergence rate [40]. Additionally, within the realm of embedded methods, Grey Wolf Optimizer (GWO) algorithm stands out. This technology is considered a swarm-based technology by nature and belongs to the broader class of metaheuristic algorithms [41].

2.8.2.1 Particle Swarm Optimization (PSO)

Nature has lengthily acted as a powerful source of motivation, showing endless opportunities for learning and innovation. One area that draws heavily from natural phenomena is Swarm Intelligence (SI). The most important thing about artificial intelligence is inspired by the joint behavior seen in social organisms. Among the most widely adopted SI algorithms is Particle Swarm Optimization (PSO). One of the axes that we will work on [42].

PSO is a promising Evolutionary Computation (EC) technique for addressing feature selection challenges. In this context, the task is approached as a binary optimization problem, where each dimension corresponds to a feature. The value assigned to a dimension—either 1 or 0—indicates whether the feature is selected or not. Initially, each feature in the swarm has an equivalent likelihood (50%) of being chosen or banned, meaning the selection probability distribution is symmetric.

As the algorithm progresses, however, features that are part of an optimal subset become more likely to be selected. This shift results in an asymmetric distribution, with a greater chance of selecting the most relevant features [40].

2.8.2.2 Grey Wolf Optimizer (GWO)

This is an optimizer that relies primarily on the hunting intelligence of gray wolves. It is considered from Swarm Intelligence (SI), combining the natural behavior and hunting strategies of gray wolves. It models the leadership hierarchy observed in wolf packs, categorizing wolves into four kinds: “alpha, beta, delta, and omega”. Each wolf is assumed to have excellent knowledge of the prey's location, while all other wolves are classified as omega [43].

The ranks of the Omega wolves in the hierarchy of their pack are established by adjusting in similar patterns that are highly associated with the ranks that are held by the top three ranking

wolves, ranked mainly based upon their intelligence, primarily in circling and trapping prey. A detailed review of the literature that is available tells us that considerable researches have made extensive use of the new and innovative Grey Wolf Optimization algorithm (GWO) that is a highly efficient tool to solve a very wide range of issues, enabling the attainment of the solution at optimal levels. It has been noted that researchers have extensively made use of the use of the GWO algorithm in a wide range of applications in various departments, not limited to, like in medical diagnosis and image segmentations, thus registering its ubiquitous usage. One of the rather complex and time-consuming issues encountered in ad-hoc systems is associated with vehicular network clustering, where efficient structuring and organization are of paramount importance to attain operational efficiency. In this respect, researchers have made successful use of the GWO algorithm to determine optimum clusters within the ad-hoc networks to improve their efficiency. In addition to this, the vital issue of balancing the workload in cloud computing is known to be of utmost necessity and has encouraged researchers to make use of GWO to make optimum and efficient use of the resources. The updation of the positions that are made by the Omega wolves are done continuously and to a large extent are highly dependent upon the ranks held by the top three ranking wolves. This updation of positions is especially made to enhance maximum efficiency of trapping prey in the process of hunting. The top three wolves that hold positions in the top three ranks of the hierarchy system play a fundamental and pivotal role, and the rest of the other ranks are placed in the fourth level of the hierarchy ranks. In this hierarchy-based social ordering system, the updation positions made by the second-level ranks—those pertaining to level four and above, grouped under the category of the Omega wolves—are solely dependent upon the instructions that are offered by the top three ranking wolves. Due to this dependence upon instructions, the Omega wolves tend to unintentionally gather at the same positions and participate collectively in the hunt that is to capture the prey [44].

2.8.3 Filter Methods

Filter approaches use feature ranking as a criterion for feature selection assessment. Features are typically ranked in decreasing order using a range of statistical tests that calculate the extent to which they are relevant to the target class. Features under a certain limit are discarded while features above the limit are selected. Once a subset of features is obtained, the subset is then passed on to the applicable classifier algorithm.

Filter approaches vary from embedded and wrapper feature selection methods since they operate independently of the respective classifier algorithm (See Figure 2.9). The implication of this independence is that filter approaches are not susceptible to the biases of the respective classifier, thereby preventing overfitting [39]. This independence comes, however, at a price of neglecting the relationship with the classifier during feature selection. The resulting set of features is hence more general and not optimized for a single use with a certain classifier. This lack of optimization results in models that display lower predictive prowess compared to models generated using the wrapper or embedded approaches. The key advantage of filter approaches over other forms of feature selection is their tendency to use less computational power, thus being able to effectively operate on datasets with very high dimensionality [45].

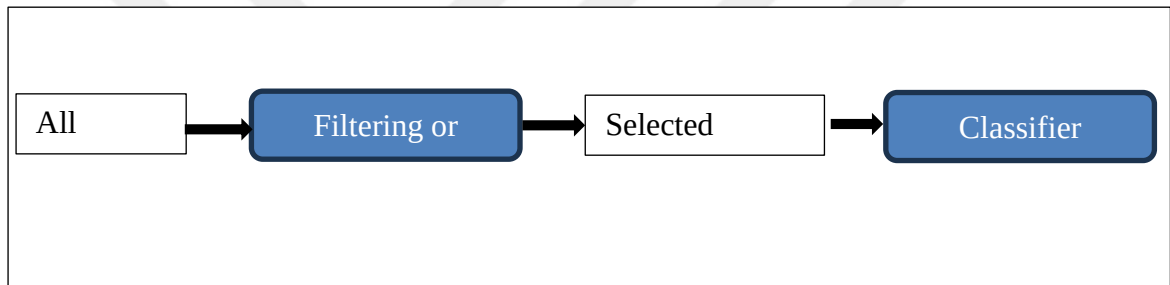


Figure 2.9: Schematic of Filter Method.

2.8.3.1 Correlation coefficient

Correlation is a well-known and highly explored measure of similarity between two random variables. When two random variables have a linear dependence on each other, it therefore follows that the correlation coefficient between them will be either +1 or -1, signifying a perfect positive or perfect negative linear relationship, respectively. When the variables under consideration are uncorrelated, i.e., there is no detectable linear relationship between them, then the correlation coefficient is equal to 0. Note that the correlation coefficient is invariant, i.e., it does not change even if the data is scaled or translated. Consequently, it is totally possible that two different features, each with different variances, could produce the same result when computed using this specific measure of correlation. For the purpose of discussion, let us assume a set of n d -dimensional feature vectors [46].

2.9 RELATED WORK

Discussion of Intrusion Detection Systems with the details previously mentioned in this chapter about the priorities of intrusion detection systems and attacks, in this section, we will explain the applications of intrusion detection systems on a variety of datasets. It will also discuss different algorithms on a dataset IoTID20.

2.9.1 Summary of Comparison of Previous Works Conducted on Different Types of Datasets

Hossein Keipour et al. [47] A dataset attacking Internet of Things applications is presented in multi-hop networks. This dataset is not only bigger but also includes more diverse network topologies compared to earlier datasets. Results obtained show that training various machine learning algorithms on this dataset leads to highly effective intrusion detection systems.

The researcher used deep neural networks (DNNs) with support vector machines using radial basis function (SVM-RBF), achieving good accuracy rates of 95% and 96%, respectively. A Random Forest Classifier (RFC) achieved significantly higher performance, achieving accuracy rates of 98% and 96%. RFC models offer the advantage of low complexity, making them suitable for deployment on low-power, resource-constrained Internet of Things (IoT) networks. In contrast, DNNs and SVM-RBF models typically require integration with cloud computing resources due to their higher computational requirements. All machine learning models in this study were trained using our proposed dataset.

Muhammad Ahmad et al. al[48] introduced a feature clustering approach using the UNSW-NB15 dataset, organizing features into categories such as Flow, Message Queuing Telemetry Transport (MQTT), and Transmission Control Protocol (TCP). Their method addresses key challenges like overfitting, high dimensionality, and data imbalance.

They applied several algorithms, including Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Networks (ANN) to these clusters. All of these are supervised algorithms. With RF, they achieved 98.67% accuracy for binary classification and 97.37% for multi-class tasks.

Further experiments using cluster-based techniques yielded 96.96% accuracy utilizing RF on Flow and MQTT, 91.4% on TCP, and 97.54% using top features selected from both

clusters. Their findings demonstrate that this clustering-based feature selection method not only improves classification accuracy but also reduces training time when compared to other advanced supervised ML models.

Himani Tyagi and Rajendra Kumar [49] proposed an IDS that utilizes a novel feature set derived from the BoT dataset IoT. This system is designed to efficiently and accurately distinguish between benign and not good network traffic. The proposed feature set is both IoT-specific and lightweight, consisting of seven independent features that effectively capture attack behavior.

The study demonstrates the strength of this minimal feature set in detecting a broad range of cyberattacks, including DDoS, DoS, reconnaissance, and information theft. It also validates the practicality of various supervised machine learning algorithms—such as KNN, Logistic Regression (LR), Support Vector Machine, MLP, Decision Tree, and Random Forest—for IoT security.

Performance was evaluated using several metrics, including accuracy, precision, recall, F-Score, and ROC curves. Although both the Decision Tree (DT) and Random Forest classifier achieved identical accuracy rates of 99.9%, Random Forest showed superior performance in terms of training and testing efficiency.

Triya Agustina et al. [50] The researchers used a comparative performance analysis for the feature selection process, using a classification algorithm, Random Forest (RF), on the UNSW-NB15 dataset. The researchers aimed to compare which is better at detecting anomalies in network traffic depend on the two evaluated models: the combination of Random Forest with a filter-dependent selection technique, and the combination of Random Forest with a wrapper-dependent technique.

The results showed that applying Random Forest with a filter achieved an effective accuracy (0.8950), an F1 score (0.8333), a ROC score (0.8928), and a recall value of (0.8347). On the other hand, applying the wrapper approach achieved better results on all metrics, with an effective accuracy: (0.9151), an F1 score: (0.8510), a ROC score: (0.9136), and a recall value (0.8637). These results clearly demonstrate that applying the wrapper technique significantly increases the performance of the random forest algorithm alone, making it a good choice for improving the system's effectiveness in identifying network traffic

anomalies. Its strength lies in its consistent performance and ability to capture subtle data patterns.

Faisal H. et al. [51] A two-stage machine learning framework was proposed to block and detect botnet attacks in IoT. First, ResNet-18 deep learning model was trained to identify scanning behaviors during the early stages of an attack, enabling proactive prevention. Second, another ResNet-18 model was specifically trained to recognize DDoS attacks to facilitate the detection of persistent IoT botnet threats. The results of this dual approach are very promising, achieving accuracy, precision, recall, and F1 score : 98.89%, 99.01% 98.74% and 98.87%, demonstrating its effectiveness in preventing and detecting IoT botnet activity.

Maryam Mahsal Khan and Mohammed Alkhathami [52] investigated the effectiveness of supervised machine learning algorithms in detecting anomalous behavior using a newly released dataset, CCIoT2023. This dataset includes 33 distinct IoT attack categories across 46 features, with varying sample sizes, organized into seven primary groups. The data was preprocessed and class-balanced to ensure unbiased training of several supervised models, including Random Forest, Adaptive Boosting, Logistic Regression, Perceptron, and Deep Neural Networks.

To further improve the model, the study included several preliminary steps: first, extracting positively correlated features, second, reducing dimensionality, third, reducing overfitting, and finally, accelerating training times. The random forest consistently outperformed the others in classifying binary and multi-class attacks, achieving 99.55% accuracy in each of the feature scenarios. Additionally, we found a reduction in computational response time, a significant factor in helping detect threats in record time.

Hosam El-Sofany et al [53]. In this work, the researchers proposed the intrusion detection system (IDS) on BoTNet-L01-IoT data. Their model was tested on a dataset comprising more than 23 different kinds of attacks. To further evaluate the accuracy of the detection application, the work also included the NSL-KDD data, and the system was evaluated in a realistic smart building environment. Many machine learning algorithms: (Random Forest, Naive Bayes, Decision Tree, Backpropagation Neural Network (BPNN), XGBoost, AdaBoost, and a combination of RF-BPNN) were used to identify the best classifiers for

application in an AI-driven response system. This system was designed to recognize attack behaviors and traffic patterns in IoT. The results showed that relying on the machine learning algorithms proposed in this work to detect intrusion achieved high efficiency and accuracy compared to other algorithms, reaching about 99% and was better than previous algorithms.

Table 2.2: Summary of Comparison of Previous Research Conducted on Different Types of Datasets.

No	Author and	Year	Dataset	Algorithm	Accuracy
1	G. Negi, A., and et al	2021.	Our dataset	DNN, SVM-RBF algorithms and RFC	(95%),(96%) respectively. RFC also achieves 98%
2	Muhammad Ahmad, Qaiser Riaz, Muhammad Zeeshan, Hasan Tahir, Syed Ali Haider and Muhammad Safeer Khan,	2021	UNSW-NB15 dataset	RF, SVM, and Neural Networks on the clusters	98.67% binary 97.37% multi-class
3	H. Tyagi and R. Kumar,	2021	BoT-IoT dataset algorithms(DT, RF)	(DT, RF) algorithms	Accuracy (99.9%) and (99.9%)
4	T. Agustina,	2024	UNSW-NB15 dataset	RF with the filter method and RF with the wrapper method	Filter method 0.8950 and Wrapper method 0.9151
5	F. Hussain, S. G. Abbas, I. M. Pires, G. A. Shah, and F. Shahzad,	2021	ResNet-18	deep learning model	98.89%
6	M. M. Khan and M. Alkhathami ,	2024	CCIoT2023	RF, Adaptive Boosting, Logistic Regression, Perceptron, Deep Neural Network	RF an approximate accuracy of 99.55%
7	H. El Sofany, S. A. El Seoud, O. H. Karam, and B. Bouallegue,	2024	BoTNet-IoT-L01 dataset	RF Naive Bayes, DT, Back propagation NN, XGBoost, AdaBoost, Ensembled RF-BPNN	The presented ML-based model is found to have a good accuracy rate of 99%

2.9.2 Summary of Comparison of Previous Research Conducted on Dataset IoTID20

Qaddoura et al. [54] proposed an extensive three-pronged approach that successfully combines the process of clustering with a critical reduction phase, an oversampling phase, and finally, a classification phase utilizing a Single Hidden Layer Feed-Forward Neural Network (SLFN). In one of these approaches, the training data is put through a k-means clustering algorithm that is utilized to form three distinct and separate clusters on the basis of the inherent patterns within the data. After this preliminary clustering procedure, there is the strategic reduction of 10% that is carried out on the formed clusters, and it is followed by the unification of the three obtained reduced clusters into a consolidated form for further analysis. In the subsequent step of the process, the collected and aggregated data are then applied with an oversampling technique known as the SVM-SMOTE oversampling technique, based on a properly determined oversampling ratio of (0.9). Then, the third step is to construct and develop an effective classification model that is precisely tailored to the oversampled data, and this is realized through the application of the SLFN classification method. Upon evaluating the suggested model on the testing data, the results indicated a high accuracy rate of (93.51%) and a G-mean of (94.53%) upon application to the IoTID20 dataset.

The study was done by Alkahtani and Aldhyani [55]. Many deep learning models were applied to classify anomalies using the IoTID20: a Long Short-Term Memory (LSTM), a Convolutional Neural Network (CNN), and CNN-LSTM algorithm. To reduce the complexity of IoTID20, a Particle Swarm Optimization (PSO) algorithm was used to identify the most important features. These selected features were then processed by relevant deep learning models. The experimental results showed that the CNN model achieved an accuracy of (96.60%), the LSTM achieved (99.82%), and the CNN-LSTM hybrid achieved (98.80%).

Song et al. [56] The researchers evaluated the effectiveness of a stacked autoencoder model across different configurations. Their experiments were conducted using three different datasets, NSL-KDD, IoTID20, and N-BaIoT to derive from traditional and IoT data. The results showed that proposed system's application in NIDS intrusion detection is significantly affected by the autoencoder's capacity and the size of the latent space. Among the tested models, configuration built on NSL-KDD dataset achieved the highest F1 score of (0.895).

Reddy et al. [57] demonstrated a fully operational system that was systematically designed with the prime aim of providing suitable facilitation to different devices that are strategically placed at fog nodes. The primary aim of this novel new system is to ensure the automatic and precise detection of malicious activities and hence preventing critical infrastructures from being hacked or targeted by such new threats.

A detailed and extensive analysis was done diligently to determine the optimal parameters that are essentially crucial to ensure effective learning upon the usage of a large size range of machine learning classifiers specifically to the IoTID20 dataset. In binary classification, all of the models were found to have astounding performance with an incredibly high rate of accuracy that exceeded 98% in their capability to identify anomalies and typical actions. In analyzing a particular case that involved multi-classification, a set of algorithms comprising Random Forest (RF), Decision Tree (DT), Bagging, Gradient Boosting (GB), and XGBoost all demonstrated an extremely high level of accuracy with a remarkable result reaching a very impressive rate of 99% in measuring the targeted feature known as Category labeled. Besides that, in another remarkable observation, XGBoost was able to achieve a classification accuracy rate of 82% with respect to the targeted feature known as SubCategory, efficiently and effectively classifying a wide range of violations

.In their research, Alahasan and Sama [58] implemented a two-layer optimisation technique of improve the effectiveness of a streamlined deep autoencoder model. Their approach involved simultaneously selecting the most relevant input features, optimal training samples, and the appropriate number of hidden neurons.

The proposed system achieved an accuracy of 99% when tested on the N-BaIoT intrusion detection dataset using 30 input features. Additionally, when evaluated on the IoTID20 dataset, the model reached 98.12% accuracy using only 25 input features.

Ullah et al. [59] A study was presented by researchers on an Intrusion Detection System (IDS) and implemented on a deep convolutional neural network (DCNN). The architecture includes two convolutional steps, followed by three densely connected steps. The main goal was to achieve high detection performance while reducing computational cost. The model's performance was measured by applying it to the IoTID20 dataset, achieving very good

results, reaching 99.84% accuracy in binary classification, 98.12% in class-level classification, and 77.55% in subclass-level classification.

Alsulami et al. [60] In this study, researchers developed a predictive learning model whose primary goal is to detect and classification network traffic within IoT systems. The model classified traffic into many different types: normal, DoS, scanning, and MITM. To evaluate its performance, five supervised learning were applied: Decision Tree (DT), kNN, SNN, Support Vector Machine (SVM), and Batch Tree (BT). The IoTID20 dataset was used for training and evaluation. Additionally, feature engineering techniques were utilized to enhance the model's accuracy. Experimental results demonstrated strong performance, with classification accuracy ranging between 99.4% and 99.9%.

Krishnan et al. [61] The IoTID20 dataset was preprocessed by removing redundant and highly correlated features, reducing the number of features by half to 33 from 83.

After applying feature selection techniques using forward and backward processing, only eight features were retained. Three classification algorithms were then applied: Support Vector Machine (SVM), Random Forest (RF), and XGBoost. These eight selected features were then applied to the three training classifiers, resulting in excellent classification performance.

Fatani et al. [62]proposed a feature engineering strategy for intrusion detection systems (IDS) that incorporates swarm intelligence (SI) techniques. To evaluate the effectiveness of their approach, they used four widely recognized public datasets: KDD99, CIC2017, BoT-IoT, and NSL-KDD.

Table 2.3: Summary of Comparison of Previous Research Conducted on Dataset IoTID20.

No	Study	Years	Technique	Accuracy
1	Qaddoura et al. [54]	2021	SLFNand SVM-SMOTE	93.51%
2	Alkahtani, Aldhyeni[55]	2021	CNN, LSTM and CNN-LSTM	99.82%, 96.60%, And 98.80%.
3	Song et al [56]	2021	NIDS	95.2%
4	D.K.Reddy et al [57]	2021	RF, DT, Bagging, GB, and XGBoost	99%
5	H. Samma et al [58]	2022	deep autoencoder with kNN	98.12%
6	S.Ullah et al., [59]	2022	CNN	binary, category, sub-category = 99.84% ,98.12%, and 77.55%
7	A.AAlsulami et al[60]	2022	DT,ANN, kNN and SVM	99.4 to 99.9%
8	A. Neyaz, et al [61]	2021	SVC, RF, and XGBoost	N/A
9	A. Daho et al [62]	2022	swarm intelligence with CNN	99%

2.10 SUMMARY

The relevant sections of this research, which explain the IoT intrusion detection system, were presented. Similar work on intrusion detection systems was presented, demonstrating that most current IDS applications have not yet limited the ability of attackers to intercept IoT traffic. Furthermore, most algorithms are slow in dealing with large numbers of attackers. On the other hand, the decision tree was demonstrated as a fast classification algorithm with several positive aspects. To further improve this algorithm, this work combined it with the coronavirus algorithm to create a hybrid IoT intrusion detection technique.

3. RESEARCH METHODOLOGY

This work aims to develop a hybrid algorithm for intrusion detection using decision tree and coronavirus algorithm to help identify and classify attacks. Therefore, the work begins with data processing and then identifying features through multiple filtering and embedding algorithms. The decision tree algorithm is a fast classification algorithm. Delays in detecting breaches and classifying attack types are common. In this study, achieving these goals requires a robust methodology capable of achieving clear understanding, which is the focus of this chapter. In order to define the objectives of this research, this chapter explains the design research methodology (DRM). The chapter begins with a comprehensive explanation of the methodology, as will be discussed in Section 3.1. Section 3.2 describes all the steps of the study's methodology, and Section 3.3 describes the applications and treatments. A summary is provided at the end of this chapter.

3.1 RESEARCH APPROACH

This work aims to create a system to identify intrusions on a smart home dataset, whether they are anomalous or normal. After determining that an attack is anomalous, it is classified. To date, various research have been conducted on breach detection systems. The IoTID-20 dataset contains significant traffic, with an estimated 625,783 Internet of Things (IoT) traffic, including 83 features. The damage increases if attacks are not quickly classified. In this research, we relied on supervised classification after multiple feature selection and used a hybrid decision tree classification to classify attacks. This research is designed based on the definitions provided by developed by Blessing and Adeeb in [63] and [64]. To achieve the main objective of this research, a basic explanation of the problem and how an IoT attack detection system works is crucial. This helps identify the attack and classify the type of IoT breach that can mitigate the aforementioned problems. Figure 3.1 below illustrates the rights management framework. The figure illustrates the relationship between the work phases, the methods to be followed in the study, the main procedures to be performed, and the results. The weak arrows indicate the process flow, while the strong, bold arrows indicate the main methods and results.

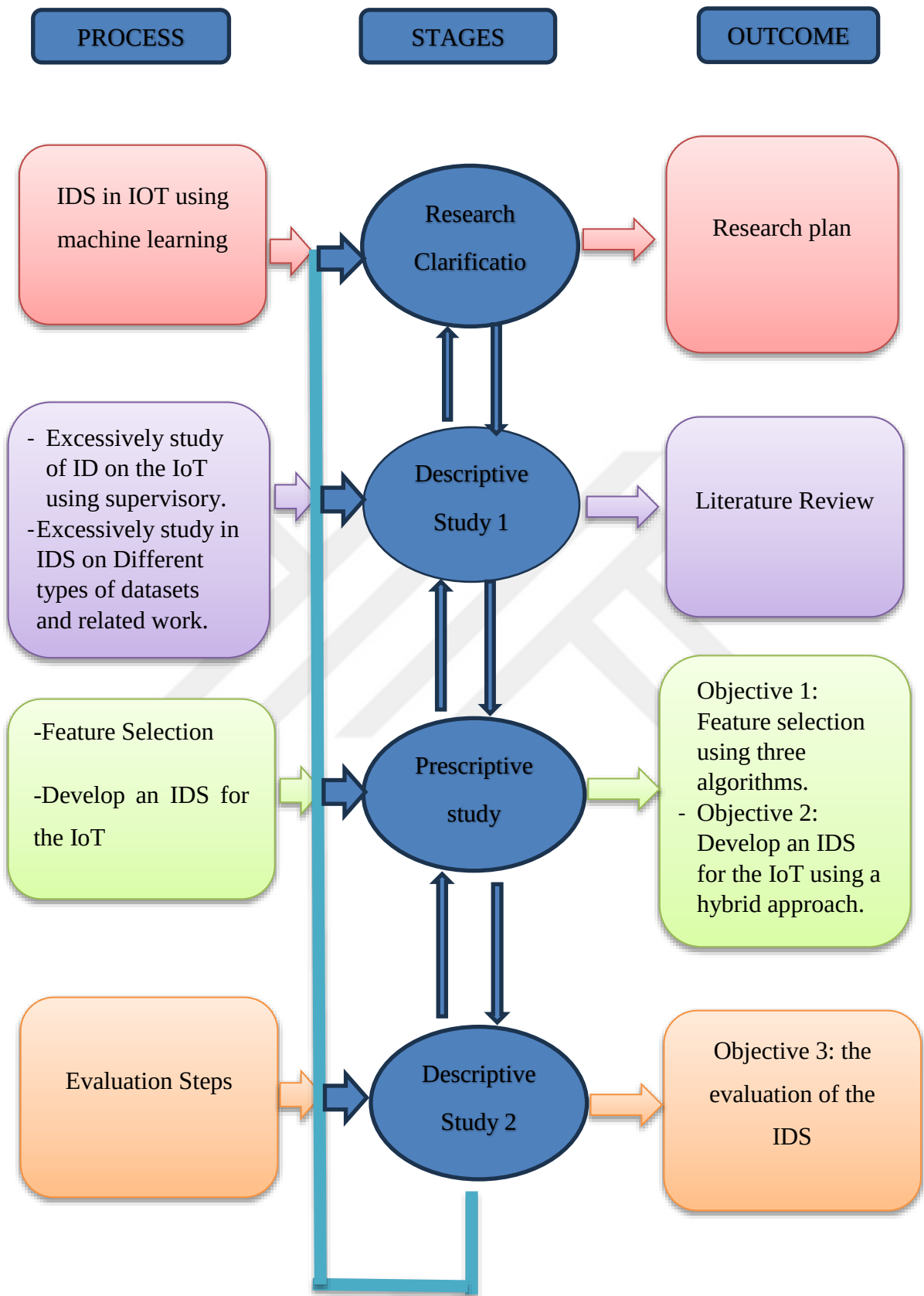


Figure 3.1: Research Approach.

3.2 STAGES OF DESIGN RESEARCH METHODOLOGY

The four stages of digital risk management are covered in the following subsections. We begin by explaining the research (RC) in Section 1, 3.2.1, then move on to Descriptive Study 1 (DS1) in Section 2, 3.2.2. In Section 3, we will explain Descriptive Study (PS) 3.2.3. In Section 4, the final section of Risk Management, 3.2.4, we will explain Descriptive Study 2 (DS2).

3.2.1 Research Clarification

The RC stage is the initial DRM step. At this point, literature study reports and previously assessed relevance can be used to define the issue and establish the goals. This section offers a strict and necessary foundation for the subsequent research procedure. As seen in Figure 3.2, RC involves six processes.

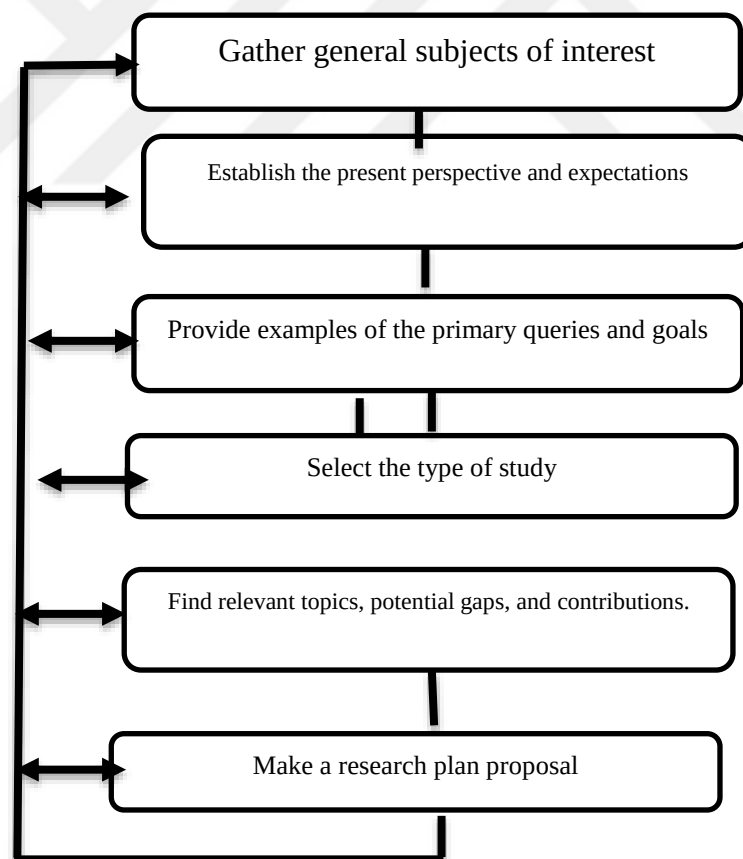


Figure 3.2: The Research Clarification Steps.

The following are the results of the RC stage:

- a. The purpose and emphasis of the research on intrusion detection systems.
- b. The study challenge and work questions concerning the Intrusion Detection System with Supervised Learning research aims.
- c. The scope, nature, and methodologies of the IDS research in the IoT
- d. The anticipated deliverables and research contribution for IDS in IoT.

3.2.2 Descriptive Study1

The overall research plan, which will be achieved at completion of the RC stage, will serve as a strong foundation for the DS-I, the second stage of the research. Critical comprehension of the existing state of affairs would have been acquired by the end of the DS-I stage. This step also includes experimental experiments and a critical evaluation of the study domain's literature review. As this stage progresses, several specifics on the primary functions of the intrusion detection system and supervised learning will be emphasized, taking into account its important problems and potential fixes based on earlier work in the mode division multiplexing method. This stage's steps are displayed in Figure 3.3 below.

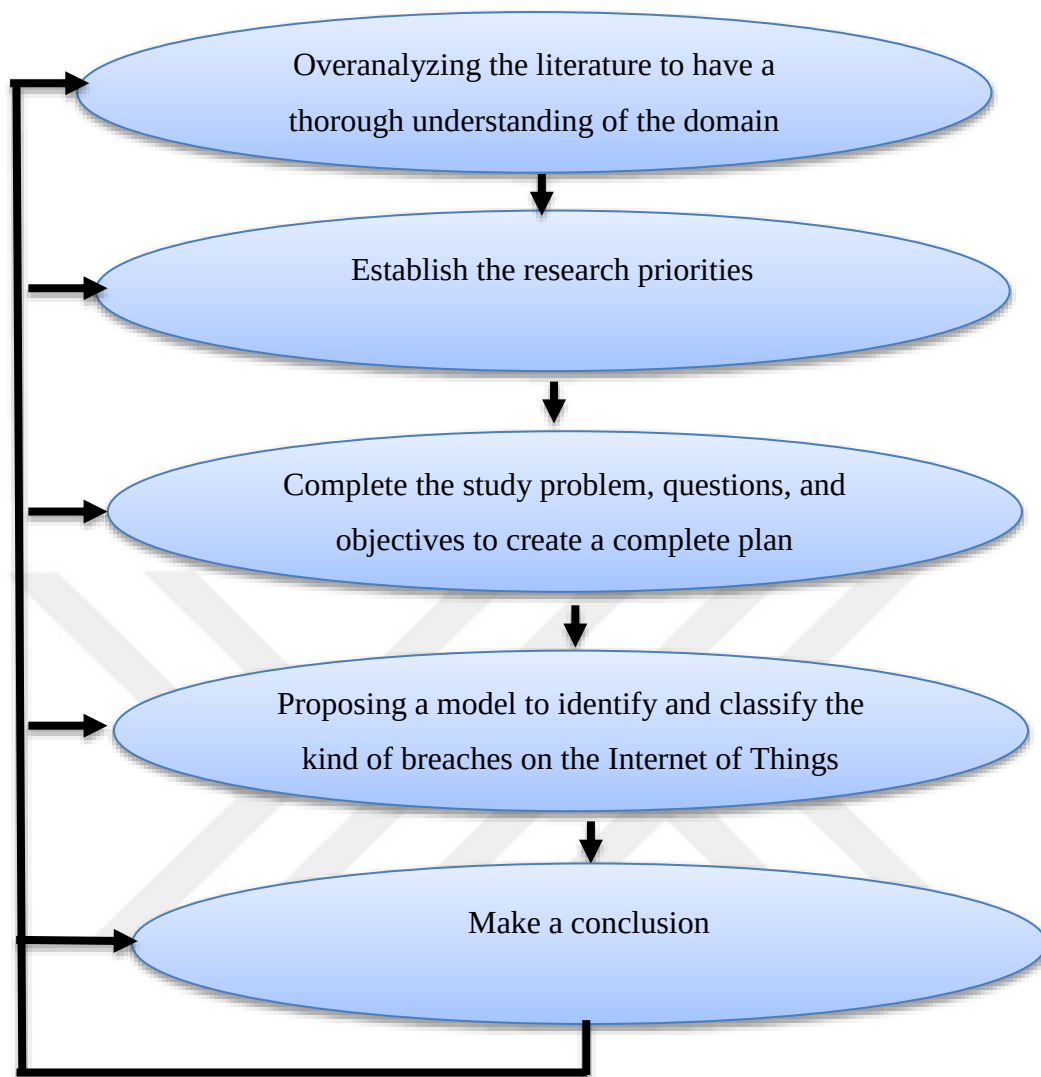


Figure 3.3: Key Procedures for DS1.

3.2.3 Prescriptive Study (PS)

The primary DRM stage outlines the implementation procedures required for creating and building intrusion Detection and Attack Identification System for the Internet of Things. Figure 3.4 below displays a block diagram that outlines the primary PS operations in accordance with this study technique.

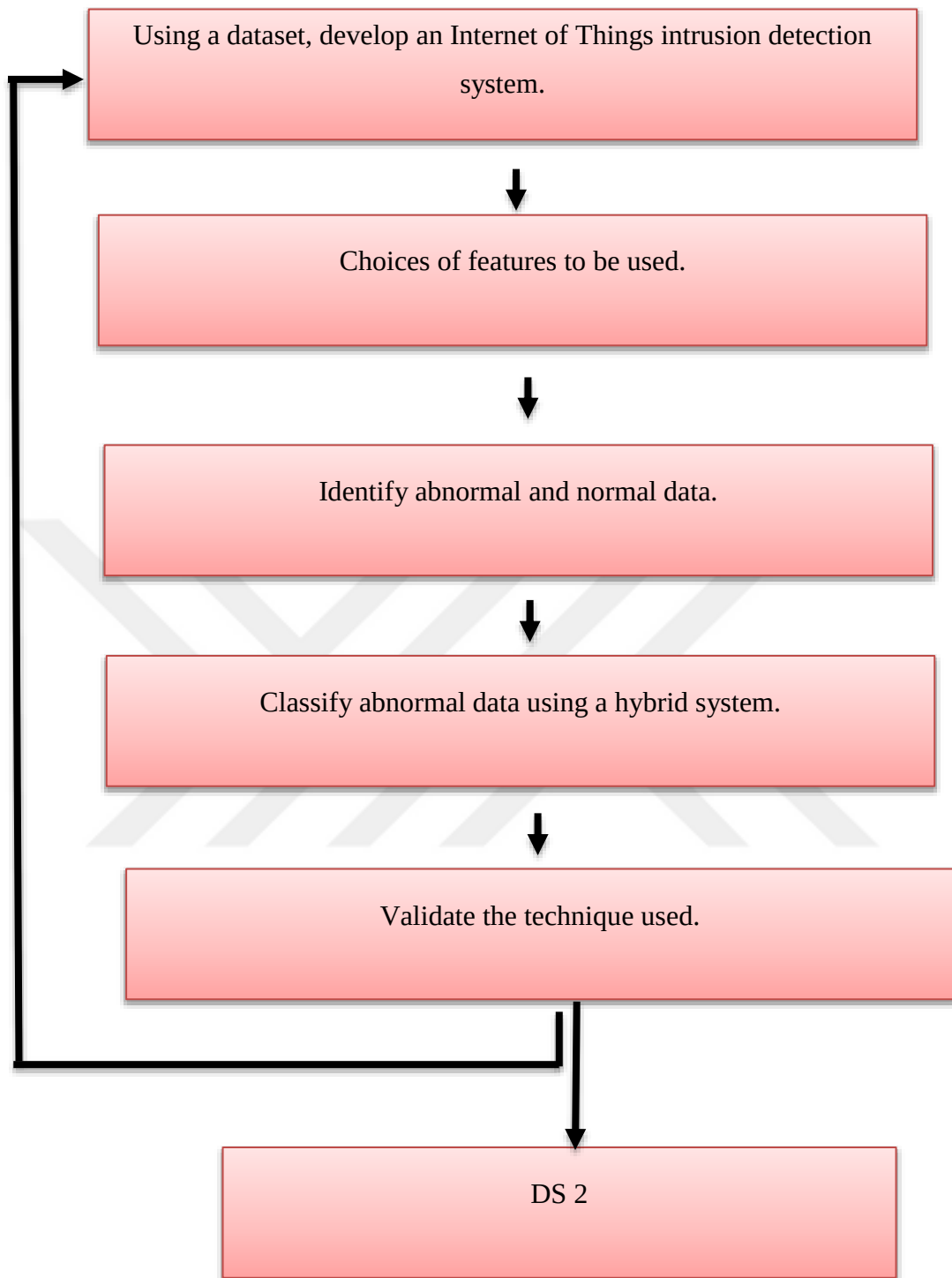


Figure 3.4: PS Steps.

3.2.3.1 Experimental design

Step one is to identify the datasets we need that has a big volume of traffic. After collecting the Internet of Things traffic, these packets must be analyzed to identify unnecessary data, conflicting data, and noisy network traffic. This is done by applying the necessary processing to these packets and manually excluding them. We then perform feature selection on the remaining traffic to select usable features using three selection algorithms using the Python programming language. The second step is to further refine the proposed methodology by entering the remaining traffic into the Coronavirus Identification (CHIO) algorithm. With all these selected features, we classify the remaining traffic using a decision tree algorithm, also using Python programming.

3.2.3.2 Verification and validation

The fourth and last stage in this phase is Verification and Validation (V&V). Making sure the created model can be changed as needed and with the necessary accuracy is known as model validation. Applying the suggested model to many popular classification methods and contrasting them with the suggested system in this study will serve as a validation study of the intrusion detection system. Various scenarios based on various independent variables will be carried out throughout the validation process. Using the same IoTID-20 dataset, the procedure will be repeated on the algorithms previously described in Chapter 2 to confirm the work.

3.2.4 Descriptive Study 2 (DS2)

The DS-II model demonstrates the intrusion detection system. Completing this phase achieves the third objective of this research. According to [63], performance evaluation is a vital stage in evaluating any research. Figure 3.5 below illustrates the main steps of this phase.

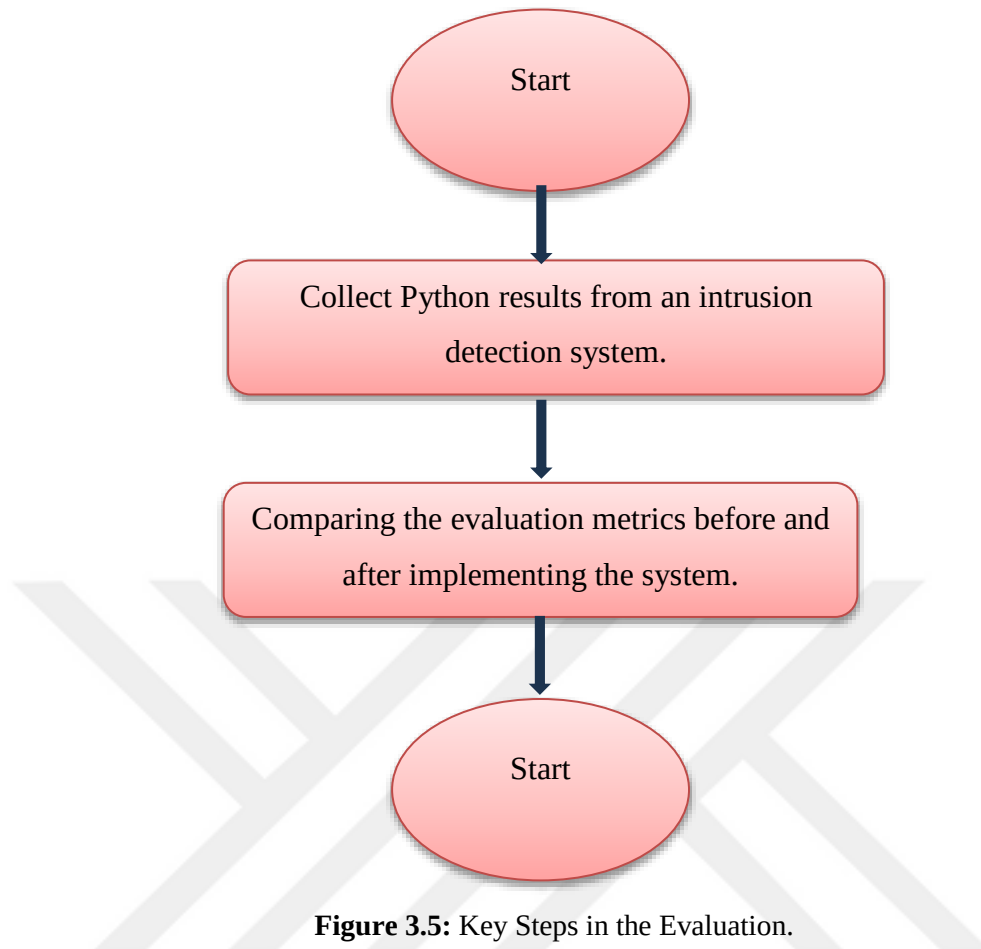


Figure 3.5: Key Steps in the Evaluation.

3.3 TECHNIQUES FOR IMPLEMENTATION AND EVALUATION

Implementation and performance evaluation are regarded as crucial components of communication systems. Three primary methods can be used separately or in combination to complete this phase. Analytical modeling, measurement, and simulation are the three methods[65].

3.3.1 Options for Implementation and Evaluation

For the execution and assessment of research, three primary approaches are frequently employed. Using mathematical equations to depict the system is known as analytical modeling. Analytical modeling can be done for basic systems, but it is not advised for complex systems since it requires more labor to simplify and establish assumptions [66], which alters how the real system is represented. However, measuring is an additional strategy that may be applied to both implementation and assessment. There are two categories of measurements: the first involves working on actual living systems, and the

second involves utilizing a specifically created laboratory setting to carry out the necessary studies. Although the second approach is more popular, it is not appropriate for complex and large-scale systems. Simulation is used for the third implementation and assessment. Simulations rely on specialized software that can depict and explain events as though they were happening in a real-world setting. Many academics now use this as their primary method for implementation and assessment because to the flexibility provided by simulation software and the latest credibility simulations[67].

3.3.2 A Proposed Technique for Designing an Internet of Things Intrusion Detection System Using the IoTID20 Dataset

Many researchers and designers use intrusion detection systems across multiple datasets in their work for educational, commercial, and industrial purposes. These systems are essential to the rapidly evolving Internet traffic, paving the way for designers and researchers to detect Internet attacks in real time. As mentioned in Chapter 2, several types of datasets are used, as are several programming languages. Here, we will use Python programming language and the IoTID20 dataset. To design an intrusion detection system using the IoTID20 dataset, we need a powerful programming language that can handle large volumes of traffic, detect Internet attacks, and determine the type of attack based on the data provided. Therefore, Python was used in this work. Python is one of the three multiple popular and advanced programming languages on earth, and it has applications in machine learning, deep learning, and many other applications[68].

3.3.3 Evaluation Metrics

The system will be evaluated based on the good results it has demonstrated compared to other systems and the speed of data classification using a decision tree algorithm, which is fast at classifying data. We will examine this in Chapter 4 of this research.

3.4 SUMMARY

To ensure the study's objectives are achieved and implemented as intended, this section presents a comprehensive explanation of the research procedures used to develop a digital rights management (DRM) approach. This work focuses on developing a multi-stage hybrid attack detection system, along with intrusion detection, to detect data traffic between Internet of Things applications, based on the four stages of DRM.



4. EXPERIMENTS, RESULTS AND EVALUATION

The necessary guidelines for achieving the main objectives of this work were established in Chapter Three. This chapter proposes a system for detecting intrusions and identifying attack types on traffic between IoT applications using a multi-feature selection system and a hybrid classification system based on the IoTID20 dataset. This system classifies anomalous and good traffic across the Internet and identifies its type.

As mentioned in Chapter Two, many systems have been used to date, but the results have not been sufficient to identify intrusions in the Internet of Things, especially given the evolution of IoT traffic. This chapter is divided into several sections, as follows: Section 4.1 provides a general explanation of the intrusion detection system. Section 4.2 then presents the first step, feature selection, while Section 4.3 explains how to implement the hybrid algorithm. Section 4.4 presents the results and evaluation of the developed algorithms. Section 4.5 summarizes the findings of this chapter.

4.1 OUTLINES THE INTRUSION DETECTION SYSTEM IN GENERAL

As shown in Figure 4.1 , designing and implementing a new IDS system involves four steps: Collecting Internet of Things application traffic, data processing, feature selection (feature engineering), and classification.

That a big amount of IoT network traffic must be collected using any sniffing tool. After collecting the IoT network traffic, these packets must be analyzed to identify unnecessary data, conflict data, and noisy network traffic to remove noise, apply necessary processing to these packets, or exclude them from the next step. Therefore, we implement multiple feature selection techniques on network traffic data to select salient features that can be utilized as discriminating parameters between different types of network traffic data, a process called feature engineering or feature selection. Finally, a classification model is built by organizing the collected network traffic data into three parts: testing, validation, and training. This model uses the training and validation data to select the best hyperparameter values, achieving higher efficiency than other intrusion detection systems during the testing phase.

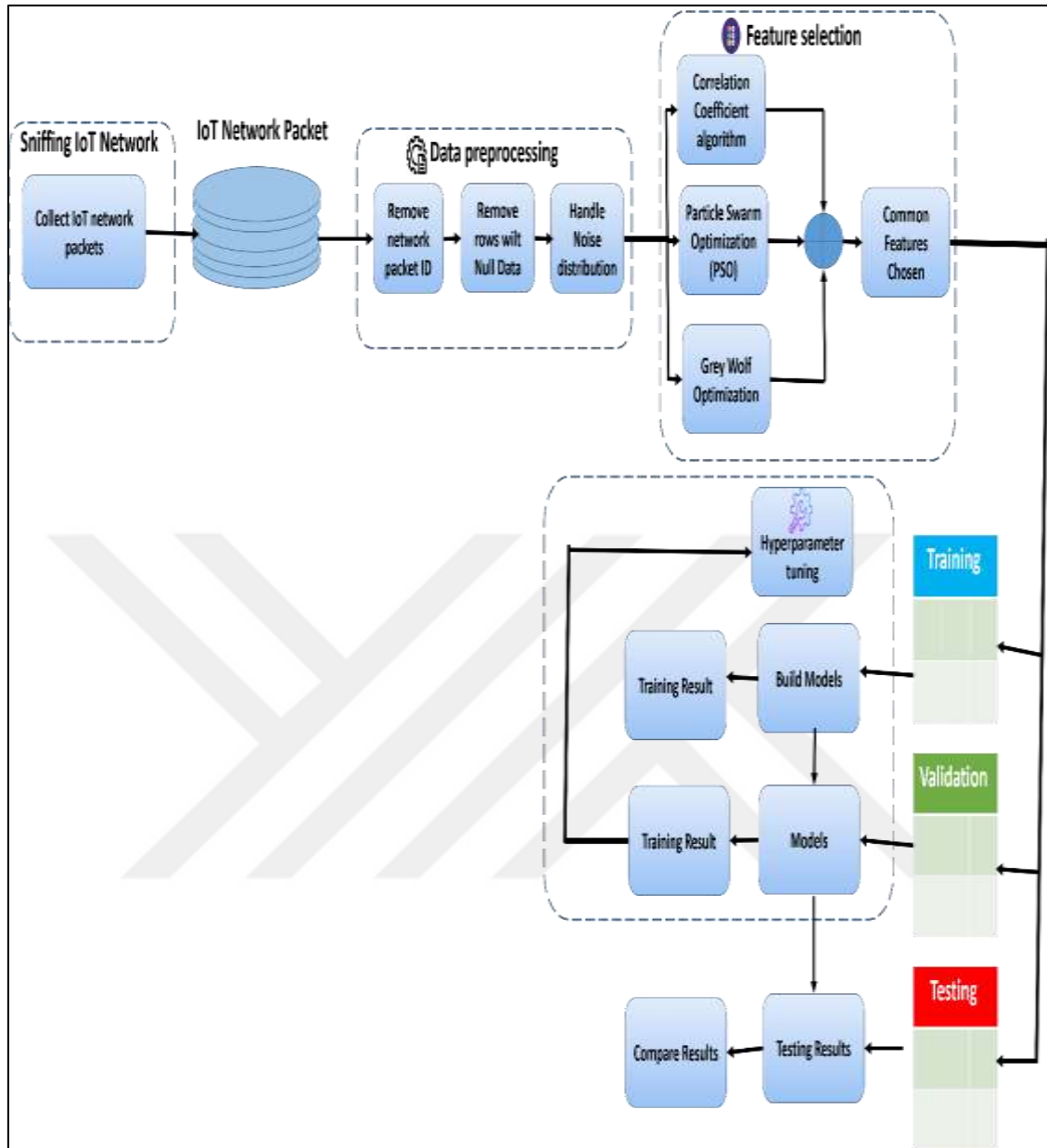


Figure 4.1: General Architecture of the IDS.

In this work, the general architecture is followed with the aim of improving system efficiency and response time. Figure 4.2 shows the architecture of the proposed system.

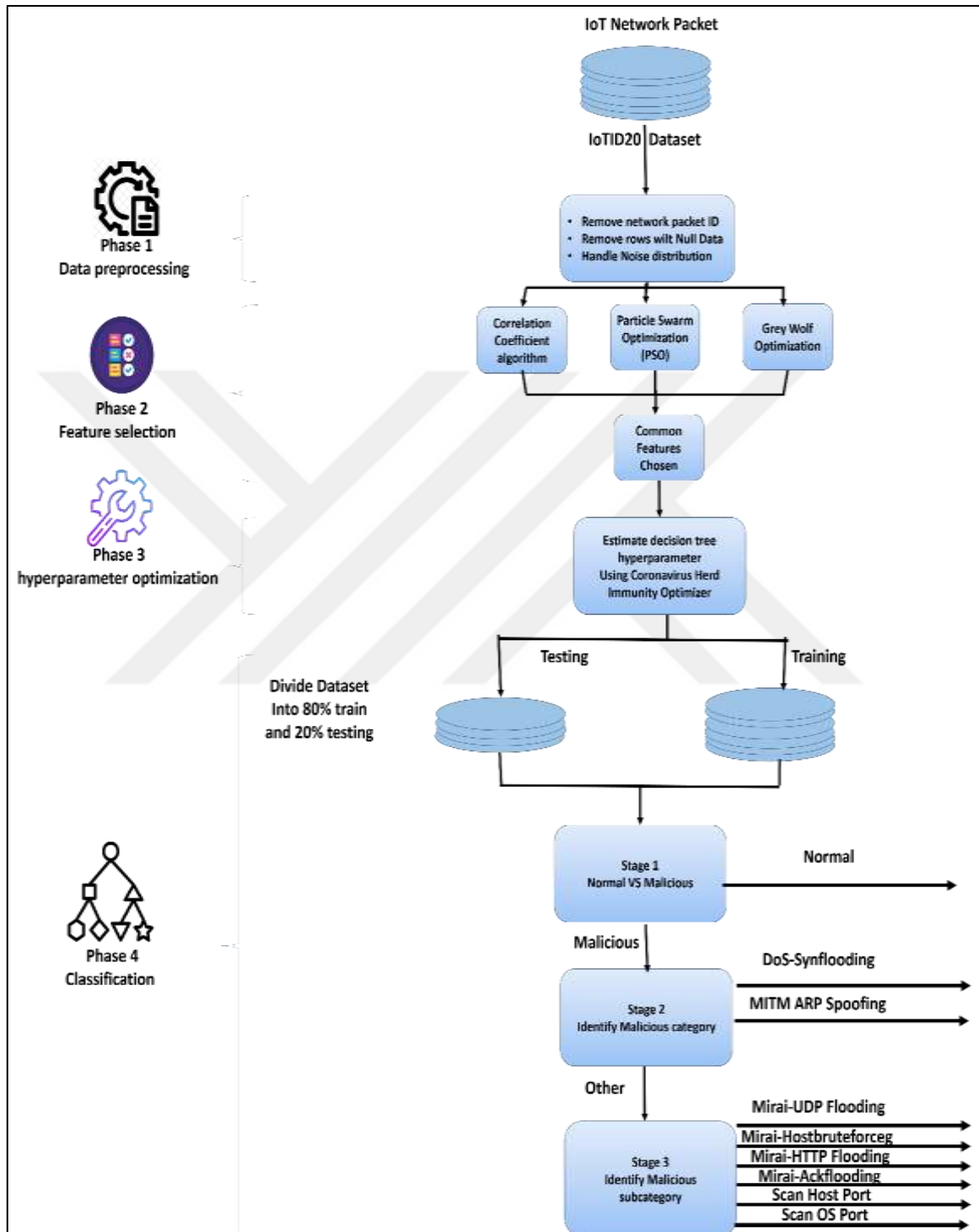


Figure 4.2: Proposed System Architecture.

Figure 4.2 above that the four stages of the general architecture are modified to achieve good accuracy. The proposed system consists of four steps: data processing, feature selection, hyperparameter optimization, and classification (Figure 18). Therefore, the sniffing stage is not performed because it consumes more time to collect network traffic, and many datasets already contain many network packets with different types of IoT network traffic. The proposed system uses the IoTID20 Dataset to classify the IoT network traffic type. The goal of data preprocessing removes noise and empty data and optimizes the dataset. This process leads to feature selection utilizing optimization techniques such as correlation coefficient evaluation, PSO, and GWO. Furthermore, the optimization process continues by applying a systematic hyperparameter optimization methodology, harnessing the COVID-19 Herd Immunity Optimizer (CHIO). The classification stage uses the decision tree method to classify each network traffic using these chosen features.

4.2 DATA PROCESSING AND FEATURE SELECTION

The steps listed in the flow chart below (Figure 4.3) are followed in the implementation of this section:

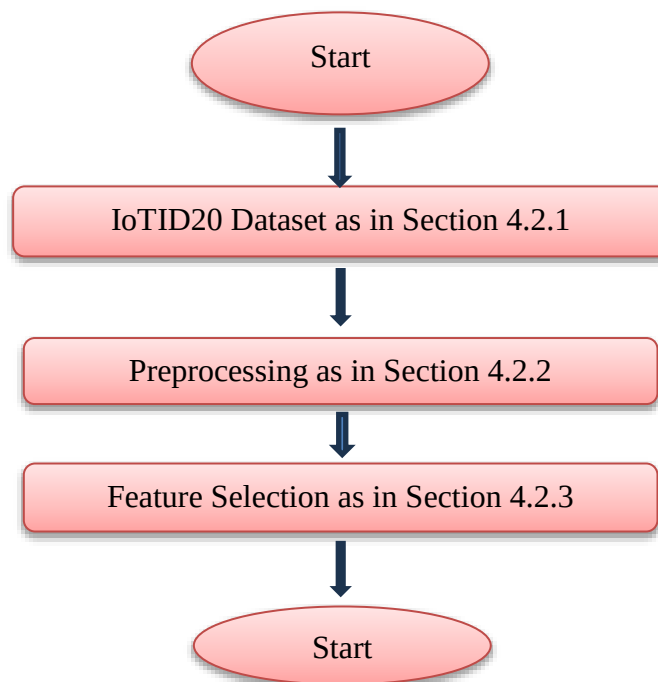


Figure 4.3: Data Processing and Feature Selection Steps.

4.2.1 IoTID20 Dataset

This dataset was gathered within a standard smart home setting. 86 characteristics and 625783 samples make up the IoTID-20.[69] The dataset consists of eight attacks and nine categories. A thorough overview of the Binary, category, and subcategory of the IoTID20 Dataset is provided in Figure 4.4: Binary, category, and subcategory of IoTID20 dataset.

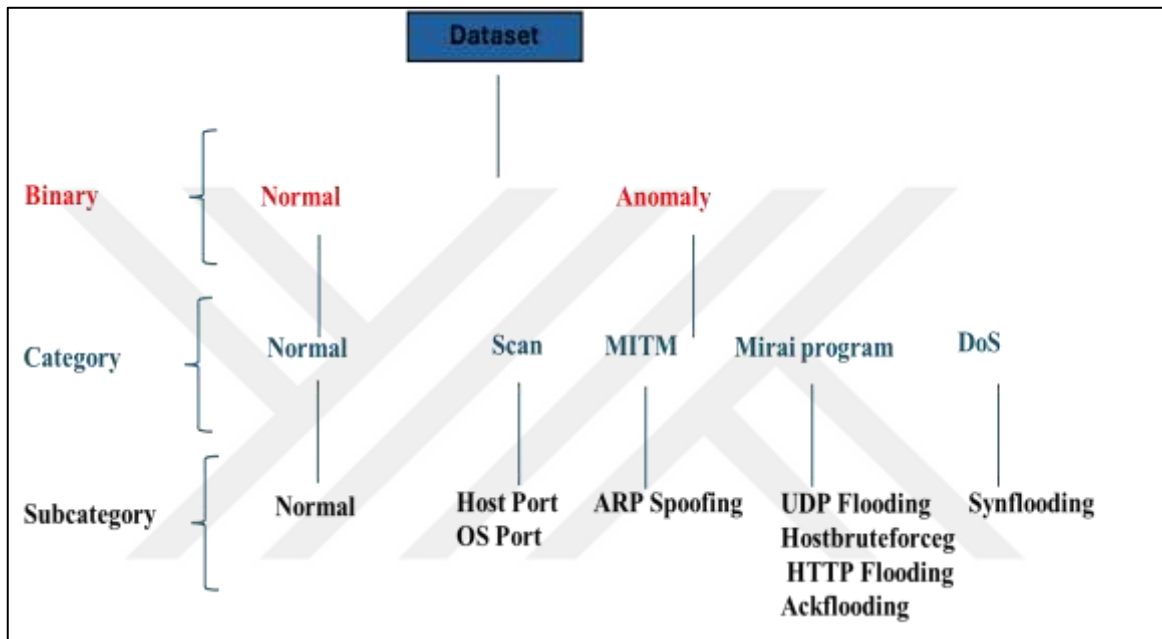


Figure 4.4: Binary, Category, and Subcategory of IoTID20 Dataset.

The analysis data used for collection included a wide variety of intrusion types, including notable examples like Denial of Service (DoS) attacks, Man-In-The-Middle (MITM) attacks, Distributed Denial of Service (DDoS) attacks, and also the method known as active scanning. These types of intrusions were identified and closely monitored, as they are considered to be the most common threats in the data collection. Synchronized (SYN) packets, which are a critical part of network communications, are commonly employed to initiate TCP connections between various communicating entities. This behavior thus reserves precious resources at both ends of the connection, including not just ports but also buffers needed for the successful execution of the transmission of data. It should be noted that DoS-synflooding attacks have high potential to target and impact the accessibility of servers and/or computers belonging to the victim and thereby undermine their effectiveness

in operating. There are several methods for simulating attacks. Among these methods that have been utilized to simulate distributed denial of service attacks are techniques and protocols, hypertext, and datagrams, which have shown positive results in distributed denial of service attacks, especially in the Internet of Things. In another work, a brute force attack [70] was used to decrypt data, demonstrating a significant breach of data confidentiality. Another work, a man-in-the-middle (MITM) attack, was implemented by interfering with the Address Resolution Protocol (ARP) table, which maps the Media Access Control (MAC) address to the Internet Protocol (IP) address of a router that occupies a central position in network communications. The result of this interference is that an unauthorized individual successfully masquerades as the network router itself and is in a position to interrupt communication between various entities on the network. The aim of this adversarial attack is to eavesdrop, seize, or even alter data as it is transmitted over the network, thereby violating the integrity and confidentiality of the data.

4.2.2 Data Processing

A systematic approach is used to perform some preprocessing in network traffic collected in IoTID-20 data that will improve the quality of the dataset. Before a preprocessing step is performed, each network traffic in the dataset contains 86 features. The preprocessing stage passes through two main steps. First, the “Flow_ID,” “Src_IP,” “Dst_IP,” “Timestamp” features are excluded from the dataset because they are used to identify each network traffic. Therefore, we perform a statistical analysis that shows the distribution of features to identify the attributes that have noisy distribution and null values. This analysis shows that eight attributes: “Fwd_PSH_Flags”, “Fwd_URG_Flags”, “Fwd_Byts/b_Avg”, “Fwd_Pkts/b_Avg”, “Fwd_Blks/b_Avg”, “Bwd_Byts/b_Avg”, “Bwd_Pkts/b_Avg”, “Bwd_Blks/b_Avg”, “Init_Fwd_Win_Byts”, and “Fwd_Seg_Size_Min”, have noisy distributions and are not useful for discriminating between anomaly and normal packets. Therefore, these eight attributes were excluded and removed from the dataset. With a standard deviation of zero, these characteristics showed consistently low volatility, indicating that they maintained a constant value across the entire dataset and lacked discriminatory power. In addition, the mean, minimum, and maximum Further evidence of these traits' lack of fluctuation came from the fact that their values were constant. The statistical distribution of these eight traits is displayed in Table 6. Additionally, the

statistical study reveals two characteristics: “Flow_Byts/s” and “Flow_Pkts/s” appear as they have null values, so we removed them from the dataset. By concentrating on the characteristics that can clearly reveal underlying patterns and behaviors, the aim of the data processing phase is to simplify the dataset and, in turn, improve the efficacy of further analysis and modeling attempts.

Figure 4.5 shows the distribution of the features that do not have any noisy distribution and will appear as good features that may be used to discriminate between anomaly types in the next steps of the proposed system.

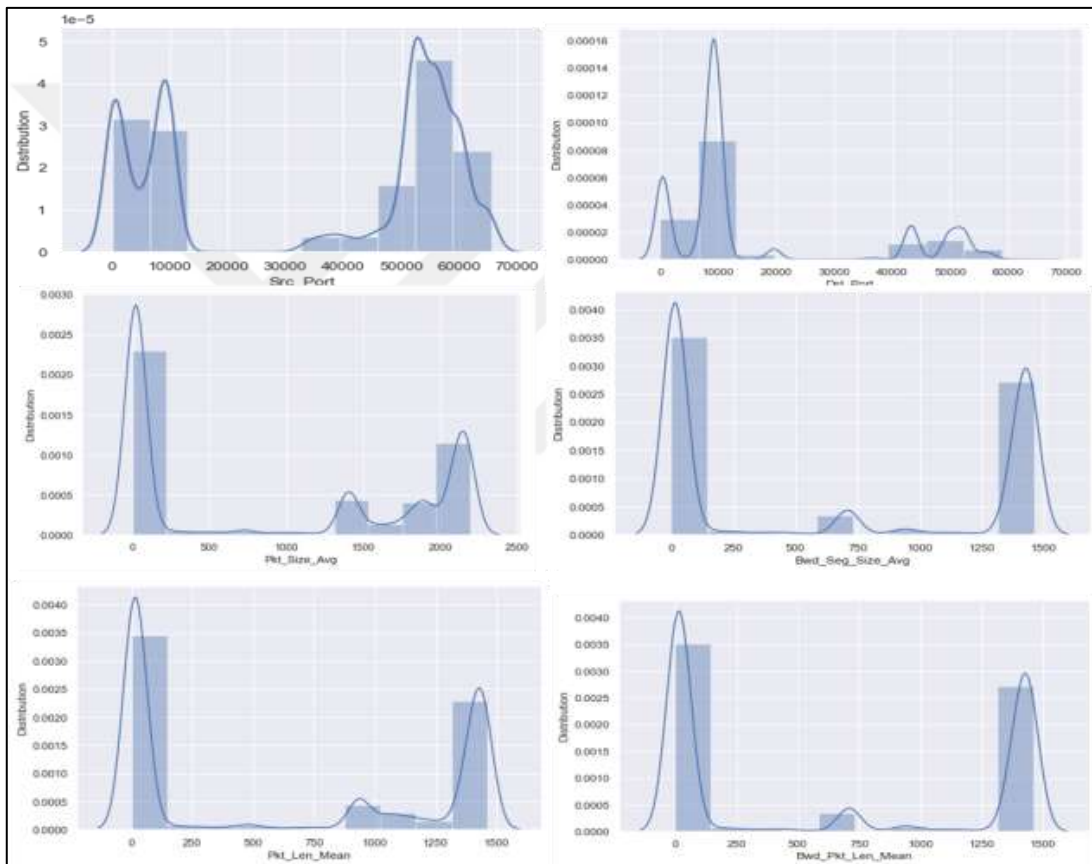


Figure 4.5: Best Feature Distribution.

In the data preprocessing step, 16 features were removed. These included four identification features, ten features with noisy distributions, and two features with null values. While the result shows, the dataset has 60 features after the preprocessing step.

Accurate categorization depends on the traits that remain after preprocessing. Heatmaps showing the connections between the top 10 features and the three target classes are shown

in Figures 4.6, 4.7, and 4.8. These graphic aids provide insightful information about the relationships between target classes and attributes. The significance of these variables in differentiating across groups is further shown by the heatmap patterns, which suggest possible strong correlations.



Figure 4.6 : Heatmap of the Label Target's Ten Most Label.

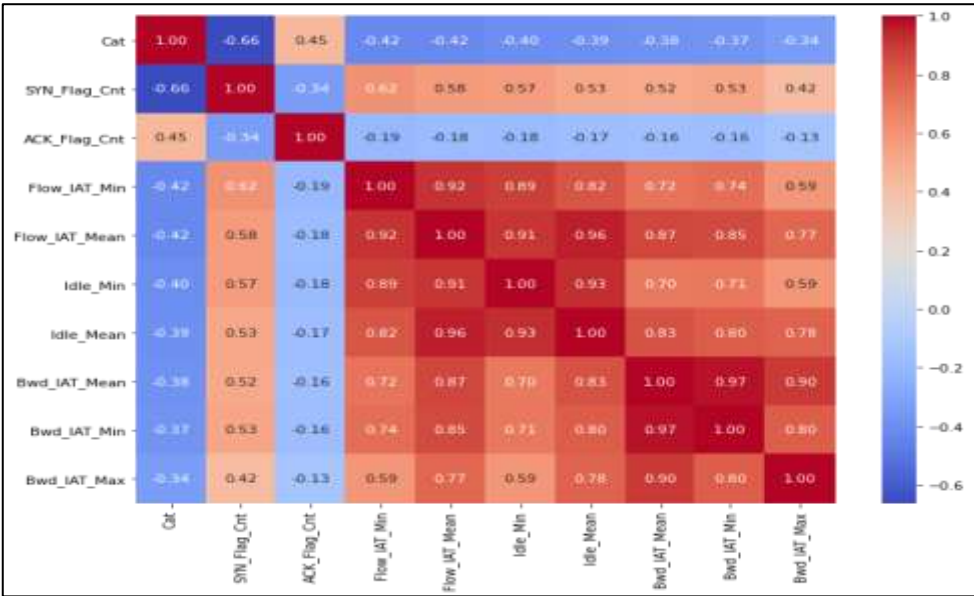


Figure 4.7: Heatmap Shows the Target Category's Ten Most Crucial Attributes.



Figure 4.8: Heatmap Displaying the Chosen Subcategory's Top ten Key Attributes.

4.2.3 Feature Selection

Often relying on data analysis and machine learning, feature selection is chosen because it helps find and extract characteristics that might be discriminative for classification labels. Additionally, characteristics that might add noise, repetition, or insignificance to categorization labels are removed. Feature extraction is classified into three types: embedded methods [22], wrapper methods [23], and filter methods [24]. The proposed system uses three feature extraction algorithms or techniques: the correlation coefficient, the PSO, and the GWO are classified as embedded methods, while the correlation coefficient technique belongs to the filtering method category.

4.2.3.1 Correlation Coefficient

Because the Correlation Coefficient algorithm is a filter method, It assesses features' significance using information-theoretic metrics or statistical characteristics. A technique for assessing the strength and direction of the linear relationship between characteristics and the target variable is the correlation coefficient. It assigns a feature value between -1 and +1 to each feature after evaluating it. There is a significant positive correlation between the feature and the target variable when the value is close to +1, and a strong negative correlation when

the value is close to 1. When the linear relationship between variables is weak or not statistically significant, the correlation value is close to zero. To calculate the correlation coefficient, we use (1).

$$r = \frac{n(\sum XY - \sum X \sum Y)}{\sqrt{n \sum X^2 - (\sum X)^2} \sqrt{n \sum Y^2 - (\sum Y)^2}} \quad (1) \quad (4.1)$$

While n indicates is the number of instances of the variables X and Y stand for the two variables. For each class label in the dataset, the correlation coefficient identifies the top 35 features with a higher correlation between these features and the target class.

4.2.3.2 Particle Swarm Optimization

This algorithm is considered is embedded method that depends on a heuristic approach that simulates the behavior of particles in flocks to find the best solution. Structure of the PSO Algorithm. A swarm of particles that updates from iteration to iteration is utilized by PSO to carry out searching. Each particle travels toward the swarm's global best ($gbest$) and previously best ($pbest$) positions in order to get the best answer. One has

$$pbest(i, t) = \arg \min_{k=1, \dots, t} [f(P_i(k))], \quad i \in \{1, 2, \dots, N_p\}, \quad (4.2)$$

$$gbest(t) = \arg \min_{\substack{i=1, \dots, N_p \\ k=1, \dots, t}} [f(P_i(k))], \quad (1) \quad (4.3)$$

The terms refer to each unknown in the rates above: NP the "total quantity". i denotes the "particle index", t the "current iteration". f the "fitness function", and P the "position". The particle's velocity V and position P are updated by the following equations:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pbest(i, t) - P_i(t)) \quad (2)$$

$$+ c_2 r_2 (gbest(t) - P_i(t)),$$

$$P_i(t+1) = P_i(t) + V_i(t+1), \quad (3)$$

(4.4)

The terms refer to each unknown in the rates above: V the “velocity”, ω is the “inertia weight” utilized to balance the global exploration and local exploitation, $r1$ and $r2$ are uniformly distributed random variables within range $[0, 1]$, and $c1$ and $c2$ are positive constant parameters called “acceleration coefficients.”

Setting an upper bound for the velocity parameter is a typical practice [71]. A technique called "velocity clamping" was employed to restrict the number of particles that might leave the search area. Another approach is the "constriction coefficient" technique, which was put out after a theoretical study of swarm dynamics. This strategy, the velocities are also constrained.

The prior velocity, or "inertia," is represented by the first portion of formula (2) and gives particles the momentum they need to move freely throughout the search space. Each particle's own thought process is represented by the second section, which is referred to as the "cognitive" component. It pushes the particles in the direction of their best locations thus far. Third element, referred known as the "cooperation" component, illustrates how the particles work together to identify the best possible global solution.

Pseudocode for the price function. Let $f: \mathbb{R}^N \rightarrow \mathbb{R}$ be the cost function that we want to minimize. The function takes a candidate solution to a vector NP of real numbers, and the output of this function is a real number that denotes the value of the cost function. Either the gradient of f is unknown or we cannot compute it. The solution is to find the global minimum x^* [71]. Algorithm 1 illustrates Steps used to select attributes

Algorithm 1 : Particle Swarm Optimization (PSO)

Input: Size of the population (N), Search space boundaries, Maximum number of iterations ($maxIter$), Termination criteria

Output: Global best position ($gbest$)

Initialization: Randomly initialize particle positions and velocities within the search space;

Evaluate the fitness of each particle and set personal bests; set the *best* to the best particle's position;

termination *criteria not met* do for each *particle* do

Update the velocity according to the world best, personal best, and current velocity.

Adapt position to velocity. Assess the new position's suitability.

if fitness is better than personal best, then

Update personal well position.

End.

If your level of fitness surpasses the world's finest, then

Update *gbest*;

End.

End.

End.

Return *gbest*;

Algorithm above shows that PSO attempts to find best feature that facilitates the differentiation between class labels by updating the position and velocity variables, reflecting how the feature can be used as a discriminative factor between anomaly categories. After applying the PSO algorithm to the dataset, it selects 34 features for each label of the three target classes.

4.2.3.3 Gray Wolf Optimization

The GWO algorithm is an optimization technique that simulates the behaviors of wolf "packs" in hunting and leadership. It attempts to find the best solution by applying three operators: exploration, exploitation, and adaptation, which are iteratively applied until the good solution is reached. The algorithm consists of three types of wolves: the alpha wolf (the leader), beta wolves (who report to the leader), and delta wolves (who report to both alpha and beta), which are used to search for the solution. Each wolf computes the fitness function, which determines how specific features can be used as discriminative factors among the three class labels. This computation is based on the position variable. The mathematical representation of each phase is given below:

a. Searching for Prey (Exploration Phase)

In this step, the wolves randomly explore the global search space for searching of prey. This makes their movement explore the varied regions because randomly chosen individuals drive it. This is mathematically defined as:

$$A \rightarrow = 2.a \rightarrow .(r_1) \rightarrow - .a \rightarrow \quad (4.5)$$

$$C \rightarrow = 2.(r_2) \rightarrow \quad (4.6)$$

$$D \rightarrow = |C \rightarrow .W_Random - W_i| \quad (4.7)$$

$$[W_i]^{new} = W_Random - A \rightarrow .D \rightarrow \quad (4.8)$$

where:

W_Random represents a randomly chosen wolf (ensuring random search).

$a \rightarrow$ decreases linearly from 2 to 0 over iterations, controlling exploration and exploitation balance.

$(r_1) \rightarrow$, $(r_2) \rightarrow$ are random numbers distributed in range of [0,1], ensuring stochastic behavior.

b. Encircling the Prey

Once the prey has been identified, wolves slowly close in on the prey, and the search space diminishes. The following equations mathematically represent this encircling mechanism:

$$D \rightarrow = |C \rightarrow .W_\alpha - W_i| \quad (4.9)$$

$$W_i]^{new} = W_\alpha - A \rightarrow .D \rightarrow \quad (4.10)$$

where:

W_α is the best solution (alpha wolf) found.

W_i represents the position of an individual wolf.

$A \rightarrow$ and $C \rightarrow$ are coefficient vectors controlling step size and search radius.

This equation ensures wolves gradually approach the best solution, refining their movement.

c. Attack Phase (Exploitation and Convergence)

Once wolves have their prey cornered, they go into exploitation. They restrict their search area and intensify searching around the best solution by employing either shrinking encircling or a spiral movement strategy:

a. Shrinking Encircling Mechanism

When $|A| < 1$, wolves move closely toward the prey, mimicking an intensified search:

$$[W_i]^{new} = W_{\alpha} - A \cdot D \quad (4.11)$$

Since $|A|$ is reduced over iterations, the search focuses on fine-tuning solutions.

b. Spiral Movement Mechanism

If the prey moves unpredictably, wolves diverge from direct movement and follow a spiral path:

$$[W_i]^{new} = (D' \cdot e^{bl \cos(2\pi l)} + W_{\alpha}) \quad (4.12)$$

where:

b controls the shape of the spiral movement.

l is a random number in $[-1, 1]$, introducing stochastic behavior.

$D' = |W_{\alpha} - W_i|$ ensures movement around the best solution.

The conventional WOA suffers from several limitations, such as premature convergence, where solutions get trapped in local optima due to insufficient search diversity. It also suffers from an imbalanced exploration-exploitation tradeoff, which results in either excessive randomness or slow convergence. Furthermore, high sensitivity to parameter selection makes manual tuning crucial for performance, while computational overhead increases for large-scale problems. Lastly, WOA is inefficient in discrete search spaces, limiting its application in feature selection

Algorithm 2 outlines the steps used to apply the GWO algorithm to select features that are discriminative between categories.

Algorithm 2: Gray Wolf Optimization (GWO)

Input: Population size (N), Search space boundaries, Maximum number of iterations ($maxIter$), Termination criteria

Output: Global best position ($gbest$)

Initialization: Randomly initialize the wolf positions within the search space;

termination *criteria not met*

for each *wolf* do

Update position on the basis of how features discriminate anomaly types;

Evaluate the fitness of the new position;

end

Update wolves on the basis of fitness;

end

Return $gbest$;

By choosing 63 characteristics for label, 64 for category, and 63 for subcategory, the GWO for feature selection demonstrated a discriminating approach. The somewhat differing feature counts among target classes indicate that the GWO adjusts to the unique features of each class, highlighting its versatility.

After calculating the features for each hyper-labeling technique or algorithm the correlation coefficient and the results of PSO and GWO for each target was computed, a crossover between the top 35 features across the three feature selection algorithms was calculated, revealing the follow and the results are shown at table 4.

Table 4.1: Feature Selection Results.

Target	Length	Features
Lab.	17 Features	“(‘Dst_Port', 'SYN_Flag_Cnt', 'Down/Up_Ratio', 'Bwd_Pkt_Len_Mean', 'Init_Bwd_Win_Byts', 'Fwd_Pkt_Len_Mean', 'Flow_IAT_Mean', 'Fwd_Pkt_Len_Std', 'Fwd_Header_Len', 'Pkt_Len_Var', 'Pkt_Len_Max', 'Flow_IAT_Min', 'Bwd_PSH_Flags', 'Bwd_Seg_Size_Avg', 'Fwd_Pkt_Len_Max', 'Src_Port', 'Protocol')”
Cat.	16 Features	“(‘Idle_Max', 'Bwd_Pkt_Len_Max', 'Flow_Duration', 'Pkt_Len_Var', 'SYN_Flag_Cnt', 'Src_Port', 'Fwd_Pkt_Len_Max', 'Bwd_Pkt_Len_Mean', 'Active_Max', 'Dst_Port', 'Init_Bwd_Win_Byts', 'Bwd_Pkt_Len_Std', 'Idle_Mean', 'Bwd_Seg_Size_Avg', 'ACK_Flag_Cnt', 'Pkt_Len_Min')”
Sub -cat.	22 Features	“(‘Flow_IAT_Max', 'Subflow_Bwd_Pkts', 'Subflow_Fwd_Pkts', 'Flow_IAT_Std', 'Src_Port', 'Bwd_IAT_Max', 'ACK_Flag_Cnt', 'Fwd_Header_Len', 'Protocol', 'Tot_Fwd_Pkts', 'Bwd_Seg_Size_Avg', 'Bwd_Pkt_Len_Std', 'Bwd_IAT_Mean', 'Pkt_Len_Var', 'Idle_Mean', 'Dst_Port', 'Idle_Max', 'Active_Min', 'Idle_Min', 'Flow_IAT_Min', 'Pkt_Len_Max', 'Flow_Duration')”

4.3 DEVELOPING AN IDS FOR THE IOT

The following section outlines the steps for developing an IoT intrusion detection system. It begins with Section 4.3.1, and 4.3.2 which implements a hybrid application on data traffic using the features we learned in the previous step to identify and classify attack types. The second step in Section 4.3.3 is to implement the hybrid algorithm and obtain the results.

4.3.1 Hyperparameter Optimization (Coronavirus Herd Immunity Optimizer)

To enhance the efficiency of our proposed system in identifying anomaly categories in IoT network traffic, we use the CHIO algorithm to determine the optimal values for the hyperparameters of the DT model. Hyperparameter estimation is considered to be the most important stage in this system because it identifies the best values for each parameter in the DT model, thereby enhancing its performance.

CHIO is an optimization technique used in this study because it enhances convergence speed, balances the exploration-exploitation trade-offs, and finds optimal or near-optimal solutions for the hyperparameters. The basic steps of the CHIO algorithm are illustrated in algorithm 3.

Algorithm 3: Coronavirus Herd Immunity Optimizer (CHIO)

Input: Population size (N), Search space boundaries, Maximum number of iterations (maxIter), Termination criteria

Output: Optimal parameters for the DT

Initialization: Randomly initialize the DT parameters within the search space;

termination criteria not met

for each individual

Update DT parameters based on DT training and Best accuracy

Evaluate the fitness of the new DT parameters using a population dynamics model;

end

Update the global best DT parameters based on fitness;

end

return Optimal parameters for DT

CHIO's adaptive nature of CHIO and its innovative method for selecting optimal hyperparameters have significantly improved accuracy. CHIO improved the model's capacity to identify complex patterns in the data by adjusting its parameters. This enhancement improved the model's resilience in a variety of settings in addition to increasing its predictive power.

This optimization technique proposes to rely on herd immunity, based on COVID-19 technology. The CHIO algorithm was applied in this work to improve the decision tree algorithm. The CHIO algorithm is discussed step by step above. Meanwhile, the figure below shows a flowchart in the form of symbols.

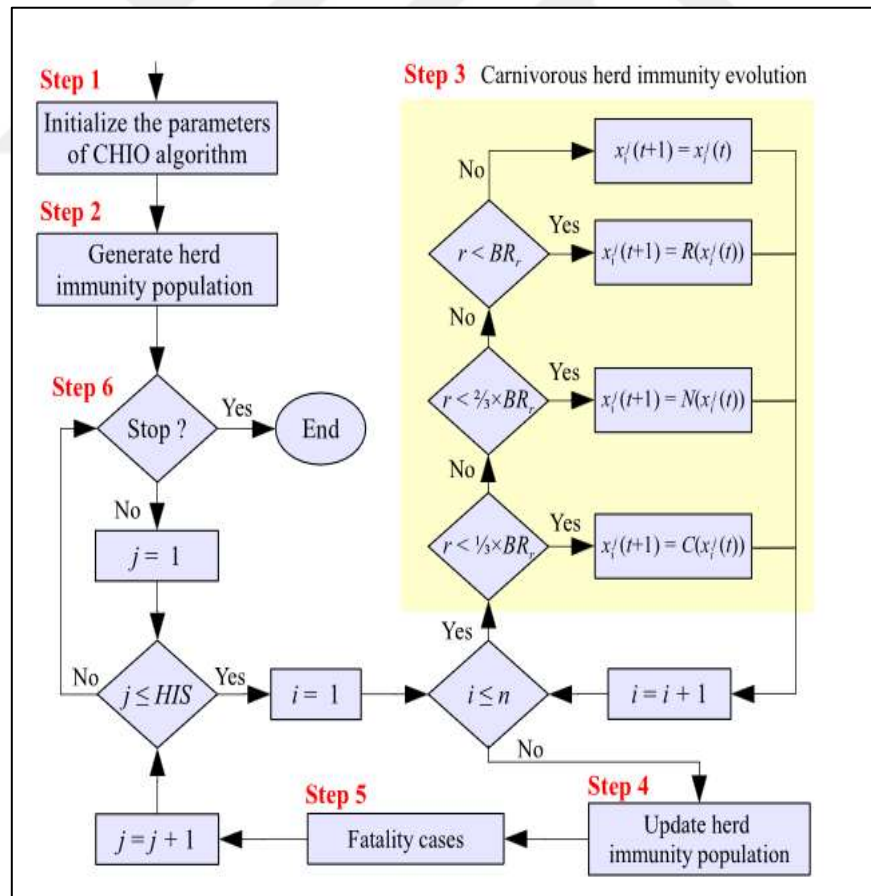


Figure 4.9: Flowchart of CHIO Algorithm[72].

4.3.2 Classification

Machine learning models employ classification algorithms to anticipate the appropriate label for given data, making classification a predictive modeling process. AI models get better at recognizing different kinds of data, spotting patterns, and producing more precise predictions as they gain experience analyzing and categorizing data in their training datasets. Test data is used to assess the model's performance after the training phase. The model is exposed to unseen real-world data once it regularly performs well.

In the classification stage, the DT machine-learning model was used in this work. During this stage, many other machine learning algorithms were also used, such as SVM [40], kNN [41], and Extreme Learning Machine (ELM) [42]. Among these, the DT outperformed all other algorithms.

4.4 EXPERIMENTAL RESULTS AND EVALUATION

The experimental results were obtained using an i5 CPU, 12 GB RAM, and a 2 GB NVIDIA GeForce MX130 graphics card. The Software environment, however, provides Python 3.8.8. Our method makes use of the IoTID20 dataset, a thorough and varied compilation of actual IoT network traffic data. This fundamental decision guarantees that our system is based on empirical findings, increasing its applicability and relevance to real-world operating situations.

Next, two subsets of the IoTID20 dataset are created: an 80% training and a 20% testing. The classifier was trained utilizing the training, and its effectiveness was evaluated utilizing the testing set. Figure 4.10 displays the outcomes of classification using SVM, kNN, DT, and ELM utilizing all 67 features.

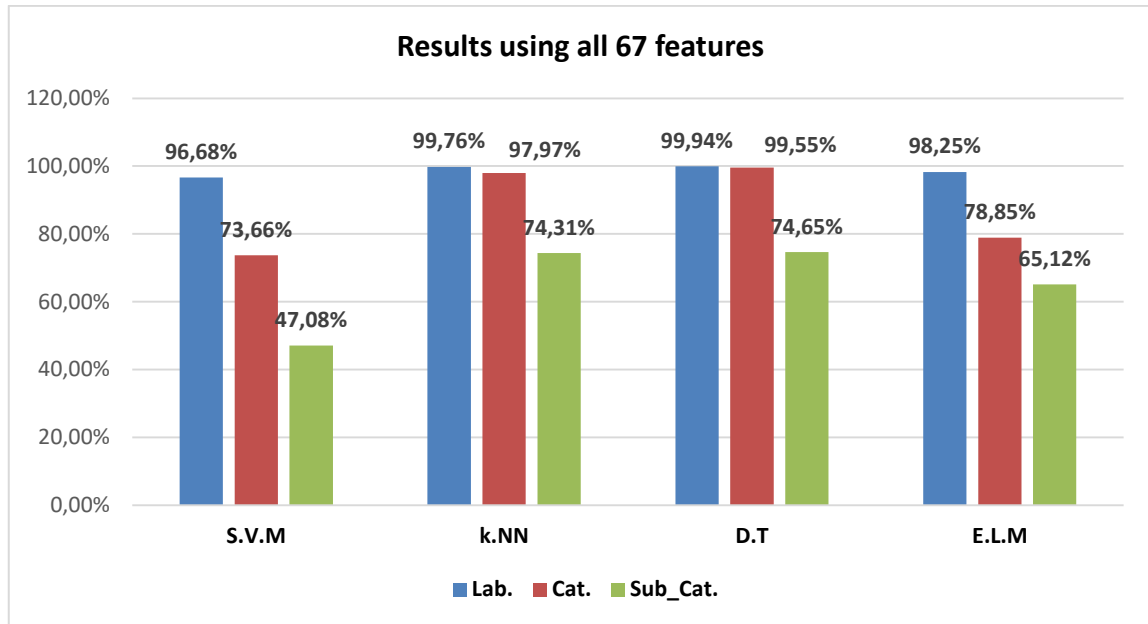


Figure 4.10: Results Using all 67 Features.

Following data preprocessing, the 67 reduced features served as input for the three feature selection methods—PSO and GWO. Figure 4.11 displays the classification results using the chosen features.

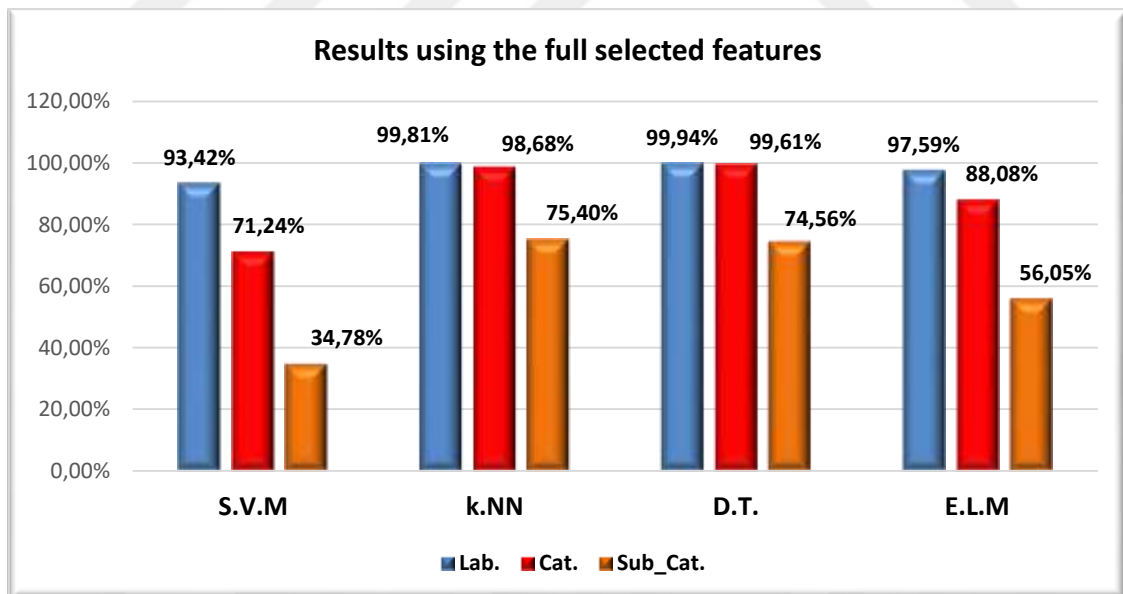


Figure 4.11: Results Using the Full Selected Features.

This small set of features makes our system well suited for deployment in IoT ecosystems where resource efficiency is paramount. The four features are selected and then input for hyperparameter tuning of the decision tree model. Finding the ideal values for the two hyperparameters—the "criteria" and the "max_depth"—is the first step in the optimization process. The criteria is a mathematical function that evaluates a decision tree split's quality. The entropy Shannon information gain and the Gini impurity are the two options. The maximum depth of the tree is established by the option max_depth. The hyperparameters of the decision tree after optimization utilizing CHIO are explained in Figure 4.12.

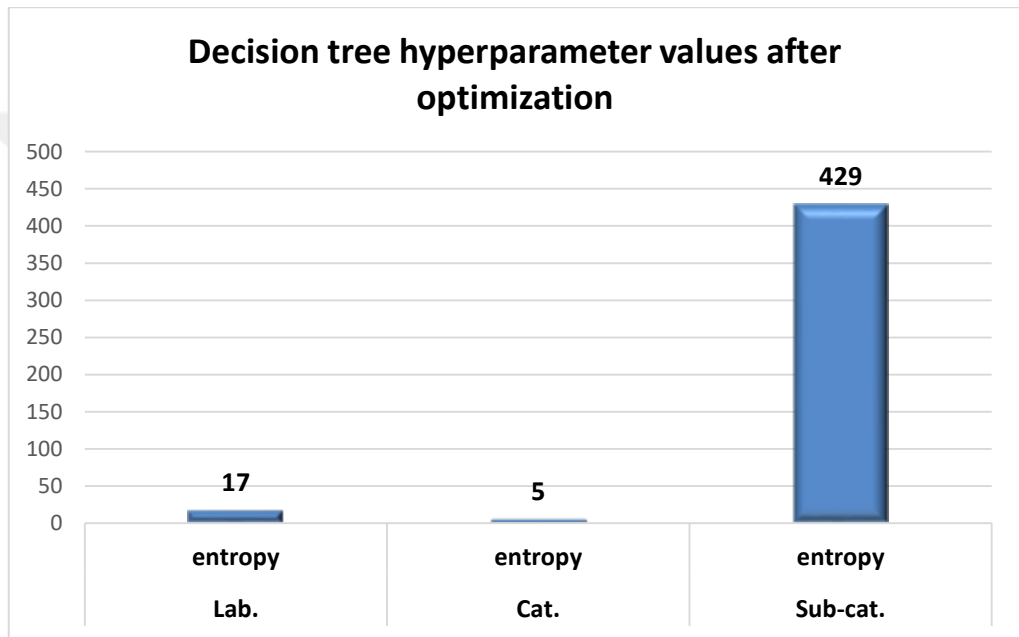


Figure 4.12: Decision Tree Hyperparameter Values After Optimization.

The decision tree model is then trained on the training data utilizing the hyperparameters. The label, category, and subcategory of the testing data are then predicted by the trained model, and the model is assessed depending on the outcomes. The accuracy of the suggested system was 99.96% for the label, 99.56% for the category, and 77.6% for the subcategory. The decision tree results are summarized in Table 5 and Figure 29 The results confirm that the feature selection and optimization processes contributed to the improved results.

Table 4.2: Decision Three Results.

	Features	Accuracy
Lab.	67	%99.76
	17	%99.94
	4	%99.96
Category	67	%97.97
	16	%99.61
	4	%99.56
Subcategory	67	74.31
	22	74.56
	4	77.6

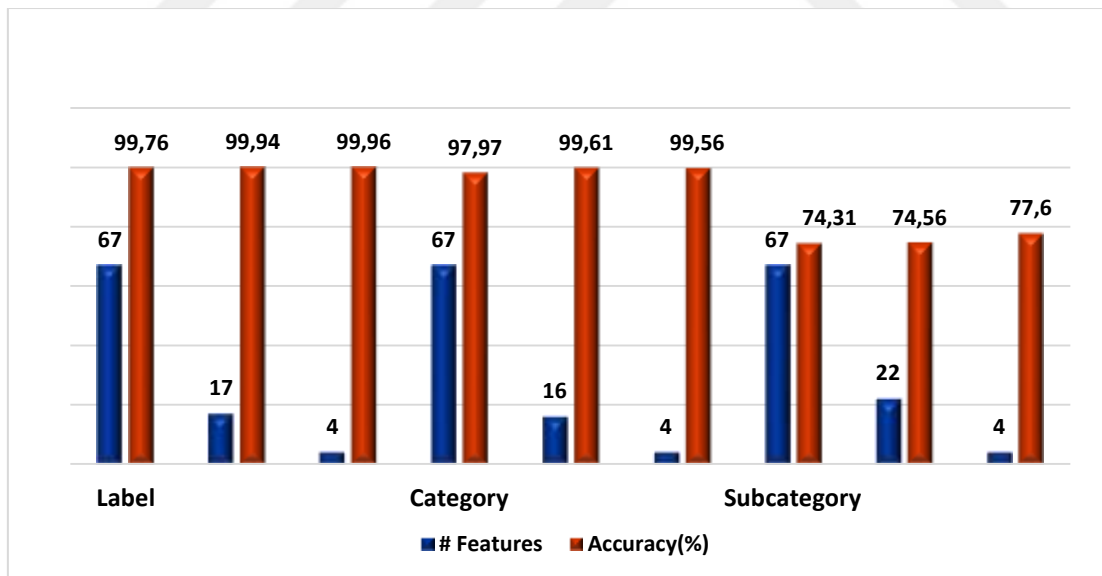


Figure 4.13: Decision Three Results.

Table 4.3 below a comparison between the proposed system and related work using the IoTID-20 dataset. The proposed system achieves promising results.

Table 4.3: Comparison With Related Work.

Study	Technique	Accuracy		
		Label	Cat	Sub_Cat
<u>Qaddoura</u> et al . [54]	SLFN	N/A	93.51	N/A
Alkahtani and Aldhyani [55]	LSTM	99.82	N/A	N/A
D.K.kReddy et al. [57]	RF and DT	98	99	82
Lahasan and Samma [58]	autoencoders	98.12	N/A	N/A
S.Ullah et al. [59]	CNN	99.84	98.12	77.55
Proposed System	DT	99.96	99.56	77.6

The suggested IDS for IoT contexts is strengthened by a number of meticulously designed actions. Every stage of the process, from data collection to feature selection, classification, and hyperparameter optimization, is carefully planned to use the distinct advantages of the selected methodologies. Using this all-encompassing approach not only ensures the system's resiliency but also positions it as a potent tool for defending IoT networks against potential security flaws. An effective and efficient follow-up process must be established once the system has successfully detected and classified possible intrusions. When an intrusion is discovered, it is stopped, and the network administrator is promptly notified about the threat's characteristics and suggested countermeasures. To help with post-event analysis and regulatory compliance, thorough logs and reports will also be produced to record the occurrence.

4.5 SUMMARY

This chapter highlights the stages of developing an intrusion detection system in the Internet of Things, from data processing and feature selection using three algorithms in two different ways, which helps the classification parameters to work faster and classify better, and then applying hybrid algorithms to identify and detect attacks on Internet of Things traffic, so that

we can then obtain the results in the final section and evaluate them by comparing them with other algorithms.



5. CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

An IDS specifically designed for IoT environments is shown in this paper. The IoTID20 dataset, which has three class labels and 83 features, is used by the system. The algorithm deftly separates typical network traffic from possible abnormalities and then further refines its classification, breaking down anomalous data into several categories such as DoS, Mirai, Scan, and MITM assaults. The third-class name, which distinguishes particular kinds of Mirai and Scan assaults, enhances this degree of specificity even more. The whole process is guided by a methodical approach that includes important steps including feature selection, data preparation, hyperparameter tuning, and classification.

Through intelligent deletions of identifier features, careful management of zero values, and skillful analysis of poor feature distributions, data preparation improved the integrity of the dataset. This effort resulted in a simplified feature set of 67 features. The simplified feature set of 67 is the result of this effort. Refined feature subsets of 17 to the label, 16 to the category, and 22 characteristics were obtained for subcategory classification, through the intersection of three selected technologies, namely: the First one, correlation coefficient, the second one, PSO, and the last one, GWO approaches. This collection was further reduced to four essential attributes using a strategic commonality approach: "Bwd_Seg_Size_Avg," "Pkt_Len_Var," "Src_Port," and "Dst_Port." By using CHIO to refine the decision tree model, the system's efficacy was further increased, resulting in high accuracy rates for label, category, and subcategory categorization of 99.96%, 99.56%, and 77.6%, respectively. The application of the system in this work shows the increase in security in networks and also motivates us to further develop Internet of Things technologies and preserve data within these networks.

5.2 FUTURE WORKS

There are a number of encouraging directions for further research and development of the suggested IDS designed for IoT settings. The first is the incorporation of real-time reaction and monitoring capabilities. There is a lot of promise in putting in place dynamic, adaptable systems that can react quickly to new risks in the always changing IoT environment.

Furthermore, adding anomaly detection algorithms—like unsupervised learning techniques—could improve the system's capacity to recognize attack patterns that haven't been noticed before. Last but not least, given how quickly IoT technologies are developing, it will be essential to update and reassess the system's performance on a regular basis in order to keep it successful in protecting IoT networks from new threats and improvements in IoT device capabilities.



REFERENCES

- [1] S. Challenges, “Internet of Things: Evolution, Concerns and Security Challenges,” 2021.
- [2] A. Alghuried, “A model for anomalies detection in internet of things (IoT) using inverse weight clustering and decision tree,” *Masters’s Thesis, Dublin Inst. Technol. Dublin, Irel.*, p. 64, 2017.
- [3] L. Graham, “Number of Connected IoT Devices Will Surge to 125 Billion by 2030, IHS Markit Says,” *IHS Markit Ltd*, 2017. .
- [4] M. binti Mohamad Noor and W. H. Hassan, “Current research on Internet of Things (IoT) security: A survey,” *Comput. Networks*, 2019, doi: 10.1016/j.comnet.2018.11.025.
- [5] J. Gao, S. Chai, B. Zhang, and Y. Xia, “Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis,” *Energies*, 2019, doi: 10.3390/en12071223.
- [6] “A Survey on IPS Methods and Techniques,” *Int. J. Comput. Sci. Issues*, 2016, doi: 10.20943/01201602.3843.
- [7] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li, and J. Wang, “A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization,” pp. 1–34, 2023, doi: 10.7717/peerj-cs.1569.
- [8] B. Abhishek and A. K. Tyagi, “An Useful Survey on Supervised Machine Learning Algorithms: Comparisons and Classifications,” *Lect. Notes Electr. Eng.*, vol. 881, pp. 293–307, 2022, doi: 10.1007/978-981-19-1111-8_24.
- [9] C. Li, J. Wang, S. Wang, and Y. Zhang, “Neurocomputing A review of IoT applications in healthcare,” *Neurocomputing*, vol. 565, no. May 2023, p. 127017, 2024, doi: 10.1016/j.neucom.2023.127017.
- [10] T. Sasi, A. Habibi, R. Lu, and P. Xiong, “A comprehensive survey on IoT attacks : Taxonomy , detection mechanisms and challenges,” *J. Inf. Intell.*, vol. 2, no. 6, pp. 455–513, 2024, doi: 10.1016/j.jiixd.2023.12.001.

- [11] A. Heidari, "Internet of Things intrusion detection systems : a comprehensive review and future directions Internet of Things intrusion detection systems : a comprehensive review and future directions," *Cluster Comput.*, no. November, 2024, doi: 10.1007/s10586-022-03776-z.
- [12] D. J. Atul, R. Kamalraj, G. Ramesh, K. S. Sankaran, S. Sharma, and S. Khasim, "A machine learning based IoT for providing an intrusion detection system for security," vol. 82, 2021.
- [13] P. Rana and B. P. Patil, "TH," no. June, 2024, doi: 10.3233/JHS-222042.
- [14] G. Obaido, B. Ogbuokiri, C. W. Chukwu, F. J. Osaye, O. F. Egbelowo, and M. I. Uzochukwu, "An Improved Ensemble Method for Predicting Hyperchloremia in Adults With Diabetic Ketoacidosis," *IEEE Access*, vol. 12, no. January, pp. 9536–9549, 2024, doi: 10.1109/ACCESS.2024.3351188.
- [15] I. D. Mienye and N. Jere, "A Survey of Decision Trees : Concepts , Algorithms , and Applications," *IEEE Access*, vol. PP, p. 1, 2024, doi: 10.1109/ACCESS.2024.3416838.
- [16] E. O. Abiodun, A. Alabdulatif, O. Isaac, M. Alawida, A. Alabdulatif, and R. S. Alkhawaldeh, "A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities," *Neural Comput. Appl.*, vol. 33, no. 22, pp. 15091–15118, 2021, doi: 10.1007/s00521-021-06406-8.
- [17] A. A. Abdulhameed, "An optimized model for network intrusion detection in the network operating system environment," vol. 4, no. 3, pp. 75–85, 2024.
- [18] R. Sharma, G. K. Mahanti, C. Chakraborty, G. Panda, and A. Rath, "An IoT and Deep Learning-Based Smart Healthcare Framework for Thyroid Cancer Detection," 2023, doi: 10.1145/3637062.
- [19] H. I. Classification, "A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges," 2023.
- [20] T. Hillel, M. Bierlaire, M. Elshafie, and Y. Jin, "A systematic review of machine learning classification methodologies for modelling passenger mode choice," no. McFadden 1981, pp. 1–40.

- [21] B. T. Jijo and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," vol. 02, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.
- [22] V. Rani, S. Tufael, N. Munish, K. Ajay, and M. Krishan, "Self - supervised Learning : A Succinct Review," *Arch. Comput. Methods Eng.*, vol. 30, no. 4, pp. 2761–2775, 2023, doi: 10.1007/s11831-023-09884-2.
- [23] S. Albelwi, "Survey on Self-Supervised Learning : Auxiliary Pretext Tasks and Contrastive Learning Methods in Imaging," 2022.
- [24] N. Saran and N. Kesswani, "A comparative study of supervised Machine Learning classifiers for Intrusion Detection in Internet of Things," *Procedia Comput. Sci.*, vol. 218, pp. 2049–2057, 2022, doi: 10.1016/j.procs.2023.01.181.
- [25] A. R. Abdulla and N. G. M. Jameel, "A Review on IoT Intrusion Detection Systems Using Supervised Machine Learning: Techniques, Datasets, and Algorithms," *UHD J. Sci. Technol.*, vol. 7, no. 1, pp. 53–65, 2023, doi: 10.21928/uhdjst.v7n1y2023.pp53-65.
- [26] M. A. Bouke, "E2IDS : An Enhanced Intelligent Intrusion Detection System Based On Decision Tree Algorithm," vol. 3, no. 1, pp. 1–16, 2022.
- [27] R. Konieczny and R. Idczak, "Mössbauer study of Fe-Re alloys prepared by mechanical alloying," *Hyperfine Interact.*, vol. 237, no. 1, pp. 1–8, 2016, doi: 10.1007/s10751-016-1232-6.
- [28] N. S. Labib, M. R. Brust, and P. Bouvry, "The Rise of Drones in Internet of Things : A Survey on the Evolution , Prospects and Challenges of Unmanned Aerial Vehicles," *IEEE Access*, vol. 9, pp. 115466–115487, 2021, doi: 10.1109/ACCESS.2021.3104963.
- [29] M. Humayun, N. Z. Jhanjhi, A. Alsayat, and V. Ponnusamy, "Internet of things and ransomware : Evolution , mitigation and prevention," *Egypt. Informatics J.*, vol. 22, no. 1, pp. 105–117, 2021, doi: 10.1016/j.eij.2020.05.003.
- [30] N. Mishra, "Internet of Things Applications , Security Challenges , Attacks , Intrusion Detection , and Future Visions : A Systematic Review," pp. 59353–59377, 2021, doi: 10.1109/ACCESS.2021.3073408.

- [31] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, “A Comparative Analysis of Supervised Machine Learning Algorithms for IoT Attack Detection and Classification,” *J. Comput. Theor. Appl.*, 2025.
- [32] A. Belenguer, J. Navaridas, and J. A. Pascual, “A review of Federated Learning in Intrusion Detection Systems for IoT,” pp. 1–13.
- [33] I. Networks and A. Awajan, “A Novel Deep Learning-Based Intrusion Detection System for,” 2023.
- [34] I. Ullah and Q. H. Mahmoud, “A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks,” 2020, doi: 10.1007/978-3-030-47358-7_52.
- [35] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of tor traffic using time based features,” 2017, doi: 10.5220/0006105602530262.
- [36] M. A. Hall, “Correlation-based Feature Selection for Machine Learning,” no. April, 1999.
- [37] S. Bashir, I. U. Khattak, A. Khan, F. H. Khan, A. Gani, and M. Shiraz, “A Novel Feature Selection Method for Classification of Medical Data Using Filters , Wrappers , and Embedded Approaches,” vol. 2022, 2022, doi: 10.1155/2022/8190814.
- [38] M. Kadhum, S. Manaseer, A. Latif, and A. Dalhoum, “Evaluation Feature Selection Technique on Classification by Using Evolutionary ELM Wrapper Method with Features Priorities,” vol. 12, no. 1, pp. 21–28, 2021, doi: 10.12720/jait.12.1.21-28.
- [39] U. F. Njoku, A. Abelló, B. Bilalli, and G. Bontempi, “Wrapper Methods for Multi-Objective Feature Selection,” *Adv. Database Technol. - EDBT*, vol. 26, no. 3, pp. 697–709, 2023, doi: 10.48786/edbt.2023.58.
- [40] J. Yang, C. Chen, J. Li, D. Liu, T. Li, and Z. Zhan, “SS symmetry Compressed-Encoding Particle Swarm Optimization with Fuzzy Learning for Large-Scale Feature Selection,” 2022.
- [41] S. M. Almufti, H. B. Ahmad, R. B. Marqas, and R. R. Asaad, “Grey wolf optimizer : Overview , modifications and applications,” vol. 1, no. 1, pp. 44–56, 2021.

- [42] A. G. Gad, *Particle Swarm Optimization Algorithm and Its Applications : A Systematic Review*, vol. 29, no. 5. Springer Netherlands, 2022.
- [43] A. Seyyedabbasi and F. Kiani, "I - GWO and Ex - GWO : improved algorithms of the Grey Wolf Optimizer to solve global optimization problems," *Eng. Comput.*, no. 0123456789, 2019, doi: 10.1007/s00366-019-00837-7.
- [44] M. Fahad *et al.*, "ARTICLE IN PRESS," vol. 0, pp. 1–18, 2018.
- [45] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Front. Bioinforma.*, vol. 2, no. June, pp. 1–17, 2022, doi: 10.3389/fbinf.2022.927312.
- [46] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos, "Feature selection based on mutual correlation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4225 LNCS, no. November, pp. 569–577, 2006, doi: 10.1007/11892755_59.
- [47] I. Conference and U. Security, "Postprint Generalizing Supervised Learning for Intrusion Detection in IoT Mesh Networks," no. UbiSec, pp. 214–228, 2021.
- [48] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW - NB15 data - set," *EURASIP J. Wirel. Commun. Netw.*, 2021, doi: 10.1186/s13638-021-01893-8.
- [49] H. Tyagi and R. Kumar, "Revue d ' Intelligence Artificielle Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches," vol. 35, no. 1, pp. 11–21, 2021.
- [50] T. Agustina, "Performance Analysis of Random Forest Algorithm for Network Anomaly Detection using Feature Selection," vol. 8, no. 2, pp. 1116–1124, 2024.
- [51] F. Hussain, S. G. Abbas, I. M. Pires, G. A. Shah, and F. Shahzad, "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.

- [52] M. M. Khan and M. Alkhathami, "Anomaly detection in IoT - based healthcare : machine learning for enhanced security," *Sci. Rep.*, pp. 1–16, 2024, doi: 10.1038/s41598-024-56126-x.
- [53] H. El Sofany, S. A. El Seoud, O. H. Karam, and B. Bouallegue, "Using machine learning algorithms to enhance IoT system security," *Sci. Rep.*, pp. 1–19, 2024, doi: 10.1038/s41598-024-62861-y.
- [54] R. Qaddoura, A. M. Al-Zoubi, I. Almomani, and H. Faris, "A multi-stage classification approach for iot intrusion detection based on clustering with oversampling," *Appl. Sci.*, 2021, doi: 10.3390/app11073022.
- [55] H. Alkahtani and T. H. H. Aldhyani, "Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms," *Complexity*, 2021, doi: 10.1155/2021/5579851.
- [56] Y. Song, S. Hyun, and Y. G. Cheong, "Analysis of autoencoders for network intrusion detection†," *Sensors*, 2021, doi: 10.3390/s21134294.
- [57] D. K. K. Reddy, H. S. Behera, J. Nayak, B. Naik, U. Ghosh, and P. K. Sharma, "Exact greedy algorithm based split finding approach for intrusion detection in fog-enabled IoT environment," *J. Inf. Secur. Appl.*, 2021, doi: 10.1016/j.jisa.2021.102866.
- [58] B. Lahasan and H. Samma, "Optimized Deep Autoencoder Model for Internet of Things Intruder Detection," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3144208.
- [59] S. Ullah *et al.*, "A new intrusion detection system for the internet of things via deep convolutional neural network and feature engineering," *Sensors*, vol. 22, no. 10, p. 3607, 2022.
- [60] A. A. Alsulami, Q. Abu Al-Haija, A. Tayeb, and A. Alqahtani, "An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering," *Appl. Sci.*, vol. 12, no. 23, p. 12336, 2022.
- [61] S. Krishnan, A. Neyaz, and Q. Liu, "IoT network attack detection using supervised machine learning," 2021.

- [62] A. Fatani, A. Dahou, M. A. A. Al-Qaness, S. Lu, and M. A. Elaziz, "Advanced feature extraction and selection approach using deep learning and aquila optimizer for iot intrusion detection system," *Sensors*, 2022, doi: 10.3390/s22010140.
- [63] L. T. M. Blessing and A. Chakrabarti, "DRM, a design research methodology," *DRM, a Des. Res. Methodol.*, pp. 1–397, 2009, doi: 10.1007/978-1-84882-587-1.
- [64] A. Habbal, *TCP SINTOK: Transmission Control Protocol with Delay-based Loss Detection and Contention Avoidance Mechanisms for Mobile Ad hoc Networks*. 2014.
- [65] R. Jain, "THE ART OF COMPUTER SYSTEMS PERFORMANCE ANALYSIS Techniques for Experimental Design, Measurement, Simulation and Modeling," *16th Int. Comput. Meas. Gr. Conf. C. 1990*, no. April 1991, pp. 1360–1361, 1990, doi: 10.1093/comjnl/35.3.207.
- [66] W. D. Kelton, *Simulation Analysis.*, vol. 1. 1983.
- [67] M. S. Ummah, "AN INTRODUCTION TO NETWORK MODELING AND SIMULATION FOR THE PRACTICING ENGINEER," *Sustain.*, vol. 11, no. 1, pp. 1–14, 2019, [Online]. Available: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI.
- [68] A. J. Dhruv, R. Patel, and N. Doshi, "Python: The Most Advanced Programming Language for Computer Science Applications," no. Cesit 2020, pp. 292–299, 2021, doi: 10.5220/0010307902920299.
- [69] F. Rustam, W. Aljedaani, M. S. Elsayed, and A. D. Jurcut, "FAMTDS: A novel MFO-based fully automated malicious traffic detection system for multi-environment networks," *Comput. Networks*, vol. 251, no. May, p. 110603, 2024, doi: 10.1016/j.comnet.2024.110603.
- [70] İ. Kara, "Detection, Technical Analysis of Brute Force Attack," *Sak. Univ. J. Comput. Inf. Sci.*, 2019.

- [71] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/931256.
- [72] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. Abu Doush, "Coronavirus herd immunity optimizer (CHIO)," *Neural Comput. Appl.*, 2021, doi: 10.1007/s00521-020-05296-6.

