March 2020

Investigation of Machine Learning Algorithms for Intrusion Detection System in Cybersecurity

Mohmmed Alrowaily
University of South Florida

Investigation of Machine Learning for Intrusion Detection Systems in Cybersecurity

by

Mohammed Alrowaily

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Zhuo Lu, Ph.D.
Nasir Ghani, Ph.D.
Ismail Uysal, Ph.D.
Srinivas Katkoori, Ph.D.
Kaiqi Xiong, Ph.D.

Date of Approval:
April 1, 2020

Keywords: Hybrid Feature Selection, Metaheuristic Optimization, Ensemble Classification,
Data Reduction, Network Security

Dedication

To my beloved family, who has always been behind me in every step of my life. Without their inspiration, trust, and support, this work would not have materialized.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my academic advisor, Dr. Zhuo Lu, for his expert guidance and unlimited support throughout my Ph.D. journey. His thoughtful and insightful comments were always the source of my inspiration and ideas. I am deeply indebted to him for not only giving me the opportunity to be part of his research group in the Communications, Security, and Analytics (CSA) Lab, but also provided me with an excellent environment for doing research.

Besides my advisor, I would like to thank my committee members, Dr. Nasir Ghani, Dr. Ismail Uysal, Dr. Srinivas Katkoori, Dr. Yao Liu, and Dr. Kaiqi Xiong for their feedback and serving in my committee despite their overwhelmingly busy schedule.

I am grateful to the people of the first step and penultimate step, to those who during the lean years have been a rainy cloud. Special thanks to my closest friends Mohammed Aldalaan, Abdulaziz Alanazi, Saud Aldossari, Abdulrahman Alsolami, and Omar Alanazi.

Table of Contents

List of Tables

List of Figures

Abstract

The proliferation in usage and complexity of modern communication and network systems, a large number of trustworthy online Services and systems have been deployed. Even so, cybersecurity threats are still growing. An Intrusion Detection System (IDS) plays a vital role in ensuring the security of communication networks, and it is taken into account as the subsequent security gate after the firewall. The IDS informs the system or network administrator in order to take specific actions to evade the suspicious activities. Three significant contributions are made during the course of this research to illustrate the feasibility of these IDS approaches. In the first contribution, we investigate the effectiveness of using conventional machine learning techniques based on intrusion detection systems. The second contribution proposes an ensemble learning algorithm for cybersecurity threat detection. The third contribution proposes a hybrid feature selection approach for improving network attack detection. All presented algorithms were evaluated on the recent public CICIDS2017 dataset, which consists of benign and the most cutting-edge common attacks, and compared with other approaches. This research considers several machine learning classifiers and feature selection techniques in order to study their classification performance under attack over different metrics. The empirical results of the three implemented systems conclude that the chosen minimized features provide promising performance to develop IDS that is effective and efficient for network intrusion detection. Moreover, these models not only improves the classification accuracy but also reduces the false alarm rate in the classification of IDS attacks.

Chapter 1: Introduction and Background

1.1 Cybersecurity in the Modern Age

As the globe is encompassed in the digital transformation, cybercrime develops alongside as well. Consequently, new measures and techniques in cybersecurity need to be developed at an equally rapid pace [1]. For effective cybersecurity measures, it is essential to remain agile in identifying opportunities that may help to improve security as well as to adapt to the threats that are evolving. Currently, professionals in the Information Technology field possess advanced tools provided by cloud technologies that not only give real-time visibility into cybersecurity but also allow them to proactively prevent threats before they escalate into issues [2]. Cyber-attacks are the significant reasons why various companies end up experiencing losses in the due process of running the business as well as why individuals may fall victim to cyberbullying. Therefore, the importance of cybersecurity in the modern world cannot be downplayed. Cybersecurity not only helps to protect data through data encryption and the addition of security layers, but it also helps to protect the reputation of a company or an individual. Computer proficiency, coupled with hacking tactics, are currently being used to steal information as well as economic resources, particularly money, which may result in ruined relationships and reputations. These malware and viruses that attack the system to steal essential aspects from the system also play a role in slowing down the system [3]. Therefore cybersecurity is important for maintaining and increasing the speed of the websites. Ultimately, cybersecurity is cost-effective. Cyber-attacks are associated with so many dangers, and defense from these attacks is arguably the best option. Cybersecurity is cost-effective because it also enables systems to seamlessly and frequently update without the need for reconfiguration, which for a company may mean remaining compliant with the regulatory guidelines [4].

1.2 A Brief History of Intrusion Detection

In 1980, Anderson J P. introduced the principle of intrusion detection. He specifies intrusion or threat as an attempt triggered the system to be not available or unreliable by illegitimate accessibility information or operating information [5]. Anderson J P. suggested an intrusion detection concept according to an audit record of the operating system. However, researchers have paid little interest to this strategy, concentrating instead on file encryption and rejection of accessibility to the data from a verified host [6]. In 1985, Denning D E. proposed an IDS prototype which is named as Intrusion Detection Expert System (IDES). The IDES model is composed of an object, host, profile characteristic, anomaly record, audit record, and activity rules [7]. This system is independent from system platform, application environment, system weakness, and types of attack. In 1988, Lunt T enhanced the system and developed a real-time IDS that identified attacks as data was received [8]. Lunt's model is employed to determine intrusions behavior for an individual host. In 1989, Heberlein L T provided network-based intrusion detection and put forward Network Security Monitor (NSM), which identifies unusual behavior by observing network data in local area networks, rather than examining audit record in the host [9].

1.3 Aims

This work aims to provide researchers with transparent results for some witnessed machine learning algorithms when deployed for an intrusion detection system, using evaluating metrics and recording their efficiency when they undergo optimization methods. Optimization methods such as data cleaning or modifications, ensemble classifications, feature selection. All an effort to exhibit and provide good practices within the field of attack detection, serve to help future works utilizing machine learning remedies turn into simple to comprehend in this research area.

1.4 Objectives

- Performing a literature survey related to intrusion detection systems by considering different IDS tools and researches conducted utilizing machine learning-based solutions to significantly improve them.

- Finding an intrusion detection dataset for training and evaluation, which contains benign and the most up-to-date common attacks, and resembles the real-world data (PCAPs), as the proposed solutions will put to use some machine learning and data mining techniques.

- Research which subset of traffic features that appropriate for detecting different kinds of malicious traffic, by making use of many feature selection algorithms for the sake of improving system efficiency.

- Develop an intrusion detection system that can deliver maximum classification accuracy, minimum false alarms, highest precision-recall, as well as reducing the amount of time taken to detect cyberattacks.

- Measuring the performance of the system that detects several attacks using different evaluation metrics based approach.

1.5 Dissertation Structure

- Chapter 1 begins by presenting the importance of cybersecurity in the modern age, followed by a brief history of intrusion detection. It also discusses the aims and desired objectives of this research. The remaining sections of this chapter cover the information on the background as an introduction to the topics of the thesis, such as intrusion detection systems, feature selection techniques, a comprehensive review of machine learning as well as the classification of IDS. The applied CICIDS2017 dataset, besides its statistical observations and attack scenarios, are likewise detailed in this chapter.

- Chapter 2 investigates the effectiveness of machine learning algorithms for the intrusion detection system. Several experiments were performed on seven machine learning classifiers, and make use of public intrusion detection dataset (CICIDS2017), which includes benign and the most cutting-edge common attacks. Since achieving the highest accuracy does not necessarily signify that the classifier accurately predicts with high reliability. As a result, we extensively evaluate our system over different performance metrics to examine the reliability of the proposed system results.

- Chapter 3 proposes a heuristic methodology that combines the benefits of correlation-based feature selection (CFS) and bat algorithm (BA) to overcome this issue. An ensemble approach was used to enhance the predictive performance by combining decisions from multiple classifiers (C4.5, RF, and CSForest) based on the average of probabilities (AOP) combination rule.

- Chapter 4 presents a hybrid feature selection approach that combines the strengths of both the filter and the wrapper to select the optimal feature subset from the original feature set. Symmetric Uncertainty (SU) acts as a filter to remove redundant features and Consistency Subset Evaluator (CNS) with Flower Pollination Algorithm (FPA) and Random Forest classifier as a wrapper to select the ideal feature subset from the remaining features.

- Chapter 5 summarizes the main contributions made by the entire thesis and highlighting possible extensions and future work.

Network attack is an endeavor which attempts to jeopardize the typical operating of a computer network. To reduce the effects of cyber attacks, we need to design a system called intrusion detection, which is an approach to alleviate or alerts these intrusions. Nevertheless, it eventually becomes problematic to oversee and determine intrusions at extremely high network speed. For that reason, it ends up being an essential element for organizations to gear up themselves versus impending network attacks. With traditional intrusion detection techniques, we have striven to keep a watch on the networks effectively. To conquer these obstacles, in recent years, there have been numerous efforts to propose effective and reliable intrusion detection system (IDS). IDS is an application that keeps track, detects, and prevents any set of actions that compromise the confidentiality, integrity, and availability (CIA) of a system's resources [10]. It contains supervising of undesirable usage of the network resources, maintaining it accessible for the authorized users and hindering data loss to intruders.

1.6 Cybersecurity Triad

The objectives of cybersecurity known as the CIA triad are pinpointed on the first measurement of cybersecurity cube and are commonly used as criteria for protecting and evaluating cybersecurity system. These objectives, involving confidentiality, integrity, and availability (CIA) [11], as explained below.

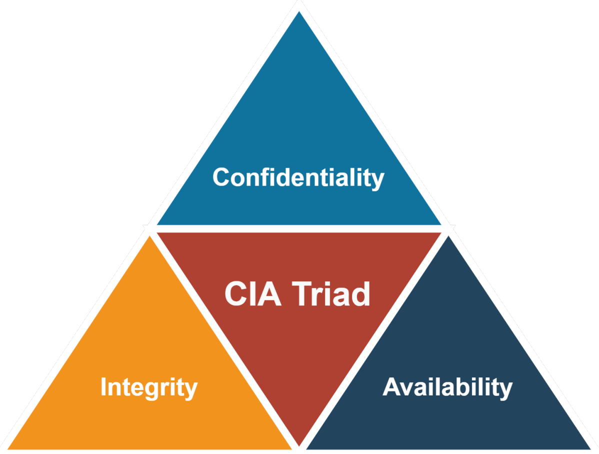


Figure 1.1: Cybersecurity CIA triad

- Confidentiality: Maintaining accredited restrictions on information accessibility and disclosure, featuring strategies for securing individual privacy and valuable proprietary information. A loss of confidentiality is the illegal declaration of information.
- Integrity: Defending against improper data alteration or damage, ensuring information non-repudiation and legitimacy. A loss of integrity is the unauthorized tampering or devastation of data.
- Availability: Ensuring that authorized parties have timely and reliable access to and use of resources when they are needed. A loss of this property is the interruption of accessibility to and usage of system resources.

1.7 Intrusion Detection System

An IDS is a software application or a device that helps to monitor either a network or systems for policy violations or any malicious activities, which makes it a crucial component of promoting cybersecurity [12]. Security information and event management system centrally collect and report any policy violations or malicious activity detected. Primarily, IDS systems are designed to ensure that the systems can promptly detect when a network attack or intrusion might be occurring. When an IDS is strategically placed at a point or various points within a network for monitoring traffic going to or leaving from all the connected devices on the network, it will analyze the traffic as well as match the traffic that is passed on the subnets to a library of known attacks. However, noise, as referred to in signal processing, can greatly limit the effectiveness of an IDS, where unwanted modifications such as bad packets that software bugs generate may result in significantly higher rates of a false alarm from the IDS [13]. IDS systems are vital in contemporary networked business environments that require high-security levels for ensuring a trusted and safe communication of information between several organizations. These IDS systems act as safeguard technology that can be adapted for the security of the system in an event where the traditional technologies have failed. A major challenge that is rapidly evolving technology as this poses is that cyber-attacks are increasingly becoming more sophisticated learning newer ways of bypassing the security measures of these IDS systems through, for instance, fragmentation, where fragmented packets are sent, and these will allow attackers to remain under the radar, ultimately bypassing the IDS capability to detect the attack signature of the attacker [14].

1.8 Feature Selection

Feature selection (or attribute selection) refers to a subset of relevant features selection process for use in the model construction [15]. Feature selection techniques are used to simplify models and make them easier for users and researchers to interpret, evade dimensionality in ML, enhance generalization through reducing variance, and to shorten the time for training as well. When applying feature selection technique to data, often the data contains traces of features either irrelevant or redundant, which can be removed without losing much information. Feature selection techniques are often confused with feature extraction. However, the two differ in that feature extraction uses functions of the original features to create new features; on the other hand, feature selection techniques presents a subset of the original features [16]. Feature selection techniques are widely used in domains entailing an array of samples and few data points to act as comparing samples [17]. The feature selection algorithm is the combining of a search technique to be used to propose new feature subsets with an evaluation measure that helps to score these different feature subsets. The data features selected to be deployed for training a machine learning model have a significant impact on the maximum performance that can be achieved. Redundant, partially relevant, or completely irrelevant data features will impact the model performance negatively hence making the optimal feature selection technique a vital machine learning concept that impacts the performance of a model greatly [18]. Feature selection can be classified into the following three categories.

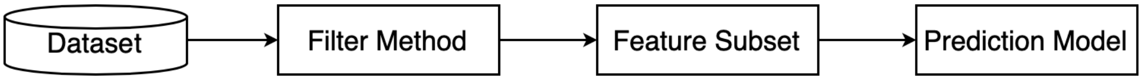


Figure 1.2: Filter method

1.8.1 Filter Method

Filters can be taken into account as opting for the top features providing the most information about the classes, based upon a statistic criterion. Given that the attributes are frequently assessed individually from each other, these methods are fast. They are reliable to diminish the attributes space, mainly when the total number of attributes in the dataset is large. Nevertheless, they may not wholly remove redundancy since the existence of one attribute might minimize the effect of some others on the class attribute. As they are not customized to any particular classifier, the chosen attributes should be utilized as input for another processing procedure instead of as the final feature subset for classification [19].

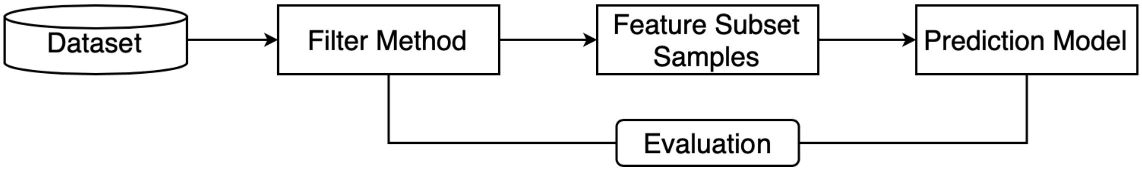


Figure 1.3: Wrapper method

1.8.2 Wrapper Method

Wrapper approaches conduct a search amongst the dataset to choose an optimum set. A predictive model is used to evaluate the attribute subsets and appoint a score based upon the classifier accuracy. The method is computationally costly for high dimensional datasets. However, wrappers tend to perform much better in choosing features as they keep in mind the model hypothesis into account [20].

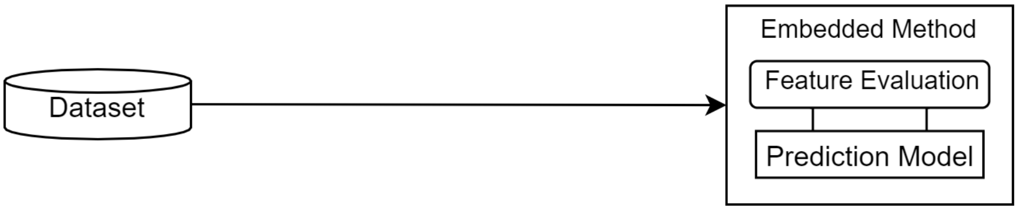


Figure 1.4: Embedded method

1.8.3 Embedded Method

Unlike the filter and wrapper approaches, the embedded method of attribute selection does not come apart from the learning from the feature selection component. In embedded structure, the feature selection algorithm is embedded into the learning procedure of the classifier construction process, thus minimizing the computational costs caused by the classification algorithm required for each subset. An example of such a model is the decision tree induction algorithm [21], wherein at each branching node, an attribute needs to be chosen.

1.9 Machine Learning

Machine learning (ML) [22] is a subset of artificial intelligence that utilized in computer systems with the ability to learn without being explicitly programmed to do specific tasks. A procedure which corresponds to that of data mining is being followed in machine learning. A machine learning scheme includes two main phases, namely, training and testing [23]. In the training phase, the training data samples are the input wherein by taking advantage of a learning algorithm the features are learned. In the testing phase, an implementation engine is employed by the learning algorithm in order to make a prediction for the unknown testing data. The classified data is granted as the output by the learning system. Currently, Machine Learning is thriving in the processing of natural language and image recognition, but it is also making moves in cybersecurity as well [24]. The challenge, however, remains that cyber terrorists are always in pursuit of weaknesses in the algorithms or a system to be able to bypass security. Other importances of this concept are that it provides an efficient solution in the modern era of large data amounts, it helps in solving regression through predicting of the next values based on data from the previous values, it classifies by grouping things in different categories based on their similarities, it is used for association learning where it recommends things based on a user's previous experience and ML also does dimensionality reduction where it conducts generalization of features in an array of examples [25]. Applying these applications of ML to cybersecurity, a regression can be used to detect fraud by noticing a change in location for activity, classification in cybersecurity deploys spam filters for classifying messages differently from spam. Applying associate rule learning to cybersecurity involves the process where a system learns a response to an incident, and when risk values are assigned to these incidents, association rule learning can be used to offer risk management solutions. Dimensionality reduction in cybersecurity is vital for handling unlabeled data in complex systems such as in areas of face detection [26].

With the rapid developments in cyberattacks and accessibility of massive amounts of malevolent data in cyber frameworks, machine learning and data mining and various other interdisciplinary abilities are often utilized to manage the obstacles of cybersecurity. Hence based on the data provided to the algorithm, there are two major types of machine learning, namely supervised learning and unsupervised learning [27].

1.9.1 Supervised Learning

The supervised methods are based on existing pre-defined knowledge, which requires two datasets, training, and testing datasets. Firstly, the training data had been correctly classified and labelled. Labelling data includes tagging the instances in the training dataset, for instance, into normal and attack class. At the next stage, the model can anticipate any future expectations. This method learns from the pre-defined input instances, and yield a classifier that can be utilized to map unseen data among one the two formerly specified classes. The computational cost associated with the supervised learning methods is high as a result of using the labelling process.

1.9.2 Unsupervised Learning

Unsupervised learning methods performed without having pre-defined knowledge [27]. Thus, the system primarily concentrates on discovering statistical relationships between the data instances and classify instances depended on how strongly they correlate with each other. The cost of this approach is low since it does not entail outsourcing knowledge like labelling.

1.10 Classification of Intrusion Detection System

Intrusion detection methods are classified in several categories. Depending on the location of the network system infrastructure. IDS are categorized into two types: network-based and host-based [28]. On top of that, pertaining to the main detection technique, IDS can be divided into two types: anomaly-based [29] and signature-based [30]. An overview of an IDS is depicted in Figure 1.5.

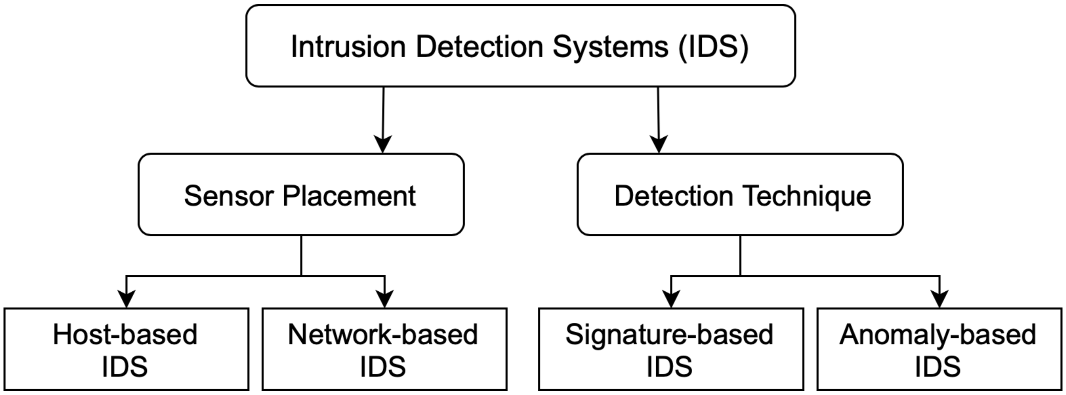


Figure 1.5: Overview of intrusion detection system

1.10.1 Network-Based IDS

NIDS have become significant security tools used in many modern network environments. They can be used to detect and classify all the traffic traversing between computers and devices in a network. Since the intrusions typically take place as irregular patterns, this type of IDS examines traffic to determine the incidence of regular traffic and abnormal activities.

1.10.2 Host-Based IDS

The main purpose of a HIDS is gathering information pertaining to the security of a certain single system or host. Even with the fact that NIDS monitors all traffic on the network, HIDS only watch intrusions based on the host system internals such as file systems and operating system. These agent hosts are described as sensing units, which are implemented on a machine that is most likely to be susceptible to possible intrusions.

1.10.3 Anomaly-Based IDS

Anomaly detection is built based on the conception of a baseline profile standing for normal network behavior, by recognizing any significant deviation of the system falls outside of a predefined set of normal network behaviors. Nonetheless, the weakness of anomaly-Based IDS is the high amount of false positive rate.

1.10.4 Signature-Based IDS

Signature-based techniques, also known as misuse detection, are based on pattern matching techniques to find a known attack. To put it simply, it analyzes network activities by using a set of well-known signatures or patterns of a previous intrusion that already exists in the signature database. Whenever an undertaking matches with the signature of a previous intrusion that already exists in the IDS database, an alarm signal is triggered.

1.11 IDS Evaluation Dataset

This research uses the CICIDS2017 benchmark dataset [31] in our experiments, which is one of the most neoteric publicly available datasets and exemplifies a data set that satisfies the 11 important criteria [32] for generating a valid IDS dataset. The CICIDS2017 dataset is asserted to be most upgraded with all common attacks and also real-world web traffic. This dataset covers seven prevalent families of attacks, namely botnet, DoS attack, infiltration attack, DDoS attack, web attack, brute force Attack, and heartbleed attack.

1.11.1 Statistical Observations

The CICIDS2017 data were collected for five days, and 78 features extracted from generated network traffic, an overview of the available features is shown in Table 1.1. The CSV version of the dataset divided into eight files for machine learning purposes and contains

2,830,743 rows. Each row is labeled as Benign or one of the other 14 types of attacks. The first two columns of Table 1.2 present the class labels and their corresponding counts. This number of attack labels is moderately large, where some labels are sufficiently smaller than others, this in fact what makes analyzing the CICIDS2017 dataset still an open issue and there is always a space for improvements in the existing or new machine learning algorithms.

Table 1.1: Network features of CICIDS2017

No.	**Feature label**	**No.**	**Feature label**	**No.**	**Feature label**
1	Destination Port	27	Bwd IAT Mean	53	Average Packet Size
2	Flow Duration	28	Bwd IAT Std	54	Avg Fwd Segment Size
3	Total Fwd Packets	29	Bwd IAT Max	55	Avg Bwd Segment Size
4	Total Backward Packets	30	Bwd IAT Min	56	Fwd Avg Bytes/Bulk
5	Total Length of Fwd Pck	31	Fwd PSH Flags	57	Fwd Avg Packets/Bulk
6	Total Length of Bwd Pck	32	Bwd PSH Flags	58	Fwd Avg Bulk Rate
7	Fwd Packet Length Max	33	Fwd URG Flags	59	Bwd Avg Bytes/Bulk
8	Fwd Packet Length Min	34	Bwd URG Flags	60	Bwd Avg Packets/Bulk
9	Fwd Packet Length Mean	35	Fwd Header Length	61	Bwd Avg Bulk Rate
10	Fwd Packet Length Std	36	Bwd Header Length	62	Subflow Fwd Packets
11	Bwd Packet Length Max	37	Fwd Packets/s	63	Subflow Fwd Bytes
12	Bwd Packet Length Min	38	Bwd Packets/s	64	Subflow Bwd Packets
13	Bwd Packet Length Mean	39	Min Packet Length	65	Subflow Bwd Bytes
14	Bwd Packet Length Std	40	Max Packet Length	66	Init_Win_bytes_forward
15	Flow Bytes/s	41	Packet Length Mean	67	Init_Win_bytes_backward
16	Flow Packets/s	42	Packet Length Std	68	act_data_pkt_fwd
17	Flow IAT Mean	43	Packet Length Variance	69	min_seg_size_fwd
18	Flow IAT Std	44	FIN Flag Count	70	Active Mean
19	Flow IAT Max	45	SYN Flag Count	71	Active Std
20	Flow IAT Min	46	RST Flag Count	72	Active Max
21	Fwd IAT Total	47	PSH Flag Count	73	Active Min
22	Fwd IAT Mean	48	ACK Flag Count	74	Idle Mean
23	Fwd IAT Std	49	URG Flag Count	75	Idle Std
24	Fwd IAT Max	50	CWE Flag Count	76	Idle Max
25	Fwd IAT Min	51	ECE Flag Count	77	Idle Min
26	Bwd IAT Total	52	Down/Up Ratio	78	Label

1.11.2 Description of Attack Scenarios

The CICIDS2017 consists of state-of-the-art attack scenarios based on the most updated list of commonly used attack families, which can be explained as follows.

Table 1.2: Statistics for distribution CICIDS2017 dataset

Class	**Raw**	**Filtered**	**Difference**	**Proportion(%)**
Benign	2273097	1893223	379874	0.167
DoS Hulk	231073	173791	57282	0.247
Port Scan	158930	1956	156974	0.012
DDoS	128027	128020	7	0.000
DoS GoldenEye	10293	10286	7	0.000
FTP-Patator	7938	6093	1845	0.232
SSH-Patator	5897	3360	2537	0.430
DoS Slowloris	5796	5385	411	0.070
DoS Slowhttptest	5499	5242	257	0.046
Botnet	1966	1437	529	0.269
Web-Brute Force	1507	37	1470	0.975
Web-XSS	652	652	0	0.000
Infiltration	36	36	0	0.000
Web-SQL Injection	21	21	0	0.000
Heartbleed	11	11	0	0.000
Total	2830743	2229550	601193	2.448%

- Web Attack: Three web attacks have been implemented in their dataset. First, SQL Injection is an application security vulnerability in which an attacker interferes with the queries that an application makes to its database, to let the unauthorized users view the data. Second, Cross-Site Scripting (XSS) which is happening when the attacker injects malicious code into the victim's web application. Last, Brute Force which tries a probabilistic entire possible passwords to decode the administrator's password.

- Botnet Attack: A collection of internet-connected devices such as a home, office or public systems, contaminated by harmful malicious code called malware. It can enable the attacker access to the device and its connection for stealing, taking down a network and IT environment. Botnets attack are remotely controlled by cybercriminals and have turned into one of the most significant threats to security systems today.

- Heartbleed Attack: is a severe bug in the implementation of OpenSSL, an open-source implementation of the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. This vulnerability allows malicious hackers to read portions and steal data from the memory of the victim server.

- Brute Force Attack: is a dictionary attack method that generates many successive estimates as to access encrypted data. This type of attack is commonly used for cracking passwords, locating the hidden web page or content, and decoding Data Encryption Standard (DES) keys.

- DoS Attack: is a type of cyber attack on a network that is designed to prevent legitimate users temporarily from accessing computer systems, devices, or other network resources due to malicious cyber activities.

- DDoS Attack: is one of the most popular cyber weapons, in which attempt to exhaust the resources available to an online service and network by flooding it with traffic from several compromised systems, deny legitimate users access to the service.

- Infiltration Attack: is a piece of malicious that attempts to enter or damage the inside of the network which is generally manipulating a susceptible software like Adobe Acrobat/Reader.

Chapter 2: Effectiveness of Machine Learning Based Intrusion Detection Systems

2.1 Introduction

Intrusion is an intense problem in security and a prime complication of data breaches, given that a single circumstance of intrusion may steal or even delete information coming from a computer as well as network units in a few seconds. Intrusion can quickly also destroy system equipment. Additionally, the intrusion may trigger significant reductions economically as well as weaken the IT crucial facilities, thereby causing info inferiority in cyberwar. For that reason, intrusion detection is necessary, and also its prevention is required [33]. The appearance of cutting-edge attacks drives the commercial enterprise and academic community to look into for unique approaches, which manage to tightly keep track of this competition and fine-tune rapidly to the transformations in the field [34].

Network security can be attained by employing a software application called an Intrusion Detection Systems (IDS) that helps to withstand network breaches. The objective of these systems is to have a shield wall that prevents such types of attacks. It identifies the illegal activities of a network or a computer system. Generally, there are two major categories of IDS, namely Anomaly detection and Misuse detection. The former learns from recorded normal behavior to identify new intrusion attacks. Any variance from existing baseline patterns is determined as attacks and alarms are triggered. Nevertheless, misuse detection detects the intrusion based on the repository of attack signatures but has no false alarm.

Machine learning approaches have been extensively utilized in determining different sorts of attacks, which is a powerful tool to enhance network security. In addition, it can assist the network's monitoring team in taking the necessary countermeasures for protecting against intrusions.

In this chapter[1], we utilize the public real-world intrusion dataset CICIDS2017 [31], which includes benign and the most sophisticated attacks and presenting results of seven machine learning classifiers, such as AdaBoost [36], Naive-Bayes (NB) [37], Random Forest(RF) [38], Decision Tree [39], Multi-layer perceptron (MLP) [40], K-Nearest Neighbors (KNN) [41], and Quadratic Discriminant Analysis (QDA) [42].

The main contributions of this chapter at hand are as follows:

- Presenting a discussion of various existing literature studies for building an IDS using different machine learning classifiers, emphasizing on the detection mechanism, applied feature selection, and attack detection efficiency.

- We examine the CICIDS2017 dataset that includes benign and the most cutting-edge common attacks. We also carried out various machine learning algorithms to analyze the detection performance of IDS.

- We extensively evaluate our system over different performance metrics such as accuracy, precision, recall, and F1-score, training, and prediction time.

The structure of this chapter is described next. Section 2.2 presents a literature review of the related works that only uses the same CICIDS2017 dataset for intrusion detection. Section 2.3 introduces the implemented dataset in detail with the explanation of the attack scenarios. Section 2.4 gives a brief overview of machine learning classifiers. Section 2.5 discusses the performance results of the classifiers over different evaluation metrics. Finally, the conclusion to our work is given in Section 2.6.

[1] The content of this chapter was published in [35], 2019. Permission is included in Appendix A.

2.2 Related Works

Over the last few years attempts to attacks on determining sizable data have revved up. In this part, different research studies employing machine learning for intrusions detection have been analyzed. In each research study, the applied machine learning algorithms and system performance are provided. When selecting these research studies, the focus was on the ones that used different machine learning algorithms on the CICIDS2017 dataset.

In reference [31], they proposed a new dataset named as the CICIDS2017. Their IDS experiments were performed over seven well-known machine learning classifiers, namely AdaBoost, Random Forest, Naive Bayes, ID3, MLP, KNN, and QDA. They claim that the highest accuracy was achieved by KNN, RF, and ID3 algorithms, but this work was lack of accuracy rate results.

In another study [43], a hybrid IDS using the CICIDS2017 dataset is proposed, which combines the classifier model based on tree-based algorithms, namely REP Tree, JRip algorithm, and Random Forest. They claim that their proposed system experimental results prove superiority supremacy in terms of false alarm rate, detection rate, accuracy, and time overhead as compared to the existing state-of-the-art schemes. The results obtained show that their system was able to detect different attacks with an accuracy rate of 96.665%

The authors in [44] describe and optimize the CICIDS2017 dataset using Principal Component Analysis (PCA), which results in dimensionality reduction without losing specificity and sensitivity. Hence, decreasing the overall size and bring on faster IDS. This work has been employed on the recorded data of Friday and Thursday, which targeted various attacks, namely DDoS, botnet, port-scan, web attacks, and infiltration. The dataset is examined using three classifiers, including KNN, C4.5, and Naive Bayes. The highest detection rate for DDoS was achieved by Naive Bayes, and KNN classifiers are 90.6% and 99%, respectively. As a result, Naive Bayes has an elevated false alarm rate of 59%, which in turn classifies

KNN with a false alarm rate of 1.9% as a sufficient classifier for a DDoS attack. The number of attributes had notably been lowered, roughly by 75%, of the total attributes number.

In [45], the authors have proposed a machine learning Multi-Layer Hidden Markov (HMM) model-based intrusion detection. This multi-layer approach factors a substantial issue of large dimensionality to a discrete set of reliable and controllable elements. Moreover, it can be broadened further than two layers to capture multi-phase attacks over long periods of time. The portion of Thursday morning records in the CICIDS2017 dataset was used, which comprises of brute force, SSH Patator, and benign traffic. The proposed system reveals an excellent performance among all evaluation metrics as 98.98% accuracy, 97.93% precision, 100% recall, and 98.95% F-measure.

Reference [46] outlines an IDS using supervised learning techniques and the Fisher Score feature selection algorithm on the CICIDS2017 dataset for benign and DDoS attacks. Their work was performed on Support Vector Machine, Decision Tree, and K-Nearest Neighbours machine learning algorithms. The performance measurements show that the KNN performed much better outcomes with 30 features; the examination scores did not change for the Decision Tree algorithm. Alternatively, SVM's outcomes did not fulfill both 80 and 30 features. After using the Fisher Score feature selection, the dataset was reduced by 60%. As an accuracy outcome of this study, 0.9997% KNN, 0.5776% SVM, 0.99% DT accomplished when selecting 30 features.

Authors [47] presented a machine learning approach based on DDoS attack detection via NetFlow analysis. Different machine learning classification algorithms were primarily evaluated, namely C4.5 Decision Tree, Random Forest, AdaBoost, and Support Vector Machines against their NetFlow collected data. This DDoS detection approach was secondarily evaluated by using public dataset CICIDS2017 to prove its validity. The experiment consequences indicate that this approach obtains an average accuracy of 97.4% and a false positive 1.7%.

The authors of [48] have proposed an intrusion detection approach, named XGBoost. In the study, the relevant system created by employing the Wednesday recorded dataset that consists of various sorts of DoS attacks from the CICIDS2017. The classification of DoS attacks obtained an accuracy of 99.54%.

In the relevant works, it is witnessed that research studies employing the same dataset are presenting excellent results. However, when the research studies examined, it is observed that most of the authors partially used the CICIDS2017 dataset to build the IDS models, which therefore indicates that their IDS are only exposed to some of the attacks in the subject dataset.

2.3 Data Preprocessing and Analysis

The process of analyzing any given dataset to develop an IDS should certainly involve understanding the dataset in hand, cleaning, then carrying out some powerful statistical methods that assure achieving the study's goals, along with their predetermined performance metrics. This section shows the evaluation metrics along with the process of analyzing and preprocessing the CICIDS2017 dataset.

2.3.1 Benchmark Dataset

CICIDS2017 Dataset [31] generated by the Canadian Institute for Cybersecurity at the University of New Brunswick. Benign and the most sophisticated widespread attacks, for instance, real-world data (PCAPs), are featured in the CICIDS2017 dataset. This dataset includes five days records stream on a network generated by computer systems using updated operating systems (OS), which provides for Windows Vista/ 7/ 8.1/ 10, Mac, Ubuntu 12/16, and Kali. Monday records consist of benign traffic. The employed attacks are Brute Force SSH, Brute Force FTP, Infiltration, Heartbleed, Web Attack, DoS, Botnet, and DDoS.

The formerly available network traffic datasets suffer from the absence of traffic diversity, volumes, anonymized packet information payload, constraints on the range of the attack, the lack of the feature set and metadata. Therefore, the CICIDS2017 came to conquer these concerns like different protocols, including HTTP, HTTPS, FTP, SSH, and also e-mail protocols, which in turn were not offered in the dataset previously.

2.3.2 Evaluation Metrics

Our work subject to different evaluation metrics, which are accuracy, precision, recall, F1-score, training time, and prediction time. Since achieving the supreme accuracy does not essentially signify that the classifier properly predicts with high reliability. As a result, we utilize other strategies to examine the reliability of the proposed system results. Table 2.1 shows the description of confusion matrix.

Table 2.1: Confusion matrix

		Predicted Class	
		Classified as Normal	**Classified as Attack**
Actual Class	**Normal**	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

The evaluation metrics are specified based on the following explanations:

- True Positive (TP): describes the number of attacks correctly detected.
- True Negative (TN): describes the number of normal correctly detected.
- False Positive (FP): describes the number of normal wrongly detected.
- False Negative (FN): describes the number of attacks wrongly detected.

Afterward, we calculate the evaluation metrics from the following formulas as follows.

- Precision: the proportion of correctly predicted attack relative to all data classified as the attack.

$$Pr = \frac{TP}{TP + FP}, \tag{2.1}$$

- Accuracy: the proportion of records are correctly determined as attack and normal.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.2}$$

- F1-Score: a combination that measures the harmonic average of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{2.3}$$

- Recall: indicating the proportion of correctly predicted attack to all attack data.

$$Rc = \frac{TP}{TP + FN}, \tag{2.4}$$

- Training time: represents the time consumed for a particular algorithm to train the model for the entire dataset.
- Prediction time: represents the time consumed for a particular algorithm to predict the entire dataset as benign or attack.

2.3.3 Data Cleansing

We observed that the CICIDS2017 dataset includes some significant pitfalls which cause the classifier to be biased, and the goal of this study is to address those imperfections and apply machine learning classification properly to make more accurate results. It might be an essential step to make some modifications to the dataset employing it in practice, rendering it more reliable. For this purpose, in this part, some pitfalls of the CICIDS2017 dataset are remedied, and some data are modified. The dataset contains 2830743 records and 86 features. The updated distribution of this dataset can be shown in Table 1.2. When we examine these records, it can be noticed that 601193 are faulty records. The first step in the data pre-processing will be to remove these undesirable records.

An additional change that requires to be made in the dataset is that we remove all rows with features "Flow Bytes/s" and "Flow Packets/s" that have either "Infinity" or "NaN" values. Furthermore, we remarked that some features have zero values for all rows, namely Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, and Bwd Avg Bulk Rate; hence, they are also excluded.

We noticed that the attack label with small counts still maintains that count before and after cleaning the data. By looking at the proportion column, a tiny proportion of each attack type was deleted during the data cleaning process. Lastly, the first column "Destination Port" is also excluded, even though when it was included, we noticed an improvement in the performance of the classifiers. Therefore, the data size used for the analysis is 2230983 records by 69 features. After the removal of these features, the dataset is randomly split into two parts, 70% was used for training, and 30% was used for testing the model, in order to evaluate their performance in the intrusion detection system.

2.3.4 Random Forest Feature Selection

Since the main purpose of applying feature selection technique is to eliminate irrelevant or redundant features from a high dimensional dataset. The Random Forest feature importance was used to measure and rank the features based on their importance. Then, we select the most efficient 10, 30 features that can distinguish the information in the most significant way.

2.4 Overview of Machine Learning Algorithms

This section presents an overview of the utilized machine learning approaches.

Adaptive Boosting (AdaBoost): a boosting approach, is a machine learning algorithm designed to enhance classification efficiency. The fundamental working concept of boosting algorithms can be described as follows; the data are initially sorted into groups with rough draft rules. On any occasion the algorithm is run, new rules are contributed to this rough draft rules. In this manner, several feeble and low-performance rules called ”basic rules” are acquired.

Multi-Layer Perceptron (MLP): is a category of artificial neural networks (ANN). ANN is a machine learning technique that takes motivation from the method the human brain works. The objective of this approach is to mimic the human brain properties, for instance, making decisions and obtaining new information. While the human brain is comprised of interconnected nerve cells, ANN is comprised of interconnected artificial cells.

Decision Tree (DT): is the most potent tool in classification and prediction. A Decision Tree is a flow diagram such as tree structure, where each tree includes leaves, branches, and nodes. It divides a dataset into scaled-down subsets while simultaneously an associated decision tree is incrementally formed. The final outcome is a tree with leaf nodes and decision nodes.

Naive Bayes (NB): is a family of probabilistic classification techniques that benefits from probability theory and the Bayes' Theorem for predictive modeling, which presumes that all attributes are statistically independent. It computes the probabilities for each factor in order to single out the result that has the highest probability.

K-Nearest Neighbors (KNN): is a versatile and sample-based method. It depends on in which the data points are separated into multiple classes; in other words, similar things are near to each other, in order to determine the K-nearest neighbors.

Quadratic Discriminant Analysis (QDA): is a discriminant analysis method that is utilized to identify which variables differentiate between two or more naturally taking place groups; it may have a predictive or a descriptive goal.

Random Forest (RF): is a machine learning approach that utilizes decision trees. Herein method, a ”forest” is produced by putting together a substantial number of various decision tree structures that are created in various ways.

2.5 Performance Analysis

The implementation of data preprocessing, feature selection, and classification were coded using Python programming language. The experimental results of machine learning classifiers are given in Table 2.2. Based on the values of precision, recall, and F1-Score, the KNN has the best performance among other classifiers, followed by the MLP and Random Forest classifiers. Then, the performance of the Decision Tree, AdaBoost, and Naive Bayes is ranked as fourth, fifth, and sixth, respectively. The QDA algorithm has the lowest performance results.

Table 2.2: Performance examination results of machine learning algorithms

Algorithm	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Training(s)**	**Prediction(s)**
Random Forest	0.946	0.957	0.948	0.957	348.6	5.80
KNN	0.995	0.995	0.995	0.995	2590.6	1358.1
Naive Bayes	0.795	0.848	0.779	0.848	4.60	7.70
Decision Tree	0.882	0.904	0.892	0.904	19.90	0.20
MLP	0.964	0.970	0.966	0.970	103.7	1.10
AdaBoost	0.857	0.917	0.885	0.917	607.6	15.50
QDA	0.720	0.848	0.779	0.848	15.20	10.00

The training and predicting times were also computed during the process, and given by Table 2.2. It can be noted that the KNN algorithm requires significantly more time during the training and testing process; this could be a drawback of the classifier as it memorizes all the training flows. Naive Bayes has the lowest training and predicting times among other classifiers, but, as mentioned earlier, it performed as a second-worst classifier on the CICIDS2017 dataset. Thus, it is a trade-off between performance and prediction time. On the other hand, the MLP classifier has a good balance between its performance and the prediction time.

Since the total number of features after the data cleaning process is 68, the feature importance based on the Random Forest classifier was computed, which helped to rank the 10 and 30 most important features, respectively. The subject machine learning classifiers were carried out on the reduced CICIDS2017 dataset, and the results are given by Table 2.3. The results indicate similar performance consistency of the classifiers when using only 10 and 30 most important features, respectively. Nevertheless, the performance of the classifiers was higher when considering all the 68 features.

Experimental results have demonstrated that the K-Nearest Neighbors (KNN) classifier found to be the best performer among all four evaluation metrics. However, it has the longest training and predicting times. The MLP algorithm achieved the second-highest performance, and it maintained reasonably small training and prediction times. The chosen machine learning classifiers excluding KNN have trained their models in a reasonable time. The feature selection based on the random forest classifier did not support the classifiers to perform better compared to the usage of all features after the data cleansing process. There is no significant difference in the performance of the Naive Bayes and QDA classifiers based on the evaluation metrics, where both have the worst overall performance, regardless of their small training and predicting times.

Table 2.3: Experimental results of the selected features

Algorithm	**Selected features**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest	30	0.9395	0.9484	0.9382	0.9485
	10	0.9287	0.9401	0.9283	0.9401
KNN	30	0.9944	0.9945	0.9941	0.9946
	10	0.9690	0.9675	0.9675	0.9676
Naive Bayes	30	0.7958	0.8487	0.7794	0.8488
	10	0.7204	0.8488	0.7794	0.8488
Decision Tree	30	0.8816	0.9025	0.8907	0.9026
	10	0.9282	0.9417	0.9305	0.9418
MLP	30	0.9536	0.9625	0.9557	0.9626
	10	0.9347	0.9460	0.9356	0.9460
AdaBoost	30	0.8578	0.9173	0.8854	0.9173
	10	0.8692	0.8901	0.8576	0.8901
QDA	30	0.7204	0.8488	0.7794	0.8488
	10	0.7204	0.8488	0.7794	0.8488

2.6 Summary

In this chapter, several IDS experiments were carried out to examine the efficiency of seven machine learning classifiers, namely AdaBoost, Random Forest, Naive Bayes, Decision Tree, MLP, KNN, and QDA. We use a publicly available intrusion detection evaluation dataset (CICIDS2017), which includes benign and most sophisticated popular attacks. The experimental results attest to the superiority of the K-Nearest Neighbors (KNN) classifier in terms of various performance metrics such as precision, recall, accuracy, and F1-score, among other machine learning algorithms. However, all of the selected machine learning classifiers, excluding KNN, trained their models in an acceptable time.

Chapter 3: A Heterogeneous Ensemble Machine Learning Approach for Cyber Attack Detection

3.1 Introduction

Intrusion Detection Systems (IDS) have been identified as a type of system that widely used strategy for uncovering and refuting malicious undertakings in networks. As the number of malicious attacks is continuously rising, IDS are much compelled to handle the prevention of such attacks ahead of they trigger extensive devastation. Furthermore, today's rise of the Internet of Things (IoT) gadgets and solutions have indeed extremely transformed our daily life. A plurality of applications based upon advanced IoT solutions is effectively constructed and implemented, which includes smart city, smart healthcare, smart home, and also automotive networks [49], these systems open more opportunities for malicious attackers. As stated in [50], security is a significant obstacle to the application of IoT networks and services, this is because IoT uses diverse requirements and protocols, thus creating heterogeneous networks.

The research in the intrusion detection domain has been primarily concentrated on anomaly-based and misuse-based detection techniques for a long time. While misuse-based detection is generally preferred in industrial product lines because of its predictability and high accuracy, in academia, anomaly detection is usually conceived as a much more effective approach due to its theoretical possibility in addressing unique invasions [51].

Nowadays, ensemble learning techniques have obtained growing attention in the field of predictive modeling [52]. It is a reliable method that incorporates various learning algorithms, so in order to enhance the overall prediction accuracy. The ensemble approach deals with the idea that a team of experts offers more precise decisions than a single expert. Thus, ensemble modeling incorporates the collection of classifiers to develop a single model to improve the performance of machine learning [53].

An ensemble is an efficient approach that integrates different learning algorithms, so in order to gain better prediction accuracy [54]. Ensemble methods emerged as a popular method from the past two decades in the spot of classification and prediction. Ensemble techniques have been successfully used in numerous fields. An Ensemble puts together the predictions of individually trained classifiers when classifying unique instances.

This chapter introduce a heterogeneous ensemble method based on Decision Tree [55], Random Forest [38], and Cost-sensitive classifier [56]. For this intention, we employ J48, RF, and CSForest algorithms and combine their prediction by making use of the average voting rule. The main objective of the proposed network intrusion detection mechanism to identify different types of attacks. Our method makes use of an ensemble classifier mixed with a nature-inspired feature selection technique based on the bats' behavior to reduce the bias of a machine learning model as well as the computational complexity. The feasibility and effectiveness of the suggested model are investigated under several statistical metrics such as accuracy, precision, recall, f-measure, and false alarm rate.

The major contributions in this chapter are summarized below:

- We propose a machine learning approach that integrates the benefits of feature selection and ensemble classifiers intending to offer a highly effective and precise intrusion detection system.

- We utilize the CFS-BA as a feature selection technique to reduce the dimension of feature vectors and time cost. This method comprises of Correlation-based Feature

Selection (CFS) and Bat Algorithm (BA), bearing in mind the fact that not all features are considered significant or perhaps pertinent in determining attacks.

- To further enhance the performance of classification, an ensemble model based on assembling several learning algorithms (C4.5, RF, and CSForest) is implemented. Then, the average of probabilities (AOP) combination rule is incorporated into the model for the decision step.
- We perform extensive experimental evaluations by making use of the latest public intrusion detection dataset, which demonstrates a promising performance gain on all statistical metrics when compared to various state-of-the-art methods. The experimental results show that the CFS-BA ensemble technique achieves the highest detection accuracy with 94.8%, lower false alarm rate 2.1%, and training and testing time decreased when feature selection is deployed before classification.

The structure of this chapter is described next. The organization of this chapter is as follows. Section 3.2 dedicated to briefly review relevant work, Section 3.3 describes proposed methodology used in our research study. Section 3.4 and 3.5 focuses on the experimental setup and results discussion. The last section concludes the chapter by the summary of contributions.

3.2 Related Works

Nowadays, ensemble learning algorithms are extensively applied in the research areas of image processing, finance, medicine, and biology [57–60]. The application of ensemble approaches for building intrusion detection systems is likewise a hot topic over the last few years [61], due to the fact that the use of several weak classifiers assists in lowering the overfitting and conquering the imperfections of single classifiers. Also, in the case of network traffic, which varies in nature, using several classifiers serves to help in pointing out a wide range of attacks, which in turn would be challenging for an individual algorithm. Several studies have introduced the benefits of ensemble learning approaches versus the individual base learner with regards to detection performance and false alarm rate curtailment. Designing an optimum machine learning based-detection systems routes research study to investigate the efficiency of various intrusion detection models built by ensemble machine learning techniques. Numerous algorithms carried out by scholars in the field of feature selection and classification are explained below.

Authors [62] introduced an FVBRM technique that utilizes a method based on feature vitality with three feature selection methods, namely correlation-based feature selection, gain ratio, and information gain. After that, apply an efficient naive Bayes classifier on reduced datasets to identify important reduced input features for an IDS systems. As compared to the single Naive Bayes classifier, this application provides a high detection rate. Nevertheless, the constraint of this technique is that more false positives are created.

In reference to [63], they proposed an ensemble method for intrusion detection utilizing the AdaBoost algorithm, which integrates the remedy of the following classifiers: SVM, KNN, decision tree, multilayer perceptron(MLP), and Naive Bayes. The AdaBoost algorithm initializes the distribution of data, trains the classifiers, assesses errors, and also designates weights to each of them. After that, the combination of classifiers is linear and based upon a weighted voting technique.

Study by [64] used stacking, boosting, and bagging ensemble approaches to the intrusion detection to improve the accuracy and minimize its false positive rate. As base classifiers for these ensemble approaches, they employed Naive Bayes, decision tree, JRip, and also KNN. The use of stacking, boosting, and bagging revealed no substantial gain in accuracy. While stacking was the only technique that led to a considerable decrease in false positive rates, it also has the longest execution time and therefore is as well ineffective to be practical for the intrusion detection.

In [65], the authors have implemented an ensemble distributed classifier for network IDS based upon a new tree-level method for combining the individual classifiers' decisions. The approach relies on utilizing ensembles of neural networks designed through genetic programming-based ensembling (GPEN). GPEN automatically develops a program utilizing genetic programming operators to show how to integrate the component networks' predictions to get a reliable ensemble prediction. This research study varies from others dealing with the traditional ensemble, considering that it offers the partial acquiring of adaptive results by distinct classifiers denied of an ensemble classifier.

The authors in [66] have presented the use of ensemble learning by employing a Random Tree and Bayesian Network as base classifiers. These algorithms were combined with meta-learning algorithms putting to use Random Committee. After that, voting was executed for the classification process. In this work, the authors stated that the KDDcup99 dataset was utilized. Among one of the major purposes in this work was dealing with the out of balance nature of this dataset applying ensemble learning. The model is assessed using receiver operating characteristic (ROC) curves. The authors also have computed the area under the ROC curves (AUC). In the results, it was obtained that the ensemble model outmatches the individual-based models.

The authors of [67] have proposed a model that utilized different feature selection methods to remove the irrelevant attributes in the dataset and built a classifier that is much more sturdy and also effective. The methods that were utilized combined with classifiers are Information Gain, Correlation, Relief, and Symmetrical Uncertainty. Their empirical work was divided into two portions: The initial one is constructing a multiclass classifier based upon different decision tree techniques including ID3, CART, REP Tree, REP Tree as well as C4.5. The further one is using the feature selection technique on the very best system obtained, which was the C4.5 method. Their experiential evaluation was conducted utilizing the weka toolbox.

Authors [68] evaluated the performance of the initial data pre-processing impact on attack detection accuracy by utilizing of ensemble approach, that relies on the idea of combining multiple weaker learners to develop a stronger learner model of four varied classifiers, namely J48, C5.0, Naive Bayes and PART. Min-Max normalization and Z-Score standardization were applied in the pre-processing stage. They compared their suggested model with as well as without pre-processing techniques utilizing greater than one classifier. Their results showed that their classifier ensemble model generates even more accurate results.

From this inspiration, we aim to construct an intrusion detection model by integrating correlation-based feature selection with the ensemble of C4.5, RF, and CSForest machine learning algorithms. In this work, our proposed method is designed to increase efficiency and minimize the false alarm rate by selecting the most relevant features subset with optimal output performance.

3.3 Proposed Methodology

This section presents the details of the proposed methodology. In the first step, we deploy the preprocessed CICIDS2017 dataset by applying data filtration, creating a well-balanced dataset, and normalization methods. In the second step, CFS with a Bat search algorithm was applied during the feature selection process in order to evaluate the correlation of the selected features and beneficial in terms of optimizing overall prediction accuracy. In the third step, ensemble classifiers (Decision Tree, Random Forest, and CSForest) are trained with the train set. Figure.3.1 shows an overview of the proposed ensemble intrusion detection framework followed during this work.

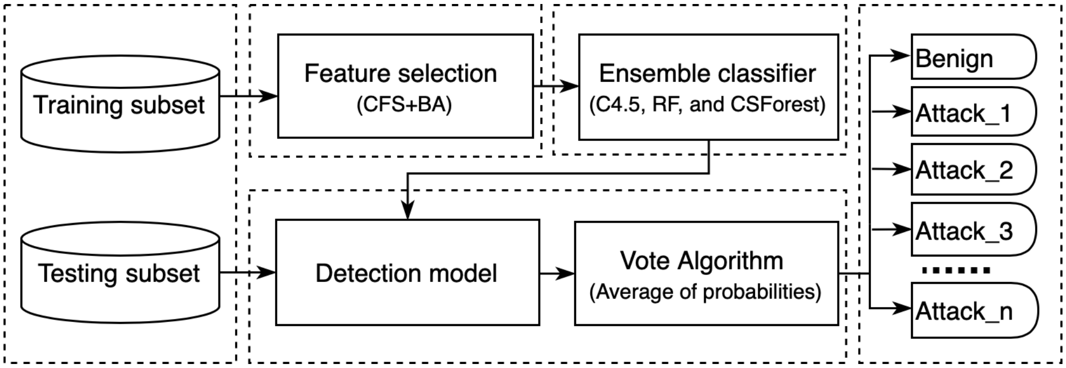


Figure 3.1: Construction process of the ensemble classification

3.3.1 Dataset Preprocessing

Data preprocessing is a crucial step in classification for substantial amounts of data. Redundant information needs to be eliminated, as discussed in section 2.3.3, and normalization constructs a piece of well-balanced dataset throughout preprocessing.

Table 3.1: Training and testing subsets distribution

Label	**Raw**	**Filtered**	**Train**	**Test**
Benign	2273097	1893223	20000	20000
DDoS	128027	128020	2700	3300
DoS Slowloris	5796	5385	1350	1650
DoS Slowhttptest	5499	5242	2171	1169
DoS Hulk	231073	173791	4500	5500
DoS GoldenEye	10293	10286	1300	700
Heartbleed	11	11	5	5
PortScan	158930	1956	3808	4192
Botnet	1966	1437	936	624
FTP-Patator	7938	6093	900	1100
SSH-Patator	5897	3360	900	1100
Web-Brute Force	1507	37	910	490
Web-XSS	652	652	480	160
Web-SQL Injection	21	21	16	4
Infiltration	36	36	24	6
Total Attack	471454	336327	20000	20000
Total	2830743	2229550	40000	40000

3.3.1.1 Balanced Dataset

To create a balanced dataset, we obtain both training and testing subsets based upon the data distribution presented in Table 3.1. In each subset, we attempted to involve rows that comprise of all the attacks, but the identical row can not show up in both of these subsets. For the training subset, we select the first rows of each kind. Then, For the testing subset, we single out randomly the rows after the suppressing of the training subset rows [69].

3.3.1.2 Min-Max Normalization

The numerical values in the dataset is of different ranges, which postures several obstacles to the classifier throughout training to compensate these distinctions. Hence, it is essential to normalize the values for each feature, to make all data points scaled within the range of [0, 1]. This method gives more uniform values to the classifier while sustaining relevancy amongst the values of each feature. Each feature value should be normalized as follows.

$$\overline{x} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{3.1}$$

whereas x_{min} and x_{max} represent the initial minimum and maximum values of feature x, and $\overline{x}$ the normalized feature value in the range [0,1].

3.3.2 Feature Selection

Feature Selection (FS) can be employed to eliminate the irrelevant and redundant attributes from the high dimensional attributes. The process of feature selection plays an indispensable role in data pre-processing step in classification procedures as it enhances the quality of data and, as a result, strengthens the predictive efficiency of the prediction models.

To select the optimal feature from the CICIDS2017 intrusion dataset, we implement subset evaluation (CfsSubsetEval) with the bat search algorithm using weka toolbox.

3.3.2.1 Correlation-based Feature Selection

Correlation-based Feature Selection (CFS) [70] is a well-known filter technique that select attributes according heuristic (correlation-based) function. The predilection of this function is to select subgroups that contain attributes extremely associated with the class, but uncorrelated with one another. Unessential attributes should be ignored because they will have a low relationship with the class. In contrast, recurring attributes are screened out as they will be exceptionally connected with at least one of the rest of the features. The recommendation of an attribute will rely upon the level to which it predicts classes in territories of the instance space not already predicted by different attribute [71]. CFS's feature subset evaluation function [72] is as:

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k\left(k-1\right)\overline{r_{ff}}}}, \tag{3.2}$$

where M_s denotes the heuristic merits of a feature subset S with k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f{\in}S$), and $\overline{r_{ff}}$ is average inter-correlation value of feature-feature.

3.3.2.2 Bat Algorithm

The Bat Algorithm (BA) heuristic search method was developed based on the echolocation behavior of bats [73]. This nature-inspired algorithm has diverse applications. It can be applied for classification process [74], optimization problems such as single-objective optimization and multi-objective optimization [75], for data prediction, and so forth. Bat algorithm imitates the manner when the bat looks for its prey based on the echolocation strategy. Using echolocation, the bat alters its path and speed based on the sound that strikes back after contacting the target. It updates its speed arbitrarily to reach its prey in the fastest span. Previous research studies expose that bat algorithm outperforms both the genetic algorithm (GA) and particle swarm optimization (PSO) in offering solutions to the unconstrained optimization issues [76].

Each bat fly randomly with velocity (v_i) at position in space (x_i). These parameters are computed based on a fixed sound pulse frequency (f_{min}) at iteration (t), loudness (A_i), and varying wavelength (λ) in space during the search for prey (or optimal solutions), their updating rules can be written as.

$$\begin{aligned} f_i &= f_{\min} + (f_{\max} - f_{\min})\beta \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_*)f_i \\ x_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \tag{3.3}$$

where $\beta \in [0,\ 1]$ is a random vector drawn from a uniform distribution. For each iteration (t) in BA, the loudness (A_i) and pulse emission rate (r_i) are adjusted using the updating rules by the following equations:

$$\begin{aligned} A_i^{t+1} &\propto A_i^t \\ r_i^{t+1} &= r_i^0.[1 - e^{-\gamma t}], \end{aligned} \tag{3.4}$$

Similar to other types of meta-heuristic algorithms, random walk is implemented in the Bat Algorithm [75], which in turn would improve the variability of the possible solutions. Predominantly, BA selects one solution amongst the most reliable bats, and then a new solution for each bat is generated locally using a random walk.

$$x_{new} = x_{old} + \varepsilon A^t, \tag{3.5}$$

where $\varepsilon \in$ [-1,1] stands for a random number drawn from a uniform distribution, while A^t is the average loudness of all bats at time t.

3.3.2.3 CFS-BA Algorithm

In this segment, we propose the CFS-BA approach filter based attribute selection method, which is applied to assess the relevancy and redundancy of the selected subset of features. For a feature subset M with n features, $M = (m_1, m_2, m_3 \dots m_n)$, CFS examines the average inter-correlation amongst features and the mean feature-class correlation by making use of Eq.(1) for the sake of increasing classification accuracy and reducing overfitting. Even though CFS can conveniently select the subset of separately desirable features, this subset of features may not be the most efficient combination due to the redundancy in between features. The elimination of redundant features produces impacts on applications, for instance, accelerating a data ML algorithm, enhancing learning accuracy, and resulting in better model comprehensibility. In order to reduce the dimensionality of network data and remove its redundant features, the Bat Algorithm (BA), which inspired by the fascinating capability of microbats was utilized. In BA, every candidate solution of the problem is denoted by the bat's position, which can be represented as a vector of binary coordinates.

3.3.3 Ensemble Classification

The ensemble learning system integrates the group of classifiers to develop a singular composite model that provides much better accuracy. Ensemble approaches can be described as a committee, classifier fusion, combination or aggregation, etc. Research study reveals that prediction from a compound model offers far better results as contrasted to a single model prediction. Ensemble techniques can be classified as Heterogeneous and Homogeneous ensemble approaches. Heterogeneous ensemble approaches employ an assortment of learning algorithms and also manipulate training datasets to make numerous models. A few of the Heterogeneous techniques are stacking, voting, and so on. While Homogeneous ensemble approaches work with a single learning algorithm on various training datasets to build a wide range of classifiers such as Bagging, Boosting and Random Forest and a lot more [77].

3.3.3.1 Decision Tree

A Decision Tree C4.5 (J48) [55] is a tree-like framework in which each inner node serves as a decision based upon a feature value. C4.5 belongs to the most preferred algorithms for creating decision trees. In constructing the tree, instances are split right into small-scale subsets by a feature with the highest info gain. The splitting actions recursively continue on each subset until all the instances in a subset characterize the same class.

3.3.3.2 Random Forest

Random Forest (RF) [38] is primarily thought to be the ultimate "off-the-shelf" classifiers for high-dimensional data. Random forest is a combination of tree predictors such that each tree relies on the values of a random vector sampled autonomously and with the same apportionment for all trees in the forest. The generalization error of forests assembles to restrict the number of trees in the forest ends up being huge, which also counts on the strength of the specific trees in the forest and also the association between them. Various subsets of the training data are chosen, with replacement, to train each tree. The residual of training data is employed to error evaluate and variable significance. Class assignment is made by a variety of votes from the whole trees, and for regression, the average of the results is utilized [78].

3.3.3.3 Cost Sensitive Decision Forest

A Cost-Sensitive Decision Forest (CSForest) classification algorithm was introduced in [56], which acquires the benefits of conventional decision tree algorithm, and additionally conquers the traditional decision tree problem for ignoring numerous costs occurring from the high cost of classification [79]. This method can compute the classification cost via cost-sensitive voting based upon the positive and negative costs of labeling [80].

3.3.3.4 Voting Algorithm

Voting is a meta-algorithm that performs the decision procedure by combining several individual classifiers and analyzing their output using combination rules. This algorithm has several combination rules, such as the average of probabilities, multiplication of probabilities, minimum probability, maximum probability, and majority voting. In the average of probabilities (AOP) approach [81], the class label can be determined based upon the maximum value of the average of predicted probabilities. C4.5, RF, and CSForest classification techniques are combined via the Voting scheme available on Weka for the classification of the samples.

3.4 Experimentation

In this work, an ensemble machine learning-based network intrusion detection system is proposed and evaluated. The first step was data preprocessing, followed by dimensionality reduction using feature selection techniques, building and training ensemble tree-based, and finally, attack recognition. The CICIDS2017 dataset was chosen to build the ensemble machine learning classifiers using the Weka toolbox. In this research study, the concentration is primarily on the assessment of the ensemble classification model, as the evaluation of individual machine learning classifiers has already been carried out in our former work [35]. The comprehensive implementation and examination of this model conducted in four primary stages. As demonstrated in Figure 3.1, it consisted of feature selection, model building, attack recognition, and evaluation of the ensemble proposed approach.

3.4.1 Weka Environment

Waikato Environment for Knowledge Analysis (Weka) [82] was selected for this study, which is an open-source software providing a collection of state-of-the-art machine learning algorithms and pre-processing data tools. Weka toolkit supports various data analysis tasks such as data cleaning, classification, regression, clustering, visualization, and feature selection. All the experiments from the process of selecting features to the classification process were performed on Weka 3.8.3.

3.4.2 Evaluation Metrics

We adopted two sets of popular performance metrics to examine the classification performance of employed IDS. The first set is comprised of true positive rate (TPR), false positive rate (FPR), precision, recall, f-measure, Matthews correlation coefficient (MCC), and receiver operating characteristic (ROC) for benign traffic or other fourteen attacks. The second set contains correctly classified instances (CC), incorrectly classified instances (IC), training, and prediction time. These metrics are calculated using a confusion matrix, which offers four measures as shown in Table 2.1.

3.5 Performance Analysis

This section discusses the results of the proposed ensemble model. Comparisons of the ensemble IDS approach are provided with each other and with different models. To carry out different parts of the developed system, we have used Python for the above-mentioned data preprocessing. Then, the used filters and classification have been implemented in an updated version of Weka environment (3.8.3). The experiment was conducted using the CICIDS2017 dataset. At the outset, important attributes were determined by employing the proposed correlation-based feature selection evaluator with a bat algorithm to examine the stability of the minimized attribute subset in the feature selection phase. In general, 16 features were chosen from the 70 original features for the subsequent phase. Table 3.2 reveals the names of selected features and index numbers. By using the proposed CFS-Bat algorithm individually, the technique was observed to lower the dimensionality significantly and eliminate the unimportant attributes of the CICIDS2017 dataset.

Table 3.2: Selected features for CICIDS2017 dataset

Feature No.	Feature Name	Feature No.	Feature Name
1	Destination Port	37	Min Packet Length
7	Fwd Packet Length Max	38	Max Packet Length
8	Fwd Packet Length Min	40	Packet Length Std
13	Bwd Packet Length Mean	52	Avg Fwd Segment Size
14	Bwd Packet Length Std	53	Avg Bwd Segment Size
18	Flow IAT Std	58	Init_Win_bytes_forward
25	Flow IAT Min	59	Init_Win_bytes_backward
36	Bwd Packets/s	66	Idle Mean

Furthermore, to considerably enhance the classification efficiency of the proposed network intrusion detection model, an ensemble classifier that consists of several decision tree classifiers was utilized in a voting algorithm. The experimental results of different evaluation metrics are shown in Table 3.3. As per the results obtained, it was seen that both TPR and FPR achieved satisfactory results in most of the classifications. However, SQL Injection performance was low at most of the evaluation metrics such as TPR, FPR, precision, recall, f-measure, and MCC caused by a relatively small proportion of this attack in the entire dataset.

Table 3.3: Classification results of proposed method

Class	TPR	FPR	Precision	Recall	F-Measure	MCC	ROC
Benign	0.969	0.390	0.961	0.969	0.965	0.930	0.989
DDoS	0.995	0.000	0.996	0.995	0.995	0.995	1.000
DoS Slowloris	0.922	0.500	0.882	0.922	0.902	0.898	0.996
DoS Slowhttptest	0.769	0.020	0.904	0.769	0.831	0.829	0.993
DoS Hulk	0.972	0.010	0.991	0.972	0.982	0.979	0.999
DoS GoldenEye	0.597	0.003	0.783	0.597	0.677	0.679	0.893
Heartbleed	1.000	0.000	1.000	1.000	1.000	1.000	1.000
Port Scan	0.997	0.008	0.936	0.997	0.965	0.962	0.997
Botnet	0.492	0.006	0.573	0.492	0.529	0.524	0.975
FTP-Patator	0.997	0.000	0.987	0.997	0.992	0.992	1.000
SSH-Patator	1.000	0.000	0.998	1.000	0.999	0.999	1.000
Web-Brute Force	0.704	0.003	0.719	0.704	0.711	0.708	0.998
Web-XSS	0.163	0.004	0.157	0.163	0.160	0.156	0.992
Web-SQL Injection	0.000	0.000	0.000	0.000	0.000	0.000	0.999
Infiltration	0.500	0.000	0.500	0.500	0.500	0.500	0.995

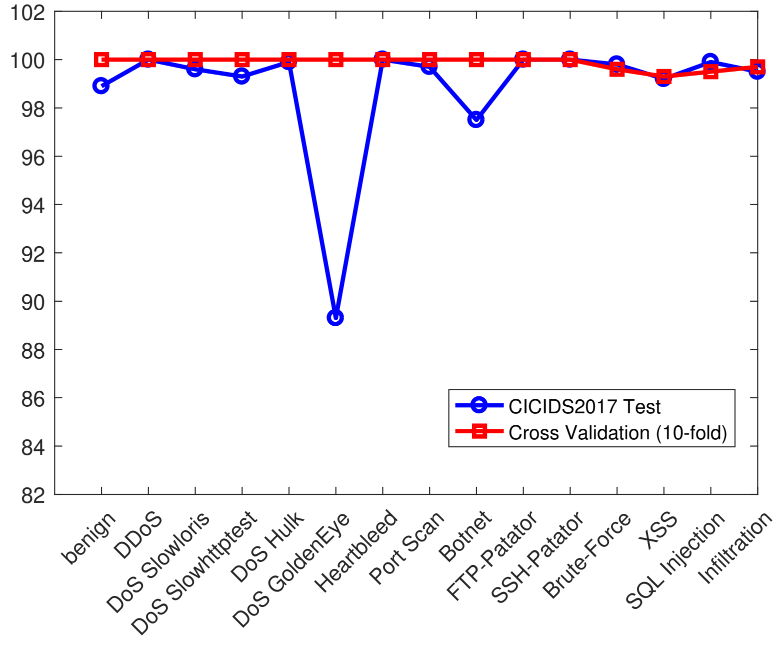


Figure 3.2: ROC area of each class for cross-validation and CICIDS2017 test

To evaluate how the results of a statistical analysis will generalize to an individual dataset, 10-fold cross validation (CV) is carried out on the training dataset, and the area under the

ROC curves as shown in Figure 3.2 is computed for each attack classes in the dataset based on two test options cross-validation and CICIDS2017. The findings likewise show that the ROC values are fluctuating with both DoS GoldenEye and Botnet attacks but have virtually the same values for the majority of attacks in both test options.

Table 3.4: Performance results of the original features and selected features

Classifier	**Accuracy(%)**	**CC**	**IC**	**Training(Sec.)**	**Prediction(Sec.)**
Decision Tree	79.10	31678	8322	7.00	1.03
Random Forest	91.80	36748	3252	27.46	2.76
CSForest	92.50	37032	2968	742.07	4.92
Ensemble	87.00	34831	5169	810.91	7.39

(a) Results of the original features (70 features)

Classifier	**Accuracy(%)**	**CC**	**IC**	**Training(Sec.)**	**Prediction(Sec.)**
Decision Tree	90.80	36353	3647	5.15	0.32
Random Forest	92.30	36945	3055	15.09	1.18
CSForest	89.80	35953	4047	280.74	2.86
Ensemble	94.80	37942	2058	475.31	7.71

(b) Results of the features selection approach (16 features)

As an efficient technique to address the curse of dimensionality, is to project a high-dimensional data into a smaller sized dimension without eliminating features that matter for classification. Thus, we compare the results of the proposed feature selection approach to results when using all original features. This comparison was performed based on the second set of evaluation metrics under base and ensemble classifiers. Table 3.4 displays the influence of applying the suggested model on the detection performance for each type of classifier. As expected, the use of selected features yielded the highest accuracy of 94.80% compared to all the original features that only achieved a classification accuracy of 87%. It was noted that our ensemble classifier with the 16 selected features had alleviated the building and prediction times substantially compared to the ensemble model working with all original 70 attributes. On the other hand, the employment of the CFS-BA approach helps the model to maximize the number of correctly classified instances (CC) and minimize the percentage of incorrectly classified instances (IC). Notably, the training time was considerably lowered from 810.91 to 475.31.

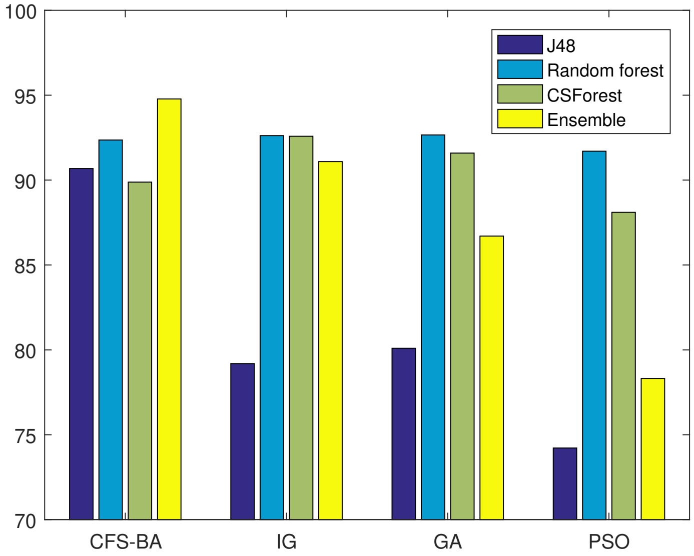


Figure 3.3: Accuracy of different feature selection methods

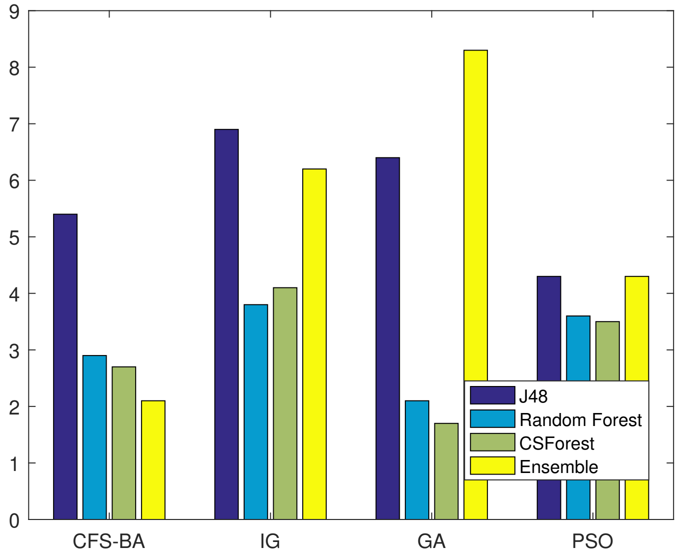


Figure 3.4: False alarm rate of different feature selection methods

To further examine our proposed feature selection method, we make a comparison between the proposed model and three popular attribute selection approaches, including Information Gain [83], Genetic Algorithm [84] and Particle Swarm Optimization [85]. We investigate the effectiveness of our system using two metrics, namely classification accuracy and false alarm rate. Initially, as displayed in Figure 3.3, the accuracy of our proposed model outshines all other features selection approaches and attains the highest accuracy rate of 94.8%. In the case of the ensemble classifier, the best system accuracy of the attribute selection techniques was 91.09% IG, 86.7% GA, and 78.31% PSO. Similarly, Figure 3.4 depicts that our proposed CFS-BA demonstrates more desirable false alarm rate results than IG, GA, and PSO, and its value varies from 2.1% to 5.4%. For an intrusion detection task, malicious activities are expected to be correctly determined, and benign activities are prepared for not to be misclassified. Consequently, higher detection accuracy and a lower false alarm rate are intended.

Table 3.5: Comparison study on the balanced CICIDS2017 subsets

Authors	**Year**	**Feature selection**	**Accuracy(%)**
Kevric et al. [86]	2017	No	61.41
Chen et al. [87]	2018	No	79.84
Zhu et al. [88]	2019	No	84.18
Proposed Method	2020	No	86.68
Aguileraa et al. [89]	2013	Yes	85.47
Sornsuwit and Jaiyen [90]	2019	Yes	75.51
Tama et al. [91]	2019	Yes	92.02
Proposed Method	2020	Yes	94.80

Table 3.5 illustrates the comparison study between the accuracy of our ensemble model and the peers, which is divided into two parts, where the first one compares our ensemble approach with others that do not apply feature selection techniques. Then, we make another comparison between the ensemble method after applying feature selection with other existing ensemble methods that use a feature selection technique in the second part. For a fair comparison, we adopt all of these methods using the same well-balanced CICIDS2017 dataset. A

comparative analysis of these results indicates that the proposed ensemble algorithm shows an improvement in performance accuracy over these existing approaches.

3.6 Summary

we propose a network intrusion detection system based on feature selection and tree-based ensemble classifiers. This detection system integrates both correlation feature selection and bat algorithm to reduce the number of irrelevant features and automatically determine the optimal number of features. We also adapt the ensemble classifier based on Decision Tree, Random Forest, and CSForest for building the system. Our method is evaluated using an intrusion detection dataset named CICIDS2017 and pulled off remarkable results when it comes to TPR, FPR, precision, recall, f-measure, MCC, as well as ROC area among benign activities and malicious data flows. On the other hand, we witnessed that the propositioned CFS-BA ensemble approach has minimized the training and time significantly against using all authentic features. For additional assessment of the algorithm mentioned above, we compare it with three popular feature selection techniques, such as information gain, genetic algorithm, and particle swarm optimization under each sort of classifier. Compared with individual classifiers, our model is more suitable for the real system due to its sturdier robustness in various evaluation metrics. On top of that, the experimental results have shown the CFS-BA ensemble technique achieves the highest detection accuracy with 94.8%, and lower false alarm rate 2.1% in comparison with the ensemble models of other attribute selection methods.

Chapter 4: Hybridized Feature Selection Framework for Enhancing Network Attack Detection

4.1 Introduction

Machine learning (ML)is a subset of artificial intelligence that has a capability of machines to learn without being explicitly programmed [22], has proved to be appealing in addressing real-world issues. It includes the making of algorithms that can gain from and make predictions on data fed into it [92]. Machine learning approaches can be divided into two primary groups, which are supervised and unsupervised learning. In supervised learning, a labeled set of training data is utilized to estimate or map the input data to the desired output. On the other hand, under the unsupervised learning techniques, no labeled are offered, and there is no learning procedure.

Feature selection (FS) is referred to as attribute selection, plays a crucial role in machine learning, considering that it aims to single out a small subset of attributes out of the large feature space [93]. This technique enhances the classification accuracy and minimizes the classification building time by disposing of irrelevant and redundant attributes [94]. High-dimensional data in the input space considerably increase time and space complexity. Furthermore, because of the existence of irrelevant or redundant features, the learning algorithms tend to over-fit and end up being less precise in the large data environment. Choosing an optimum feature subset has a prime significance in the area of machine learning and data mining. Feature selection techniques assist for a better understanding of the domain by supporting pertinent attributes based upon some validity criterion [95].

Feature selection (FS) can be divided into three categories: filter, wrapper, and embedded method. In the filter method [96], the individual features are ranked, and a subset is selected without utilizing a learning algorithm. In contrast, the wrapper method [97] employs a learner to evaluate the attribute subset to be chosen. Filter approaches are quite faster, while wrapper methods offer more high classification accuracy for particular classifiers with a higher computational expense. In embedded method [19], feature selection is performed as part of the model construction process, which is specific to the applied learning algorithm.

In this work, a new filter-wrapper based feature selection algorithm is proposed to select the most important features that contribute much to enhance the performance of intrusion detection systems. The wrapper takes all the possible mixes of feature subsets and ultimately selects the best subset, which carries out well for a given classifier. Hence, it needs a substantial time for processing and ends up being more difficult when applied directly to the intrusion detection dataset. For that reason, in order to conquer this concern, a filter based pre-processing is performed initially before utilizing a wrapper method. Symmetric Uncertainty(SU) [98] is a filter method that can effectively remove redundant and irrelevant features from the CICIDS2017 intrusion dataset. Consistency Subset Evaluator (CNS) [68] with Random Forest Classifier, then use Flower Pollination Algorithm (FPA) [99] as the wrapper method that to select the ideal feature subset from the remaining features.

The novelty of the proposed system are highlighted as follows:

- Proposing a novel hybridization feature selection method by combining filter and wrapper techniques for predicting network attacks, which reduces

- Presenting a comparison of the approach with different search methods that have proved to be successful tools in the network intrusion detection field.

- Conducting experiments on the CICIDS2017 dataset to investigate the feasibility and effectiveness of the proposed hybrid feature selection framework, which includes many types of novel attacks and high-dimensionality features.
- Extensively examining the performance of the selected features over several evaluation metrics to deliver a proof of concept, which is sustained by empirical results as well as exhibits its potentiality for additional expansion.

The structure of this chapter is described next. The next section provides an overview of the past related works. Details of our proposed methodology on the hybrid approach and explanation of the two phases involved in the feature reduction technique have been discussed in Section 4.4. The adopted experimental setup is presented in Section 4.5. The experimental results and discussions of the empirical study are elaborated in Section 4.6. Finally, Section 2.6 concludes the chapter by summarising the presented results and their contributions.

4.2 Related Works

Recently, several hybrid feature selection approaches have been proposed and examined in the literature to greatly boost the accuracy of IDS. Each approach has its own strengths and weaknesses. In addition to the performance of each strategy differs in terms of the evaluation metrics.

The authors in [100] came up with a hybrid feature selection method that takes away the unimportant and redundant attributes by utilizing two filter approaches, namely, F-score and information gain. The feature sets produced by these techniques are integrated to form a candidate features. These candidate feature sets are passed through a wrapper approach for picking out the final feature subset by using a sequential floating search method (SFSM).

In reference [101], the authors propose the hybrid feature selection model by combining the filter and wrapper based approaches. In the first stage, the filter method is used for ranking of attributes from the most relevant to the least relevant. In the second stage, the attributes that have a high ranking are employed in the wrapper approach with genetic algorithms (GA) and particle swarm optimization (PSO). The result revealed that the hybrid technique is effective in attaining an optimum and smaller subset of attributes with higher accuracy and lower computational cost.

Another filter ranking was implemented by [102], who proposed a filter-wrapper feature selection algorithm for threat detection. This hybrid system consists of two stages. Mutual information was implemented in the first stage to remove irrelevant and redundant features. In the next stage, the least-square support vector machine (LS-SVM) was used as a classification algorithm to find the best one subset from many feature subsets.

Authors of [103] consider a hybrid feature selection methodology based on mutual information (MI) and genetic algorithm (GA) for the intrusion detection system. Basically, the hybrid approach used an MI-based filter method to minimize the search space, and GA has been utilized as a wrapper method to choose the best reduction. They experimentally showed that an SVM-based classifier succeeds in achieving better performance than an artificial neural network (ANN).

Study by [104] developed a hybrid feature selection approach based on attribute ranking technique for unsupervised learning. This approach hybrid model operates in two phases. In the filter phase, the attributes are ranked based on Laplacian score, and the top-ranked attributes are selected based on some threshold value. In wrapper phase, the selected attributes are indexed by utilizing the Calinski-Harabasz index.

Further study [105] applied a hybrid feature selection technique for defect prediction on an extensive legacy software system in telecommunications. They first utilized a feature ranking to reduce the search space and then apply a feature subset selection. The study explored the effectiveness of seven feature ranking methods and four feature subset selection methods on a private defect dataset.

Despite the benefits abovementioned hybrid feature selection approaches for function optimization and feature selection. One might perhaps question the objective of building a new hybrid FS method. This question can be answered using the *No Free Lunch (NFL)* theorem, which reasonably confirms that any single algorithm is not efficient in addressing all optimization problems [106]. This implies that there is plenty of room for performance improvement by developing new algorithms to deal with function optimization dilemmas along with attribute selection challenges in a more effective way. Motivated by earlier works on IDS, in this study, another hybrid FS approach is proposed for tackling practical optimization and feature selection problems.

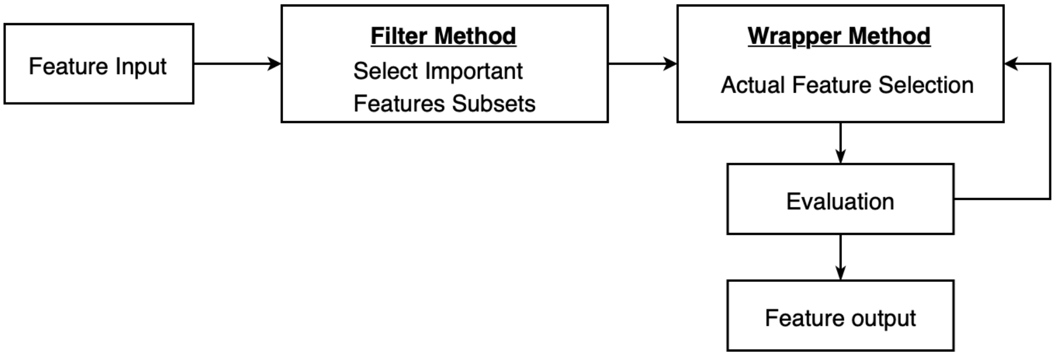


Figure 4.1: Hybrid method

4.3 Hybrid Method

The concept of the hybrid approach is based on leveraging the strengths of both filter and the wrapper methods [107]. Hybrid approaches have emerged as a popular technique that deploys a combination of approaches in filtering and choosing the most significant attributes seeking for better accuracy and performance of the used classification system. Moreover, these hybrid methods research studies play an essential role not only in motivating of using a variety of approaches but likewise in reducing the usage of resources, effort, and time for later phases. Fig.4.1 shows the general flowchart of a hybrid feature selection approach.

4.4 Framework Design

The hybrid feature selection approach was used in conjunction with incorporating filter and wrapper methods to select the optimum features. The approach explained herein was tailored towards addressing the performance problems associated with the wrapper method by lowering the total number of features through the filter method. Hence, the results were ultimately higher in detection accuracy and efficiency of computation. The proposed framework is displayed in Figure 4.2, comprised of two phases. As soon as the dataset is packed, in Phase 1, feature selection is carried out using the proposed hybrid attribute selection model, which is comprised of two stages. The first stage is the attribute selection filter. In this stage, the top-ranked features are selected using the greatly enhanced feature selection filter based upon Symmetrical Uncertainty (SU). Then, we use the Naive Bayes classifier that helps us to select the best possible feature set by eliminating the irrelevant features based on the global minimum number of incorrectly classified instances, which results in a total of 62 features. The second stage is the wrapper function selection, where the top 62 features from the filter stage are used to pick out the final optimum set of features that are used for classification.

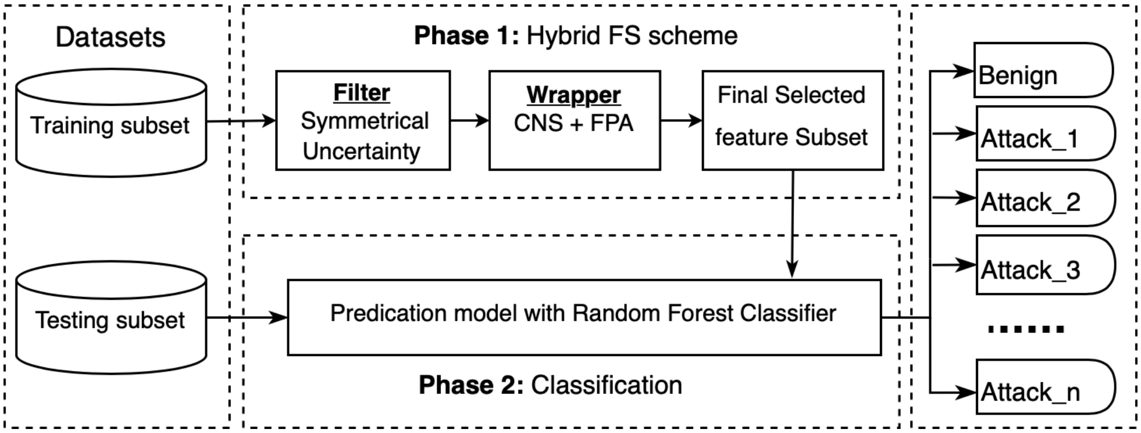


Figure 4.2: The proposed hybrid feature selection approach

4.4.1 Hybrid Filter-Wrapper Algorithm

In the initial phase, the two-stage feature selection scheme has been developed to improve the performance of intrusion detection system is described below.

4.4.1.1 Filter Feature Selection

In this first stage, we adopt a filter approach based on Symmetrical Uncertainty (SU) [98], which is a feature weighting algorithm that records the SU score of an attribute then assigns the rank to each feature. SU score with high value has a top rank, and less value has the least rank. For attribute selection, leading ranked attributes can be selected relying on the requirement and type of the issue on which it will be used. The measurement of SU was defined to measure the redundancy as follows.

$$IG(X/Y) = H(X) \ - \ H(X/Y), \tag{4.1}$$

$$SU(X,Y) = 2\left[\frac{IG(X/Y)}{(H(X)+H(Y)}\right], \tag{4.2}$$

where $IG(X/Y)$ is the information gain of feature X, that is an independent attribute and Y is the class attribute. $H(X)$ is the entropy of feature X and $H(Y)$ is the entropy of feature Y.

4.4.1.2 Consistency Subset Evaluator

In this second stage, the Consistency Subset Evaluator (CNS) [68] is employed as a wrapper method with Flower Pollination Algorithm (FPA) to select the best subset of features. FPA is utilized on the top 62 features from the filter in the first stage as the technique to search for an effective subset of features. This process is continued on all the subsets of features produced by the search. Finally, the features subset with the highest accuracy is chosen as the final set of optimum features for classification.

Flower Pollination Algorithm (FPA) [99] is a bio-inspired optimization search algorithm that models the rules of the pollination process of flowering plants in nature. FPA can be divided into two mechanisms: self-pollination and cross-pollination. Self-pollination occurs between pollen in the same flower or another flower of the same plant; this abiotic pollination method is called local pollination in FPA. Cross-pollination is biotic and is accomplished by physical factors when the pollen is delivered to other plants over long distances through pollinators (e.g., birds, bees, insects). Thus, cross-pollination is a representation of global pollination and can be expressed as follows. on and can be expressed as follows.

$$x_i^{t+1} = x_i^t + L(x_i^t - g_*), \tag{4.3}$$

where x_i^t is pollen i or the vector solution x_i at iteration t, and g_* represents the current optimal solution obtained among all solutions at this iteration. The parameter L is the pollinator random step size obeying the Lévy distribution, which satisfies the following Lévy distribution equation.

$$L \sim \frac{\lambda\Gamma(\lambda)sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}},\ (s \gg\ s_0 > 0). \tag{4.4}$$

here $\lambda = 1.5$, $\Gamma(\lambda)$ denotes the standard gamma function, and this distribution is valid for large steps $s > 0$. Both s and s_0 reflect the Levy flight step size and minimum step size, respectively. The mathematical description of local pollination is implemented by the following formula.

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t), \tag{4.5}$$

where x_j^t and x_k^t represent the positions of pollens from the same plant species with different flowers, and ϵ is the random walk that is uniformly distributed on [0,1].

4.4.2 Classification

In the classification phase, we use the Random Forest (RF) classifier which is another decision tree-based algorithm proposed by [38], which works by constructing multiple decision trees from a different subset of the input samples during training time and find mode of all classes output of the individual trees as the final class. Then it employs the bootstrap aggregation algorithm to tree learners, leading to a better performance model by reducing the variance, while the bias remains the same. One of the main benefits of random forests is in their ability to eliminate the overfitting of the training dataset after combining many decision trees and improve the predictive accuracy compared to other machine learning algorithms. RF functions efficiently on a high dimensional dataset and able to manage unbalanced and missing data. On the other hand, RF appears to result in robust and efficient detection of a multitude of attacks, no matter their kind. Furthermore, the training and implementation time of the RF method was substantially reduced with a low false positives rate. In RF, a collection of tree-structured classifiers can be defined as:

$$\{h(x, \theta_k), k = 1, 2, \dots i \dots\} \tag{4.6}$$

where h is the random forest classifier, $\{\theta_k\}$ are identically distributed random vectors and each tree has a unit vote for the most popular class at input x.

4.5 Experimentation

In this section, a discussion is made on our proposed hybrid feature selection approach for network intrusion detection. The CICIDS2017 dataset was selected to build the hybrid system using the Weka machine learning toolbox. The detection framework of the proposed method is performed in four primary stages, as shown in Figure 4.2, which includes feature selection, model building, attack detection.

4.5.1 Dataset Preprocessing

Data preprocessing step, which involves three phases, including data filtration, as mentioned earlier in section 2.3.3. While creating balanced subsets and applying min-max normalization has been discussed in Section 3.3.1. In order to build the proposed model, we create balanced subsets out of the CICIDS2017 dataset as displayed in Table 3.1.

4.5.2 Evaluation Metrics

We utilized two sets of evaluation metrics to examine the classification performance of the proposed hybrid feature selection approach for IDS. The first set is comprised of true positive rate (TPR), false positive rate (FPR), recall, and receiver operating characteristic (ROC) for benign traffic or other fourteen attacks. The second set was used to compare the performance of the suggested model with different search methods which contains accuracy, root mean squared error (RMSE), mean absolute error (MAE), kappa statistic, correctly classified instances (CC), incorrectly classified instances (IC), training and prediction time. These metrics are calculated using a confusion matrix, which offers four measures as displayed in Table 2.1.

Table 4.1: Training and testing subsets distribution

Label	**Raw**	**Filtered**	**Train**	**Test**
Benign	2273097	1893223	20000	20000
DDoS	128027	128020	2700	3300
DoS Slowloris	5796	5385	1350	1650
DoS Slowhttptest	5499	5242	2171	1169
DoS Hulk	231073	173791	4500	5500
DoS GoldenEye	10293	10286	1300	700
Heartbleed	11	11	5	5
PortScan	158930	1956	3808	4192
Botnet	1966	1437	936	624
FTP-Patator	7938	6093	900	1100
SSH-Patator	5897	3360	900	1100
Web-Brute Force	1507	37	910	490
Web-XSS	652	652	480	160
Web-SQL Injection	21	21	16	4
Infiltration	36	36	24	6
Total Attack	471454	470365	20000	20000
Total	2830743	2827876	40000	40000

4.6 Performance Analysis

This section presents the results of the proposed hybrid FS approach for intrusion detection system. We have made use of Python for the prior data preprocessing steps, and then, the proposed hybrid feature selection algorithm and classification were conducted on the CICIDS2017 dataset that has been executed in an updated version of Weka environment (3.8.3). The results are listed in the tabular format together with competent charts. Table 4.2 summarizes the classification results of the proposed IDS in the context of TPR, FPR, recall, and ROC. The results of the experimental analysis are discussed in terms of comparison with other search methods, improvement in classification accuracy, and the reduction in dimension of features.

Table 4.2: Overall classification performance

Class	**TPR**	**FPR**	**Recall**	**ROC**
Benign	0.984	0.053	0.984	0.989
DDoS	0.987	0.002	0.987	0.999
DoS Slowloris	0.923	0.000	0.923	0.999
DoS Slowhttptest	0.900	0.002	0.900	0.998
DoS Hulk	0.951	0.000	0.951	0.998
DoS GoldenEye	0.660	0.001	0.660	0.839
Heartbleed	0.400	0.000	0.400	1.000
Port Scan	0.996	0.010	0.996	0.996
Botnet	0.244	0.001	0.244	0.974
FTP-Patator	0.997	0.000	0.997	1.000
SSH-Patator	0.998	0.000	0.998	1.000
Web-Brute Force	0.898	0.004	0.898	0.999
Web-XSS	0.063	0.001	0.063	0.988
Web-SQL Injection	1.000	0.000	1.000	1.000
Infiltration	0.000	0.000	0.000	0.995

Based on the obtained experimental results, it was noted that both TPR and FPR achieved satisfactory results in most of the classifications. However, Infiltration performance was low at most of the evaluation metrics such as TPR, FPR, and recall caused by a relatively small proportion of this attack in the entire dataset.

4.6.1 Comparison with Different Search Methods

Table 4.3 shows comparative results for the classifier performance of the proposed Flower pollination algorithm (FPA) compared to different search methods after generating their selected subset of features for detection attacks on the well-balanced CICIDS2017 dataset. The results of the numerical examples can be concluded in the following points:

Table 4.3: Results comparison of different search methods

Search Method	Acc(%)	RMSE	MAE	Kappa	CC	IC	Training	Prediction
70 features	91.80	0.115	0.040	0.884	36748	3252	42	2.28
Random Search	61.30	0.168	0.050	0.337	24534	15466	3887.68	1.16
PSO Search	93.60	0.097	0.027	0.910	37449	2551	32.54	3.40
Genetic Search	91.40	0.110	0.037	0.876	36596	3404	31.41	3.34
Ant Search	94.00	0.112	0.040	0.913	37602	2398	44.92	3.62
Bat Search	92.70	0.100	0.026	0.898	37108	2892	36.9	3.28
Elephant Search	92.80	0.103	0.031	0.897	37129	2871	31.21	3.23
Firefly Search	93.40	0.110	0.036	0.906	37364	2636	30.41	3.40
Wolf Search	91.20	0.108	0.036	0.870	36480	3520	284.74	4.34
Flower Search	95.43	0.097	0.034	0.934	38175	1825	29.89	3.40

- Accuracy: the observation from the experimental results is that the Random Forest classifier could produce only 91.87% accuracy without incurring a feature selection technique. Selecting all 70 features for training and testing will reduce the performance of the classifier, and it also escalates time complexity. Feature selection using the wrapper method CNS with FPA as the search method selects 18 relevant features from the dataset and produced an accuracy of 90.22%. However, the proposed hybrid feature selection method with FPA as the search method provides maximum accuracy of 95.43% by selecting 21 features with SU as ranking criteria and CNS as the wrapper method compared to all other search algorithms listed in Table 4.3.

- RMSE and MAE: when it comes to the root mean squared error (RMSE) and mean absolute error (MAE) [108], both metrics should always be less for better prediction. We can see that the both PSO Search and FPA methods gave the lowest RMSE value among all other search methods as 0.097. However, their achieved MAE values were 0.027 and 0.034, respectively.

- Kappa: according to Cohen's Kappa value, it can be observed that the suggested FPA search approach is superior to other search techniques, and it produced a promising result of 0.934. For example, Kappa of Random Search is 0.337, which is even worse than Wolf Search, whose Kappa is 0.87, this indicates that the Random search still does not allow for a sufficiently significant classification. The results also demonstrate a slight increase in Kappa Statistic in comparison to classification without applying any feature selection technique.

- CC and IC: by testing and classification of 40000 instances of records from the CICIDS2017 dataset. The total number of classified records for each selected algorithm are shown in Table 4.3. Based on these results, the proposed FPA algorithm also provides significantly higher correctly classified records (CC) and lower incorrectly classified records (IC).

- Training time: another issue could be the time required for building the classifier training models. Based on the experiments, Flower Pollination Algorithm (FPA) built a training model in the fastest time. In contrast, Random and Wolf search algorithms are more computationally expensive since they have the longest building time of 3887.68 and 2844.74 seconds, respectively.

4.6.2 Reduction in Features

Initially, the Symmetric Uncertainty (SU) filter method is used in order to rank each feature. Out of 70 attributes constituting the dataset, only 62 are selected as the best possible feature set after this method eliminating eight features with the help of Naive Bayes classifier. Following this, the Consistency-based Subset Evaluator (CNS) with Flower Pollination Algorithm (FPA) is applied as a wrapper method to highly ranked features from the filter in the first stage. A feature sub-selection is performed to search for the best subset of features that represents the dataset in order to reduce classification complexity and minimize the training time.

Table 4.4: Selected features for CICIDS2017 dataset

Feature Name	**Feature No.**	**Importance**
Fwd Packet Length Max	7	0.4828
Init_Win_bytes_backward	59	0.4586
Average Packet Size	51	0.4546
Destination Port	1	0.4244
Bwd Packet Length Std	14	0.4025
Fwd IAT Max	24	0.3818
Flow IAT Max	19	0.3701
Flow Duration	2	0.3654
Fwd IAT Std	23	0.3592
Fwd IAT Total	21	0.3583
Fwd Packet Length Min	8	0.3513
Fwd IAT Mean	22	0.3499
Min Packet Length	37	0.3448
Fwd Packet Length Std	10	0.3281
Flow IAT Mean	17	0.3243
Subflow Bwd Packets	56	0.3091
Bwd Packet Length Min	12	0.3067
Fwd IAT Min	25	0.2524
Flow IAT Min	20	0.2166
FIN Flag Count	42	0.0320
Fwd Header Length	33	0.0033

Table 4.4 reveals the 21 selected features generated by CNS, and these features are ranked in decreasing order according to their importance. As shown in the table, it can be clearly observed that “Fwd Packet Length Max” is ranked 1^{st} with the highest weighted score of 0.4828. Also, An average of 70% of feature reduction was observed after the hybrid feature selection method.

4.6.3 Classification Improvement

Accuracy has been thought about to evaluate the efficiency of a classifier. The efficiency of different well known classifiers such as JRip [109], Random Forest [38], Random Tree [110], LAD Tree [111], NB Tree [112], Simple Cart [113], FURIA [114], and Bayesian Network [115] prior and after applying the proposed hybrid feature selection model has been graphically depicted in the Table 4.5. Also, the improvement of accuracy has been examined and noted. The results convey that our attribute selection method triggers better efficiency of a classifier. The maximum accuracy gain was revealed by Random Forest Classifier 95.43% with the improvement of 3.56 %.

Table 4.5: Accuracy improvement during feature selection

Classifier	**Before FS(%)**	**After FS(%)**	**Improvement(%)**
Random Forest	91.87	95.43	3.56
Random Tree	61.41	75.82	14.41
JRip	71.76	80.04	8.28
LAD Tree	75.47	82.25	6.78
Simple Cart	68.83	79.4	10.57
Bayesian Network	67.17	81.57	14.4
NB Tree	49.38	79.93	30.55
FURIA	85.03	91.8	6.77

4.7 Summary

This study aims to propose a hybrid feature selection approach that capitalizes on the strengths of both filter and wrapper approaches as the pre-processing stage for network intrusion detection. A two-stage feature selection method is consisting of Symmetric Uncertainty (SU) as a filter to remove redundant features. Then apply the Consistency Subset Evaluator (CNS) with Flower Pollination Algorithm (FPA) and Random Forest classifier as a wrapper to select the optimal feature subset from the remaining features. We also compare the Flower Search Algorithm (FPA) with several existing search methods in terms of its accuracy, RMSE, MAE, Kappa, correctly classified records (CC), incorrectly classified records (IC), and model training time. From this comparison, we found that the FPA search method obtains superior performance, defeating the baseline performance in both accuracy and model building time. The proposed hybrid approach achieved not only select the most significant features but also maximize the classification accuracy while eliminating redundant and noisy features. From the experimental results on the CICIDS2017 dataset, an average of 70% reduction in features has been observed, which leads to achieve accuracy of 95.43% and reduction in time for building a classifier.

Chapter 5: Conclusions

This chapter will outline all the groundwork endeavours presented in the previous chapters, along with their entire accomplishment to the aim and objective of the thesis. Subsequently, all available adjustments, which might significantly improve the functionality of the introduced research remedies, definitely will be explained as future work directions.

5.1 Summary of Thesis

Chapter 1 provides a brief history of IDS, along with the aims and objectives of this thesis. Followed by a survey of the intrusion detection system and feature selection techniques. Feature selection is an essential part of this section. Then, we introduce the importance of using machine learning algorithms in the cybersecurity field and the classification of IDS. The implemented CICIDS2017 dataset, besides its statistical observations and attack scenarios, are explained in this chapter.

Chapter 2 investigates the effectiveness of machine learning approaches for intrusion detection. A range of experiments has been carried out on seven machine learning algorithms, namely AdaBoost, Random Forest, Naive Bayes, Decision Tree, MLP, KNN and QDA. After evaluating the system, KNN is the best performer which obtained an accuracy of 99.55%, a precision of 99.53%, and a recall of 99.55%. However, all the machine learning classifiers except KNN build their models in adequate training time. Thus, KNN has the longest execution time, which is a severe drawback in the intrusion detection field.

Chapter 3 introduces a network intrusion detection system based on feature selection technique and tree-based ensemble classifiers. For this intention, we employ Decision Tree, Random Forest, and Cost-sensitive algorithms and combine their prediction by making use of the average voting rule. The main objective of this mechanism is to detect different types of attacks with high accuracy and low false alarm rate. This methodology integrates both correlation feature selection (CFS) and bat algorithm (BA) to reduce the number of irrelevant features. The feasibility and effectiveness of the suggested model are investigated under several statistical metrics. The results indicate that the proposed ensemble method provides higher accuracy of 94.8%, a lower false positive rate of 2.1%, and efficiently detects various types of attacks compared to individual base classifiers.

Chapter 4 proposes a hybrid feature selection approach that capitalizes on the strengths of both filter and wrapper methods for intrusion detection. This approach is consisting of Symmetric Uncertainty (SU) to remove redundant features, and the Consistency Subset Evaluator (CNS) with Flower Pollination Algorithm (FPA) and Random Forest classifier to select the optimal feature subset from the remaining features. We also compare the FDA with several search methods over different evaluation metrics. Empirical results show that an average of 70% reduction in features has been observed, which leads to achieve an accuracy of 95.43% and reduction in time for building the model.

5.2 Future Directions

The future research directions in this field should involve the following aspects.

- Detecting zero-day attacks have been the main objective of cybersecurity, specifically intrusion detection for a long time. Machine learning is thought to be a supportive approach to address that concern. Various systems have been proposed however a practical remedy is still yet to find, mostly because of the restriction brought on by the outdated open datasets readily available. For that reason, we will take an in-depth evaluation of the CICDDoS2019 dataset [116] with some well-known machine learning algorithms for detecting zero-day intrusions. The training subset is comprised of twelve types of attacks, whereas the testing subset includes seven different types of intrusions. The testing subset, involving zero-day real-life intrusions and benign traffic flows gathered in real research development network are and then applied to test the efficiency of the selected classifiers.

- Modern technology growth about studying high-speed environments, namely Big data, Hadoop, Spark, and cloud computing, has assured to bring the Internet application to a new level. The application of these technologies provides faster handling yet more substantial data consideration, which will considerably enhance the effectiveness of our proposed approaches. Our future work is likewise to execute our systems utilizing these solutions to extend their ability.

- We plan to develop an IDS model by applying artificial neural networks on a new cyber defense dataset (CSE-CIC-IDS2018) which obtained through cloud computing for intrusion detection. The attacking framework is composed of 50 machines, and the victim organization has five departments which include 420 machines and 30 web servers [31].

- Finally, all the proposed approaches in this thesis could only be utilized for a static dataset of intrusion detection systems. None of them could be employed on a real-time intrusion detection system. Real-time IDS needs action in time, and it ordinarily handles data according to their instances. Machine learning techniques could be applied to real-time IDS; nevertheless, it requires much more enhancements and screening in practical.

References

[1] A. Rudskoy, A. Borovkov, P. Romanov, and O. Kolosova, “Reducing global risks in the process of transition to the digital economy,” in *IOP Conference Series: Materials Science and Engineering*, vol. 497, no. 1. IOP Publishing, 2019, p. 012088.

[2] K.-K. R. Choo, “A cloud security risk-management strategy,” *IEEE Cloud Computing*, vol. 1, no. 2, pp. 52–56, 2014.

[3] F. Cohen, “Computer viruses: theory and experiments,” *Computers & security*, vol. 6, no. 1, pp. 22–35, 1987.

[4] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, “A survey on cyber security for smart grid communications,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[5] J. P. Anderson, “Computer security threat monitoring and surveillance,” *Technical Report, James P. Anderson Company*, 1980.

[6] C. Kolias, G. Kambourakis, and M. Maragoudakis, “Swarm intelligence in intrusion detection: A survey,” *computers & security*, vol. 30, no. 8, pp. 625–642, 2011.

[7] D. Denning and P. G. Neumann, *Requirements and model for IDES-a real-time intrusion-detection expert system.* SRI International, 1985, vol. 8.

[8] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, and A. Valdes, “Ides: The enhanced prototype-a real-time intrusion-detection expert system,” in *SRI International, 333 Ravenswood Avenue, Menlo Park.* Citeseer, 1988.

[9] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, “A network security monitor,” Lawrence Livermore National Lab., CA (USA); California Univ., Davis, CA (USA . . . , Tech. Rep., 1989.

[10] R. Di Pietro and L. V. Mancini, *Intrusion detection systems.* Springer Science & Business Media, 2008, vol. 38.

[11] W. Stallings and M. P. Tahiliani, “Cryptography and network security: principles and practice, vol. 6,” 2014.

[12] R. G. Bace and P. Mell, “Intrusion detection systems,” 2001.

[13] G. Eschelbeck and M. Krieger, “Eliminating noise from intrusion detection systems,” *Information Security Technical Report*, vol. 8, no. 4, pp. 26–33, 2003.

[14] J. Bhatia, R. Sehgal, S. Kaur, S. Popli, and N. Taneja, “Analyzing intrusion detection system evasions through honeynets,” *6th Annual Security Conference*, 2007.

[15] Y. Huang, P. J. McCullagh, and N. D. Black, “Feature selection via supervised model construction,” in *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, 2004, pp. 411–414.

[16] D. Addison, S. Wermter, and G. Arevian, “A comparison of feature extraction and selection techniques,” in *Proceedings of the International Conference on Artificial Neural Networks*, 2003, pp. 212–215.

[17] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *Advances in neural information processing systems*, 2006, pp. 507–514.

[18] D. Koller and M. Sahami, “Toward optimal feature selection,” Stanford InfoLab, Tech. Rep., 1996.

[19] P. Q. Huy, A. Ngom, and L. Rueda, “A new feature selection approach for optimizing prediction models, applied to breast cancer subtype classification,” in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2016, pp. 1535–1541.

[20] S. Suresh and A. Narayanan, “Improving classification accuracy using combined filter+ wrapper feature selection technique,” in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2019, pp. 1–6.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern classification 2nd ed,” *John Willey & Sons Inc*, 2001.

[22] D. Michie, D. J. Spiegelhalter, C. Taylor *et al.*, “Machine learning,” *Neural and Statistical Classification*, vol. 13, no. 1994, pp. 1–298, 1994.

[23] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, “An overview of machine learning,” in *Machine learning*. Elsevier, 1983, pp. 3–23.

[24] V. Ford and A. Siraj, “Applications of machine learning in cyber security,” in *Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering*, 2014.

[25] P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[26] J. B. Fraley and J. Cannady, "The promise of machine learning in cybersecurity," in *SoutheastCon 2017*. IEEE, 2017, pp. 1–6.

[27] N. Michael, "Artificial intelligence a guide to intelligent systems," 2005.

[28] G. Karatas and O. K. Sahingoz, "Neural network based intrusion detection systems with different training functions," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 2018, pp. 1–6.

[29] A. Valdes and D. Anderson, "Statistical methods for computer usage anomaly detection using nides," *Technical report, SRI International*, 1995.

[30] G. Vigna and R. A. Kemmerer, "Netstat: A network-based intrusion detection approach," in *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217)*. IEEE, 1998, pp. 25–34.

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[32] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *2016 International Conference on Information Science and Security (ICISS)*. IEEE, 2016, pp. 1–6.

[33] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33 789–33 795, 2018.

[34] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13 546–13 560, 2019.

[35] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2019, pp. 277–288.

[36] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, 2008.

[37] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International journal of computer science and network security*, vol. 7, no. 12, pp. 258–263, 2007.

[38] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[39] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[40] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, “Intrusion detection by machine learning: A review,” *expert systems with applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009.

[41] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.

[42] T. Hastie and R. Tibshirani, “J. friedman the elements of statistical learning. chapter 6,” 2001.

[43] S. Ustebay, Z. Turgut, and M. A. Aydin, “Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier,” in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. IEEE, 2018, pp. 71–76.

[44] A. Boukhamla and J. C. Gaviro, “Cicids2017 dataset: Performance improvements and validation as a robust intrusion detection system testbed,” *International Journal of Information and Computer Security*, vol. 9, 2018.

[45] W. Zegeye, R. Dean, and F. Moazzami, “Multi-layer hidden markov model based intrusion detection system,” *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 265–286, 2019.

[46] D. Aksu, S. Üstebay, M. A. Aydin, and T. Atmaca, “Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm,” in *International Symposium on Computer and Information Sciences*. Springer, 2018, pp. 141–149.

[47] J. Hou, P. Fu, Z. Cao, and A. Xu, “Machine learning based ddos detection through netflow analysis,” in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 1–6.

[48] A. Bansal and S. Kaur, “Extreme gradient boosting based tuning for classification in intrusion detection systems,” in *International Conference on Advances in Computing and Data Sciences*. Springer, 2018, pp. 372–380.

[49] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[50] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” *Computer networks*, vol. 76, pp. 146–164, 2015.

[51] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the kdd cup 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 2009, pp. 1–6.

[52] L. Lu, S. Teng, W. Zhang, Z. Zhang, D. Liu, and X. Fang, "Error-correcting ability based collaborative multi-layer selective classifier ensemble model for intrusion detection," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD).* IEEE, 2019, pp. 4–9.

[53] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Computational intelligence magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[54] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques.* Elsevier, 2011.

[55] J. R. Quinlan, *C4. 5: programs for machine learning.* Elsevier, 2014.

[56] M. J. Siers and M. Z. Islam, "Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem," *Information Systems*, vol. 51, pp. 62–71, 2015.

[57] A. Takemura, A. Shimizu, and K. Hamamoto, "Discrimination of breast tumors in ultrasonic images using an ensemble classifier based on the adaboost algorithm with feature selection," *IEEE transactions on medical imaging*, vol. 29, no. 3, pp. 598–609, 2009.

[58] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, "Predicting short-term stock prices using ensemble methods and online data sources," *Expert Systems with Applications*, vol. 112, pp. 258–273, 2018.

[59] V. C. Korfiatis, S. Tassani, and G. K. Matsopoulos, "A new ensemble classification system for fracture zone prediction using imbalanced micro-ct bone morphometrical data," *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1189–1196, 2017.

[60] I. Partalas, G. Tsoumakas, E. V. Hatzikos, and I. Vlahavas, "Greedy regression ensemble selection: Theory and an application to water quality prediction," *Information Sciences*, vol. 178, no. 20, pp. 3867–3879, 2008.

[61] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, "Neural network ensembles: evaluation of aggregation algorithms," *Artificial Intelligence*, vol. 163, no. 2, pp. 139–162, 2005.

[62] S. Mukherjee and N. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[63] P. Sornsuwit and S. Jaiyen, "Intrusion detection model based on ensemble learning for u2r and r2l attacks," in *2015 7th international conference on information technology and electrical engineering (ICITEE).* IEEE, 2015, pp. 354–359.

[64] I. Syarif, E. Zaluska, A. Prugel-Bennett, and G. Wills, "Application of bagging, boosting and stacking to intrusion detection," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition.* Springer, 2012, pp. 593–602.

[65] V. Bukhtoyarov and V. Zhukov, "Ensemble-distributed approach in classification problem solution for intrusion detection systems," in *International Conference on Intelligent Data Engineering and Automated Learning.* Springer, 2014, pp. 255–265.

[66] Y. Wang, Y. Shen, and G. Zhang, "Research on intrusion detection model using ensemble learning methods," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS).* IEEE, 2016, pp. 422–425.

[67] H. Hota and A. K. Shrivas, "Decision tree techniques applied on nsl-kdd data and its comparison with various feature selection techniques," in *Advanced Computing, Networking and Informatics-Volume 1.* Springer, 2014, pp. 205–211.

[68] N. Paulauskas and J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using nsl-kdd dataset," in *2017 open conference of electrical, electronic and information sciences (eStream).* IEEE, 2017, pp. 1–5.

[69] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS).* IEEE, 2019, pp. 228–233.

[70] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[71] S. Singh and A. K. Singh, "Web-spam features selection using cfs-pso," *Procedia Computer Science*, vol. 125, pp. 568–575, 2018.

[72] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, "Fast correlation based filter (fcbf) with a different search strategy," in *2008 23rd international symposium on computer and information sciences.* IEEE, 2008, pp. 1–4.

[73] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010).* Springer, 2010, pp. 65–74.

[74] S. Mishra, K. Shaw, and D. Mishra, "A new meta-heuristic bat inspired classification approach for microarray data," *Procedia Technology*, vol. 4, pp. 802–806, 2012.

[75] Y. Xin-She, "Bat algorithm for multi-objective optimization," *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 267–274, 2011.

[76] G.-Q. Huang, W.-J. Zhao, and Q.-Q. Lu, "Bat algorithm with global convergence for solving arge-scale optimization problem," *Jisuanji Yingyong Yanjiu*, vol. 30, no. 5, pp. 1323–1328, 2013.

[77] I. Gandhi and M. Pandey, "Hybrid ensemble of classifiers using voting," in *2015 international conference on green computing and Internet of Things (ICGCIoT)*. IEEE, 2015, pp. 399–404.

[78] C. L. Devasena, "Comparative analysis of random forest rep tree and j48 classifiers for credit risk prediction," in *International Conference on Communication, Computing and Information Technology (ICCCMIT-2014)*, 2014.

[79] S. Feng, "A cost-sensitive decision tree under the condition of multiple classes," in *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2015)*. Atlantis Press, 2015.

[80] P. Lingden, A. Alsadoon, P. Prasad, O. H. Alsadoon, R. S. Ali, and V. T. Q. Nguyen, "A novel modified undersampling (mus) technique for software defect prediction," *Computational Intelligence*, 2019.

[81] C. Catal and M. Nangir, "A sentiment classification model based on multiple classifiers," *Applied Soft Computing*, vol. 50, pp. 135–141, 2017.

[82] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[83] B. Azhagusundari and A. S. Thanamani, "Feature selection based on information gain," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 2, pp. 18–21, 2013.

[84] S. K. Pal and P. P. Wang, *Genetic algorithms for pattern recognition*. CRC press, 2017.

[85] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[86] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, no. 1, pp. 1051–1058, 2017.

[87] X. Chen, L. Zhang, Y. Liu, and C. Tang, "Ensemble learning methods for power system cyber-attack detection," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 2018, pp. 613–616.

[88] Y. Zhu, L. Zhou, C. Xie, G.-J. Wang, and T. V. Nguyen, "Forecasting smes' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach," *International Journal of Production Economics*, vol. 211, pp. 22–33, 2019.

[89] A. Aguilera, M. Palma, and R. Mata-Toledo, "Determination of significant features to precancerous cervical classification," *AASRI Procedia*, vol. 4, pp. 275–281, 2013.

[90] P. Sornsuwit and S. Jaiyen, “A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting,” *Applied Artificial Intelligence*, vol. 33, no. 5, pp. 462–482, 2019.

[91] B. A. Tama, M. Comuzzi, and K.-H. Rhee, “Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system,” *IEEE Access*, vol. 7, pp. 94 497–94 507, 2019.

[92] H. Song, I. Triguero, and E. Özcan, “A review on the self and dual interactions between machine learning and optimisation,” *Progress in Artificial Intelligence*, vol. 8, no. 2, pp. 143–165, 2019.

[93] P. Pudil, F. J. Ferri, J. Novovicova, and J. Kittler, “Floating search methods for feature selection with nonmonotonic criterion functions,” in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*, vol. 2. IEEE, 1994, pp. 279–283.

[94] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[95] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 121–129.

[96] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[97] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[98] C. E. Särndal, “A comparative study of association measures,” *Psychometrika*, vol. 39, no. 2, pp. 165–187, 1974.

[99] X.-S. Yang, “Flower pollination algorithm for global optimization,” in *International conference on unconventional computing and natural computation*. Springer, 2012, pp. 240–249.

[100] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, “Hybrid feature selection by combining filters and wrappers,” *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, 2011.

[101] A. Kawamura and B. Chakraborty, “A hybrid approach for optimal feature subset selection with evolutionary algorithms,” in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2017, pp. 564–568.

[102] M. A. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, and U. T. Nagar, “A novel feature selection approach for intrusion detection data classification,” in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2014, pp. 82–89.

[103] R. Vijayanand, D. Devaraj, and B. Kannapiran, “A novel intrusion detection system for wireless mesh network with hybrid feature selection technique based on ga and mi,” *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1243–1250, 2018.

[104] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, “A new hybrid filter–wrapper feature selection method for clustering based on ranking,” *Neurocomputing*, vol. 214, pp. 866–880, 2016.

[105] K. Gao, T. M. Khoshgoftaar, H. Wang, and N. Seliya, “Choosing software metrics for defect prediction: an investigation on feature selection techniques,” *Software: Practice and Experience*, vol. 41, no. 5, pp. 579–606, 2011.

[106] D. H. Wolpert, W. G. Macready *et al.*, “No free lunch theorems for optimization,” *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[107] Z. Huang, C. Yang, X. Zhou, and T. Huang, “A hybrid feature selection method based on binary state transition algorithm and relieff,” *IEEE journal of biomedical and health informatics*, vol. 23, no. 5, pp. 1888–1898, 2018.

[108] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance,” *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[109] W. W. Cohen, “Fast effective rule induction,” in *Machine learning proceedings 1995.* Elsevier, 1995, pp. 115–123.

[110] D. Aldous, “The continuum random tree. i,” *The Annals of Probability*, pp. 1–28, 1991.

[111] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, “Multiclass alternating decision trees,” in *European Conference on Machine Learning.* Springer, 2002, pp. 161–172.

[112] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.” in *Kdd*, vol. 96, 1996, pp. 202–207.

[113] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and regression trees. belmont, ca: Wadsworth,” *International Group*, vol. 432, pp. 151–166, 1984.

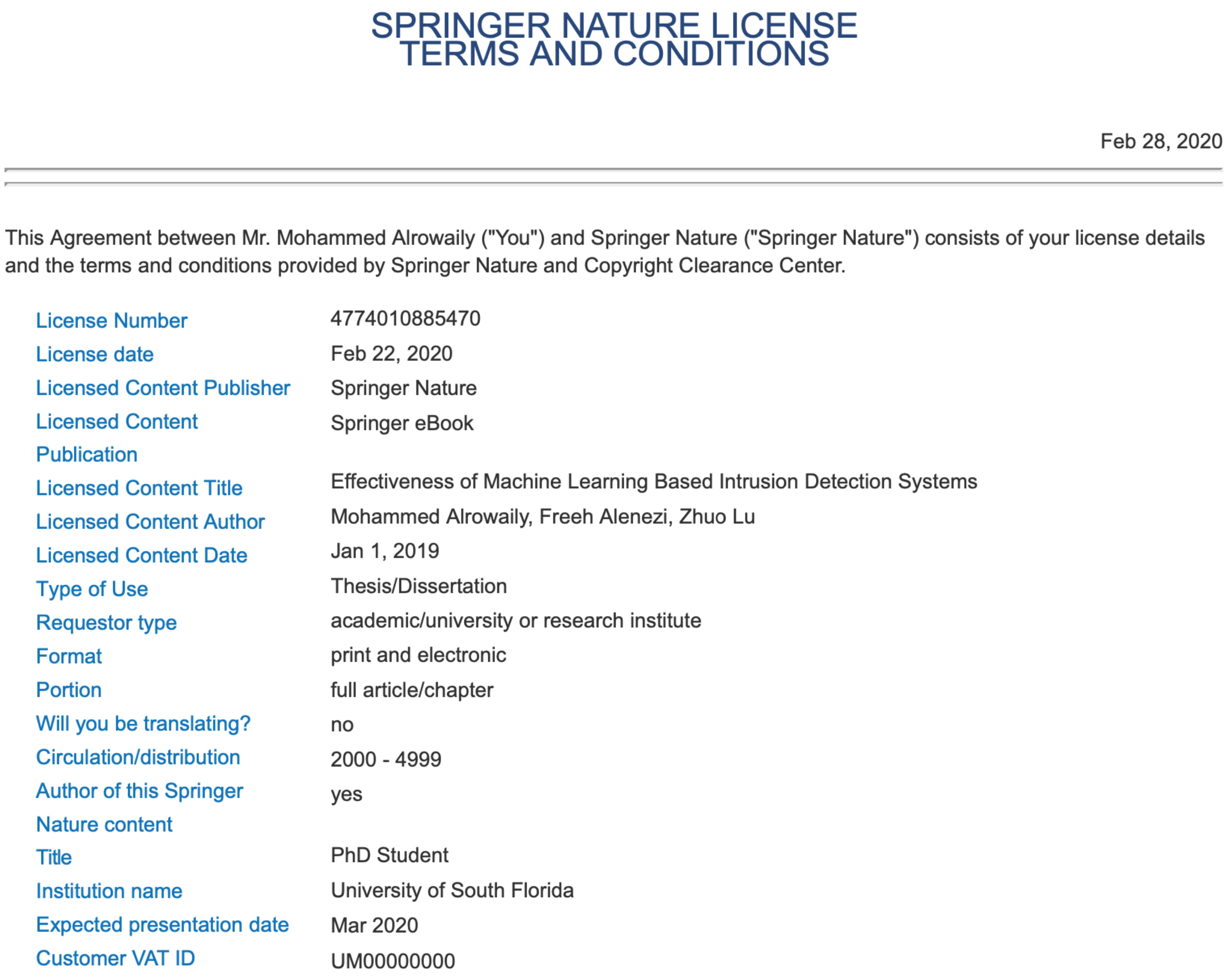
[114] J. Hühn and E. Hüllermeier, “Furia: an algorithm for unordered fuzzy rule induction,” *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[115] R. R. Bouckaert, “Bayesian network classifiers in weka for version 3-5-7,” *Artificial Intelligence Tools*, vol. 11, no. 3, pp. 369–387, 2008.

[116] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (ddos) attack dataset and taxonomy,” in *2019 International Carnahan Conference on Security Technology (ICCST).* IEEE, 2019, pp. 1–8.

Appendix A: Copyright Permissions

The permission below is for the use of the material in Chapter 2.



SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Feb 28, 2020

This Agreement between Mr. Mohammed Alrowaily ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4774010885470
License date	Feb 22, 2020
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Effectiveness of Machine Learning Based Intrusion Detection Systems
Licensed Content Author	Mohammed Alrowaily, Freeh Alenezi, Zhuo Lu
Licensed Content Date	Jan 1, 2019
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	2000 - 4999
Author of this Springer Nature content	yes
Title	PhD Student
Institution name	University of South Florida
Expected presentation date	Mar 2020
Customer VAT ID	UM00000000