Intrusion Detection System for IoT Networks for Detection of DDoS Attacks

By

Monika Roopak

In the Partial Fulfilment of
Doctor of Philosophy

School of Engineering
Newcastle University
(February 2021)

Abstract

In this thesis, a novel Intrusion Detection System (IDS) based on the hybridization of the Deep Learning (DL) technique and the Multi-objective Optimization method for the detection of Distributed Denial of Service (DDoS) attacks in Internet of Things (IoT) networks is proposed. IoT networks consist of different devices with unique hardware and software configurations communicating over different communication protocols, which produce huge multidimensional data that make IoT networks susceptible to cyber-attacks. The network IDS is a vital tool for protecting networks against threats and malicious attacks. Existing systems face significant challenges due to the continuous emergence of new and more sophisticated cyber threats that are not recognized by them, and therefore advanced IDS is required.

This thesis focusses especially on the DDoS attack that is one of the cyber-attacks that has affected many IoT networks in recent times and had resulted in substantial devastating losses. A thorough literature review is conducted on DDoS attacks in the context of IoT networks, IDSs available especially for the IoT networks and the scope and applicability of DL methodology for the detection of cyber-attacks. This thesis includes three main contributions for 1) developing a feature selection algorithm for an IoT network fulfilling six important objectives, 2) designing four DL models for the detection of DDoS attacks and 3) proposing a novel IDS for IoT networks. In the proposed work, for developing advanced IDS, a Jumping Gene adapted NSGA-II multi-objective optimization algorithm for reducing the dimensionality of massive IoT data and Deep Learning model consisting of a Convolutional Neural Network (CNN) combined with Long Short-Term Memory (LSTM) for classification are employed. The experimentation is conducted using a High-Performance Computer (HPC) on the latest CISIDS2017 datasets for DDoS attacks and achieved an accuracy of 99.03 % with a 5-fold reduction in training time. The proposed method is compared with machine learning (ML) algorithms and other state-of-the-art methods, which confirms that the proposed method outperforms other approaches.

Acknowledgements

I want to express gratitude to my supervisors, Prof. Gui Yun Tian, and Prof. Jonathon Chambers for the enormous amount of support, guidance, and crucial insights in undertaking this research, and for playing an active role in my professional development.

I want to thank Dr Martin Johnston and Dr Stephane Le Goff for their expertise and assessment of my work in helping to keep my work on the right track during the four years of my research.

I feel short of words in expressing my appreciation to Dr Yu Gong from Loughborough University and Dr Mohsen Naqvi to provide me valuable comments and corrections to improve my work presented in this thesis.

My special thankfulness goes to my colleagues Yachao Ran, Junzhen Zhu, Ruslee, Denis Ona, Adi Marindra and Lawal Umar Daura for providing their enormous help and support at difficult times during my PhD.

I am grateful for love and support given by my better half Rahul and my amazingly supportive family who have always motivated and supported me to pursue my dreams.

Finally, I want to be grateful to the Government of India for sponsoring this research work.

Table of Content

Glossary

ANN ..... Artificial Neural Network
CNN..... Convolutional Neural Network
CPU .....Central Processing Unit
DDoS .....Distributed Denial of Service
DIDS..... Distributed Intrusion Detection System
DL.....Deep Learning
DoS .....Denial of Service
ELM.....Extreme Learning Machine
FS.....Feature selection
GA ..... Genetic Algorithms
GPU .....Graphics Processing Unit
HIDS..... Host-based Intrusion DetectionSystem
HPC .....High-Performance Computer
IDS..... Intrusion Detection System
IoT .....Internet of Things
IP .....Internet Protocol
ISP .....Internet Service Provider
LSTM .....Long Short-Term Memory
ML ..... Machine Learning
MLP.....Multi-Layer Perceptron
MSE..... Mean Square Error
NIDS.....Network Based Intrusion Detection Systems
NSGA-II-aJG..... Non-dominated Sorting Algorithm with its Adapted Jumping Gene operator
RFID .....Radio Frequency Identification
RNN..... Recurrent Neural Network
SDN .....software defined networking
SLFN .....Single Hidden Layer Feedforward Neural Networks
SMOTE..... Synthetic Minority Over Sampling
SVM ..... Support Vector Machine

List of Publications

Journal Paper

1. M. Roopak, G. Tian and J. Chambers, 'Multi-Objective based Feature Selection for DDoS Attack Detection in IoT Network', to appear in *IET Networks*, 2020.

Conference Papers

1. M. Roopak, G. Y. Tian, and J. Chambers, 'An Intrusion Detection System Against DDoS Attacks in IoT Networks', in *2020 IEEE 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020.

2. M. Roopak, G. Y. Tian, and J. Chambers, 'Deep Learning Models for Cyber Security in IoT Networks', in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0452–0457.

3. M. Roopak, G. Y. Tian, and J. Chambers, 'Improved NSGA-II for Solving Multi-objective Optimization Problems, in 2018, Annual Research Conference 2018, Newcastle University.

Poster

1. Poster Presentation "A Review of Intrusion Detection Schemes for IoT" in Annual Research Conference 2017, Newcastle University, January 2017.

Chapter 1. Introduction

This chapter introduces the aim and objectives of this thesis including the research questions addressed in the thesis. The background and motivation for the presented work are specified. The three significant contributions are listed, and in the end, the organization of the thesis, including a brief introduction to each chapter of the thesis, is provided.

1.1 Overview

The Internet has emerged as IoT (Internet of Things) that has touched every corner across the globe and is helping the lives of human beings incredibly. In the IoT every object is uniquely identified and accessible to the network, its position and status are known, intelligence is added to this accelerated Internet, ultimately impacting on our professional, personal and social environments; in other words, the IoT revolves around the interconnection competencies amongst things, devices, and people. It promises to create a world in which all the devices (also referred to as smart devices) are connected to the Internet and talk with each with minimum human intervention. The purpose is to facilitate a higher and more comfortable standard of living. The fundamental components that form IoT are hardware, software, and communication infrastructure. An IoT network is a heterogeneous system that consists of varieties of devices such as RFID and Wireless Sensors, communicating over different protocols. The concept of IoT was first proposed by Kevin Ashton at MIT's AutoId lab in 1999, but the development of the IoT was limited because of the low network resources support and Internet [1]. However, with the development of the high-speed Internet, the concept of IoT has now come into practice. The IoT system is expected to grow with a projected 77.44 billion devices by the end of 2022 [2]. IoT examples extend from smart connected homes appliances to wearables healthcare devices. The IoT networks execute proper security mechanisms such as encryption, back up of data, user authentication and applications, and integrity assurance of processed and stored data in the system. The underlying architecture of the IoT network is illustrated in Figure 1.1, comprising a network including smartphones, laptops, cars, smart bulbs, home appliances such as toasters, and many more devices connected over the Internet.

In theory, the IoT networks are fully secure with all the necessary security mechanisms in place; however, it is hypothetical, and IoT security is still a big gap from the expected reality. Like any other computer network system, IoT networks are susceptible to different cyber-attacks. Recent attacks on IoT networks have revealed that cybersecurity is still a major loophole [3][4][5][6]. With the development of IoT networks, cyber-attacks against such

systems have increased in numbers significantly, especially DDoS (Distributed Denial of Service) attacks, that have disrupted many IoT networks in the recent past and have resulted in devastating losses. IoT devices are poorly secured and managed, which makes it an attractive target for a hacker to harness many nefarious purposes and intentions. The IoT devices connected over the Internet are resource constrained tiny and cheap devices, that lack in security controls as these do not have enough processing power and memory, which the hacker can easily hack and control by using various technologies and tools such as for cracking the password a brute force attack can be applied to the device and sometimes the default password of these devices is never changed. IoT networks connect thousands of smart devices, also known as smart objects connected across the globe, that facilitate DDoS attacks to reach beyond any limit in scale and potential size. Employing an IDS (Intrusion Detection System) is one of the technologies for the detection of cyber-attacks on networks. The earlier a cyberattack is detected the lesser would be the adverse consequences.

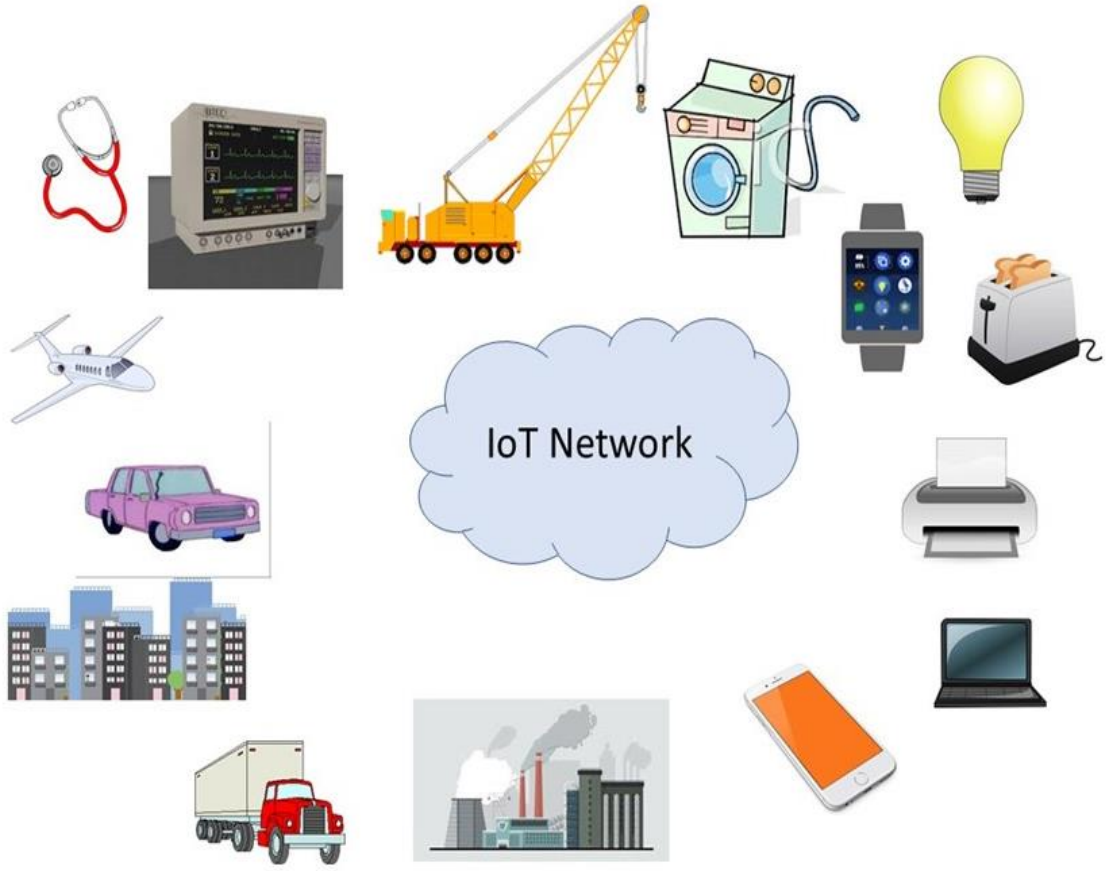

Figure 1.1 System diagram of connections to an Internet of Things network

1.2 Research Questions

In this thesis, the following research questions are addressed in the context of cybersecurity in IoT networks.

Question 1. Why cybersecurity, especially the DDoS attack, is a big problem in IoT networks?

IoT networks promise amazing benefits and advantages in easing the life of every human being. However, in the early development of IoT networks, the cybersecurity issues were ignored, that has now become a bottleneck problem. IoT networks have suffered and are still suffering adversely because of the DDoS attacks. In fact, IoT networks have increased attackers for launching and spreading DDoS attacks and targeting high profile websites and Internet based service providers. This raises the question of why cybersecurity is still a big loophole in the IoT implementation.

Question 2. How have DDoS attacks adversely affected IoT networks, and what method is used for their launch on the network? The efficiency of currently deployed IDSs in the detection of DDoS attacks should be studied.

As IoT networks have aided the attacker for launching DDoS attacks, so the statistical analysis of the adversities of the DDoS attacks should be performed. Another important aspect is the method of the launch of the attacks into the system, that would help find the present loopholes and aid in the development of the advanced IDSs. The websites and servers employ defence mechanisms generally IDSs for securing systems from the cyber-attacks. The modern cybersecurity dense mechanisms are mostly Machine Learning based systems. There arises a question of why these defence mechanisms have failed in the detection of the attacks. Some of the DDoS attacks lasted for many days without being discovered despite the modern cyber defence methods.

Question 3. What is the possible solution for developing an advanced IDS that can learn on its own for defence against the new more sophisticated cyber-attacks?

The present interest of the researcher in the field of cybersecurity has moved towards DL (Deep Learning) for developing the advanced IDSs that will be capable of detecting the sophisticated cyber-attacks. This is the main underlying concept of the IDS proposed in this thesis.

Question 4. How to improve the applicability of DL in the field of cybersecurity in IoT networks?

The DL has provided fascinating accomplishments in many fields such as image and video processing, Bigdata and Natural Language Processing. The IoT networks consist of heterogeneous devices varying from high processing objects to resource constraint devices. So, this leads to the search for the method that will help in the exploitation of DL in the field of cybersecurity.

1.3 Aim and Objectives

The aim of this work is to build an advanced Intrusion Detection System for the detection of DDoS attack in IoT networks employing multi objective optimization technique taking six most important objectives and deep learning techniques.

Specifically, the objectives of work presented in this thesis are to:

- Review the different types of cyber-attacks in context with the IoT networks. The focus is on to investigate the state of the art IDSs for IoT networks. Review of the latest research work on feature selection methods and deep learning techniques application for the detection of DDoS attacks on IoT networks.
- Propose an advanced feature selection method for reducing the dimensionality of the network data. To investigate the six most important with a multi-objective optimization method for feature selection.
- Investigate the feasibility of deep learning technique for the detection of the DDoS attack in context of the IoT networks. To find the best deep learning model in terms of the various performance metrices on CICIDS2017 datasets. Another objective is to compare the performance of deep learning technique with machine learning models in the context of the IoT networks.
- Propose an advanced IDS combining the benefits of both the multi-objective optimization algorithm and deep learning method for the detection of the DDoS attacks in the IoT networks.

1.4 Background and Motivation

Cybersecurity in IoT networks deals with providing IoT devices and network defence against cyber-attacks, alteration, and damage to network data and illegal access. Just like other communication networks, every IoT network demands three essential security features, namely confidentiality, integrity and availability [7], but these get affected by intrusions within the network. The efficient and effective identification of intrusions in a resource-restricted, scalable, continually evolving environment and the distributed network is a challenging task. Nowadays, more sophisticated and advanced cyber-attacks have been developed. DDoS is one of the attacks that has affected the IoT networks tremendously and resulted in enormous devastating consequences. DDoS attacks make target server or devices stop serving the legitimate user by denying service [4]. The attack is carried out by flooding excessive false requests that exhaust the server and consume all the bandwidth so that the target server can no longer fulfil any other request [5].

Initially, the security issues in the IoT networks were ignored with the development of the systems. But as cybersecurity has become the bottleneck problem in the success of IoT networks because DDoS attacks for example, have targeted various IoT networks; for example, on 21 October 2016 a company named Dyn server, that controls much of the Internet's DNS infrastructure in America, was hit by a DDoS attack using a new weapon called the Mirai botnet. Major sites affected by this attack were Amazon, Netflix, PayPal, Spotify, and Twitter in Europe and the US. Another incident of a DDoS attack on an IoT network was recorded in April 2017, where a new IoT botnet was discovered named Persirai, which shares Mirai's codebase and targeted over 1000 different models of IP Camera. The attack was discovered by cybersecurity researchers at Trend Micro and was affecting 122,069 IP cameras across the globe. Other of the DDoS attacks on IoT networks are discussed in more detail in Chapter 2. Some of the statistics of DDoS attacks are given in [8]:

- According to the survey research done by Cisco Visual Networking Index, the DDoS attacks will double to 14.5 million by 2020
- DDoS attacks represent up to 25 % of the USA's Internet traffic that makes the most prominent threat to service providers

- According to Kaspersky's Securelist [9], the DDoS high profile attacks have increased. The top favourite target countries for the DDoS attacks recorded are China with 63.8 % and the USA with 17.5 %
- The most significant DDoS attacks ever experienced were recorded by Imperva [10]. The attack targeted the application layer that ran for 13 days stretch and peaked at 292000 requests per second in the year 2019.
- Bulletproof's annual report on cybersecurity [11] stated that DDoS for small companies could cost up to $120000 and for large enterprise could cost more than $2 million
- According to Akami's survey presented in [12], the DDoS attacks have increased in financial service organizations between Dec. 2, 2018, and May 4, 2019, with 800 DDoS attacks on financial industries only that is more than 40 % of total DDoS attacks.
- Another research form IBM X-Force discovered that more than 80 % Mirai botnet variants in 2019 were recorded in media and information service and insurance companies [13].
- According to the survey report in [14], the current DDoS protection and mitigation in 2019 are $ 2.4 billion that will double to $4.7 billion by 2024, with an annual growth rate of 14 percent.

The motivation for this work, therefore, comes from the fact that IoT systems in recent times have suffered significantly as discussed above in some of the examples of the latest DDoS attacks, such as the Mirai DDoS attacks on the Dyn server [3]. The present detection systems are mostly statistical based or machine learning based IDSs. However, based on the record, it can be stated that with the advancement in more sophisticated DDoS attacks an advanced IDSs that learn themselves are demanded. Deep learning techniques have shown to provide impressive results in the field of image processing, video processing, Big data, and Natural Language Processing that is capable of learning on its own. Another exciting feature of deep learning is mentioned as the ability of deep learning algorithms for performing high-level feature extraction that makes them an ideal choice for novel attack detection [6]. Advancement in neural network and CPU (Central Processing Unit) and GPU (Graphics Processing Unit) [15] has facilitated the application of deep learning in the field of cybersecurity.

1.5 Research Contributions

The following are the three significant contributions of this thesis:

1.5.1 Contribution 1

The first contribution of this thesis is to propose a novel multi-objective based feature selection method for the detection of DDoS cyber-attacks in IoT networks. The main objectives of Chapter 4 are:

- Proposing and implementing a multi-objective optimization method for performing feature selection and satisfying conflicting objectives for extracting optimal attributes from the datasets for the detection of the DDoS attack,
- A method incorporating the Jumping Gene adapted NSGA algorithm is developed for optimized feature selection, considering six important objectives, namely maximize relevance, minimize redundancy, minimize the number of features, maximize classifier accuracy, maximize recall, and maximize precision,
- Investigating a method for obtaining feature subsets as Pareto-front, that facilitate the user with choice in selecting the feature set,
- Undertaking an extensive evaluation on the latest CICIDS2017 dataset using standard performance metrics and to compare the performance of the presented method with state-of-the-art algorithms.

1.5.2 Contribution 2

The second objective of this thesis is to propose and compare various deep learning models for the detection and classification of DDoS attacks in IoT networks. The main contributions of Chapter 5 are:

- Proposing four deep learning models feasible for the cybersecurity in IoT networks,
- Comparing the proposed deep learning models to discover the best model in terms of performance metrics,
- Carry out extensive evaluation of the proposed model on CICIDS2017 datasets using the standard performance parameters,
- Comparing the performance of the proposed deep learning algorithm with other machine learning algorithms in the context of DDoS attack detection in IoT networks.

1.5.3 Contribution 3

The last contribution of this thesis is to propose a novel intrusion detection system against the DDoS attack in IoT networks. The main contributions of Chapter 6 are:

- Proposing a novel IDS integrating multi-objective based feature selection and the deep learning methodology for the classification of the DDoS attack,
- Extensively evaluating our proposed IDS on the high-performance computer over several standard assessment metrics to analyse the proficiency of the proposed method,
- Comparing the proposed work with state-of-the-art-algorithms and machine learning methods f, which have been used to a great extent in the field of cybersecurity for the detection of DDoS attacks.

1.6 Organization of Thesis

Chapter 1: This chapter introduces the thesis structure. The introduction to the IoT concept and its layered architecture is discussed. The background and motivation behind this thesis are detailed, and the three main contributions of the thesis are discussed, and, in the end, the organization of the thesis is presented.

Chapter 2: This chapter delivers the literature review part of the thesis. The review conducted in this chapter offers the motivation for doing the work presented in this thesis. The various cyber-attacks that affect IoT networks are discussed. Introduction to the DDoS attacks and various types of DDoS attacks in the context of IoT networks are detailed. This chapter provides the introduction to IDS, the review of latest DDoS attacks on IoT networks, literature review of IDSs for IoT networks, review of feature selection methods for IoT networks, the review of deep learning techniques for the detection of cyber-attacks and in the end the open research issues and challenges in the context of DDoS attacks on IoT networks are discussed.

Chapter 3. In this chapter, the datasets employed for evaluating the proposed methods and the performance measures used are described. Datasets have critical importance for training and testing and intrusion detection systems. Gathering datasets and examining activities should increase consciousness and the capability to identify assaults within the future. The employed CICIDS2017 datasets are elaborated and discussed. Performance metrics employed for testing and validating an IDS are also critical as they reflect the performance of the proposed system. The employed performance evaluation metrics are also described in this chapter.

Chapter 4: In this chapter, a multi-objective optimization-based feature selection method for the detection of DDoS attacks in IoT networks is presented. The real-world measurements that form the input to an IDS are generally huge. FS (Feature Selection) is therefore required to decrease the dimensionality of data and improve the functioning of an IDS. The Non-dominated

Sorting Algorithm with its Adapted Jumping Gene operator (NSGA-ii-aJG) has been employed to solve the optimization problem and an Extreme Learning Machine (ELM) is exploited as the classifier for feature selection based on six critical objectives for an IoT network.

Chapter 5: In this chapter, the four realistic DL (Deep Learning) models for cybersecurity in networks are proposed and compared. DDoS attacks have affected many IoT networks in the recent past that have resulted in huge losses. The proposed deep learning models are evaluated using the latest CICIDS2017 datasets for DDoS attack detection, which have provided the highest accuracy as 97.16%; also, proposed models are compared with other machine learning algorithms.

Chapter 6: In this chapter, an IDS using the hybridization of the deep learning technique and the multi-objective optimization approach for the recognition of DDoS assaults in the IoT networks is proposed. In a network, the IDS is a vital tool for securing it from cyber-attacks. Detection of new emerging cyber threats are becoming difficult for existing IDS, and therefore advanced IDS is required. In this chapter, an IDS founded on the fusion of a Jumping Gene adapted NSGA-II multi-objective optimization method for data dimension reduction which is proposed in Chapter 4 and the Convolutional Neural Network (CNN) integrating Long Short-Term Memory (LSTM) deep learning techniques for classifying the attack which is proposed in Chapter 5 is established. The experimentation is conducted using a High-Performance Computer (HPC) on the latest CICIDS2017 datasets on DDoS attacks and achieved an accuracy of 99.03 % with a 5-fold reduction in training time.

Chapter 7: The overview of the critical findings of the thesis and an outlook into future work are presented in the last chapter.

Chapter 2. Literature Review

This chapter covers the literature review part of the thesis. The study conducted in this chapter gives the motivation for doing the work present in this thesis. The various cyber-attacks that affect IoT networks are discussed. The DDoS attacks introduction and types of DDoS attack in the context of IoT networks are detailed. This chapter provides the introduction to IDS including a literature review of IDSs for IoT networks, the review of latest DDoS attacks on IoT networks, review of feature selection methods for IoT networks and in the end section, the review of deep learning techniques for the detection of cyber-attacks and open research issues are discussed.

2.1 IoT networks cybersecurity vulnerabilities

Securing IoT from several possible cyberattacks is quite a complex task. However, it turns out to be handy to a certain degree when concerns are examined in a layered arrangement. Each layer has its specific hurdles and weaknesses that should be identified to confirm it's shielded by banning various categories of assaults [16]. Averting such incidents takes an appropriate defense practice and approach which can tackle current weaknesses found in IoT networks. The study needs to be done to look for the vulnerability and the way it promotes to a cyberattack. [17]. A vulnerability within network characterizes the inability of the procedure that allows the invader to find out the scope to invade the network cybersecurity. This would be dangerous that it may result in an assault when get overlooked or unnoticed. Table 2.1 gives the catalogue of existing vulnerabilities along with its supporting elements which genuinely are liable for the incidence of any cyber assault on IoT devices.

Table 2.1 Top Vulnerabilities in IoT networks

Vulnerability	Cause
Inadequate Authorization or Authentication	The default passwords are never changed in IoT devices, weak password strength, credentials not protected, absence of granular access control facilitates hacker to gain access, insecure password recovery
Unprotected web interfaces	Lack of encryption that can help the attacker to get access to data and controls, the weak password recovery procedure, easy passwords

Vulnerable network services	The network services in IoT are not protected and secured that can be used to launch attacks on devices or spread the cyber-attack to other devices
Lack of proper transport encryption and verification of the integrity	Transport encryption is crucial for securing the data within a network, in IoT that can aid a hacker to listen and view the data being shared in the network.
Insufficient physical level security	Because of the varying nature of IoT devices, the security provided at the physical level is not good. The peripheral devices, data storage devices, USB port, or memory cards can aid hackers to get access to the IoT data.
Poor security configuration	Lack of security configuration can make it easy for an attacker to enter the IoT networks and get access to the sensitive information and data in the network, or the hacker may launch and spread the cyber-attack in the network to destroy it.
Vulnerable Cloud Interface	The cloud is the backbone of the IoT network, IoT devices can send and receive data from the cloud if attackers gain control of IoT devices that can aid them to launch an attack on to the cloud. Also, the cloud provides a pay-as-you-go facility via which the hacker can enter the cloud and execute its intentions.

2.2 Cyber-attacks in IoT Networks

The cyber-attacks can be broadly classified into four categories, as shown in Figure 2.1 [18]–[20].

1. Physical Attacks: An attacker launches this kind of attack by being physically close to the network or devices [21]–[23].

- **Tampering**: In this attack, the devices within the network are physically modified by being physically present near the device
- **Fake node infusion**: In this assault, the invader controls the flow of information and data by launching a fake node near nodes.
- **Jamming or RF interference:** The attacker disturbs the communication in the network by launching DoS attacks on the sensor and RFID tags by sending noise signals over radiofrequency.

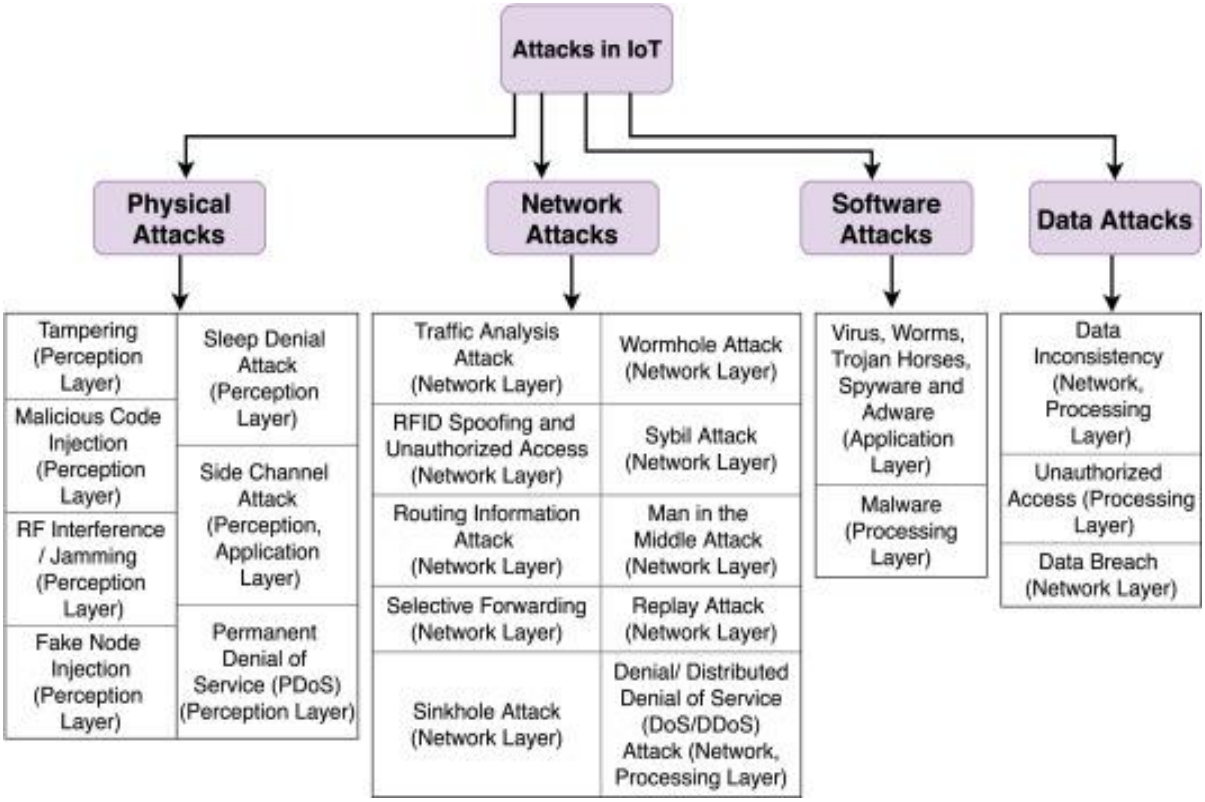

Figure 2.1 Various kinds of cyber-attacks in IoT networks**[24]**

- **Malicious code injection:** In this attack, the attacker compromises a device and launches a malicious code in the device.
- **Side-channel attack:** In these attacks, the attacker by using techniques such as fault attack, timing, etc. collects the encryption keys and later uses those keys to encrypt or decrypt the secure data within the network.

2. Network Attacks: These kinds of attacks are launched by the manipulation of the IoT networks. The attacker does not need to be present near the network or device; these attacks can be launched to far away devices and networks across the globe[25].

- **RFID (Radio Frequency Identification) Spoofing**: The attacker gets access to RFID tags information by spoofs the RFID signal and later sends malicious data using the stolen tag ID.
- **Traffic Analysis Attack:** The attacker manages to listen to the data and information flowing among the devices in the network for stealing the network information.
- **Routing Information Attack**: The routing information in the network is spoofed by the attacker to disturb the network by sending error messages, creating routing loops, etc.

- **Sinkhole attack:** In this type of attack, the attacker makes a node as a sink by compromising that node and attract other nodes in the network to flow network traffic towards that node [26], [27].
- **Wormhole attack**: In this attack, the attack launches a low latency link and makes packets to tunnel through that link from one point to another within this link.
- **Sybil attack:** It is another unusual type of attack in which the attacker creates multiple identities of itself and places itself at various locations in the network. This makes the resource allocation within the network to be unfair [28].
- **Man, in the Middle attack:** The attacker manages to place itself between two nodes in the network and listen to the interaction between those nodes and may use the data according to their intention.
- **DDoS/DoS (Denial of Service)**: In this attack, the attacker makes the target system not serve the requests by legitimate users by denying the services[29]. This attack has affected devastatingly the IoT network, which is discussed in more detail in the next section and hence is the motivation of doing the work in this thesis.
- **Eavesdropping attack:** Another name for this attack is sniffing or snooping cyber-attack [29]. The attack takes advantage of communication over the unsecure network to listen to the data being transmitted over the network. To implement this type of attack by exploiting the weak network connection, make the network data being transferred to them by placing a sniffer which is a software for monitoring network on a server or connected client computer.

3. Software Attacks: These kinds of attacks are launched by invading a software program in the network to damage the network devices [31]. The attacker takes advantage of software and security vulnerabilities in the IoT network.

- **Virus, Adware, Spyware, Worm, Trojan horse**: The attacker infects the network by invading a malicious software for stealing of modifying the data.
- **Malware:** The IoT networks are infected by launching malware in the IoT devices that contaminate the data server or even cloud.

4. Data Attacks: Cloud computing is the backbone of the IoT networks that make it easy to manage, store, and analyze the huge IoT data. The security of data within the cloud gives rise to another challenge for authentication, software update process etc.[32]

- **Data Breach:** This refers to the leakage of confidential and sensitive information to authorized users [33].
- **Data Inconsistency:** The attack may alter the data stored within the cloud or data servers by damaging the integrity of data.
- **Unauthorized Access:** The malicious user can gain authorization to the data and may make legitimate user access to the data.

2.3 DoS/DDoS Attacks

DoS attacks have become a major danger to present computer networks and have become the toughest cyber threat on the internet at the present [34]. DoS attacks target web servers, applications, or systems by misusing the internet to make service unavailable to genuine clients. To understand the DoS attack we can imagine a situation where one is in the queue to the teller, but someone else with no intention of the bank-related transaction, cuts in front you that person and keeps the teller engaged. As a legitimate customer of the client, you are made to wait for the service. The other person becomes a malicious user in this case. As this malicious user leaves the teller another malicious user cut out in the front and keep the legitimate customer waited. This process keeps on going continuously and the legitimate user of the bank keeps on waiting for the service.

To launch a DoS attack the invaders flood the target system with massive requests that result in making the target system exhausted [35]. The exhausted system can no longer serve the requests of the genuine users that result in denial of service to the users of that system. If the attack is launched by a single computer or system to drain the target server is known as a Denial of Service attack. The big server has many resources on hand and hence they can tolerate a DoS attack, to attack such systems the invader makes use of a large number of computer systems to target such a single server to exhaust it so that it no longer serves its users. Such attacks launched with the help of many computer systems is known as a Distributed Denial of Service attacks. DDoS attacks are difficult to get detected and blocked, which demands an advanced defence mechanism to protect the server.

The DDoS attack has got more challenges as compared with the DoS attack. At the present time, the hacker has advanced and evolved. DDoS attacks are turning out to be larger, more repeated, and further advanced. To launch the attack huge terabit and gigabit scale of requests are flooded to the target system. Considering the vulnerability of the IoT system, attackers have weaponized IoT networks with multi-vector capabilities that also have slow detection system

at the server site. Even some DDoS-for-hire services are easily available on the dark networks they provide services to launch DDoS attacks. DDoS attack defence mechanism also requires intelligent Automation to speed detection and superior detection rate. DDoS attacks also become more challenging as broader protection is required as attacks are launched from diverse geographical locations, detection also required much manual intervention, usage is cost-prohibitive, and also lack granular control for a more agile response.

2.4 DDoS Attacks in IoT Networks

Figure 2.2 presents the method of implementation of a DDoS attack in IoT networks; initially, the hacker selects a DDoS master (Bot), an IoT device such as a computer, laptop, etc., by compromising these devices and taking advantage of the vulnerability of that IoT device The attacker then uses that DDoS Bot to further compromise several IoT devices (sometimes thousands) on the networks such as CCTVs, smart light bulbs, VOIP phones, etc. which are known as Zombie bots[4]. These IoT devices connected over the Internet are resource constraint tiny and cheap devices which lack in security controls as these do not have enough processing power and memory, as such the hacker can easily hack and control them by using various technologies and tools such as by cracking the password by applying brute force in the device and sometimes the default password of these devices is never changed. The attacker instructs these zombie bots via the DDoS master to send several flooding attacks to target a specific system, which results in denial of service to the legitimate users of the system. These kinds of cyber-attacks are attractive for hackers as they involve easy implementation of attacks to target large scale and popular websites to disable them. Therefore, a DDoS attack causes tremendous damage to servers and devices on the Internet and make the services unavailable to legitimate users of a system cannot access resources or services[4], [5].

2.4.1 Classification of DDoS attacks in IoT networks

Based on the implementation of attacks, the DDoS attacks can be broadly classified as [16], [36]:

(1) Bandwidth attacks: In IoT networks for the exchange of data, the bandwidth is available. DDoS bandwidth attack consumes network bandwidth by flooding the massive volume of packets on to the target server, which in result, exhausts the server. The exhausted server can longer serve the legitimate user or client and hence causes the denial of service to the user. For example, one of the ways to launch a DDoS attack is to flood massive volumes of TCP, UDP,

or ICMP packets to the target server. The attacker might spoof the source address and prevent identity so that it could not be detected by the dense mechanism in the target server[37].

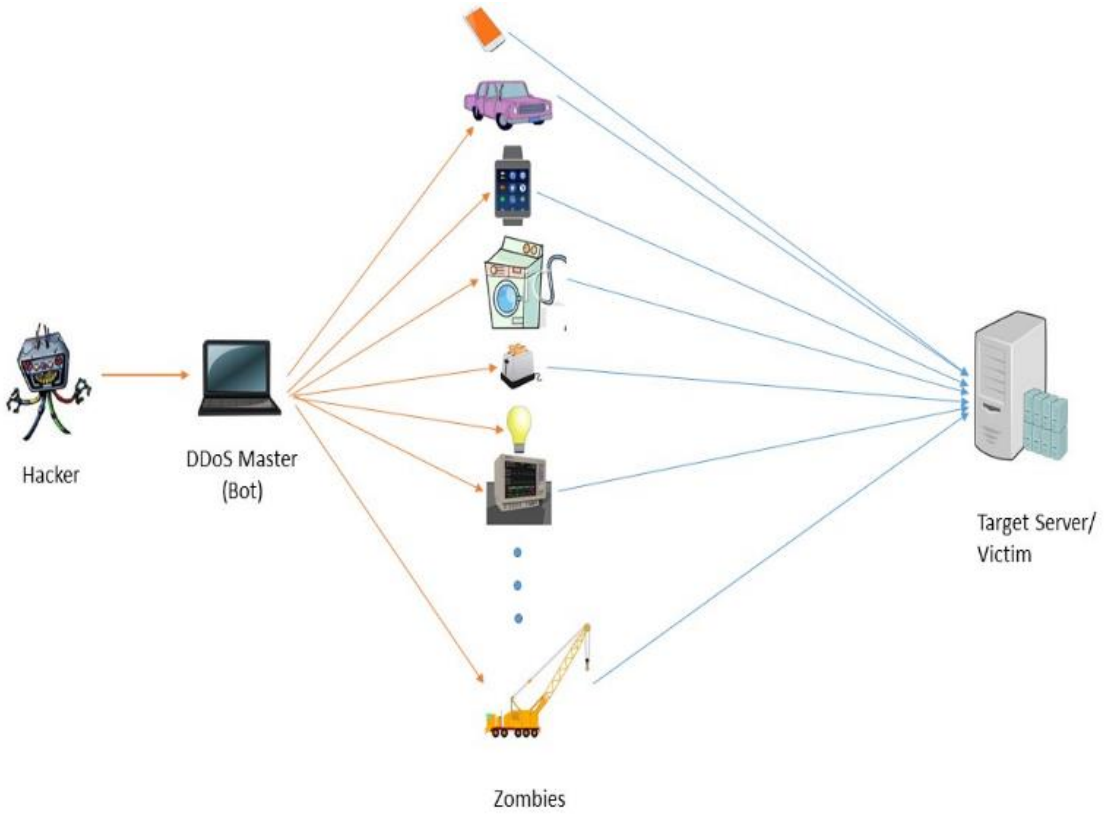

Figure 2.2 Launch of DDoS attack in IoT networks

(2) Application Layer attack: This kind of DDoS attack is launched by tying up computational resources by taking advantage of the expected behaviour of the protocols used to prevent the target server from processing legitimate requests [38]. In this type of attack, the attacker invades the application layer of IoT networks by flooding the web server by HTTP requests [16]. These attacks are generated at a lower rate, which makes it difficult to detect these kinds of attacks. Some of the examples are DNS service-based attacks and HTTP flooding etc.

(3) Infrastructure Layer attacks: Type of DDoS attack is initiated by exploiting the network layer and transport layer vulnerabilities to make the target server not accessible by the legitimate user. Depending on the implementation strategies, these attacks can be further classified as volume-based and protocol-based attacks [24]. The frequent way to launch this type of attack is by using an amplification method or reflection technique. The attacker hoaxes the IP address of initiating source to make a sent request appeared to be an unrequested reply towards the victim, which ultimately congests the victim network. Whereas in the case of the amplification

method, the bandwidth is consumed by generating large replies for smaller requests with the reflection method. It has been discovered that 65 % of the attacks are volume-based attacks; some of the examples of this type of attack are UDP/TCP flooding, ICMP flooding etc. Whereas in the case of protocol-based attacks, the attacker not only consumes the server resource but along with this the intermediate communication equipment such as load balancers and firewalls, etc. are also consumed [39]. Another name for these attacks is resource depletion attacks. Some of the examples of such attacks are namely SYN flooding, ping of death, Smurf DDoS, fragmented packet attacks etc.

(4) Zero-Day attack: These kinds of attacks are launched on day zero by exploiting the unknown vulnerabilities within the IoT networks and existing loopholes[36]. The vulnerabilities within the networks re-analysed and reviewed after the attack has taken place, and appropriate defence mechanisms are introduced against those attacks.

2.5 Recent DDoS Attack in IoT System

As the IoT as evolved, more and more devices are connected to this network; it has got attached by various DDoS attacks in recent times. The review of these DDoS attacks that took place in the recent past is advantageous for knowing tools and techniques and their impact on IoT, used for attacking network systems, may help better to develop countermeasures against these attacks. Table 2.2 illustrates the review of some of the major DDoS attacks on IoT networks. The security firm Imperva's client that belongs to the entertainment industry became the target of the DDoS attack that was launched by exploiting IoT botnet between March and April 2019. Ober 400,00 IoT devices located in Brazil were used for launching the attack over 13 days producing 292,000 requests per minute. The attack was later discovered to be an application layer attack [40]. The attackers employed the same agent as that of a company's application uses, which gave access to the authentication module. The network could not differentiate between the malicious and normal traffic that overwhelmed the server and resulted in the consumption of all the resources of the company. The company offers a security mechanism based on machine learning algorithms, for example, CounterBreach2.0 [41]. However, that DDoS attack was quite enhanced and sophisticated that was detected for 13 days.

Cloud services are the backbone of the IoT system as all the data of IoT devices are collected, processed, and analysed in the cloud. Recently, DDoS attacks have been able to target cloud computing using the important set of features of service provided by the cloud such as auto-scaling, pay-as-you-go accounts, and multi-tenancy. In February 2018, attackers succeeded in

attacking popular online code management cloud-based website GitHub [42] This is the most significant DDoS attack recorded to date with incoming traffic of 1.3 Tbps. This attack was implemented by sending 126.9 million packets per second. As GitHub was using DDoS protection services, so this was detected within 10 minutes of the attack starting. According to the survey, cloud-based services such as Microsoft Azure is the most abused platform by hackers with 38.70 % attacks originated from this, another cloud service provider Amazon Web Services (AWS) has been reported to be used 32.70 % times while Google is being used 10.78 % for flooding the DDoS attacks [43].

Table 2.2 Review of the latest DDoS attacks on IoT networks

Date of Attack	**Organization Attacked**	**Attack Tool**	**Packet format used to launch the attack**	**The device used for attacking**	**OSI Level**	**Type of Attack**	**Explanation of attack principle**
March-April, 2019 [40].	Imperva's client	Exact tool not known but Mirai like properties	Flooding traffic packets	Various IoT devices located in Brazil	Application Layer	Resource, Bandwith	Flooding at the rate of 292000 RPM, Attacker used the same agent as the company's application uses,
February 2018 [42]	Github	Flooding	UDP protocol	Memcached database server	Application Layer	Resource	Memcached database servers were exploited that support UDP protocol that does not require

							authenticatio n
Octobe r 2015 [44]	Imperv a Icapsul' s client	Bashlite malware	HTTP	CCTV and DVR	Applica tion Layer	Resour ce	Attack of web server by mass sending of requests
June 2016 [45], [46]	Bricks & mortar' Jewelry store	Telnet Flooding	HTTP	CCTV and DVRs	Applica tion Layer	Resour ce	Attack of web server by mass sending of requests
Septem ber 2016 [47]	Brian Krebs Websit e	Amplific ation Techniqu e using Mirai botnet	SYN Floods, ACK Floods, Get Floods, POST Floods, and GRE Protocol Floods	Cameras, Thermos tat and Lightbul b	Transp ort Layer, Applica tion Layer	Bandwi dth, Resour ce	Mass sending of TCP and UDP packet (not requiring a previously-established connection)
21 Septem ber 2016 [48]	OVH	Mirai	Massive torrent flood	Routers, CCTV, DVR and other smart devices	Transp ort Layer, Applica tion Layer	Bandwi dth, Resour ce	Mass sending of TCP and UDP packet (not requiring a previously-established connection)
21 Octobe r 2016 [49]	Dyn	Mirai	TCP UDP traffic over port 53	Routers, IP cameras, the baby monitor	Transp ort Layer, Applica	Bandwi dth, Resour ce	Mass sending of TCP and UDP packet (not requiring

					tion Layer		
November 2016 [50]	Deutsche Telekom	Modified Mirai	TCP, UDP	Webcam, Routers, DVR	Transport Layer	Bandwidth	Massive sending of various packets
February 2017 [51]	Unnamed American University	Bruteforce method to unlock password	DNS flooring	Light bulb, vending machine, Lamppost etc	Application Layer	Resource	Attack of a DNS server by mass sending of requests
April 2017 [52]	Various vendor's IP cameras	Persirai	UDP flooding	IP Camera	Transport Layer	Bandwidth	Mass sending of UDP packets (not requiring a previously-established connection)

In October 2015, attackers were able to compromise more than 900 CCTV cameras spread around the globe and used them to attack Imperva Incapsula's client (name disclosed) websites by launching a distributed denial-of-service attack. The target of attacks was a rarely used asset of a large cloud service, catering to millions of users worldwide. The attack was made by flooding HTTP requests up to 20000 requests per second [44]. The malware used by the attacker was the ELF binary for ARM Bashlite. A similar attack was documented in June 2016 that compromised more than 25000 DVRs and CCTV cameras. The attack was deployed by flooding about 50000 HTTP requests per second at its peak to Bricks & mortar' Jewelry stores website as the infrastructure provisioned for such websites can handle only a few hundred or thousand connections at a time, so these attacks can easily cripple a small website. It was found that botnet was distributed globally including Taiwan (24 percentage), U.S. (16 percent), Indonesia (9 percent), Mexico (8 percent), Malaysia (6 percent), Israel (5 percent), and Italy (5 percent) and at other parts of the world as well. Initially, the attacking botnet was not known, but it was claimed to be attacked by the Point of Sale (POS) Trojan [45], [46].

In September 2016, a huge DDoS attacked security consultant Brian Krebs website KrebsOnSecurity which reached 665 Gbps in size. The unknown attackers attacked using garbage web attack methods, via SYN Floods, ACK Floods, Get Floods, POST Floods and GRE Protocol Floods by hijacked internet-connected devices like cameras, thermostats, routers and lightbulbs. Attackers used common amplification techniques that to enable a small botnet made by compromising systems to turn a small attack into a larger one [47]. On 21 September 2016 French OVH hosting company became the victim of 1.5 Tbps DDoS attack largest DDoS attack ever recorded implemented using hundreds of thousands of comprised IP cameras, Routers and DVRs [48]. This attack was initiated by flooding massive torrent of traffic on 20 September 2016 to OVH's website via 152,463 hacked low powered cameras and smart devices which increased the future in the next 48 hours. The attack was made by using various types of traffic, which included Generic Routing Encapsulation (GRE) traffic, which is novel to DDoS landscape. It was announced by the founder of OVH later that servers of its company were hit by numerous incidents surpassing 100 Gbps concurrently concurring at 1 Tbps DDoS assault. One of the incidents was noticed to reach 93 Mbps and 799 Gbps.

On 21 October 2016, Dyn server, a firm that operates lots of the internet's DNS infrastructure in America, was hit by a DDoS attack by a new weapon called the Mirai botnet [49]. Major sites affected by this attack were Amazon, Netflix, PayPal, Spotify and Twitter in Europe and the USA. Mirai botnet first infects the internet-connected devices through Telnet services such as webcams, routers and CCTV that fun on BusyBox, it then using brute force relying on the small dictionary of potential username-password pairs to deduce the administrative credentials of these internet-connected devices and begins scanning the Internet for other hosts running Telnet servers. Mirai scans random public IP addresses through TCP ports 23 or 2323. Mirai botnet source code was released in public on September 30, 2016, in the hacking community forum Hackforums and on October 25th on GitHub, and since then, it caused an increase in DDoS attacks across the globe. The attack executed attacks through internet-connected devices such as printers, IP cameras, baby monitors and routers.

In November 2016, by using a modified version of Mirai botnet, 900,000 Deutsche Telekom Customers were knocked offline by launching DDoS attacks by infecting routers, which disrupted telephony and television services and internet connections causing million pounds damage to the company [50]. One unnamed American University was attacked by more than 5,000 internet-connected lightbulbs, vending machines and lamp posts on its campus in Feb 2017 to continually search for seafood [51]. The university's network connectivity became

unbearably slow and inaccessible as the IoT devices were making hundreds of Domain Name Service (DNS) lookups related to seafood every 15 minutes. It was found that the malware-infected connected devices by launching a brute-force attack to guess default passwords that had not been changed.

In April 2017 a new IoT botnet was discovered named Persirai which shares Mirai's codebase which targeted over 1000 different models of IP Camera. Attack was discovered by cybersecurity researchers at Trend Micro that was affecting 122,069 IP cameras across the globe [52]. It attempted to access the IP camera web interface via TCP Port 81, as IP cameras use Universal Plug and Play (UPnP) network protocol that allows devices to open a port on the router and act like a server. After logging into the interface, a command injection to force the IP Camera as performed by the attacker to connect to a download site and download and execute a malicious shell script, after the download and execution of samples, the malware deletes itself and run only in memory. The malware proliferates by exploiting a documented zero-day flaw that lets attackers directly obtain the password file.

2.6 IDSs for IoT Networks

IDSs have been employed extensively by online web services and other online service providers. The survey on the various types of IDSs existing is crucial for developing an advanced IDS that is capable of dealing with new upcoming types of cyber threats on Internet-connected networks. The focus of this thesis is on building a defence system for IoT networks, so a detailed review of the existing IDSs that exist specific for IoT networks would be very beneficial for developing advanced IDS.

2.6.1 Intrusion Detection System

IDS is an essential tool for providing cybersecurity to an IoT network that detects the intrusions that is any activity that violates security policies within the network. IDS works by collecting passive network traffic to monitor and analyze the flow of data in the networks and services and looks for the detection of any vulnerabilities in the network [53]. The job of IDS is to monitor network traffic and secure the network and connected devices against any intrusions affecting the confidentiality, availability and integrity of the network information. The IDSs can be divided among four categories as network-based IDSs that monitor network traffic, host-based IDSs that monitor host system log files, hybrid-based IDSs and distributed based IDSs depending on the monitoring environments[54]. The more detailed description is provided in Table 2.3.

Table 2.3 Types of IDS

IDS	Description	Advantages	Limitations
Host-based IDS (HIDS)	The IDS is installed on a particular server or mobile device for monitoring the network by analysing the operating system audit book for the detection of intrusion.	No additional hardware required Can work in an encrypted environment Capable of detecting Trojan horse Monitor network traffic at the transport layer	Can work for already known attacks Several disadvantages of being at host cause limitation to network
Network-based IDS (NIDS)	Network-based IDSs monitors network traffic and information on application protocol activity.	Detection rate better than Host-based IDS Does not affect existing infrastructure within the network so easy to deploy Cost-effective	Encrypted packets are hard to be monitored Needs to be running all the times Located far from the host
Hybrid IDS (HIDS)	These IDSs are the hybridization of NIDS and HIDS. The central agent checks the overall network whereas the mobile agent moves to each host for checking system log file	More efficient than NIDS and HIDS Facilitate to quantify attacks Protect the entire network Provide additions security	Does not prevent the attacks Massive data is collected and generated for monitoring network Every expensive Generate false positive and false negatives.

Distributed IDS (DIDS)	Many IDS are deployed at faraway sensors and hosts to detect any intrusion. The collected information is carried forward to centralized control. It can be NIDS, HIDS, or a combination of both.	Central server control and analysis and monitoring Advanced detection Utilizes traffic data from faraway sources	The intruder can interrupt between remote hosts and central control Huge data is shared regarding the network flow.

Based on the detection methodologies employed, the IDSs can be divided into three subclasses [53].

(1). Misuse detection IDS

The misuse detection IDS maintains a database that stores all the information of already known intrusions' signature and patterns [55]. The IDS uses that database to detect similar known attacks. This method of IDS needs to update and maintain the signature and pattern database. Another disadvantage of these kinds of IDS is the cost of signature matching, network packet overload and huge number of false alarms.

(2). Anomaly detection IDS

These kinds of IDSs work to detect any anomalies within the network that is some unusual behaviour caused by external intruders. The anomaly detection IDSs create a model of normal behaviour in the network and keeps on updating that by collecting data from other users and look for any behaviour that is deviating from normal behaviour. Some of the techniques for implementation of these IDSs are namely as data mining, statistical models, rule models, payload model and machine learning models. The latest development in these IDSs is the use of deep learning methods. The proposed work in this thesis is based on the deep learning techniques.

(3). Specification-based IDS

This is a monitoring and detecting system that make use of already specified security features for categorizing the normal behavior of the network. The security features are defined based on

the functions and security policies of the networks. Any operation mentioned in the security specifications is considered as an abnormal behavior that is causing security violation [56].

2.6.2 Survey of IDSs for IoT networks

In this section, the IDSs specific to IoT networks is reviewed. The brief details about the discussed IDSs are presented in Table 2.4 in a tabulated form. A hybrid IDS method using the advantages of both machine learning data mining techniques is proposed in [57] for IoT networks. The algorithms employed for development as NIDS are fuzzy c-mean (FCM) clustering and PCA (principal component analysis) algorithm. The dataset is reduced by doing feature selection using PCA and FCM is applied for clustering. The experiment results achieve a high accuracy value with a low false-positive rate. The KDD-CUP'99 datasets are used for an extensive evaluation of the proposed method for the detection of DoS, U2R, R2L, Probing, Normal data classes. The weakness of this method is that it is not tested on an IoT kind of environment and also the reconfigurable hardware devices are not tested. The evaluation is conducted on a centralized placement of IDS. [58], presents a NIDS that leverages collaboration among various IoT devices for the detection of intrusion. The model of the projected work is conducted on the Contiki operating system and the dataset is collected in the same environment. The power and time consumption are considered as performance metrics.

Table 2.4 Review of IDSs for IoT networks

IDS	Detection	Type	Datasets	Methodology Employed	Attack Detection	Limitations
Deng et al. [57]	Anomaly	NIDS	KDD-CUP'99	Machine Learning, Data mining	DoS, U2R, R2L, Probing, Normal	The method is not tested on IoT based smart environment that include reconfigurable devices such as FPGAs

						Centralized placement of IDS is evaluated
Arshad et al. [58]	Anomaly	NIDS	Dataset collected using Cooja simulator in the Contiki operating system	Distributed and collaborative	Routing and application-specific attacks	The results are not specific to a particular class of attack The dataset employed for evaluation of the proposed method is not standard. Only the power of the devices is considered for comparing the results.
Abhishek et al. [59]	Anomaly	NIDS	Dataset collected from the simulation of the IoT network in Matlab	Statistical model	Advisory present at the physical layer	The data employed for evaluation is not standard datasets Based on theoretical foundation without the need for training
Liu et al. [60]	Anamoly	NIDS	Data collected using fuzzy clustering	Machine learning, data mining	High-risk data, Low-risk data	The data collection procedure is not mentioned

						Does not include any particular cyber attack
Khan and Herrman n [61]	Anomaly	Hybri d IDS	Data collected in Matlab simulation of IoT healthcare system	Protocol model	Sinkhole, selective forwardin g, version number	Validated results on simulation-based data Proposed method valid for routing-related attacks only
Anthi et al. [62]	Anomaly	NIDS	Data collected from IoT smart Home Testbed for four consecutive days	Machine Learning	DoS	The data employed for evaluation is not standard datasets ML features like Payload, Ingoing/Outgoin g ratio not considered Not tested for other kinds of attacks
Amouri et al. [63]	Anomaly	NIDS	Data collected by simulation of IoT networks in Cooja under Contiki environmen t	Machine Learning	Blackhole	No feature selection method is suggested No standard datasets used for evaluation of the proposed method The expensive method in terms

						of computation cost
Yang et al. [64]	Anomaly	NIDS	KDD-CUP'99, AWID	Machine Learning	DoS, U2R, R2L, Probing, Normal	Results obtained are not on distributed resource-constrained devices data. The query strategy for training IDS not specified Results provided are not specific to attack types
Fu et al. [65]	Anomaly	NIDS	Data collected from testbed created using a Raspberry Pi, Android phone, router	Signature model, the protocol model	Jam-attack, false-attack, reply-attack	Proposed method evaluated on a standard dataset Data used in not real word data Does not evaluated on other attack types
Bostani and Sheikhan [56]	Anomaly, specification	Hybrid IDS	Tested on WSN simulator created on .Net framework	Machine learning, signature model	Sinkhole, selective forwarding attack, wormhole	The proposed method only valid for routing attacks The high false-positive rate

Arrignton et al. [66]	Anomaly	HIDS	Data collected on a software simulation of the smart home IoT system	Machine learning	Abnormal behavior in the system	Specific to smart home IoT applications Does not detect any external intrusion in the system Motion sensing not included for simulation Does not include real-world data
Mohan et al. [67]	Anomaly	HIDS	Data collected from IoT nodes	Rule model, the Signature model	Intrusions in the IoT networks	The results are not very efficient Limited detection based on signature Validated on limit data Does not include real-world data
QD La et al. [68]	Rule	HIDS	Tested on honeypot enabled networks	Game Theory	Intrusions	Limited results provided in support of method No real-world data
Satnam et al. [69]	Anomaly	NIDS	Data collected from	Machine learning	Attack on wi-fi protocol,	Does not include real IoT data

			software simulated networks		DNS protocol, HTML protocol	Only works for routing attacks
Villalobos [70]	Anomaly	NIDS	Data collected from network simulated using Apache Storm and Apache Kafka tools	Unsupervised Machine Learning	DDoS Attacks	No standard dataset is used for evaluation of the proposed algorithm. The proposed algorithm has been tested on limited data.
Ili Ko [71]	Anomaly	HIDS	Netflow based simulation used for collecting Data	Unsupervised Neural Networks	DDoS Attacks	The data used is not much in quantity as required for machine learning algorithms. No standard dataset employed for comparing the result of the proposed method.

In [59], a statistical method for the detection of advisory at the physical layer is presented. The simulation is conducted on Matlab and the dataset is collected from the same environment. A NIDS based on suppressed fuzzy clustering and PCA algorithm is presented in [60] for IoT networks. The proposed algorithm has collected datasets using data initialization using fuzzy clustering and applied PCA for feature selection on collected data. The data is classified as high risk and low risk depending upon the frequency of the data. [62] presents a machine learning-

based NIDS for anomaly detection for IoT networks. The Naive Bayes classifier is employed for the classification of data labels on Weka software tool. The datasets are collected by using Wireshark tool on a smart home IoT system testbed with normal and DoS attack data. The shortcomings of this method is that the proposed method is not evaluated on a standard datasets, although the results on collected data is nearly 97.7% accurate but that is collected in a controlled environment and limited number of IoT devices being in communication. A block hole attack detection method is proposed in [63] using a machine learning technique. The NIDS works on two stages; the first one is that local detection is done by employing a dedicated sniffer that works on supervised training using a decision tree algorithm to classify the instances. These classified instances in the second step are sent to the super node, where a liner regression method is applied for time-based profile generation for classifying malicious and normal nodes. The datasets are collected by simulation of the IoT network in the Cooja simulator on the Contiki environment. The accuracy of the proposed method is 100 %. The drawback of this method is that it is not evaluated on standard datasets and simulation network data are not near the real-world data also any feature selection methods is not suggested because the dataset only has 6 features. A XGBoost machine learning method based on NIDS for anomaly detection is proposed in [64] for active learning. The results obtained in the experimentation are better than other supervised training methods for IDS. The cons of the proposed method are as the results discussed are not specific to attack types and also the query strategies for the active training system presented are not specified. A hybrid NIDS is proposed in [65] using the protocol model and signature model. The data used for evaluation of the proposed method is collected by creating an IoT tested using a Raspberry Pi, Android phone and a router.

A NIDS for detection of routing attacks, namely selective forwarding and sinkhole attack, is presented in [56]. The proposed method is a hybrid of the anomaly and specification-based IDS. The proposed method works by placing agents at router nodes for analysing the behaviour of host nodes; another agent located in the root node works for the detection of an anomaly. The authors have validated their proposed method on a WSN simulator created using .Net framework written in C# language. A behaviour-based anomaly detection HIDS for IoT smart home is presented in [66] that is based on immunity inspired algorithms for differentiating normal behaviour. The proposed method incorporates a simulator including a discriminator amongst human activity orchestrated in a smart IoT system. The simulation of the smart home system id based on the placement of the sensor at various locations to collect nearby data, which is further used for detecting the behaviour in the system. [67] presents a framework to display protection threats feasible on IoT devices. The method incorporates modules such as record

size, alert generator, anomaly detector. The record capture module collects the utility level recorded, delivery and network headers of the visitors that are going into the IoT tool. The anomaly detector module makes use of the signature-based technique to stumble on threats. The proposed framework is tested on a testbed comprising of Arduino broads with wiznet ethernet defend because the IoT device communicates with Samsung Android smart cell phone over a bridge connected via Wi-Fi. Snort IDS on the bridge is employed for generating alerts for the intrusion.

In [68], a game theory version to investigate the trouble of misleading assault and dense in a honeypot enabled network within the IoT networks. A Bayesian signalling recreation of incomplete data is formulated. The method works in two stages, first stage one-shot version and then inside the repeated scenario. The Bayesian notion update scheme is used in the game theory model. Proposed game work to account for the presence of false-positive and false negatives in the defender's IDS. [69] presents a NIDS including micro AB-IDS exploiting machine learning. The method includes analysis of threat model and feature selection and exploited protocol foot printing to create a machine learning method for classifying the behaviour as normal or abnormal. The proposed method is analysed by designing micro IDSs for identification of attacks on DNS protocol, HTML protocol and wi-fi protocol. The experiment has achieved a high accuracy value with low false negative and false positive. However, the proposed method is not evaluated on real-life IoT datasets.

An advanced unsupervised method for the detection of DDoS attacks is presented in [70]. The proposed algorithm is evaluated on distributed and collaborative design for online elevated rate DDoS attacks. The proposed work focuses on countering the challenge of developing defence system with a generic attribute that can identify any type of DDoS attack in the real world, which is independent of the network layer or protocol used. An unsupervised machine learning algorithm has been exploited in their method. The authors conclude that the proposed method can be scaled and incorporated with the present network infrastructure and is independent of underlying technologies being used in the network. An Internet service provider (ISP) based DDoS detection method is proposed in [71]. The proposed method is a hierarchical two layered self-organizing map with two-fold feature extraction. The authors have claimed that a mitigation method on ISP domain which as between user and internet, is more efficient. An unsupervised neural network is exploited. Their proposed method outperforms K-Mean model by 3.04% F1 score.

2.7 Survey on Feature Selection Methods for IoT networks

Various feature selection methods have been proposed in recent publications to enhance the execution of the classifiers employed. In [19], authors have discussed major security issues existing for IoT networks and state of the art solutions. In [72], it was found that filtering methods could lead to a misleading selection of features as filtering methods compute average scores on dataset classes and predict class labels accordingly. That may result in the non-selection of a feature that might be especially relevant for a class label. So, the authors proposed a multi-objective approach for feature selection. They have considered two objectives namely relevance and redundancy of class labels for feature selection. In this work, Growing Hierarchical Self-Organizing Maps (GHSOMS) is used, which is an unsupervised clustering machine learning method that combines a new unit labelling method. DARPA/NSL-KDD datasets are used to evaluate this method. They have concluded that their method produces an efficient determination of the wining unit as output and provides a maximum detection rate of 99.8% and 99.6% with normal and anomalous traffic respectively. [73]–[77]authors have proposed a feature selection method based on limited criteria using the NSGA-II algorithm for network anomaly detection and pattern classification. They have evaluated their work in terms of classification accuracy and time of execution for different benchmark datasets.

A feature selection wrapper method is proposed in [76] based on single objectives to maximize Information Gain (IG) for the detection of DDOS attacks using Bayesian networks (BN) and decision tree (C4.5) classifiers. Their method is evaluated on the KDD'99 dataset and DDoS dataset collected by Telecom Bretagne France on real-time computer networks. In this work, authors found that massive network traffic data work high-speed IDS are challenging for efficient processing. Based on their work, they state that only important features should be used for the detection of the attack. Similar work is proposed in [77] where two wrapper methods of feature selection named RF-FSE and RF-BER have used IDS with a decision tree machine learning classifier. In their work, four objectives were used. They have evaluated their proposed methods on three benchmark datasets. In this work, they have used an RF classifier with CV-parameter selection methods to validate the performance of the proposed algorithm.

In [78], an NSGA-III algorithm which improves NSGA-II with reference points is proposed for feature selection exclusive by IDS to reduce computational complexity and improve the accuracy of the classifier focusing on the imbalance class problem of learning classifiers. The Jaccard-Index has been used for measuring the performance of their method on three benchmark datasets NSL-KDD, KDD'99, and Cure-KDD. A jumping gene adapted NSGA-II

proposed in [26], [79] which is inspired by real transposons present in DNA which can jump in and out of chromosomes for the optimization problem of an industrial low-density polyethylene tabular reactor by employing multi-objective optimization algorithm with two conflicting objectives. Different variants of jumping gene-based NSGA-II such as NSGA-II-mJG, NSGA-II-saJG, NSGA-II-aJG and NSGA-II-sJG 2 have been investigated. It is concluded in [26] that the NSGA-II-ajG algorithm works superior than the other two procedures in evaluation of computation and convergence. A DDoS attack detection method based on semi-supervised learning for an IoT network is proposed in [80] using an ELM classifier. They have used the NSL-KDD and KDDCUP'99 datasets for evaluating their algorithm, which provides better performance in comparisons with the centralized detection of attack framework in terms of accuracy. They have achieved maximum accuracy of 86.53% with a deduction in runtime by 11 milliseconds.

2.8 Review of Deep Learning technique for cyber-attack detection

A deep learning-based method for the detection of distributed attacks in fog-to-thing computing is proposed in [81]. This work illustrates the drawback of cloud computing in IoT networks, as it is centralized processing which is not appropriate for large IoT networks as it requires the managing of cybersecurity at the edge of the system. Deep learning has been proven in the field of big data areas, so for IoT networks, a fog-to-node method is appropriate for the massive IoT networks generating huge data. This work is conducted on NSL-KDD datasets by employing stacked autoencoder along with SoftMax as a classifier and compared with a shallow learning model based on performance metrics such as accuracy, false alarm rate and detection rate. The author also demonstrated the fruitfulness of distributed parallel computing employed on fog to node model as improved accuracy and efficiency of attack detection.

In [15], another fog-to-node methodology based distributed attack detection scheme for IoT networks using deep learning method is proposed. The authors have suggested that because of the growth of more and more zero-day attacks on the IoT networks, the machine learning based IDSs are facing challenges for detection of the intrusions in the networks that demand advanced IDSs based on deep learning methods to be implemented. The feasibility of the application of deep learning is suggested for practical implementation because of the improvement in the neural network algorithms and the advanced CPUs and GPU's availability at present. Another interesting feature of deep learning is mentioned as the ability of deep learning algorithms for performing high-level feature extraction that makes it an ideal choice for novel attack detection. The apache-spark software has been employed for doing distributed programming, both deep

model and shallow model experiments are conducted for distributed processing and the results of the experiments are convincing for the application of deep learning for attack detection.

An intelligent intrusion detection system employing deep learning approach is proposed in [82] named as scale-hybrid-IDS-AlertNet. The author has proposed a deep neural network for the detection of various kinds of cyber-attacks. The experimentation has been performed conducted for two kinds of IDSs, first for NIDS and second for HIDS. The evaluation is conducted on KDDCup 99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS 2017 datasets for 1000 epoch with varying learning rates between 0.01 to 0.5. The proposed method is also compared with machine learning algorithms.

In [83] author has proposed a self-taught deep learning based autoencoder in combination with SVM (Support Vector Machine) for intrusion detection in the network. Deep learning is employed as a feature selection method in an unsupervised manner to reduce training and testing time and also improve the performance by increasing the accuracy of the SVM classifier. The author has compared the proposed method for both binary and multiclass classification along with a comparison with other shallow machine learning algorithms such as random forest, Bayesian naïve and J48. The proposed method has provided better results in terms of performance such as accuracy in comparison with other proposed methods. The authors have compared the proposed method with machine learning algorithms and conducted experiments to test the method on the distributed environment. An IDS based on two-stage deep learning model based on a stacked auto-encoder with soft-max classifier has been proposed in [84] In the first stage, probability score value has been used as initial response for classifying network traffic as normal and abnormal, the output of the first step is used as additional feature for the detection of attack in the second step. A low-cost DSAE method for NIDSs for feature selection is also introduced This work is evaluated on KDD99 and UNSW-NB15 dataset, with achieving 99.99% and 89.13% accuracy respectively. In [85], a different approach for simulation of the network is introduced by exploiting software-defined networking (SDN). It is mentioned that the SDN is a cost-effective, adaptable, manageable and dynamic architecture ideal for dynamic and high bandwidth network applications. Another deep learning-based method for traffic monitoring and detection is presented in [86]. A method for NIDSs involving deep learning using nonsymmetric deep autoencoder for supervised feature learning is introduced in [87].

In [88], the authors have provided a unique application of deep learning Autoencoder for feature selection of the datasets for doing feature selection to reduce the dimensionality of the data. The DBN consisting of multilayer Restricted Boltzmann Machines and a single

backpropagation layer has been employed as a classifier for the detection of the malicious code. The proposed algorithm is evaluated on KDDCUPP'99 datasets for five categories, namely DoS, Normal data, the user to root, probe and remote to local, and is also compared with single DBN. The results demonstrate that their proposed method has achieved better accuracy for detection with a reduction in time complexity. A novel deep learning-based IDS is proposed in [89] for in-vehicle network security. Another autoencoders based method for anomaly detection with nonlinear features reduction is proposed in [90]. An anomaly detection approach using deep learning with autoencoders is proposed in [91].

A hybrid image-based deep learning approach for the detection of malware is proposed in [92]. The authors suggest that because of the wide variety of devices connected over the Internet are producing massive data; the security risks are increasing as the malware attacks are increasing and suggest having improved new methods to math complexity of the data-intensive environment — this method using machine learning and deep learning model for distinguishing benign and malware binaries. The proposed methodology consists of three subsystems, one system is based on an unsupervised learning model and the other two are based on supervised learning models. This proposed method has achieved 99.00% accuracy. Similar image-based deep learning including comparison with gist-based approaches are proposed in [93]. The findings in this paper suggest that both gist-based and deep learning performance is similar to malware detection. [94] introduces a deep learning-based method for malicious code variant detection. [95] also presents an image-based deep learning approach for malware identification. Similar image-based deep learning methods have been proposed in [96], [97] they all are efficient for the detection of DDoS cyber-attack achieving high accuracy values

A comparison of shallow and deep neural networks has been proposed in [98]. The author has used KDDCup-'99' dataset to train and test the proposed method with a learning rate of 0.1 and compared the results obtained with other machine learning methods using recall, precision and accuracy as performance metrics. In their research author conclude that deep learning is a promising technology for the cybersecurity field and in conducted work deep neural network model with three layers performed best in comparison with other models.

A deep learning model based on RNN (Recurrent Neural Network) by employing Bidirectional Long Short Term Memory based (BLSTM-RNN) for the detection of the botnet is proposed and compared with LSTM which is an RNN model in [99]. The author has generated a dataset for this work for including four attack vectors as used by Mirai botnet. They have tested and validated their proposed method on four attack vector as Mirai, UDP, DNS and

ack. The proposed method has shown to be performing well for Mirai, UDP and DNS attack vectors with accuracies as 99%, 98% and 98% respectively but in case of for ack attack vector the performance based on the results obtained is not efficient that need more data for training.

A robust malware detection IDS for IoT networks in battlefield deploying deep learning is proposed in [100]. The proposed method exploits the OpCode sequence of IoT devices by transmuting it into a vector space for applying a deep eigenspace training method for the classification. The dataset used by evaluation of the proposed method is collected by setting an ARM-based IoT network with 1078 instances of benign and 128 instances of malware samples generated using the VirusTotal platform. [101] presents a real-time IDS to secure IoT networks based on deep learning by detecting malicious traffic. The proposed method is flexible enough to be deployed on various networks using different communication protocols. The evaluation of the proposed method is conducted on real-world data traces, including evaluation with scalability. IoT network using Raspberry Pi is implemented for the data collection for doing experimentation. The tool named Scapy that is an open-source tool is employed for penetration for trying out the structure, to obtain these functions by shedding down every network package. Five million network instances are trimmed by entering data pre-processing to get an input dataset of 59529 instances. The simulation begins network operations and then other simulation with a combination of nasty network communications.

An illustrated literature survey and a brief tutorial on machine learning and deep learning methods for cybersecurity are given in [102]. They have discussed various problems existing in datasets available for IDS training and testing and also the challenges in employing machine learning and deep learning for cybersecurity. The author has raised the problem of training both the methods as the network data update very fast, and this led to the retraining of the models so the author has suggested lifelong training as future work.

An RNN based IDS named RNN-IDS is proposed in [103]. The author has compared the proposed method with other traditional classification methods such as J48, naïve bayesian and random forest. The authors have evaluated the proposed technique on the NSL-KDD dataset and have accomplished 99.81 % accuracy with 80 hidden nodes and a learning rate of 0.1. In [104], a novel IDS is proposed named DFR (Deep-Full-Range) which is a combination of CNN, LSTM and Stacked Auto-Encoder. The method is evaluated on ISCX VPN-nonVPN and ISCX 2012 IDS dataset which as achieved 95.69% precision value, 98.52% recall and 97.1% F1-Score value. DeepDefense is proposed in [105], a recurrent deep learning approach that can automatically extract high-level features for the detection of DDoS attacks. The author has

evaluated the proposed method on the ISCX2012 dataset. In their work authors have compared different deep learning models, among which LSTM based model have recorded to achieve the highest accuracy of 97.966% whereas CNN combined with LSTM has achieved an accuracy of 95.896 %. ALSTM based method in the fog-to-thing network for IoT networks is proposed in [106]. In the proposed architecture, the training and detection function is locally performed on each IoT node and coordinating node compute and distributes the update of models and parameters to each node. This method is evaluated on the ISCX dataset with 128 batch sizes in 15 epochs. This has achieved 99.91 % accuracy in the case of binary classification and 98.22 % in case of multiclass classification. An interesting feature selection based method is proposed in [107]. In this method authors have used principal component analysis (PCA) on dataset for selecting features in the first step and applied LSTM-RNN for the classification of the attack. The proposed work in evaluated on NSL-KDD dataset, which has achieved 98.85% accuracy.

2.9 Artificial Neural Network

ANN (Artificial Neural Network) is a network structure inspired by the biological neural network [108]. Figure 2.3(a) presents the structure of the natural neurons; several dendrites are present in the neuron structure that serves to receive input as a signal through synapse and carries the information to the neurons. If the input signal surpasses the threshold value, the neuron gets activated and emits signals through the axon. The signal is sent to another synapse, and so on, this way the signals are transmitted in the brain. Figure 2.3 (b) illustrates the structure of the artificial neuron; it consists of input, activation function with weights and output. The input function resembles the synapse of biological neuron denoted as x1 and x2, the activation function which is a mathematical function presented as f(e) calculate the threshold using weights w1 and w2. The input is multiplied by weights so if the weights are higher the stronger will be the input signal. The value of weights could be positive or negative. In the case of the negative weights, the input signal is multiplied by weights that result in the inhibition of the input signal. The desired output can be obtained by adjusting the weights of the neurons. The process of adjusting the weights is carried out by using algorithms and is known as training or learning. The output function denoted as ‘y’ computes the output of the neurons.

2.10 Feed Forward Neural Network

The ANN is broadly classified into two categories, namely feed-forward networks and recurrent networks. The information flows in one direction, from input to output in case of feed-forward networks, whereas in the case of recurrent networks, the information can flow in the

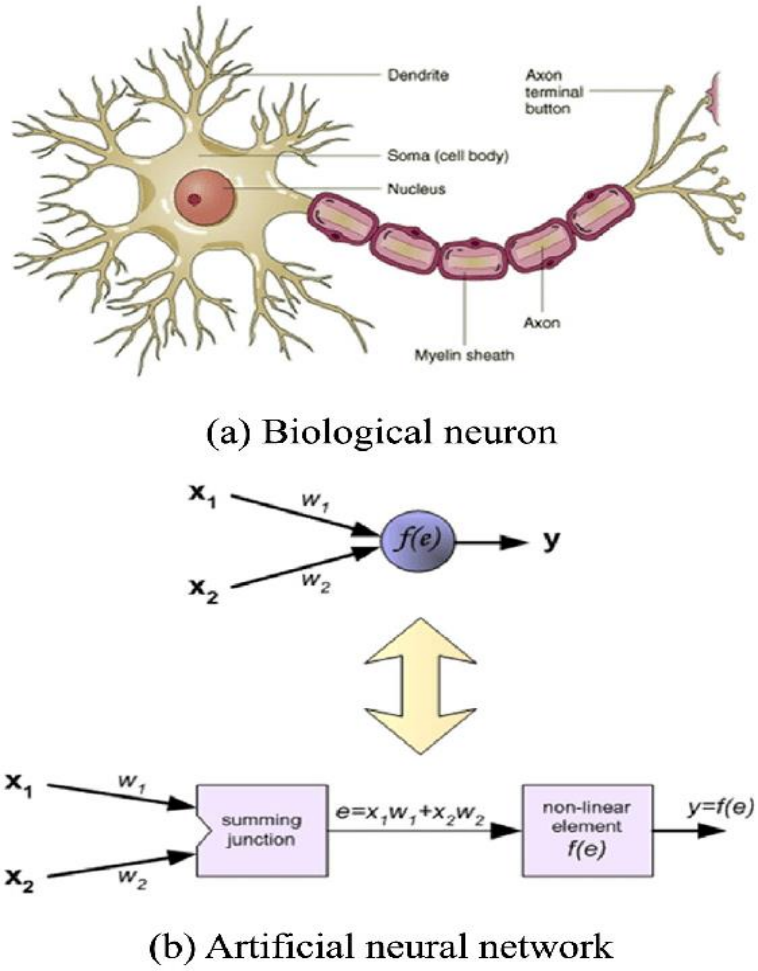

Figure 2.3 (a) Biological neuron structure (b) Artificial neuron structure [109]

opposite direction as well. The structure of the feed-forward neural network is presented in Figure 2.4, and this network consists of a total of three layers with four neurons in the input layer, three hidden layer neurons and one neuron in the third output layer. The input to neuron and neuron to neuron connections are adjusted by a weight w. Each neuron has an extra input with a constant value of one. This extra input is modified by weight known as bias. Equation (2.1) represents the processing on hidden layer on getting input and output result O_c to the neuron of the next layer.

$$O_c = h_{Hid}\left(\sum_{p=1}^{P} i_{c,p} w_{c,p} + b_c\right) \tag{2.1}$$

$where\ h_{Hid}(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function of the neuron, c is the neuron in the current layer, the previous layer is represented as p, and n is a next layer neuron. $i_{c,p}$ is input to current layer c, $w_{c,p}$ is the weight of connection from input p to c, b_c is the bias.

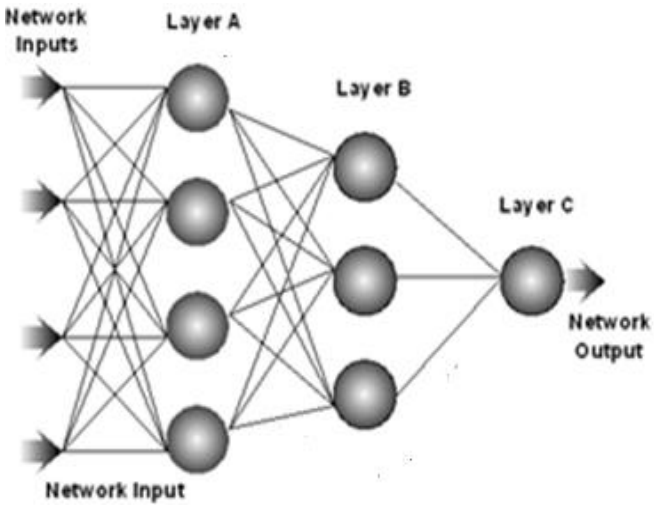

Figure 2.4 Feedforward neural network with three layers

The output layer calculations are depicted in equation (2.2) on getting inputs and produce the output result O_c.

$$O_c = h_{Out}\left(\sum_{p=1}^{P} i_{c,p} w_{c,p} + b_c\right) \tag{2.2}$$

$where\ h_{Out}(x) = x$ is liner activation function, c is the neurons in the current layer, p is a number of neurons in the previous layer, and n is the number of neurons in the next layer. $i_{c,p}$ is input to current layer c, $w_{c,p}$ is the weight of connection from input p to c, b_c is the bias.
The hyperparameters described below are important to understand for building and improving the performance of the DL models.

2.10.1 Loss Function

The loss function evaluates the functioning of the neural network by comparing the output of the neural network with the actual value of the output. The pictorial representation of the loss function is presented in Figure 2.5. The obtained output of the neural network is Y_Output, and the actual output is Y. If the difference between both the output is high, the value of loss function will be high, and it will be low if the value of variables will be similar. During the training process, if the value of loss function comes out to be low than the weights of the neural network will not change much but if the value of the loss function is high than the weight of the network will change farther than usual.

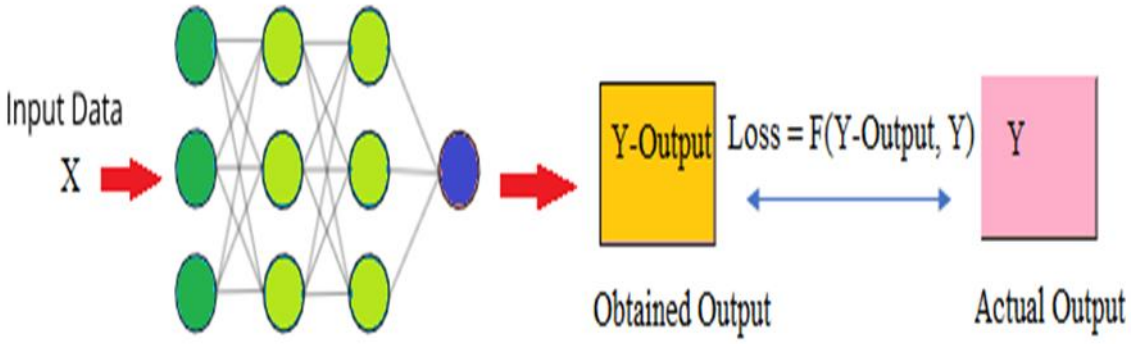

Figure 2.5 Loss function in a neural network

There exist many loss functions that are used according to the requirement in the application of neural networks. In this work, a binary classifier is developed, so here the loss function for the same classification is discussed. The activation function is critical component in a neural network that determine whether a neurron would be activated or not. It do so by evaluating the weighted sum and add bias with it. The output layer in a binary classifier has only one neuron node which output 0 or 1 so, in this case, the sigmoid function comes out to be the best activation function in this case. Sigmoid activation function is a nonliner unction,it takes real value number as input and produce output in the range of 0 and 1. The sigmoid functions ar quite useful for the last output layer of deep learnng model particulary for the binary classiciation.

2.10.2 Binary Cross-Entropy

The graph of the binary cross-entropy for output 1 is illustrated in Figure 2.6, and for negative output of 0 is depicted in Figure 2.7.

For positive target output, the loss is represented mathematically as

$$\text{Loss} = -\log(Y\_\text{Obtained}) \tag{2.3}$$

For negative target output, the loss is represented as

$$\text{Loss} = -\log(1-Y\_\text{Obtained}) \tag{2.4}$$

The binary cross-entropy loss function is presented as

$$\text{Loss} = (Y)(-\log(Y\_\text{Obtained})) + (1-Y)\,(-\log(1-Y\_\text{Obtained})) \tag{2.5}$$

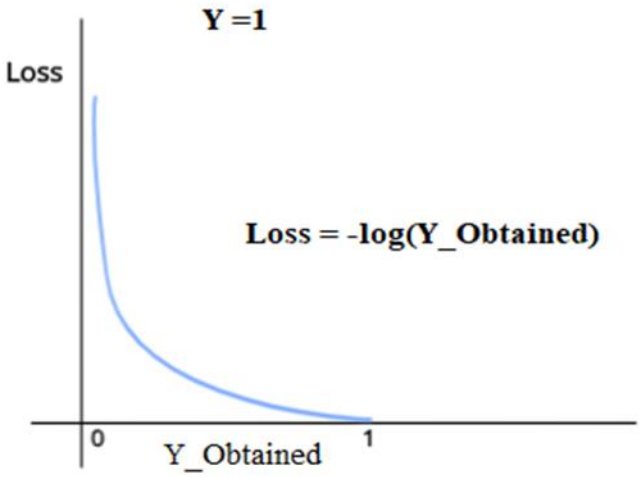

Figure 2.6 Binary cross-entropy loss for output 1

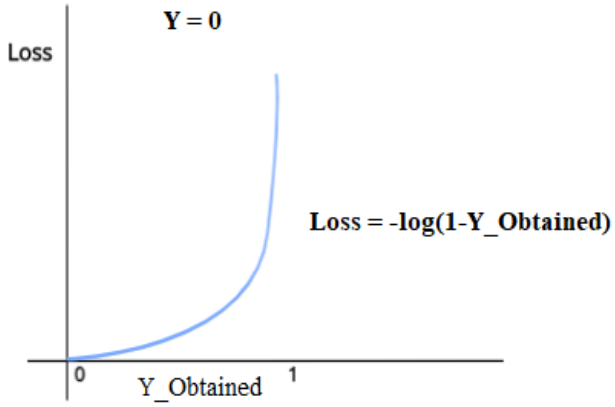

Figure 2.7 Binary cross-entropy loss for output 0

The decision boundary in case of classification problem is large, that make cross-entropy a suitable choice for binary classification. Whereas in case of Mean Square Error (MSE) are more suitable for regression problem as it does not punish misclassifications enough but is the right loss for regression as the gap between the two values which can be predicted is small. Also from probabilistic point of view, in classification problem where sigmoid and SoftMax nonlinearity exist in the output layer of the network the cross entropy becomes the choice for increasing classification accuracy. MSE suits if the target is a continues as in case of regression. The performance of the neural network depends on the value of the various parameter variables of the model being employed including minimizing the loss value. During the training, the parameters are updated depending upon the output of the model and its corresponding loss function; this process of improving the neural network model is accomplished by an optimizer employed, that follows an optimization algorithm. Optimization algorithm improves an

algorithm f(x), either by maximizing or minimizing the value of f(x). In a neural network, the optimization function help in improving the cost function of the model. The cost function in neural networks is defined as

$$C(W,b) = \frac{1}{m}\sum_{i=1}^{m} L((Y\_Obtained\ \ )^i, Y^i) \tag{2.6}$$

Where C() is the cost function, m is number of samples in the dataset, L is the loss function, Y is the true output, and Y_Obtained is the predicted output the model, W is the weights and b is the bias.

2.10.3 Optimization Function

The goal of employing the optimization function is to minimize other values of the cost function C by updating and modifying the values of W and b. The optimization algorithm can be classified into two categories, namely constant learning rate algorithms and adaptive learning algorithms.

1. Constant Learning Rate Algorithm

The Stochastic Gradient Descent (SGD) algorithm is the most widely exploited optimization algorithm with a constant learning rate. The training and learning process of the neural network is executed by the backpropagation technique. In the process, the dot product of input and corresponding weight is calculated, an activation function is then applied to sum up the products, to generate the output signal. Based on the output, the loss function is calculated which is then propagated backwards as an error in the network and helps in the updating of the weights using gradient descent. The gradient descent is the method of calculating the gradient of the error with respects to parameters of the neural network. This weight parameter is updated in the opposite direction of the loss function gradient. In the case of stochastic gradient descent, the weights are updated for each training example instead of whole data that makes it a much faster technique. The equation of SGD is presented in (2.7).

$$W^i = W^i - \eta * (\triangle C(W)) \tag{2.7}$$

Where, η is the learning rate, $\triangle$ is gradient

Although SGD is much faster in terms of computation in comparison with Gradient Decent algorithm, it has the following disadvantages**.**

- During the training process, the corrected weight value can become huge, which makes the activation function deviation to be too low. Because of this, the neural network will not train, so to solve this problem; the learning rate will be reduced which ultimately increases the training time of the neural network.
- Another major problem with SGD is getting stuck into local minima.
- SGD has another problem of convergence; the large learning rate will make the training faster but may lose predictions and can go int the wrong direction.
- SGD algorithm has shown to lead to the problem of overfitting the data

2. Adaptive Learning rate algorithms

The issue of constant learning rate algorithms is that all the parameters for the neural network are to be defined in advance. The finished learning rate is applied during the whole process of training based on which all the parameters are updated, which may reduce the performance of the model. There are many adaptive gradient descent algorithms such as Adagrad, Adam, Adadelta, RMSprop etc.in the context of work presented in this thesis here; the two optimization functions are discussed.

A. Adagrad

Adagrad applies a different learning rate at a time step for every parameter depending on the past gradients. So, there is no need to manually adjust the learning rate. Adagard makes small updates for frequent parameters and big updates for infrequent parameters. Adagard function suits best for sparse data.

$$\theta_{t+1,i} = \theta_t - \frac{\eta}{\sqrt{G_{t,i+\epsilon}}} . g_{t,i} \tag{2.8}$$

Where θ represents all the parameters, η is the learning rate, t is a timestamp $g_{t,i}$ is the gradient of the loss function at time step t for parameters θ.

- The advantage of Adagard is that the learning rate does not need to be updated manually. Generally, in most applications, a 0.01 default learning rate is implemented.
- The disadvantage of Adagard is that the learning rate keeps on decreasing during training.

B. Adam

Adaptive Moment Estimation (Adam) also apply adaptive learning rate during the training for parameters. Adam store the average of exponentially decaying past gradients and also keeps the average of decaying past squared gradients.

$$\widehat{g_{t,\iota}} = \frac{g_{t,i}}{1 - \beta 1^t} \tag{2.9}$$

$$\hat{G} = \frac{G_t}{1 - \beta 2^t} \tag{2.10}$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i+\epsilon}}} \cdot g_{t,i} \tag{2.11}$$

Where θ represents all the parameters**,** η is the learning rate, t is time stamp g(t, i) is the gradient of the loss function at time step t for parameters θ. In the proposed method, Adam is employed as an optimization function.

2.11 Distributed IDS model

Fog-to-node computing is ideal for the practical implementation and success of the IoT networks [81][110]. The fog computing is improved and extended form of cloud computing that is more suitable for distributed computing. The fog computing facilities the processing and communication near the source of the data which are IoT nodes or smart objects in case of IoT networks [111]. Fog-to-node architecture can be used for the distributed processing of proposed IDS and for detection of the cyber attack at fog node that is placed close the IoT networks.

2.11.1 Cloud Computing

The backbone of the IoT networks are cloud services, where all the data of the IoT devices are collected, processed, and analyzed. Cloud computing is a centralized computing model that facilitates the dynamic architecture for central data storage and computation resources like CPUs, Virtual Machines, software services, tools etc. Cloud provides services as three different models, 1) Software-as-a-Service, in this, the clients are offered with software via a web browser, and the hardware configuration cannot be decided by the client. 2) Platform-as-a-Service, in this, the clients are offered with a developing environment to enable developers to build, test and implement applications. And 3) Infrastructure-as-a-Service, int this the clients

are provided Virtual Machine resources with full control to configure the specifications according to the requirement of the customer.

2.11.2 Fog Computing in IoT networks

With the advancement in the use of cloud computing, new requirements also raised in terms of decentralized computing that can help with the limitations of the cloud such as high latency and no location awareness. The fog computing is the extended advanced paradigm of cloud computing that facilitates decentralized computations. In the context of IoT networks, the concept of fog computing was introduced in [112] in 2012. The idea behind the fog computation is to provide data storage and computation services close to the IoT devices, which is achieved by putting an additional layer between IoT devices and cloud named as fog node [113]. The fog nodes are capable of collecting and storing data, processing data and address the requests from the IoT networks and helps by reducing the latency, execution time and amount of data sent over the network. Another advantage of fog computing is high geographical distribution of IoT devices provided with seamless and reliable services for both stable and moving devices. Fog nodes receive requests from the IoT devices and decide whether to process request locally that depends on the capabilities of the fog nodes or propagated to the cloud. The fog computation support all types of communication networks like Wi-Fi, Bluetooth, wired and cellular networks. Following are the characteristics of fog computing model.

- **Location awareness and low latency.** The fog nodes are placed near the IoT devices the requests processing time and latency reduce significantly. The location aware services are provided by the fog nodes such as location-dependent content caching.
- **Large geographical distribution.** IoT devices are located on a large geographic area connected over the Internet, so the decentralized computation is the demand for IoT networks.
- **Mobility Support.** The fog computing model support dynamic restructures the network topology.
- **Device heterogeneity**. Fog computing provides standardized communication and virtualization of various resource required for the different types of IoT devices.

Figure 2.8 illustrates the architecture of fog-to-node model with distributed parallel computation providing intelligence to the distributed fogs by providing computation, control and storage of IDS closer to IoT network objects [15]. The global IDS is deployed on the cloud. The training of the IDS is performed locally on fog nodes and also the global model is trained

on the cloud using parameters acquired from fog nodes. The IoT devices in each network are connected to the fog node; the detection of the cyber-attack is done on the fog node where the local IDS is deployed.

2.12 Open research issues and challenges in the context of DDoS attacks on IoT networks

DDoS attacks detection and mitigation has now become one of the top priority cybersecurity issues in the IoT networks. Initially, the security issues in the IoT were ignored that later became the biggest hurdle for trusting the IoT networks. Now researchers and security experts are trying to supply the security gaps in IoT networks by addressing the challenges [16], [37], [114], [115].

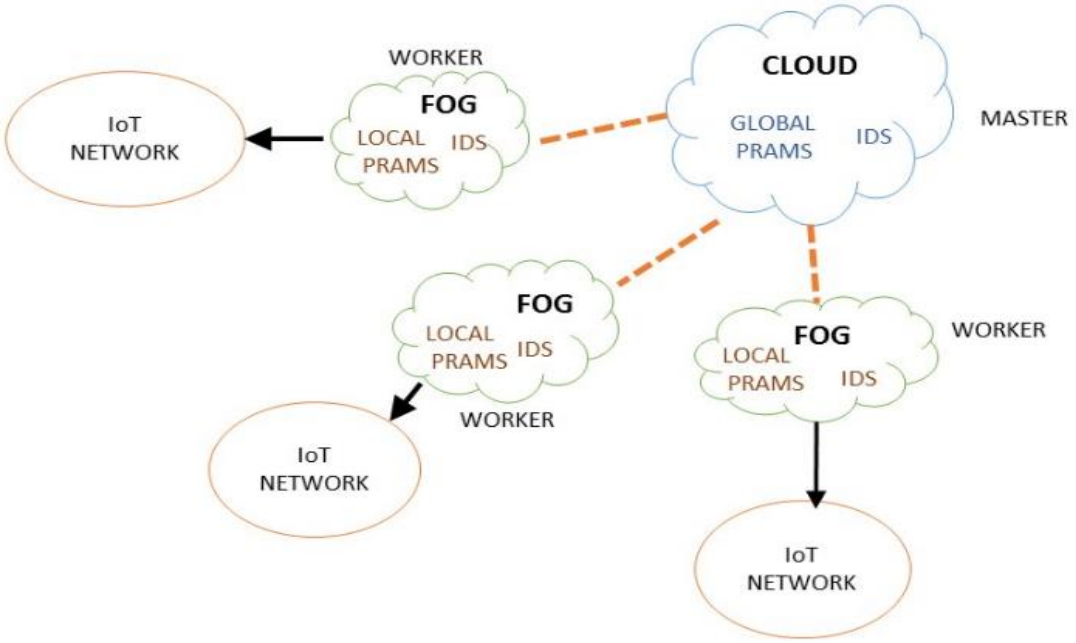

Figure 2.8 Fog-to-node architecture for IoT networks for the implementation of IDS

Some of the important issues and challenges are listed below, which should be taken into consideration for DDoS attacks IDS in IoT networks identified from the literature review.

- **IoT environment should be considered in real-time scenarios.** Most of the proposed work for the DDoS attack detection in IoT networks have overlooked the real-time environment. The IDSs should be designed based on the real-world vulnerabilities and constraints in the IoT network.
- **Quality of the datasets employed.** The datasets exploited for the training and testing of the IDSs directly influence its performance because the IDS learn from the data itself. Therefore, the training should be provided on the data as close as real-world IoT data.

Also, in the case of machine learning-based IDS, the quantity of the data being used for training effects the performance of IDSs in the real world.

- **Specific Protocol based methods.** Most of the proposed works are based on some specific protocol attacks that are not capable of detection of attacks on other protocols.
- **Learn to detect unknown attacks.** With the advancement in newer sensitive DDoS attacks, IDSs should be able to detect and mitigate those unknown attacks. So the IDSs should be able to adapt to learn new attacks on its own.
- **Quality of Services and cost.** The IDSs should not affect the quality of services in the IoT networks. Also, for the development of an IDS, the cost factor should be taken into consideration, as the goal of the IoT is to reach common people which they can afford [116].
- **No standard IoT framework.** Till now, there exists no standard for the IoT framework that is crucial for knowing the organization and involved operations. There are many proposed versions of IoT framework that exists, but they all do not follow any set protocol.
- **Complete Security.** The IoT networks should include prevention methods against any cyber threat at each layer, along with the detection methods. Most of the proposed research on the network after it has been attacked. However, there should be some mechanism to prevent any security threat as well in the system [117].

2.13 Chapter Summery

This chapter comprises the literature review conducted for doing the proposed work in the thesis. The study included the various cyber-attacks that have affected the IoT networks in the past. The introduction to DDoS and types of DDoS attack in the context of IoT networks are included. After the introduction to Intrusion Detection Systems, the review covered latest DDoS attacks on IoT networks, the survey of IDSs for IoT networks, review of feature selection methods for IoT networks and in the end section, the review of deep learning techniques for the detection of cyber-attacks is elaborated. The research challenges and problems are identified in this chapter, in Chapters 4, 5 and 6, the proposed methods are founded on these identified challenges.

Chapter 3. Research Methodology and Datasets

Research methodology and datasets employed in the proposed method is presented in this chapter. Datasets have critical importance for training and testing of an intrusion detection system. In this chapter, the datasets employed for evaluating the proposed methods and the performance measures used are defined. Gathering datasets and evaluating actions should improve realization and the ability to detect attacks within the future. However, one of the primary challenges of today's investigation studies is acquiring consultant data to obtain significant and thorough effects and suggest upon them. Awkwardly, creating a dependable standard dataset is not a clean task. Performance metrics employed for testing and validating an IDS is also critical as they reflect the performance of the proposed system. The employed metrics are also described in this chapter.

3.1 Research methodology

This thesis has three major contributions, the general methodology employed for the evaluation of the execution and evaluation is presented in Figure 3.1. The data contained in the dataset, which is discussed in Section 3.2, is preprocessed by data filtration and normalization which is discussed in detail in Sections 3.4, 3.5 and 3.6. The data is fed to the proposed method for evaluation which is detailed in Chapters 4, 5 and 6. The performance is evaluated using the standard performance metrics as discussed below.

3.1.1 Performance Measurement Metrics

The following measurement metrics are used for the performance evaluation in all the three contribution chapters. The metrics are chosen based on the references found in other published work [118], [119], and all these metrics are standard measures for the performance evaluation for classification in the field of cybersecurity.

The basic terms used for defining the performance metrics are :

False Positive (FP). The number of elements wrongly categorized as positive but is negative in actual i.e. the element's label is 'benign' but classified as 'malicious.'

True Positive(TP). The number of elements correctly categorized as positive i.e. the element label is 'malicious' and classified as 'malicious ' only.

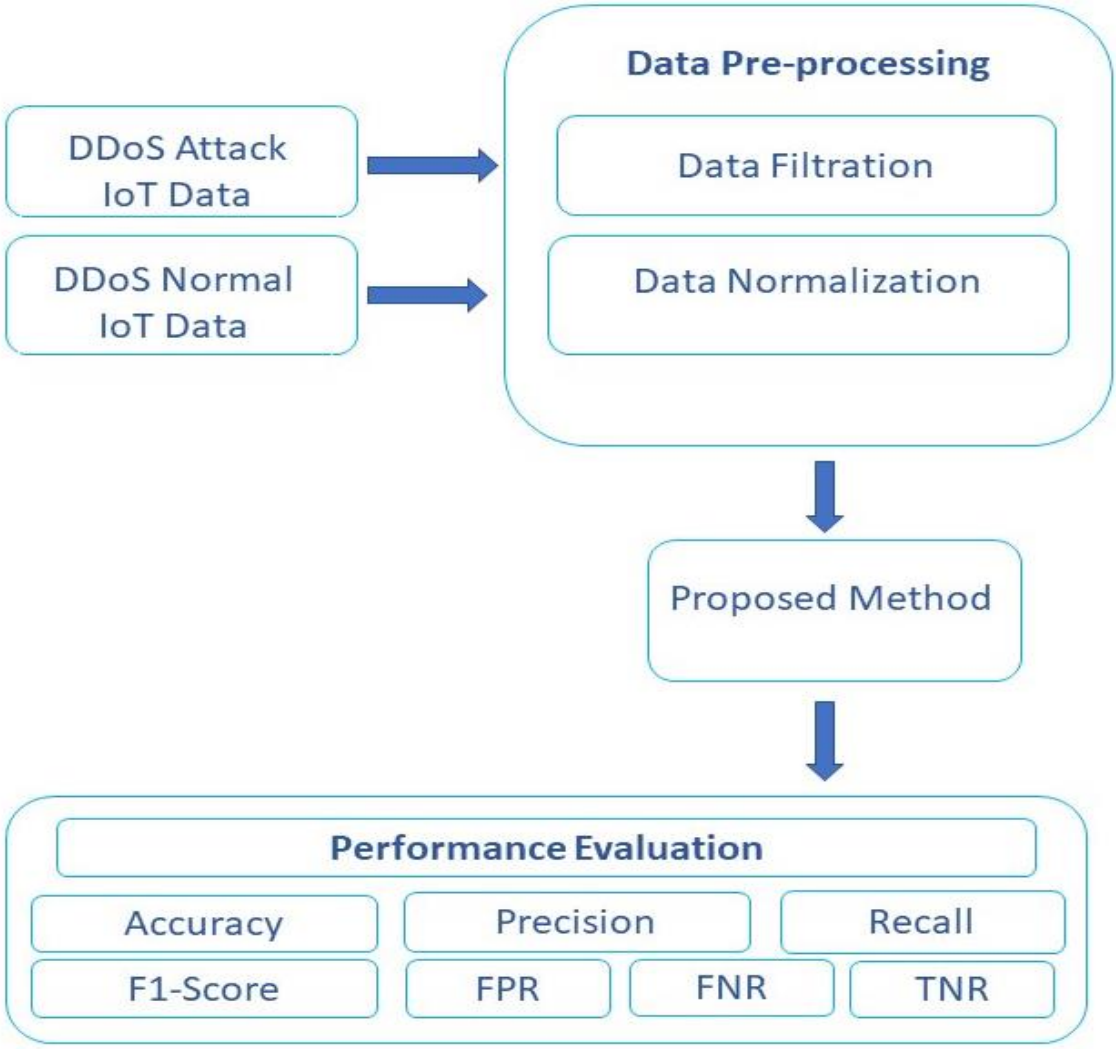

Figure 3.1 The employed research methodology

False Negative (FN). The elements that are wrongly classified as negative but are positive in actual i.e. the label of the element is 'malicious' but classified as 'benign.'

True Negative (TN). The number of elements correctly categorized as negative, i.e. the element label is 'benign' and is classified as 'benign' only.

The performance metrics exploited in this thesis are defined as:

(1). Accuracy is the ratio of true classifications done to the total number of predictions done.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

(2). Recall, it has another name as well as true positive rate, is defined as the ratio of correct classifications done to the sum of correct predictions and wrongly classified negatives.

$$Recall = \frac{TP}{TP\ +\ FN} \tag{3.2}$$

(3). Precision is the ratio of the aggregate number of accurate classifications to the total number of predictions classified as positive.

$$Precision = \frac{TP}{TP\ +\ FP} \tag{3.3}$$

(4). F1-Score is defined as a harmonic mean of Recall and Precision

$$F1 - Score = 2 * \frac{precision\ X\ recall}{precision\ +\ recall} \tag{3.4}$$

(5). False Positive Rate (FPR) is the ratio of false-positive to the sum of false positive and false negative classifications

$$FPR = \frac{FP}{FP\ +\ FN} \tag{3.5}$$

(6). False Negative Rate (FNR) is the ratio of false-negative to the sum of true positive and false negative classifications

$$FNR = \frac{FN}{TP\ +\ FN} \tag{3.6}$$

(7). True Negative Rate (TNR) is defined as the ratio of true negative to the sum of true negatives and false positive classifications

$$TNR = \frac{TN}{TN\ +\ FP} \tag{3.7}$$

(8). True Positive Rate (TPR) is defined as ratio of true positive to the sum of true positive and false negative classifications.

$$TPR = \frac{TP}{TP\ +\ FN} \tag{3.8}$$

3.2 Datasets

The CICIDS2017 dataset has been used to conduct this work. The IDS and IP (Internet Protocol) are considered the most important tools against the ever-growing network attacks, but mostly they lack in providing consistent and accurate performance; this is due to the lack of reliable test and validation datasets[120]. Most of the datasets, which include DDoS attacks, are out of relevant data that are unreliable. Dataset design suffers because of many reasons such as lack of traffic diversity, they do not contain all known attacks, and they include anonymized packet payload data which does not provide current trends[121]. The most common datasets used in other proposed work such as NSL-KDD and KDD-99 have shown limitations such as low detection rate, low true alarm, and high false positives. The CICIDS2017 labelled dataset available, which contains most up to date data network attacks resembling real-world network data[119]. The dataset has been designed by the University of New Brunswick's and Canadian Institute for cybersecurity across five days. For the generation of the datasets, the research team used a complete network infrastructure to maximize till date general attack actions. The researcher encompassed essential systems such as a firewall, switches, router, various varieties of servers, and diverse varieties of the three working systems, namely Macintosh, Windows, and Linux.

In this dataset, the inventors generated six cyberattacks profiles founded on the latest up to date listing of commonplace attack households and performed them by using the usage of related tools and codes.[122].This dataset is generated by keeping realistic background traffic as a top priority; the developer of this dataset has used a B-Profile system to profile the abstract behavior of human interactions and generated naturalistic background traffic. Intellectual actions of 25 users based on the email protocols, FPS, HTTPS, SSH, and HTTP was built. This dataset was collected for five days in 2017 on different cyber-attacks along with no attacks.

The CICIDS2017 dataset assault situations are:

1. Brute Force Attack: This sort of assault is primarily founded on a trial and error methodology concentrated on the victim system til it achieves success. The essential utilization of this cyberattack is password breaking; though, it can be utilized for finding concealed pages and content in an internet application.

2. Heartbleed Attack: It is a virus in the famous OpenSSL cryptography library, that is extensively utilized in the operation of safeguarding transmissions over webs. This assault is accomplished by delivering an abnormal heartbeat call including a tiny payload and massive size area to the weak machine to release memory contents.

3. Botnet: A variety of hacked devices linked to the internet and utilized through an outsider to implement extraordinary tasks. For example, robbing information, dispatching spam, moreover having gain access to to the device and its links.

4. DoS Attack: Besides conventional DoS assaults, the authors carried out minimal rate DoS assaults wherein a solo gadget maintains networks open with marginal bandwidth that utilizes the server resources and takes it down.

5. DDoS Attack and PortScan: Authors carried out the latest renewed listing of valuable DDoS assault software. PortScan is an assault that is performed to examine the port's status so that it will identify to be had facilities that are at present operating on a server.

6. Web Attack: This kind of assault is coming out every day focused on numerous web applications and may disclose an organization's valuable resources to the outside world. They carried out numerous web assaults, including SQL Injection, in which an attacker generates a string of SQL commands and then utilizes it to pressure the database to answer with sensitive information. Cross-Site Scripting (XSS) occurs 82 when builders do not observe their code properly to find out the possibility of script injection. In addition, Brute Force over HTTP, which involves trying a list of passwords to discover the administrator's password.

7. Infiltration Attack: It is an try to compromise the community from inside via the utilization of an inclined software. In the case of success, a backdoor will be set up on the victim's machine which leads to the overall performance of various assaults at the victim's community, for instance, IP sweep, complete port experiment and service enumerations.

This dataset contains 85 network flow features along with label attribute and a total of 225,745 instances with both attack and normal data. The feature description of the dataset is given in Appendix A given at the end of this thesis. Two versions of this dataset are available CSV file which is converted to flow recording using CICFlowMeter [123] tool and raw PCAP file. The description of the attack type in this dataset is given in Table 3.1. Figure 3.2 and Figure 3.3 illustrate the number of flow in the dataset and the number of attacks in the dataset respectively, according to the days. This dataset is highly unbalanced, so for this work, we have

modified the training dataset to balance in terms of both attack and normal data and reduced the number of instances to 83 features and divided data into training and test data. To evaluate our work, we have used data captured on July 7, 2017, which contains both normal and DDoS attack data.

Table 3.1 CICIDS2017 Dataset Overview

Date	Number of Flows	Number of Attacks	Description
Monday	529,918	0	Normal network activities
Tuesday	445,909	7938	FTP-Patator
		5897	SSH-Patator
Wednesday	692,703	5,796	DoS slowloris
		5,499	DoS Slowhttptest
		231,073	DoS Hulk
		10,293	Dos GoldenEye
		11	Heartbleed
Thursday Morning	170366	1507	Web Attack - Brute Force
		652	Web Attack - XSS
		21	Web Attack - SQL Injection
Thursday Afternoon	288602	36	Infiltration
Friday Morning	191033	1966	Bot
Friday Afternoon 1	286467	158930	Portscan
Friday Afternoon 2	225745	128027	DDoS
Total	2830743	557646	19.70 % attack data

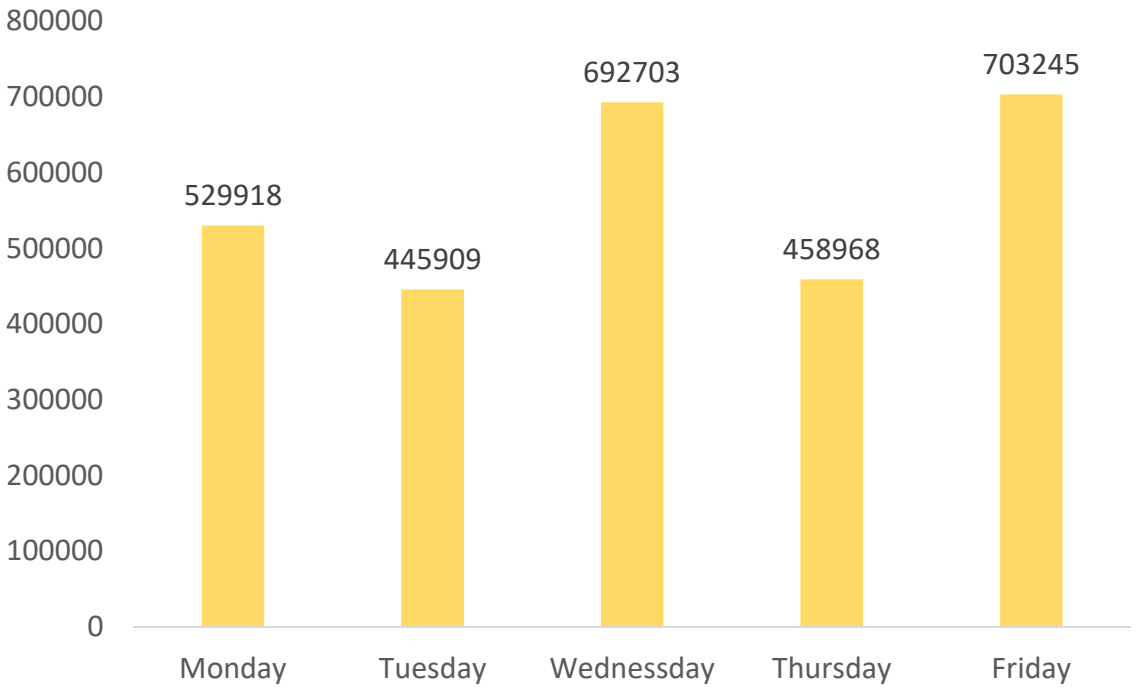

Figure 3.2 Number of flows per day in CICIDS2017 dataset

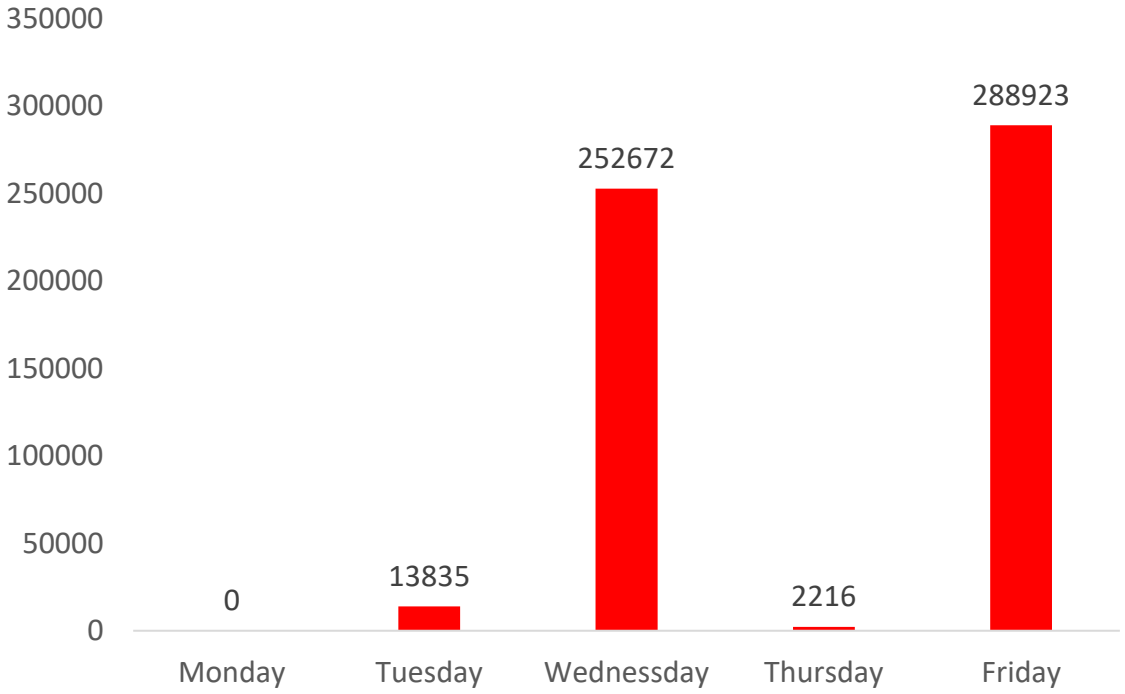

Figure 3.3 Number of attacks per day in CICIDS2017 dataset

3.3 Dataset Pre-processing

The CICIDS2017 dataset is first preprocessed by converting some of the features to numeric values. The Source, Destination and flow ID IP(Internet Protocol) [124] address are converted

to using the following equation (3.9) for an IP address. The IP address is divided into four sub blocks as w.x.y.z, where, w,x,y,z are called octet and each have a decimal value between 0 and 255. Octets are separated by periods and each sub block has different weight number each powered by 256. The numeric value is calculated as:

$$\text{IP Number} = 256^3 * \text{w} + 256^2 * \text{x} + 256^1 * \text{y} + \text{z} \quad [125] \quad (3.9)$$

The data label feature in the dataset contained DDoS and Benign classes, which are replaced by 1 and 0, respectively. The Timestamp feature is removed as it contained garbage value which cannot be processed by any detection algorithm.

3.4 Data normalization

The transformed data is then normalized to fall between [0,1]. Equation (3.10) [126] is employed to normalize the data.

$$X_i = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (3.10)$$

Where X_i is the value of a particular attribute, X_{min} is the minimum value in the column of the attribute, and X_{max} is the maximum value of the feature in that particular column.

3.5 Dataset Modification

The CICIDS2017 dataset is highly unbalanced that becomes bottleneck for the deep learning method to work efficiently. Sampling is an efficient method for tackling the unbalanced dataset [127]. There exist broadly two sampling techniques to balance an unbalanced dataset named as under sampling and over sampling. In the first technique the size of the larger class of the data is reduced to bring the dataset into equilibrium. This technique applied when the amount of the data is adequate. In this method the data of the minor class is preserved as it is and data from the major class is extracted randomly equal to the number of the element of minor class data. The second over sampling method is employed when the amount of the data in the dataset is not enough [128]. In this method the data of the minor class is increased to the amount of data of the major class. There are many methods for performing over sampling such as repetition or duplication, bootstrapping or Synthetic Minority Over Sampling (SMOTE) Technique [129]. To tackle this issue, this dataset is preprocessed and 20 % of the data is extracted from the

dataset in [130]. In [131] the to balance the data 30,000 data with attack and 30,000 data with normal data is extracted from the dataset to evaluate their proposed method. This dataset is highly unbalanced, so for this work, we have modified the training dataset employing the over sampling balancing method [132]. Duplicating technique is employed to balance in terms of both attack and normal data. The data duplication is performed manually in Microsoft Excel Software. The number of instances are reduced to 81 features and divided data into training and test data[115]. Figure 3.4 presents the pie chart of the original dataset and modified dataset. The data is initially transformed into numeric values, and then, this transformed data are then normalized before it is fed to the Deep Learning algorithm for binary classification.

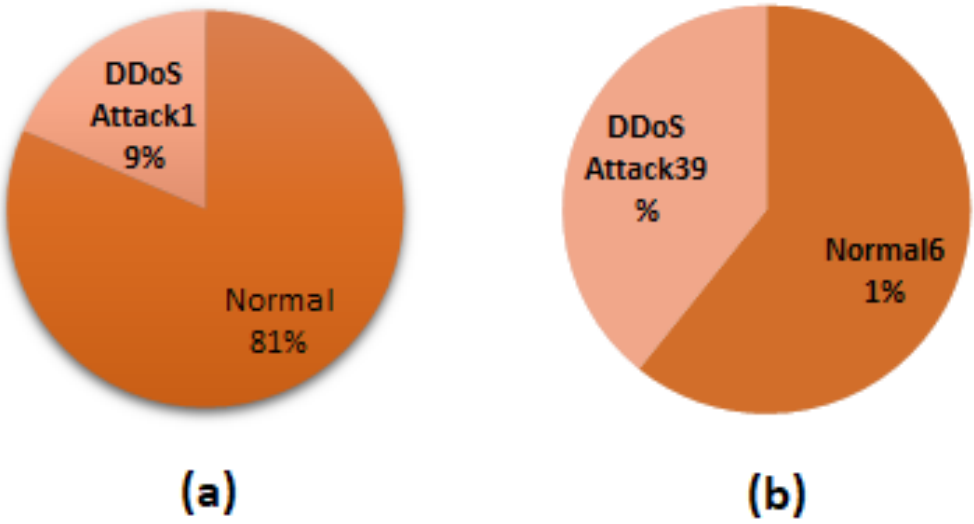

Figure 3.4 Data distribution of (a) original CICIDS2017 dataset and (b) modified dataset

3.6 Chapter Summery

In this chapter, after highlighting the research methodology, the datasets and performance evaluation metrics employed for conducting the experimentation of the three major contributions of this thesis for the evaluation of the proposed work are detailed. The latest CICIDS2017 datasets are elaborated, including the data collection environment employed for building this dataset. The dataset was preprocessing, and garbage attributes were removed from 85 to 83 total features and then normalized within the range of [0, 1]. All the standard performance metrics exploited are also discussed. The methodology including the metrics and datasets discussed in this chapter will serve for the evaluation of methods proposed in Chapters 4, 5 and 6.

Chapter 4. Multi-Objective Optimization based Feature Selection for DDoS Attack Detection in an IoT Networks

In this chapter, a multi-objective optimization-based feature selection method for the detection of DDoS attacks in IoT networks is proposed. An IDS is a vital tool for the detection of such cyber-attacks. Real-world measurements that form the input to an IDS are generally huge. FS is therefore required to reduce the dimensionality of data and improve the performance of an IDS. The critical reason for the failure of IDSs is incorrect selection of features because most of the feature selection methods are based on a limited number of objectives such as accuracy or relevance of data, but these are not enough as they can be misleading for extracting features for the detection of an attack, the contribution of this work is therefore to develop a multi-objective based approach for feature selection.

4.1 Introduction

Feature Selection is the method of reducing the dimensionality of datasets by applying some appropriate algorithm. The feature selection is a crucial step as it directly affects the performance of the machine learning classifier[133]. There are two advantages of doing FS; the first FS facilitates improving the performance of the classifying algorithm by reducing the computation cost and second, it helps with the problem of overfitting, which is caused by irrelevant features in the datasets. FS is grouped into the following three types [134]:

Filtering Methods dependent on the statistical properties of the features. Features are selected based on their relevance to provide information about different classes, as shown in

Figure 4.1. The advantage of filtering methods is that they do not demand much computation, so they are less expensive. The drawback of filtering algorithms is that they are suitable only for independent features, but for the rest, they may result in redundant features [135].

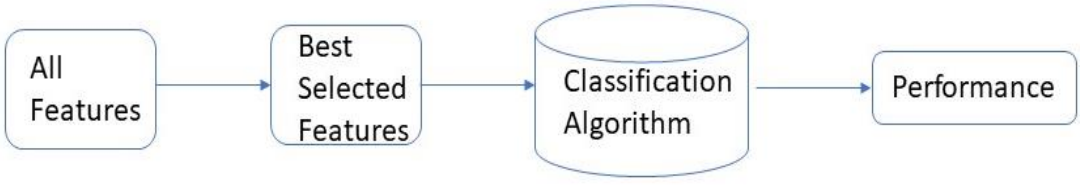

Figure 4.1 Filter FS Method

Wrapper Methods select features using the outcome of a learning algorithm. Comparatively, wrapper methods are more complex and demand more computational resources, but their performance is better than filtering methods because of more accurate results [134].

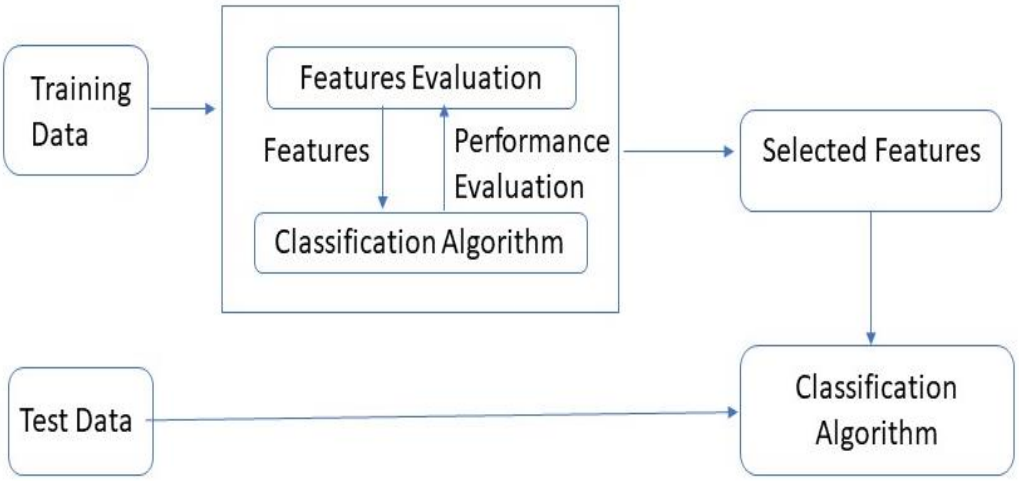

Figure 4.2 Wrapper FS Method

Hybrid Methods combine the advantages of filtering methods and Wrapper methods [135].

Various algorithms have been proposed for feature selection for various wireless networks, most of which are based on performance metrics such as accuracy, relevance, and redundancy. To solve real-time problems considering two or three objectives is not enough. Accuracy is one of the most common objectives for feature selection and attack detection, but it is questionable to rely on considering accuracy as the best model. It may be the case that accuracy is as high as 99.9%, but it is possible that the precision and recall values are low, which is because the value of false positives and false negatives is high. So, concluding performance based on one to three objectives could be misleading. There is, therefore, a demand to use the multi-objective optimization-based method for feature selection for IoT networks.

4.2 Multi-objective Optimization Problem

In the proposed method, the feature selection problem is handled as a multi-objective optimization problem. In real-world applications, the solutions to a problem are usually dependent on many conflicting objectives. That means optimizing one objective may result in the depletion of another objective; therefore, a single solution cannot be a solution for a multi-

objective problem. The multi-objective problem provides a set of solutions that satisfy all the objectives. The multi-objective optimization technique is based on the following theorems:

Theorem 1. A feasible solution 'x' is one which satisfies all the constraints and $x \in X$

Theorem 2. 'x_1' one of the feasible solutions dominate another feasible solution 'x_2' if it satisfies the following conditions

1. 'x_1' is no worse than 'x_2' for all objectives
2. 'x_1'is better than 'x_2' in at least one of the objectives

In other words, we can say that x_1 is non-dominated by x_2

Theorem 3. x_1 and x_2 two feasible solutions are incomparable if neither x_1 dominates x_2 nor, x_2 dominates x_1

Theorem 4. An x' is Pareto optimal feasible solution $x' \in X$ if there is no other solution $x \in X$ such that f(x) dominates f(x'), where f() represent objective function.

Theorem 5. The set of Pareto optimal solutions is known as Pareto set P_{True} :

$$P_{True} = \{x' \in X \qquad (4.1)$$

The Pareto front is an image of the P_{True} plotted in the objective space

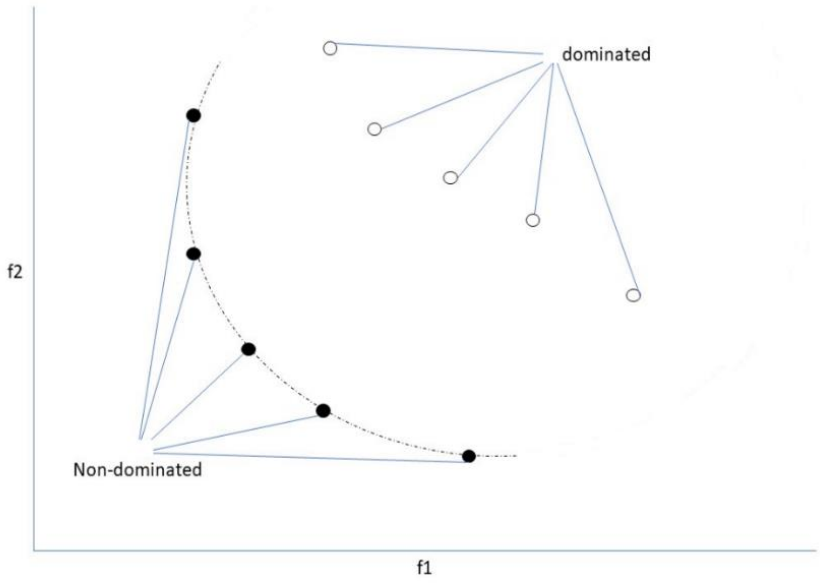

Figure 4.3: Pareto front of two objectives

The Pareto optimal front for a two objective problem is presented in Figure 4.3. The dominated solutions are represented by non-filled circles and Pareto optimal solutions are represented by black filled circles.

The multi-objective optimization problem is defined as a method to find solutions for two or more conflicting objectives with some constraints. Optimization with M number of objectives can be formulated as:

$$Minimize/Maximize\{f1(x), f2(x) \dots f_M(x)\} \quad (4.2)$$

subject to $x \in X$

where X is a set of solutions, and x is a non-dominated solution. In other words, x_1 is said to be Pareto-efficient if there exits x_2 which is dominated by x_1 if

- x_1 is no worse than x_2 for all M
- x_1 is better than x_2 in at least m_i

where M is number of objective functions and $m_i \in M$ is subset of M for i=1, 2…M)

4.3 Genetic Algorithm

The GA (Genetic Algorithms) are a heuristic search algorithm that is inspired by biological evolution and mimics the process of natural selection [136]. GA work on a set of the population through the process of selection, recombination, and mutation to produce a better generation of solutions. GA finds the optimal solution based on the survival of the fittest among all the strings. In every generation, the new generation of the string is obtained which contains the bits and pieces of the old generation strings. It makes use of a defined fitness function to evaluate the population that measures the potential of the solution to solve the problem. A pseudocode of the algorithmic concept of genetic algorithm is described below:

Algorithm 4.1: Genetic Evolutionary Algorithm

```
1   g ← 0;
2   Initialize populations p
3   Evaluate each individual in p
4   While the termination condition is not met
5   g ← g+1;
6   p' ← crossover(p)
7   P ← mutate(P')
8   Evaluate (P)
9   P_next ← select (p'U P)
```

The GA find fixed-length binary strings as optimized solutions to a problem. In every generation, a population of the chromosome is built by altering the chromosomes of the current generation. The next generation of chromosomes is randomly selected from the current generation depending on the probability of the selection of each chromosome.

In steps 1 and 2, the initialization parameters of the algorithms are set according to the problem to be solved. That creases the first generation of the chromosomes. The initialized parameters are the population size that is the number of chromosomes in each generation, the crossover probability that is the probability value of the pair of chromosomes to be crossed; mutation probability is the probability of the random mutation of a gene on a chromosome. The maximum number of generations is defined as the termination criterion. In step 3, the evaluation of chromosomes in every generation for the selection process is performed. The score of each gene in the chromosomes is added up; an average score of chromosomes is calculated to determine the elite chromosome of the generation. Step 4 defines the termination criteria of the loop. The next generation of chromosomes is selected in step 5. Step 6 performs the crossover operation on the chromosomes, which are accomplished by selecting a random site along the chromosome's length. The crossover is performed by exchanging the genes of the two chromosomes. The next mutation for developing new offspring chromosomes that are different from their parent solutions is performed in step 7. The new generation is added up to the next generation in steps 8 and 9.

4.3.1 Non-Dominated Sorting Genetic Algorithm

NSGA-II is non dominated sorting generating algorithm, which was proposed in [137] by Deb. et al. NSGA-II has the following characteristic features:

- It is a fast-non-dominated sorting method with M number of objectives, and N population size, the computational time complexity of this algorithm (MN^2).
- It is an elitism method as it preserves the promising solutions
- It eradicates the difficulty of assigning appropriate parameter values for fitness sharing function by introducing the concept of crowded distance to ensure the diversity in the populations.

The solutional to multi-objective optimization problem are mathematically expressed in terms of nondominated solutions. As explained in section 4.2 if a solution x_1 is no worse than another solution x_2 and x_1 is better than x_2 in at least one of the objective function which proves that x_1 is strictly better than x_2 or in other words we say that x_2 is dominated by solution x_1. Set

of all the solutions similar x_1 which are no dominated by any other solutions form the non-dominated pareto front. The goal of this algorithm is to find all the non-dominated solutions and reject all dominated solutions and hence this algorithm is called non dominated sorting algorithm.

The NSGA-II algorithm is represented in Figure 4.4 and has the following procedure steps:

1) The population is initialized; crossover and mutation are performed on the population to produce offspring. Parents and offspring are combined after this non-dominated sorting is applied and classified by fronts.

2) The new population is created according to fronts ranking.

3) Crowding distance, which is based on the density of solutions around each solution, is calculated and assigned to each front.

4) Tournament selection is performed to select next-generation offspring. Finally, a new generation is created by crossover and mutation operations.

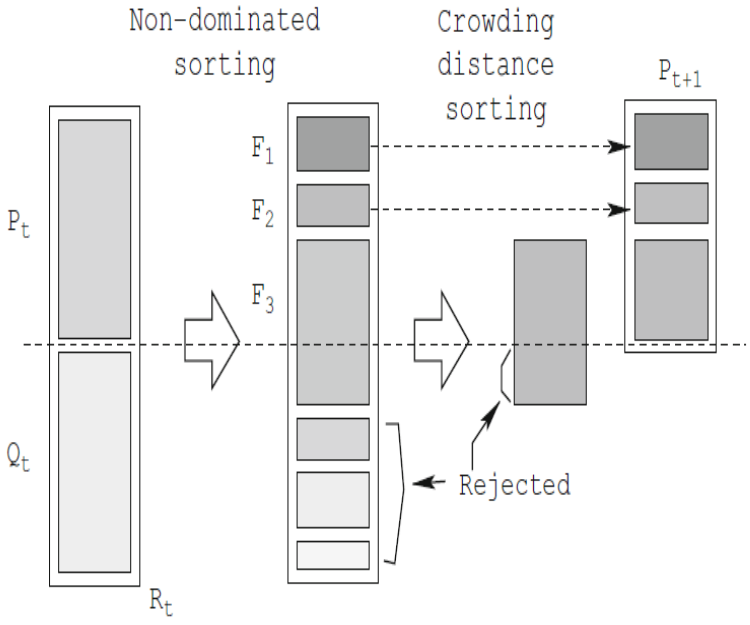

Figure 4.4: NSGA-II algorithm procedure [137].

4.3.2 Jumping Gene Adapted NSGA-II (NSGA-II-aJG)

In this work, the employed jumping gene adapted NSGA-II named as NSGA-II-aJG algorithm illustrated in Figure 4.5 for executing feature selection based on six different

objectives [26]. The Jumping gene is a concept in which a randomly generated binary string equal to the size of decision variables of the problem to be solved is used to replace a few chromosomes [79], [138]. The location to start the jumping gene replacement is chosen randomly with the condition that the chosen location is lower than the different of the total number of variables and chromosomes. The jumping gene in the NSGA-II algorithm has been employed to perform feature selection based on the concept of elitism, Pareto dominance, and crowding distance.

The main loop In the initialization step, the random population of size N is generated and put into parent P_0. The values of objective functions which are defined in section 4.4.1 are calculated on parent $P_{0,}$ and all the solutions in P_0 are sorted according to the non-dominated sorting algorithm and crowding distance evaluation which is explained in detail in the steps given below. The genetic algorithm operations as selection, crossover, and mutation are performed to obtain sorted solutions that become the offspring set Q_1 with size N.

One of the important characteristics of the NSGA-II algorithm is an elitism that preserves the best solutions of the parent set in the previous generation and those solutions remain unchanged in the current iteration. In the NSGA-II algorithm, the parent set $p_{i-1,}$ and their offspring Q_i of the i^{th} iteration is merged to form combined population R_t with size 2N. The sorting is performed on R_t based on ran of non-dominated front and crowding distance to achieve the N best solutions. The obtained solution becomes the parent as p_{i+1} for the next $i+1^{th}$ iteration [139].

Crossover. The crossover operation is executed for generating new i+1th generation from Pi parent generation. The two-parent solutions are picked randomly and crossed as in biological meiosis to form two offspring solutions. The crossover operation is provided a user predefined probability Pc , that depends on the problem to be solved. I have employed a Pc of 90.0% probability as referred in [137], [140]. The pseudocode of crossover operation is given in the algorithm. P and Q are the two-parent solutions with N= |P|, a is randomly generated 1 × N matrix containing values between 0 and 1 with |A| = |P| = |Q|, P’ and Q’ are the generated offspring of P and Q.

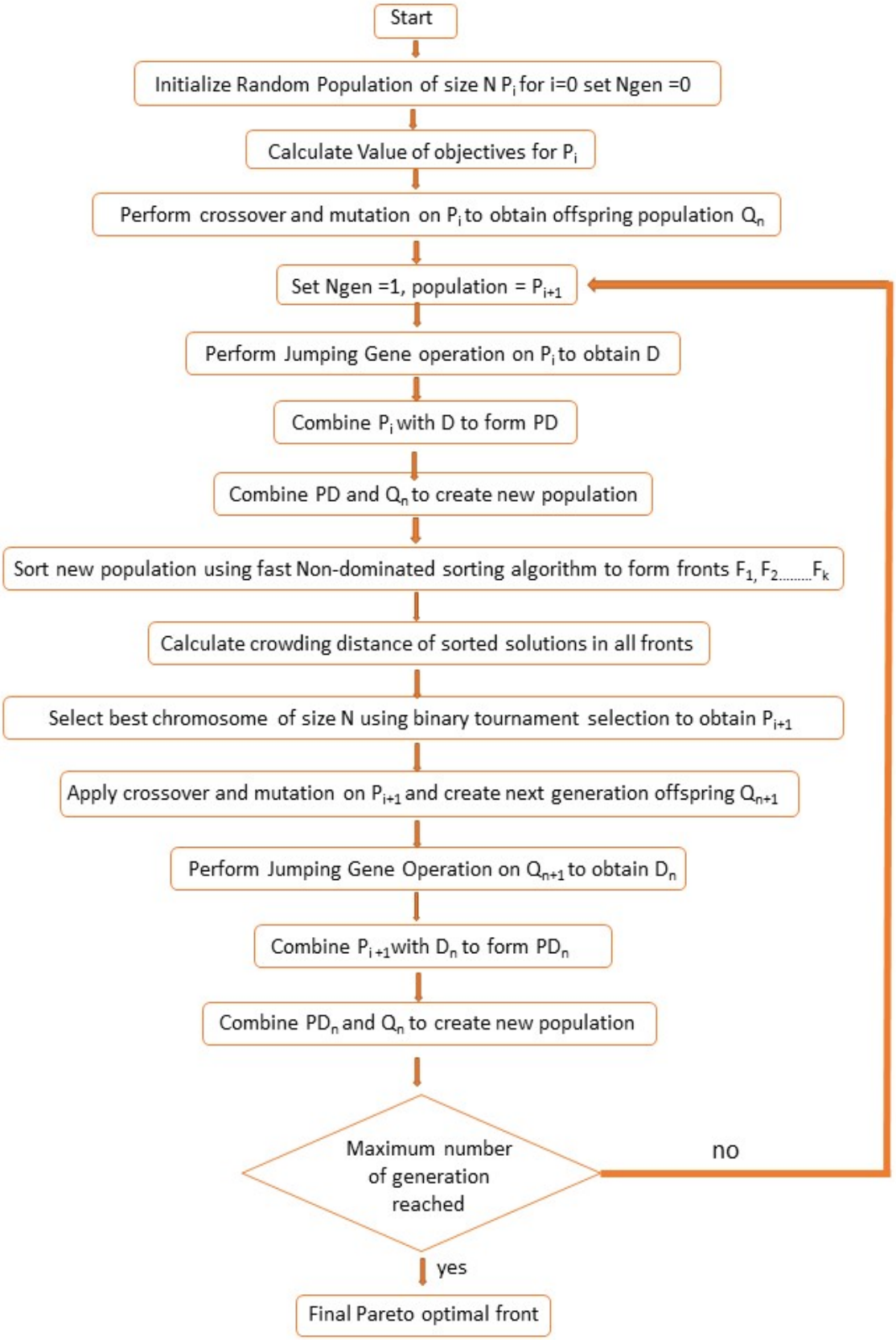

Figure 4.5 : Flow chart of the Jumping Gene Adapted NSGA-II-aJG algorithm

Algorithm 4.2. Crossover

Input P,Q
N = size(P)
A = rand(N,0,1)
P’ = A × P + (1-A) × Q
Q’ = A × Q + (1 - A) × P
Return P’, Q’

Mutation The mutation is an important process during mitosis or meiosis in genetics for producing diverse offsprings. The mutation operation in the genetic algorithm serves the same purpose. Mutation produces offspring solutions from the parent solution with different characters and properties. Similar to crossover operation mutation operation also requires a user-defined mutation probability. In[137] and [141], a mutation probability of 1/n with n number of process variables is defined and the same mutation value is used in this work. The pseudocode of the mutation process is given in the algorithm below. P is the selected parent from the set of all parent solutions, and mutation rate μ defines the number of variables to mutated for a parent solution P, mutation step size α specify the magnitude of the mutations.

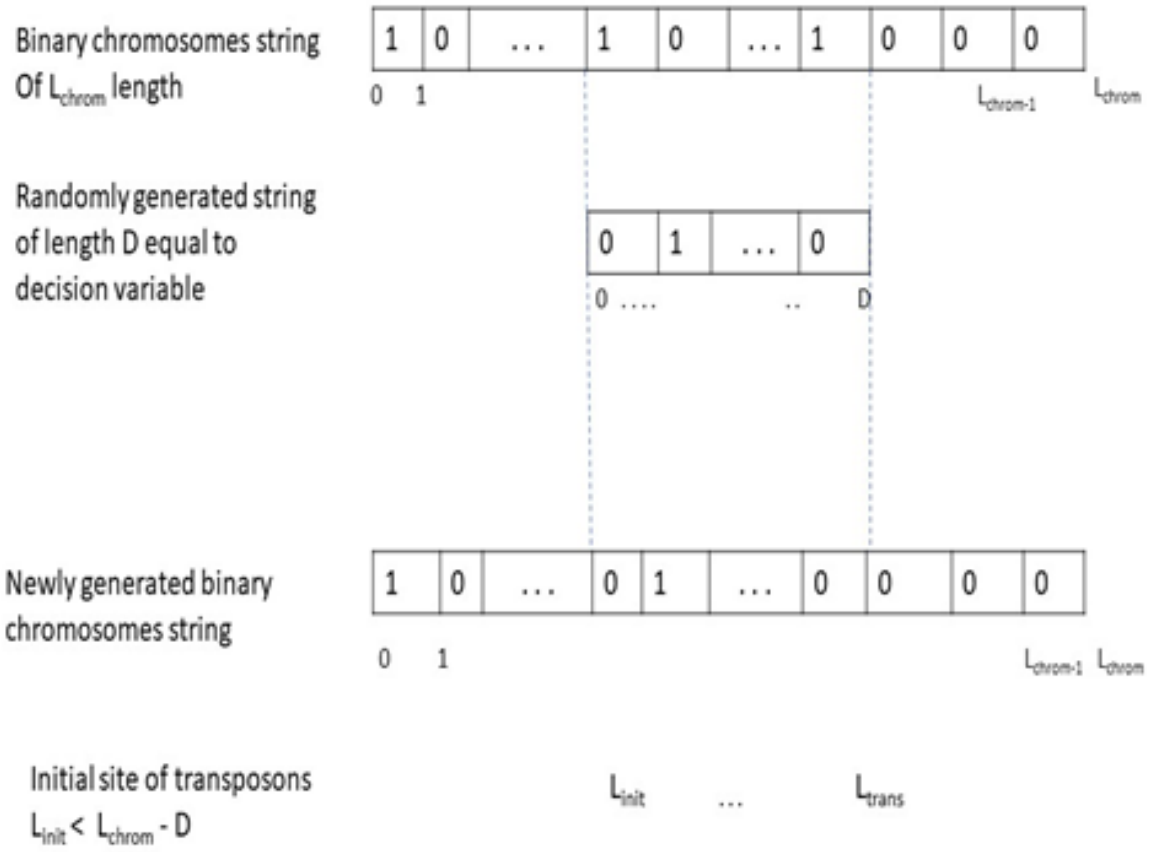

Figure 4.6: Jumping gene adaptation chromosome

Algorithm 4.3. Mutation

Input: P, μ, α
N = |P|
nVar = [μ X N]
ΔVar = randsample(N, nVar)
P' = P
For each member of ΔVar do
$P_i' = P_i' - \alpha \times rand(0,1)$
End for
Return P'

Jumping Gene. In genetic engineering, the concept of the jumping gene [142] also known as transposons [143] is quite popular. The transposons can frequently move in or out of the chromosomes and that is how it creates resistance to antibiotic drugs. An improved variant of the NSGA-II algorithm combined with the jumping gene named NSGA-II-JG along with its different variants has been used to solve a different multi-objective problem which resulted in better convergence and the reduction of CPU time. In[26], [79] it was found that the NSGA-II-aJG [138] outperforms other variants of NSGA-II on different evaluating metrics.

In this work, I have employed the NSGA-II-aJG algorithm illustrated in Figure 4.6 for executing feature selection based on six different objectives. The Jumping gene is a concept in which a randomly generates binary string equal to the size of decision variables D of the problem to be solved is used to replace a few chromosomes. The location L_{init} to start the jumping gene replacement is chosen randomly with the condition that the chosen location is lower than the different of the total number of variables and chromosomes i.e $L_{init} < L_{chrom} - D$

. **Fast non-dominated Sorting**. The fast non-dominates sorting algorithm requires $O(MN^2)$ comparisons to evaluate the non-dominated front. For every solution P in the solution set, domination count n_p, which the number of solutions that dominate the solution P, and S_p the solution dominated by P, are calculated. In the first front, the domination count of all the solutions is zero. Every member q of set S_p are visited for every solution P with $n_p = 0$. If domination count of q becomes zero then it is put in the list Q which become the second non-dominated front, then all the member of Q are visited in the same way that forms the third front. The process keeps on going until all the fronts are identified[137]. The algorithm is present in Figure 4.7: Fast non-dominated-sort[144]

```
For each p ∈ P
    S_p = ∅; n_p = 0
    For each q ∈ P; {q ≠ p}
        If (p ≺ q) then                    If p dominates q
            S_p = S_p ∪ {q}                Add q to the solutions dominate by p
        Else if (q ≺ p) then
            n_p = n_p + 1                  Increment the domination counter of p
        End
    End
    If n_p = 0 then                        p belongs to the first front (F_1)
        p_rank = 1; F_1 = F_1 ∪ {p}
End
i = 1;                                     Initialize the front counter
While F_i ≠ ∅
    Q = ∅;                                 Used to store members of the next front
    For each p ∈ F_i
        For each q ∈ S_p
            n_q = n_q − 1
            If n_q = 0 then                q belongs to the next front
                q_rank = i + 1
                Q = Q ∪ {q}
        End
    End
i = i + 1; F_i = Q
```

Figure 4.7: Fast non-dominated-sort[144]

Crowding Distance and Tournament Selection. The crowding distance is used to keep diversity and prevent the local accumulation of individual solutions. It calculates the largest cuboid as shown in Figure 4.8, without any other solution in the population. The crowding distance of i is shown by the dashed box in Figure 4.8 is the average side-length. Crowding distance produces uniform dispersion by automatically adjusting a niche [145]. Higher the crowding distance the better the solution.

After evaluating non-dominated front and crowding distance, the final solution is selected based on the so-called tournament selection. The final solution is selected based on the following two conditions until the size of offspring set to become equal to the size of the parent:

1. Solution with the lower non-dominating front are considered

2. Solutions with higher crowding distance are selected

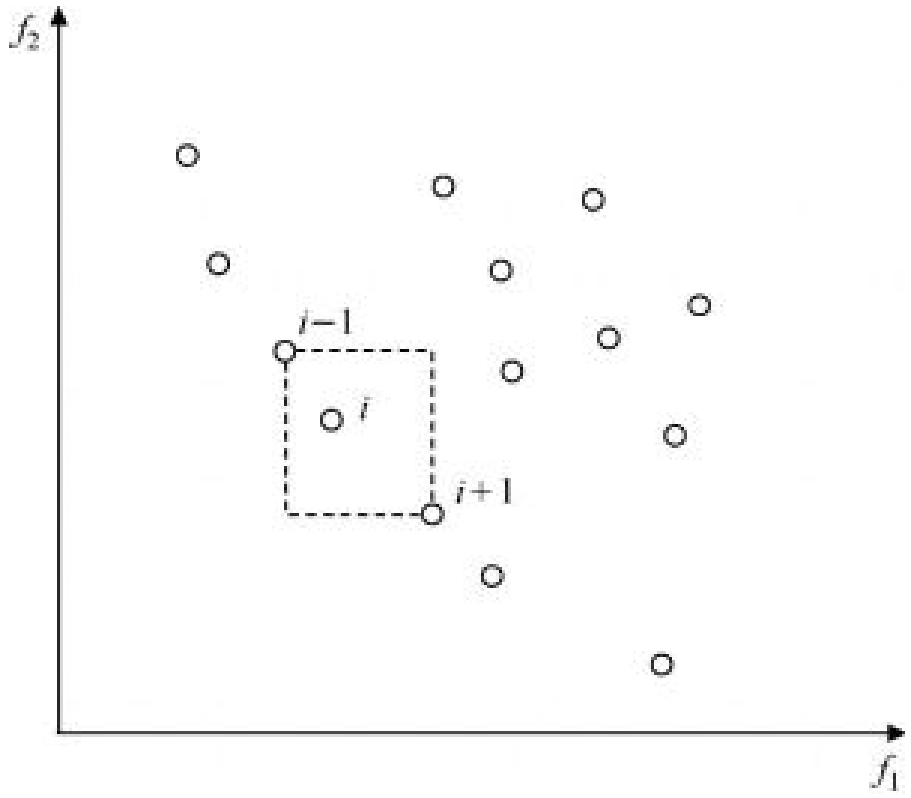

Figure 4.8: Crowding distance calculation

The all the members are sorted according to their non-dominating ranking and crowding distance which form the array R_t of size 2N i.e. twice the size of the parent, . The member of R_t is sorted in descending order of their crowding distance value. Then all the solutions are sorted in ascending order of their non-dominated rank. The first N member from the list is selected with the lowest non-dominated rank and higher crowding distance value. The remaining solutions are rejected.

4.4 Proposed Feature Selection Method

The methodology for the executing feature selection method in this chapter is shown in Figure 4.9. The proposed method starts with the collection of network data with DDoS attacks and no attack. For evaluating the proposed method, the CICIDS2017 dataset is employed. The data are transformed and normalized, which is explained in detail in Chapter 2. The dataset is divided into a 90:10 ratio, i.e., 90 percent of data for training and 10 percent of the data for validation. The normalized data become input for the employed NSGA-II-aJG algorithm exploiting six most important objectives that must be satisfied for doing feature selection in this proposed work are defined, which are explained in more detail in the subsection [146].

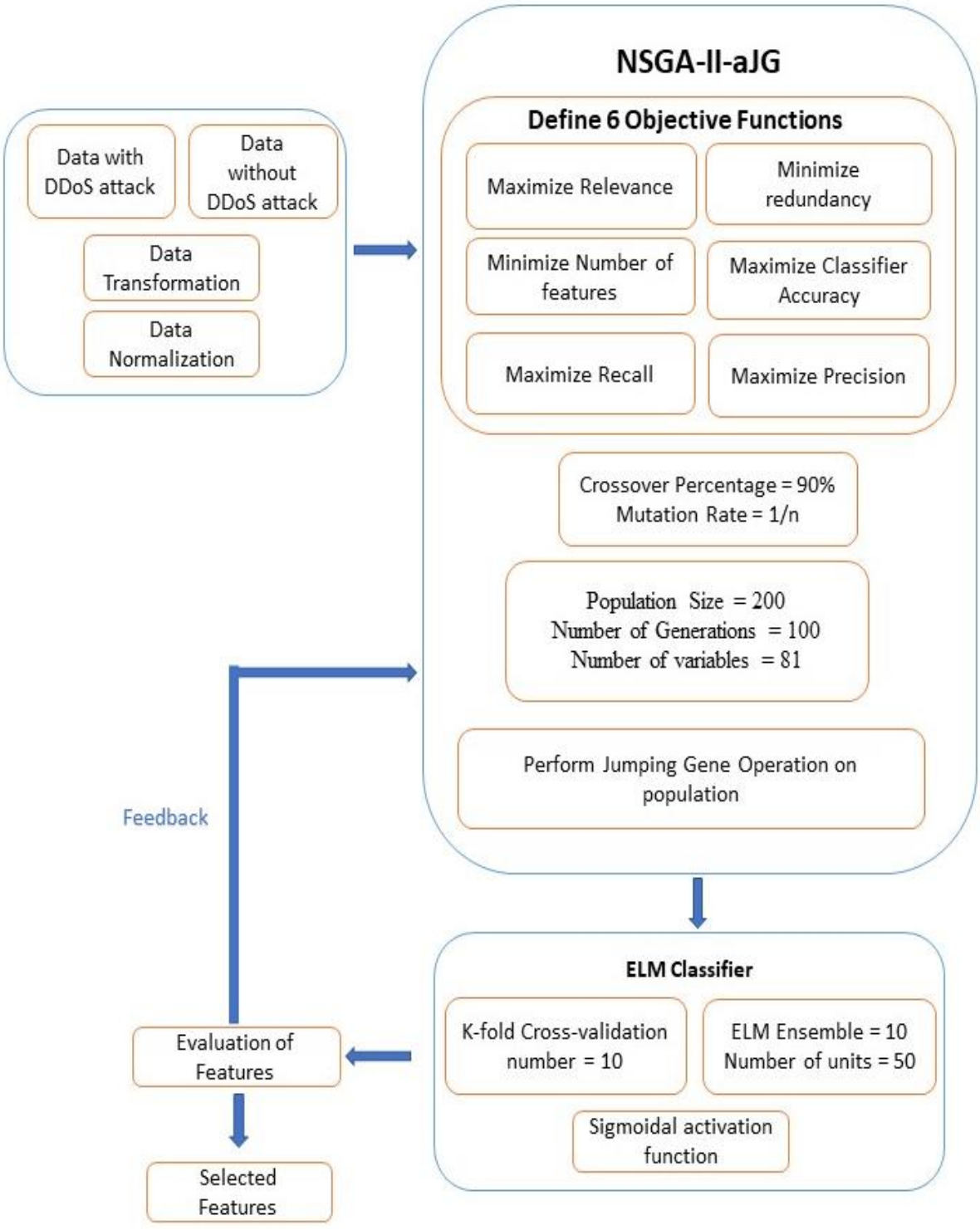

Figure 4.9 Proposed feature selection method

For evaluating and classifying features, the ELM classifier is employed, which is trained with attacked and normal data. The ELM classifier is a learning algorithm for single-hidden layer feedforward neural networks built on the idea that the input weights and hidden layer biases can be randomly assigned. The single hidden ELM has better generalization performance

than gradient-based methods, SVM, and least squares SVM and has much faster-learning speed which is desired by wrapper feature selection methods. For our work for the hidden layer, we have used a sigmoidal function as the activation function. K-fold cross-validation is repeated ten times and used for validation. Based on the feedback on the evaluation of features by the ELM classifier, the Pareto-front is generated by the NSGA-II-aJG algorithm, as explained above in detail in section 1.3, containing many sets of selected features. One of the advantages of Pareto-front is that it provides the opportunity for the user to select the feature set. The obtained results in term of different sets of features are explained in the next section

4.4.1 Objective Functions

Objective functions defined for evaluation are very critical for the feature selection for IDS. The vital cause for the failure of IDS is the wrong selection of features based on which classifier detects attacks. We have defined six important objectives which should be satisfied with the selection of features.

1) Relevance: Relevance is considered a very important criterion for selecting features; in [147]–[149]authors have used relevance as the main parameter for reducing data dimensionality. For our work, we intent to maximize the value of relevance.

Mutual Information I(X; Y) is the amount of uncertainty in X to target Y. If H(X) and H(Y) are the entropy of X and Y, respectively. Relevance is formulated as Symmetric Uncertainty is defined as:

$$\mathrm{F}\,(1): \sum_{\mathrm{x_i} \in S} \mathrm{SymUn}(\mathrm{X_i}, \mathrm{Y}) \tag{4.3}$$

$$\mathrm{SymUn}(\mathrm{X}, \mathrm{Y}) = \frac{2I(X,Y)}{H(x) + H(Y)} \tag{4.4}$$

Where S is a subset of X

2) Redundancy: Redundancy for selecting features has been proved to be a very important parameter [150]. Minimizing redundancy in data could be defined as

$$\mathrm{F}\,(2): \sum_{xi,xj \in s} SymUn(X,Y) \tag{4.5}$$

3) Number of features: The number of features within S represents the cardinality of the set. For lesser data, we expect a number of features to be as minimum as possible satisfying other objectives optimized.

$$\text{F (3): MIN(|S|)} \tag{4.6}$$

where | | denotes the cardinality of S

4) Classifier Accuracy: Classifier accuracy could be formulated as

$$\text{Max\_Accuracy: } \frac{tp + tn}{tp + tn + fp + fn} \tag{4.7}$$

where fn, tn, tp, and fp stand for false negatives, true negatives, true positives, , false positives, and, respectively.

5) Recall: Recall is one of the very important measures for attack detection in computer networks [151], [152]. The only accuracy gives the percentage of attack detection, but on its own cannot promise the correct detection of attacks as the number of false-positive and false-negative could be high. The recall is a fraction of relevant instances that are retrieved from the data. We expect recall value to be maximized.

$$\text{Max\_Recall } = \frac{tp}{tp + fn} \tag{4.8}$$

6) Precision: Similar to recall precision is also an essential measure for attack detection [153]. High precision value proves the correctness of detection of the attacks, and it can be defined as the fraction of retrieved instances that are actually relevant. We expect precision to be maximized

$$\text{Max\_Precision } = \frac{tp}{tp + fp} \tag{4.9}$$

4.4.2 Extreme Learning Machine (ELM)

Artificial Neural Network (ANN) has come out to be most efficient and popular field in machine learning area. During the training phase of ANN, the backward propagation of errors is conducted to adjust the weights of neural network. The gradient descent is backpropagated through the network literately to adjust the weights within the network. This method has got many limitations such as difficult iterative parameter tuning and slow learning rate which is due to iterative calculations [154]. Because of the these drawbacks, the training in ANN with

backpropagation require huge amount of resources and need longer time to obtain the desired result [155]. To overcome the limitation of training a ANN ELM was introduced in [156], based on least square approach. ELM perform extremely fast to map estimate and training data, input features to output targets and overcome the problem of overfitting. ELM has shown impressive performance in many real world application such as 3D human motion analysis [157], energy consumption prediction [158] and traffic flow prediction [159]. ELM has many advantages over backpropagation neural network. It generates better generalized performance because of the use of small norm of weights, without any iterative computation it calculates unique minimum norm least square solution and does not fall into local minimization problem.

The ELM classifier is a learning algorithm for single-hidden layer feedforward neural networks (SLFN) built on the idea that the input weights and hidden layer biases can be randomly assigned [160]–[162]. The single hidden ELM has better generalization performance than gradient-based methods, traditional Support Vector Machine (SVM) , and least squares SVM and has much faster-learning speed, which is desired by wrapper feature selection methods. The SVM method requires lot of time for adjusting the parameters in learning and training process such as kernel function, error-controlled parameters, and penalty coefficient. SVM also have same drawbacks as ANN, whereas SLFN only require setting the number of hidden neurons and randomly chooses the input weights and analytically determines the output weights. SLFN does not have any layer in extracting raw data into high level features from input layer before the data is processed through the outputs. For this reason, features extraction in pre-processing state is required. ELM is used as a supervised leaning method for SLFN method [163]. For our work for the hidden layer, we have used a sigmoidal function as the activation function. K-fold cross-validation is repeated ten times and used for validation.

4.5 Experimentation and Results

Experimentation in our work is done according to the methodology explained above Section. Evaluation of the proposed method is conducted in MATLAB R2017a on 64-bit Intel® Core™ i5-4690 CPU @3.50 GHz with 16 GB RAM in Windows 7 environment. Multi-objective optimization produces results as a set of Pareto-front according to the objective functions defined to be maximizing or minimizing. In our work, we have set accuracy, relevance, recall, and precision to be maximized and the number of features and redundancy to be minimized using an EML classifier as the binary classifier algorithm. Parameters values for

conducting our work are presented in Table 4.1. To optimize the performance of method, the population size should be big for a large number of features in the data, so in this work, population size is set to 200. The number of iterations is as 100, the total number of features present in our dataset is 81, including the label attribute. For the ELM classifier, the cross-validation number is set to be 10, and the number of units in ELM is considered as 50 [164].

Table 4.1 Parameter values for experimentation

Population Size	200
Number of Generations	100
Number of variables	81
K-fold Cross-validation Number	10
ELM Ensemble	10
Cross-validation number	10
Number of Units in ELM	50
ELM Kernel Activation Function	Sigmoid

The code of the program is written to produce a dump data that include the generation number, evaluate count, the total time, front count, and the average evaluation time. It includes the values of the variables every in generation and values of objective functions that are stated in the methodology section. On the specified hardware configuration system, it took 2339.76 seconds to train the algorithm.

The number of solutions obtained as Pareto-fronts in our work is more than 700 satisfying the six objective functions defined. Table 4.2 shows the obtained best subsets satisfying all six defined objectives, which have the same highest accuracy with a different number of selected features using the proposed method. Figure 4.10 illustrates the comparison of accuracy achieved against a number of selected features in a subset on validation data. The best accuracy we have achieved is 99.9%, and the least is 36.0 % with different subset sizes. The least subset size we obtained is 2 with 61.0% accuracy. The subset with minimum cardinality and highest accuracy and value of other objectives defined are as s1= {1, 7, 40, 47, 53, 62}, s2={1, 7, 17, 33, 46, 47, 53, 55, 62), s3={1, 7, 17, 46, 47, 53, 62}. The subset size having the highest accuracy value is six selected features with 99.9% accuracy; the value of relevance, recall, precision, and redundancy is 79.00%, 100%, 99.80%, and 0.19% respectively. Another best subset obtained

has nine numbers of selected features and values of relevance, recall, and precision as 79.00%, 100%, 99.80%, and 0.40 % respectively, which are the same as the previous solution but having a slightly lower value of redundancy.

Table 4.2 Subset of selected features with the highest accuracy

No. of Feature	Accuracy	Relevance	Recall	Precision	Redundancy
20	0.999	0.78	1.00	0.998	0.0653
19	0.999	0.74	0.99	0.998	0.0526
15	0.999	0.81	1.00	0.998	0.0329
12	0.999	0.74	1.00	0.999	0.0169
9	0.999	0.79	1.00	0.998	0.0040
6	0.999	0.79	1.00	0.998	0.0019

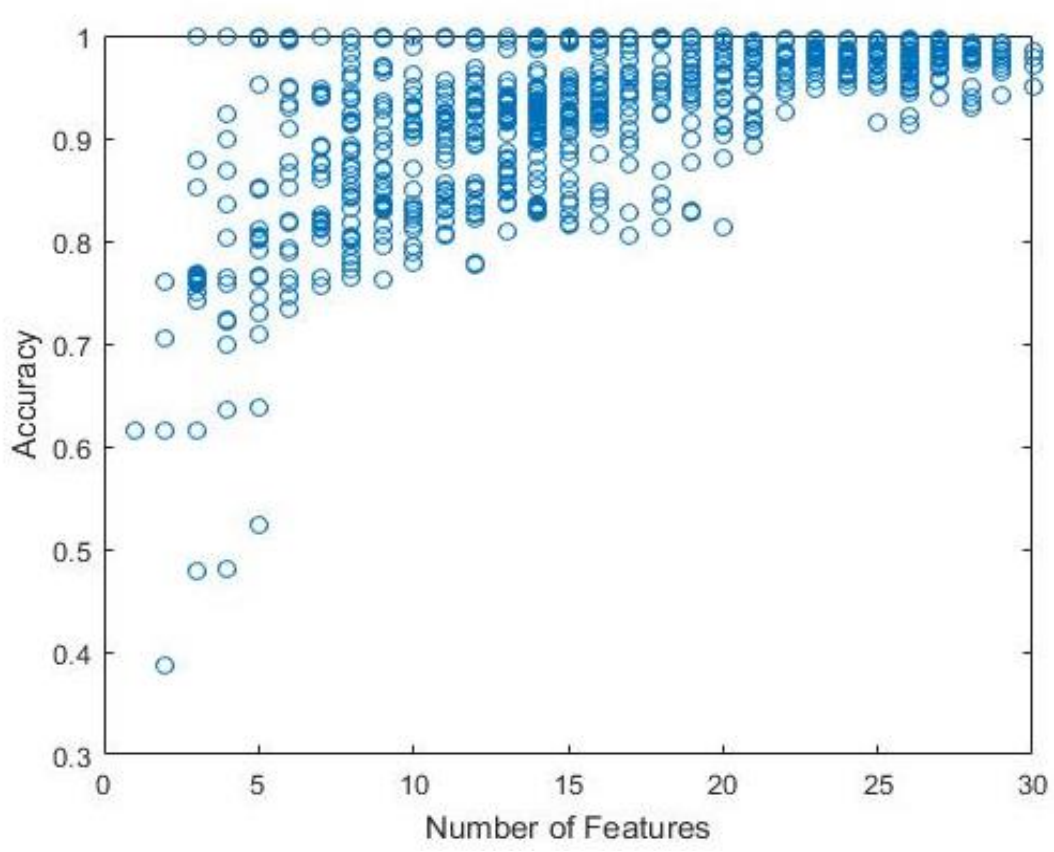

Figure 4.10: Accuracy vs. Number of Features

Figure 4.10 illustrates the accuracy values against the number of features obtained as Pareto front in Matlab. If we look for the pattern, the obtained objective values are shown in Figure 4.10. The proposed method can achieve many subsets with the highest accuracies with the low

cardinality of the subset. Another best value of accuracy obtained is 99.8% with a different number of selected features. Figure 4.11 illustrates the comparison between the precision values and the corresponding feature sets obtained as Pareto-front. The maximum precision value obtained is 99.90% with 15 number of features; the next maximum precision value obtained is 99.80% with six number of features. The majority of the Pareto fronts with high precision values are found with the number of features greater than nine. After fifteen number features, the precision value attained is mostly above 80%. Another interesting result from Figure 4.11 is with the thirty number of features; in this case, the precision value is lower than 97%, and the best precision values are falling in the range below 22 number of features.

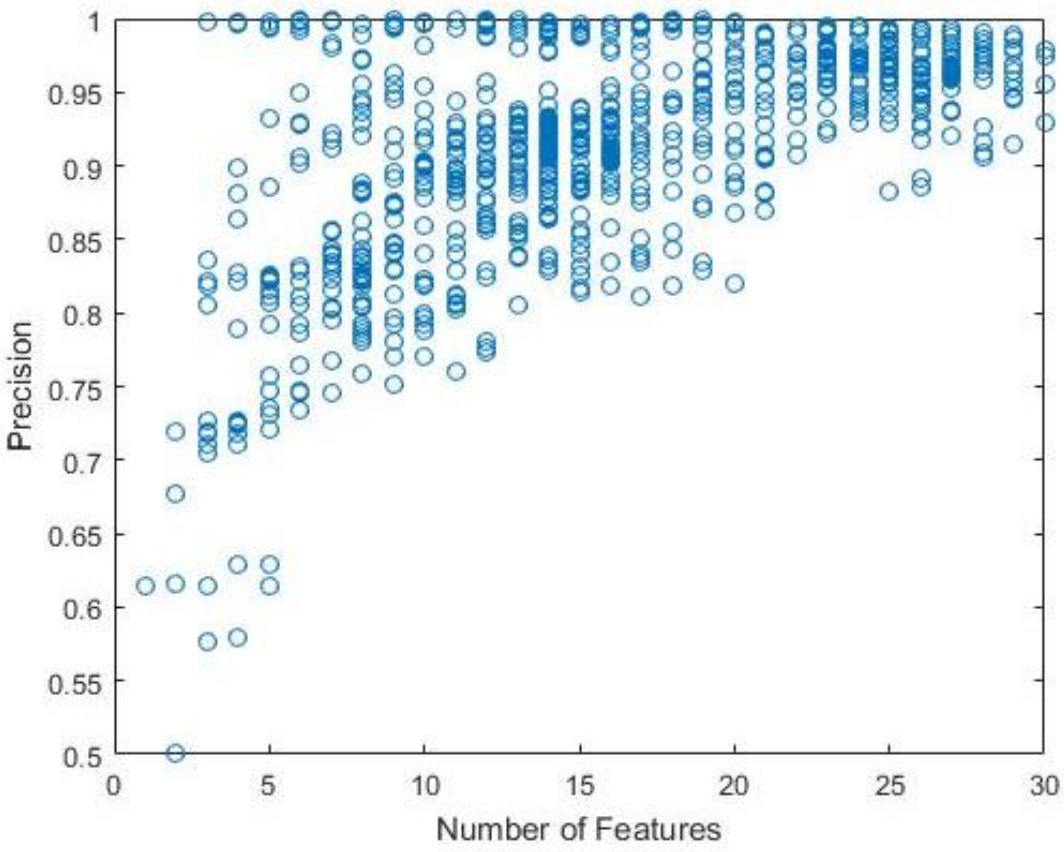

Figure 4.11 Precision vs. Number of Features

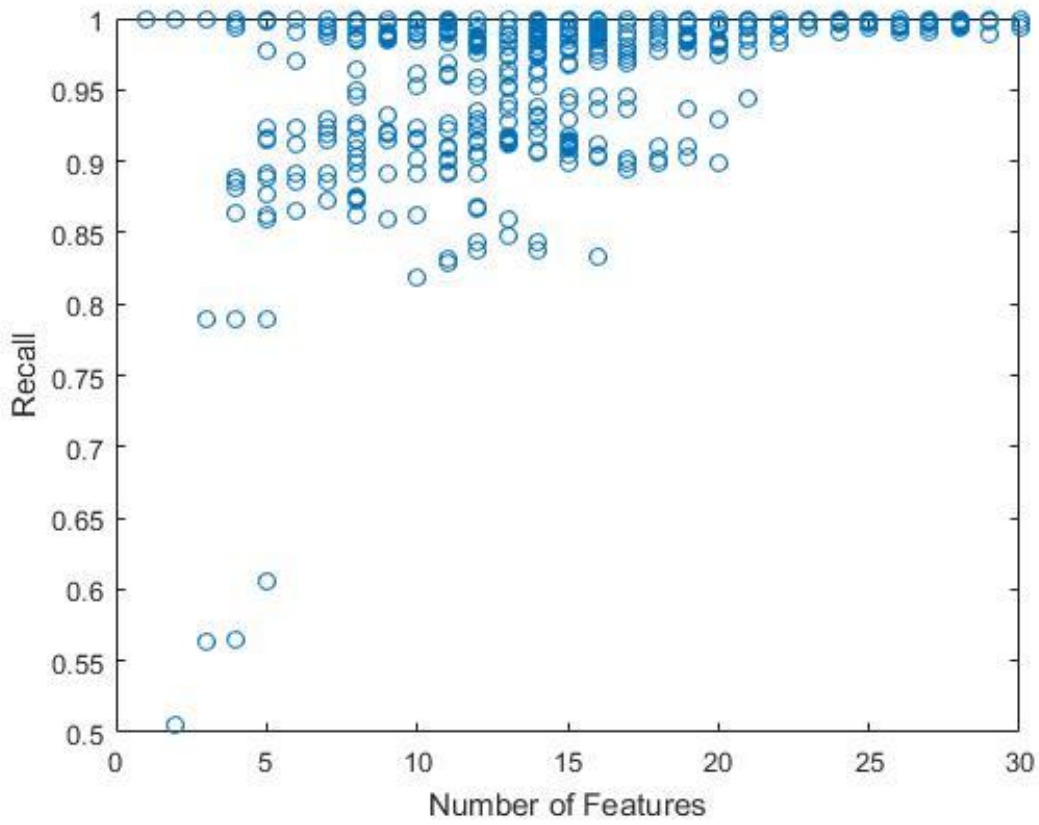

Figure 4.12 Recall vs. Feature size

Another evaluation result the proposed method has in terms of recall values. Figure 4.12 shows the relationship of the cardinality of the subset with the value of recall. It is interesting to note that in comparison with the precision values, the recall values are mostly 80 percent. The maximum recall value obtained is 99.9% with a minimum of 3 features and also with 30 features. The majority of Pareto front is falling between 7 and 20 number of features with values greater than eighty percent recall value. It is interesting to note that the majority of the Pareto front has achieved more than 94 percent recall value in the range of 10 and 17 number of features. It is interesting to note that the value of precision and recall obtained is high in the majority of subsets which confirms the correctness of the obtained accuracy that proves that the false positives and false negatives are minimized, which validates that the detection of attack is true by this method.

The comparison of redundancy and the number of features is presented in Figure 4.13; it can be seen that the redundancy is minimum with low cardinality of subsets and is increasing with the size of the subset. The minimum redundancy is falling between 7 and 18 number of features. From these results, it can be concluded that the accuracy achieved by the proposed method is on the true prediction of the DDoS attack in the dataset.

Further, Figure 4.14 shows the value of the true positive against the cardinality of the subset, and Figure 4.15 illustrates the relation of false positives with accuracy. As a result, by using our method, we have achieved a low false positives rate of nearly 0 with the least number of features such as six numbers of features. From both Figure 4.14 and Figure 4.15, it is clear that true negatives are higher, and false-positive rate is least in the majority of the subsets with high accuracy, which attains the desired aim of the proposed method.

The main objective of this paper is to find a set of Pareto-front, having the best solution satisfying all the objectives defined in section 4.4.1, which are clear from the results discussed above, is achieved by our work. The significance of our proposed method is that this has reduced the number of features from 80 to 6 selected features which are required for the detection of DDoS attacks in the CICID2017 dataset.

The purpose of the feature selection is to reduce the dimensionality of the data and find out the most important features providing correct detection of the attack in the system. Figure 4.16 showing the frequency of occurrence of features. Based on Figure 4.16 data collected, the most selected features can be seen which are able to detect DDoS attacks in the system.

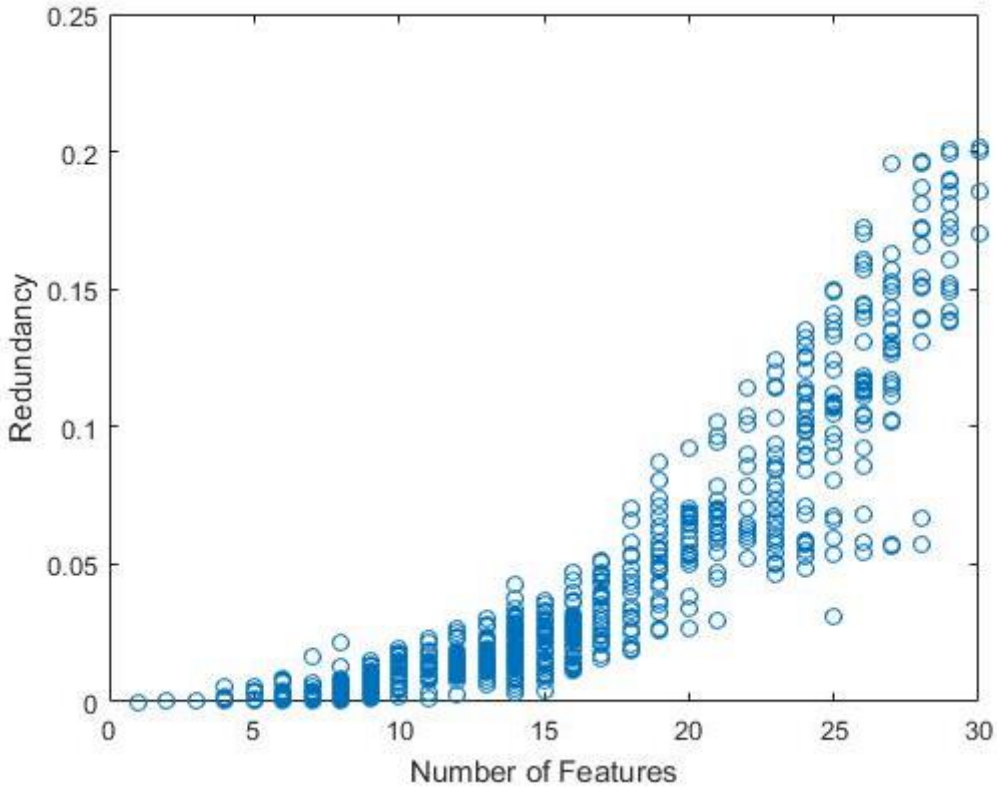

Figure 4.13 Redundancy vs. Feature size

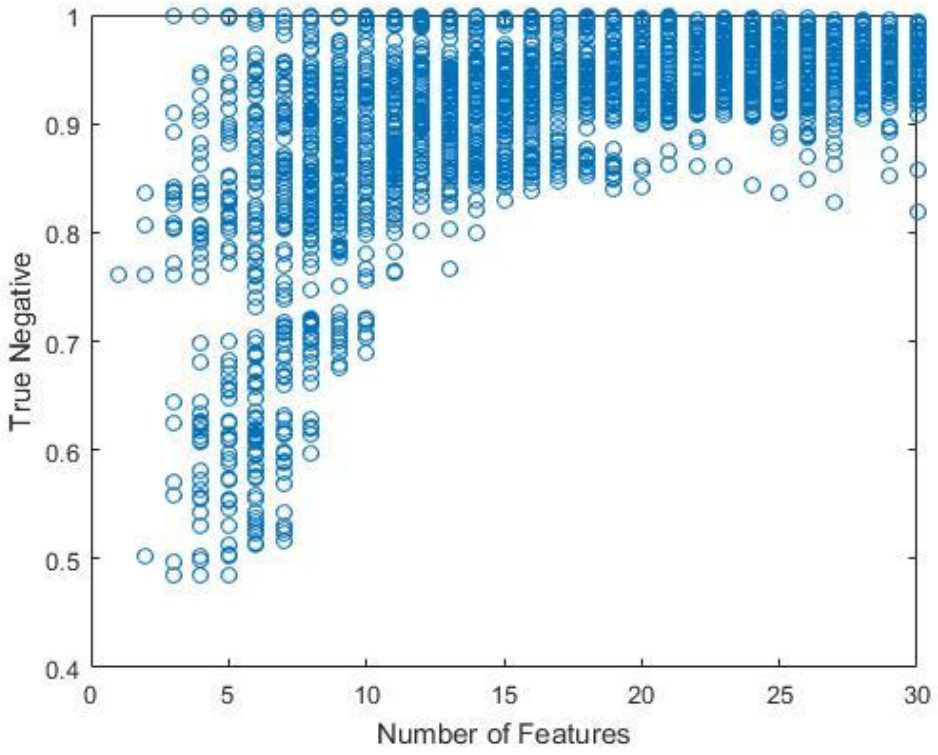

Figure 4.14. True Negative vs. Feature size

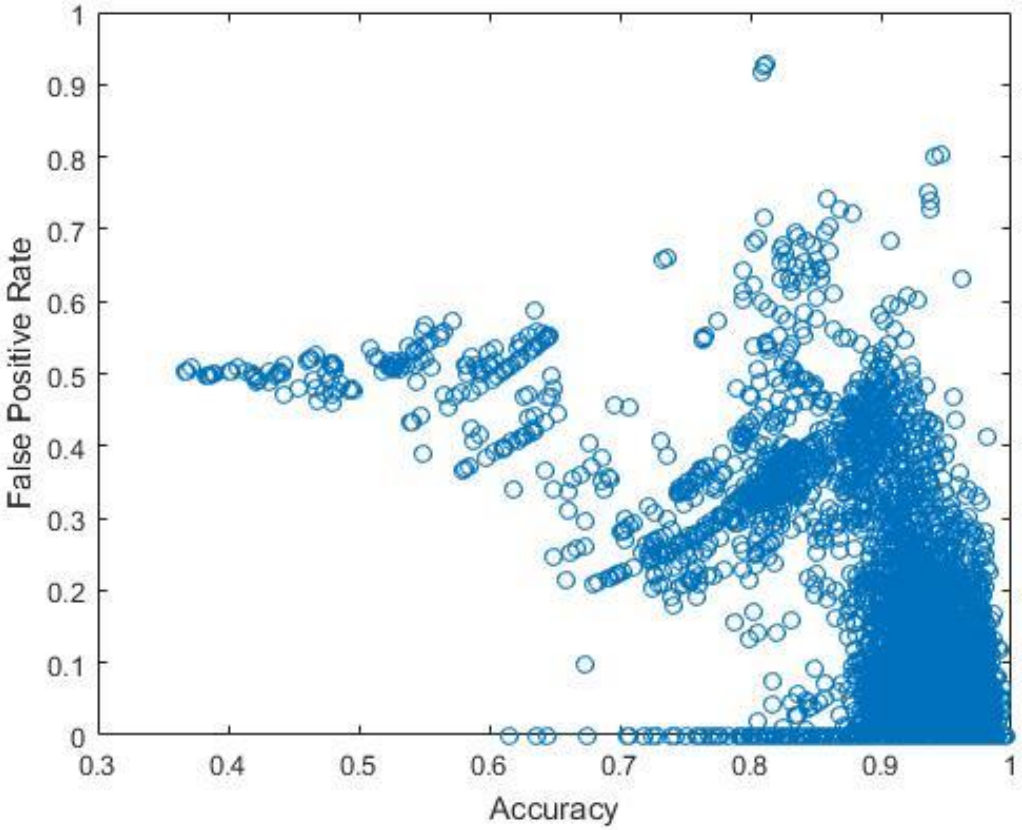

Figure 4.15 False-positive rate vs. Accuracy

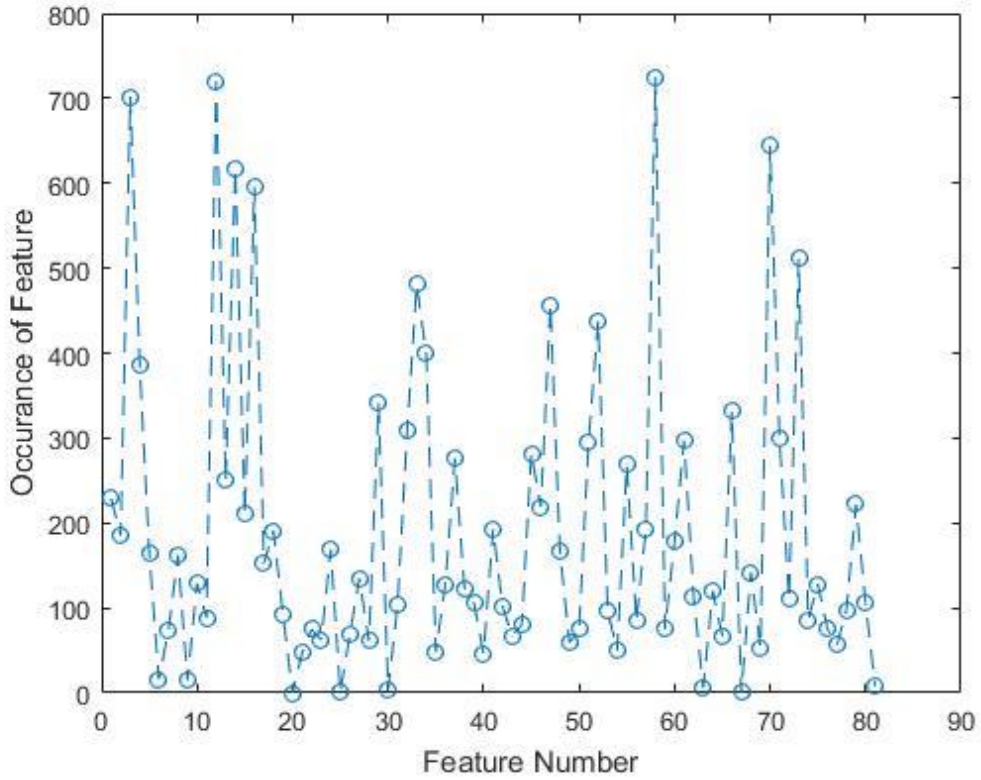

Figure 4.16 Frequency of Occurrence of features

Table 4.3 presents the top 10 features having the highest frequency of selection in different subsets. These are the most selected features which satisfy the objective functions we set for the detection of DDoS attack. On the basis of this, we can say that these features may be helpful for the detection of DDoS attacks.

Table 4.3 Top 10 most occurring features

Most selected Features in CICIDS2017 dataset
Avg Fwd Segment Size
Fwd Packet Length
Destination IP
Subflow Bwd Bytes
Fwd Packet Length Std
Bwd Packet Length Min
act_data_pkt_fwd
Bwd IAT Max
Packet Length Variance
ACK Flag Count

Another interesting result we obtained is that the proposed method could improve the runtime of the ELM classifier. The average runtime by performing feature selection is reduced as compared to the runtime without feature selection; however, the training time has increased in the case of a feature selection method. Table 4.4 summarizes the comparison of performance metrics with and without feature selection method for the ELM classifier on validation data. The highest accuracy we achieved is 97.89% with the proposed feature selection method is achieved as 99.90% and the best precision value as 96.30% without feature selection and with our method is evaluated as 99.80%. It is also interesting to see that without feature selection, the redundancy value is 0.17%, which is near to zero by implementing our proposed work. The detection runtime with FS is 0.02 seconds and without FS is 0.11 seconds whereas the training time with FS is 2339.76 seconds and without FS is 3472.36 seconds. It is clear from the data in Table 4.4; the feature selection strategy directly affects the performance of the classifier.

Table 4.4 Comparison of performance measures with and without FS

	Accuracy	Recall	Precision	Relevance	Redundancy	Runtime (Seconds)	Training Time (Seconds)
With FS	99.90	1.00	0.998	0.790	0.0019	0.02	2339.76
Without FS	96.89	0.98	0.963	0.722	0.1790	0.11	3472.36

The proposed method is compared with the other five state-of-art methods for the detection of DDoS attacks. Apriori+LSSVM [165] the method is proposed in that it adopted the Apriori algorithm with the least-squares Support Vector Machine (SVM) to perform the feature selection for the power quality event recognition system. The proposed method has achieved 98.88 % accuracy with five number of features on real power quality event data. Decision three methods [166] attained 98.38 % accuracy with 16 number of features on KDD datasets [167]. DCF+CSE [168] has employed consistency subset evaluation and DDoS characteristic features for performing feature extraction with an accuracy of 91.70% with a cardinality set of 17 selected features on NSL-KDD 2009 [169] datasets.

BN+C4.5 [74] method adopted in combined Bayesian Network (BN) with a Decision tree (C4.5) which attained accuracy as 99.80 % with 10 features Square on benchmark datasets.

Chi+ Symmetrical Uncertainty [170] that adopted chi-square and symmetrical uncertainty together with Decision tree classier has achieved 88.00% accuracy with eight extracted features on CAIDA real life network traffic datasets. Table 4.5 summarizes the comparison of our proposed method for feature selection with recent publications. It can be seen from Table 4.5 that our proposed method outperforms other proposed state-of-the-art methods achieving 99.90 % accuracy with six number of features.

Table 4.5 Comparison with other work

Method	**Number of features**	**Accuracy**
Apriori+LSSVM [165]	5	98.88
Decision Tree-based [166]	16	98.38
DCF+CSE [168]	17	91.70
BN+C4.5 [74]	10	99.80
Chi-Square + Symmetrical Uncertainty [170]	8	88.00
This paper	6	99.90

4.6 Chapter Summary

In this chapter, a multi-objective optimization method for feature selection to detect a DDoS attack is proposed. This method is different from the traditional multi-objective feature selection algorithms which are based on a few objectives only such as the number of features and a measure of classification accuracy. The latest CICIC2017 dataset which is close to real-time data and does not have the shortcoming of other benchmark datasets are exploited, for the detection of the DDoS attack in IoT networks. In this work, an NSGA-II-aJG algorithm with six different conflicting objectives as relevance, redundancy, number of features, classification accuracy, recall, and precision using ELM as binary classifier algorithm for the detection of DDoS arrack is proposed. This method searches in a larger space enabling algorithms to generate a large number of Pareto-efficient solutions. The proposed method obtained many subsets of selected features as Pareto-front each with the same and different selected features. The best solution obtained with the proposed method has reduced the total number of features from 80 to 6 numbers of selected features having highest accuracy as 99.90% Along with the highest accuracy this subset has highest value of other objectives defined such as the value of recall is 100% and precision is 99.90% and redundancy of 0.20% which is best among all the

subsets. It is proved that the performance of IDS is brought down when using full features present in the dataset, so the feature selection method is crucial for the performance of IDS in terms of both complexity and runtime. The proposed method is compared with other previously proposed state-of-the-art methods; it is found that the proposed method has achieved the best results in terms of accuracy and other critical objectives selected. On the basis of results obtained, it can be concluded that the proposed method to detect DDoS satisfying six conflicting objective functions has achieved its goal to reduce the number of features with the true value of accuracy. The achieved results are quite satisfying, so it can be concluded that the method provides the best results as the selection of features for DDoS attack detection. On the basis of the literature review conducted in Chapter 2, it can be concluded that for the development of advanced IDS, the Deep Learning is an idle choice that is capable of learning on own its own and can deal with enhanced more sophisticated DDoS attacks, based on the current challenges and requirement. In the next Chapter 5, the Deep Learning Models for detection of cyberattacks are proposed.

Chapter 5. Deep Learning Models for Cyber Security in IoT Networks

In this chapter, four deep learning models applicable to building IDSs for delivering the cybersecurity in IoT networks are proposed and compared. The overviews of the Artificial Neural Network and feed-forward neural network are detailed. The introduction of deep learning techniques, including the details on CNN, MLP, and LSTM, are covered. In the later section, the proposed deep learning models are evaluated, and the obtained results are discussed. The proposed models are compared with other machine learning algorithms. As the sensitivity and sophistication of DDoS attacks on IoT networks have increased, an advanced IDS that can learn on its own is demanded. The machine learning algorithms for the classification of attacks have been inefficient and failed in the detection of the cyber-attacks. [171]–[173] are some of the examples where machine learning based IDS failed to detect attack due to various reasons. In this chapter deep learning methods for the classification of normal and attacked data are proposed.

5.1 Introduction

Deep Learning techniques have provided excellent results in the areas such as video processing, image processing, BigData and natural language processing etc. Recently the interest of researchers in the field of cybersecurity has shifted towards the field of deep learning seeing the promising results achieved in other fields. A few years back from now, the concept of deep learning was still there but because of the demand for the high volume of data, processing power and other resources, this technique was not much used in practical applications. There exist three types of learning algorithms as supervised, unsupervised and semi-supervised learning. Deep learning is the broader subfield of machine learning which is a more extensive neural network and can be employed in all three types of learning methods. The hypothesis of deep learning was first introduced in 1980 as multi-layer artificial neural networks. However, that time the training of the deep learning models was not practically possible because of limited processing power available and so longer training time was required. Also, the older deep learning algorithms suffered from vanishing gradient problems. The concept of deep learning method was first applied as a deep belief network and it proved to be highly effective in fields such as big data, image processing, natural language processing and self-driving car etc. Limitation of deep learning techniques is the long time it requires for training. Higher the training data, higher is the training time, but deep learning methods need massive data for training for performing well. The deep learning is brought into practical applications by the advancement in hardware which is much faster and launch of GPUs as a

processing unit. At present, all the big IT companies are exploiting the deep learning technology in their product and services. This chapter proposes and compares the performance of MLP (Multi-Layer Perceptron), CNN, LSTM, and hybrid CNN+LSTM model for the detection of DDoS cyberattacks in the IoT networks. The four deep learning models are extensively evaluated to obtain the performance for the detection of DDoS attacks. The latest CICIDS2017 datasets preprocessed in Chapter 3, are employed as input to all the deep learning models. The deep learning models are compared with other extensively used machine learning algorithms in the field of cybersecurity.

5.2 Deep Learning

Deep Learning is the field of artificial intelligence that aims to develop a system that can automatically learn on its own and improve with the experience without human interference [174], [175]. Deep Learning builds a computer system that can collect enormous data and construct models to process and learn from the data to make decisions based on the obtained data. Deep learning is ANN with multiple hidden layers. The deep learning and machine learning models are broadly classified into three categories as unsupervised learning, supervised learning, and semi-supervised learning. In the case of supervised learning, the learning models are provided with labelled data for training; these trained models are used to predict data without the label. The unsupervised learning algorithm train on the data without the label. The learning model infers functions to find hidden features and structures within the data. In the case of semi-supervised learning, the models are provided with large unlabelled data and small-sized label data. The concept of deep learning method was first proposed in [176] based on deep belief network and it has offered excellent results in the fields such as image processing, natural language processing and self-driving car etc. Since then, the field of deep learning has drastically improved and has been very impressive in fields such as image processing, video processing and natural language processing, etc. [174], [175], [177].

5.2.1 Multilayer Perceptron

The *Perceptron* is a single layer ANN with only one neuron. The Perceptron is a network that computes the linear mixture of its Boolean/real-valued inputs and feeds it to a threshold activation function:

$$O = \text{Threshold}(\Sigma_{i=0}^{d} w_i x_i) \quad (5.1)$$

where x_i are the inputs $\mathbf{x}_e = (x_{e1}, x_{e2},..., x_{ed})$ from the set $\{(\mathbf{x}_e, y_e)\}_{e=1}^{N}$

Threshold is the activation function defined as follows:

$$\text{Threshold}(s) = 1 \text{ if } s > 0 \text{ and } -1 \text{ otherwise} \quad (5.2)$$

Perceptron classifies the data subject to the value of sum i.e. if the sum is greater than the threshold value $\Sigma_{i=1}^{d} w_i x_i > -w_0$ or the sum is less than the threshold value $\Sigma_{i=1}^{d} w_i x_i < -w_0$. That gives another name for Perceptron, which is the threshold logic unit (TLU). The above formulation imagines that the threshold value w_0 is the weight of an additional connection constantly held to $x_0 = 1$.

To be able to solve nonlinearly separable problems, several neurons are connected in layers to build a multilayer perceptron. In MLP, each perceptron identifies a small linearly separable section from the input fed to it. The final output of the MLP is produced by the combined output of all the perceptions from all previous layers. The input to the inner neurons is prevented by a step function, also known as a hard-limiting function for producing output. This problem is solved by replacing step function by a continuous function such as a sigmoid function in case of a binary classifier. In MLP, the neurons are arranged into an input layer, an output layer, and one or additional hidden levels as shown in Figure 5.1. The MLP follows a learning rule named the generalized delta rule or, in other words, known as backpropagation rule. During the training of MLP, on getting and input, the generalized delta rule calculates an error function which is backpropagated to precious layers. This process is repeated for all the inputs. The error function is exploited to adjust the weights of nodes in direct proportion to the error in the connected units to that node.

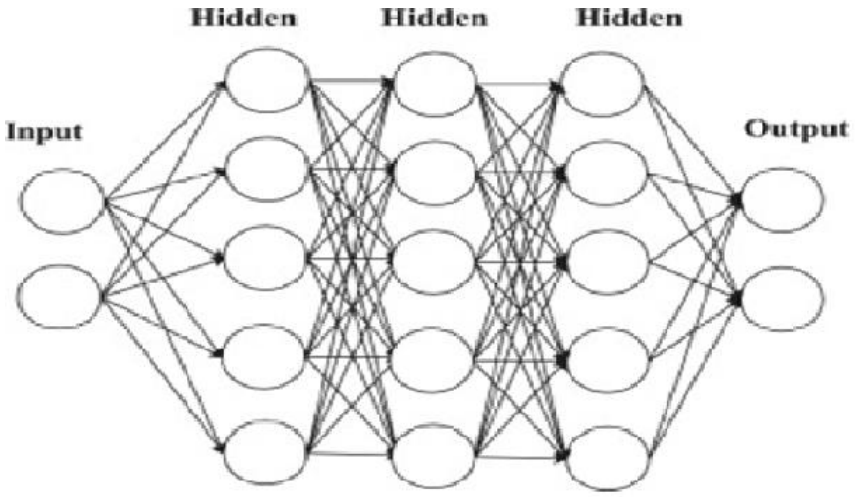

Figure 5.1 The basic architecture of MLP

Learning Difficulties in Multilayer Perceptron

The MLP may fall into a local minima situation which will make it unable to learn the correct output. So, the learning rule in MLP does not guarantee to produce convergence. Another problem with MLP is global minima, in which it finds itself in one of the local minima because of the gradient descent strategy followed.

Advantages of Multilayer Perceptron

- **Generalization**. MLP is capable of generalization so they can classify an unknown pattern based on other known patterns with the same distinguishing features. This feature of MLP makes it unique to classify noisy or incomplete input based on their similarity with complete inputs with distinguishing features.
- **Fault Tolerance**. Another feature of MLP is that they are highly fault-tolerant known as graceful degradation, so, in case of loss of neurons and its interconnections, the MLP keeps on learning even if the damage is relatively quick.

5.2.2 Convolutional Neural Networks (CNN)

CNN [178], [179] is inspired by the visual processing of the animal visual cortex. The study of the visual cortex was first presented in [180], from which we now know that the visual cortex is a very complex arrangement of cells. Each cell is sensitive to a small area of the whole visual field known as the receptive field. These receptive fields are tiled over the entire visual field. The cells exploit the natural image and look for the correlation. Two kinds of cells have been identified, the first one is the simple cell which is sensitive to specific edges such as patterns in the receptive field, the second kind of cell is named as the complex cell that is sensitive to a constant area in the pattern and generally has large receptive fields. The CNN was developed and proposed in [181] as deep feed-forward neural networks. CNN is the most significant innovation in the field of computer vision. CNN has given remarkable improvement, especially in the field of image processing and natural language processing. Some of the major companies are using CNN at the core of the services they are providing. Some of the examples are Google using for picture search; Amazon uses this for recommending products to the customers, Instagram for search feature and Facebook use for automatically tagging. The basic structure of CNN is depicted in Figure 5.2. CNN has three necessary layers as part of its structure viz. convolutional layer, pooling layer, and a classification layer.

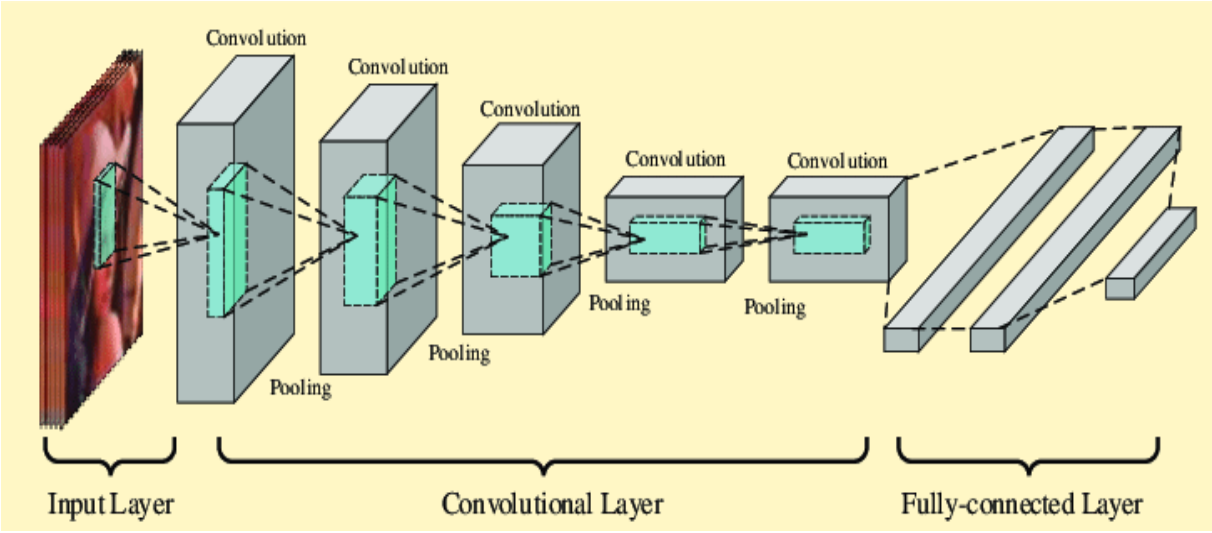

Figure 5.2 The basic three-layer architecture of CNN **[182]**

Convolutional Layer: Convolutional layer takes input as 3D shape image a (m X m X r) where m is height and width of the image, and r is the number of channels, in case of a color image the number of channels is 3 as it RGB image. Another feature of the convolutional layer is filter or kernels which also a 3D matrix. There can be any number of filters in a convolutional layer; each filter is sensitive to a particular edge in an image. The structure of the filter is the same as image as (n x n x q) with one condition that n less than m and q could be the same as the number of channels or less than that. Generally, the value of q is kept the same as that of the number of channels. The filters are convoluted over the entire image to find feature maps. The feature map is a 3D matrix of size (m-n+1), and it contains the value of the sum of the product of values in the filter matrix and the original pixel matrix over which the filter is consulting. Each map is further subsampled by another convolutional layer or a pooling layer.

$$net(i,j) = (x * k)[i,j] = \sum_{m}^{0} \sum_{n}^{o} x[m,n]k[i-m,j-n] \tag{5.3}$$

Where $net(i,j)$ is the output in the next layer, x is the input image, and k is the kernel or filter matrix and $*$ is the convolution operation.

Pooling Layer: The convolutional layer is followed by a pooling layer whose main task is sub-sampling the output from the convolutional layer and provide a lower resolution representation of the feature map. The pooling layer works by summing up similar information of the receptive field and produce a dominant response as output within the local region. The pooling operation can be represented by the equation.

$$Z_i = f_p(F_i(\mathrm{x}, \mathrm{y}) \quad (5.4)$$

Where Z_i is i^{th} output feature map, $F_{i\ x,y}$ is i^{th} input feature map, and f_p is pooling operation. The pooling layer resolves the overfitting problem and reduces the dimension of feature maps. The computation of the statistics of the activation can be done by applying max pooling, mean pooling, or weighted pooling. The most common pooling function applied in CNNs is max pooling, which can be computed as:

$$P_{cn} = \max \mathrm{cn} \in \mathrm{S}(C_{cn}) \quad (5.5)$$

Where P_{cn} is the output of pooling layer, S is pooling block size,

Fully Connected Layer. The fully connected layer consists of a layer in which all the neurons are connected to all activations in the previous layer. The classification of data into the label is performed by a fully connected layer that receives input from the convolution or pooling layer. The fully connected layer work by flattening the output from the convolution or pooling layer into a single vector of values that represents a probability of a feature belonging to a particular layer. The weights in a fully connected layer are sent back through the backpropagation process for adjusting to accurate weights.

5.2.3 Long Short Term Memory (LSTM)

LSTM is a sort of Recurrent Neural Network (RNN), which is a feed-forward neural network [183] LSTM solves the vanishing gradient problem of the RNN model, and it retains information in a gated cell. Just like computer memory, in LSTM, information can be read, written, or stored in a cell. To solve the problem of vanishing gradient, LSTM has three gates; Figure 5.3 presents the diagram of the LSTM model. The first is the Cell state or forget gate, second is the Input gate, and the third is the Output gate. The input gate read the input data from the training dataset and take the decision to update the current state. The output of the cell body is controlled by the output gate. The LSTM's self-recurrent connection is managed by the forget gate; it decides whether to store or forget the precious state vector. represents the underlying architecture of an LSTM network.

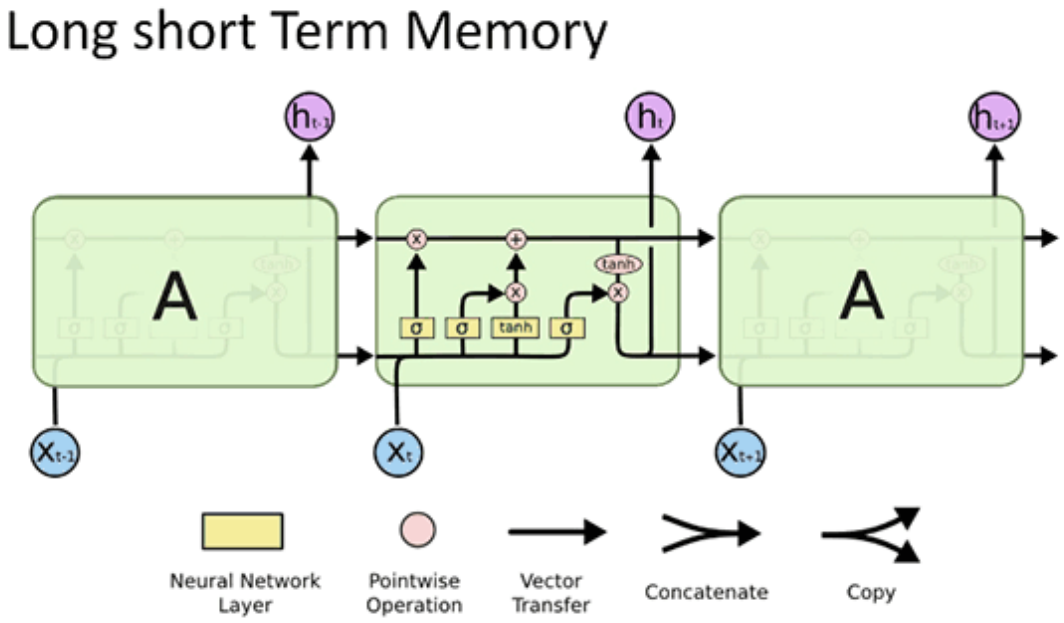

Figure 5.3 The basic architecture of the LSTM network **[184]**

Input Gate. Every cell body has two inputs, The first x_t is the current input, and the second input is the data stream from the precious cell status. The input gate will decide what new information is to be stored in the memory cell. The vector of new information candidate to be added to the state is created by a tanh layer afterwards.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.6}$$

$$\mathrm{Ct}' = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5.7}$$

Where i_t is input, W_i is the weight of the input gate x_t is input at time t, Ct' is an intermediate state, h_t is the current state.

Forget Gate. The forget gate will process the current input data; it will decide whether it should be ignored. The old cell state C_{t-1} is multiplied by vector f_t and added to i_t * ct', which can be written in the equation as

$$\mathrm{Ct} = f_t * c_{t-1} + i_t * \mathrm{Ct}' \tag{5.8}$$

The output gate. The output gate decides what information is to be produced as output from the cell memory. The sigmoid function is applied to the previous hidden state and present input which is later multiplied by tanh applied to the new memory cell.

$$o_t = \sigma(W_o[h_{t-1,}x_t] + b_o) \quad (5.9)$$

$$h_t = o_t * \tanh(C_t) \quad (5.10)$$

5.3 Deep Learning Models

The four different classification deep learning models that are elaborated in the above section, MLP, CNN, LSTM, CNN+LSTM for the detection of DDoS attacks have experimented in this work. The feasibility of DL in the field of cyber security has been explored in some recent research work. MLP has been extensively employed in the classification prediction problems. [185] [186] [187] are some of the examples where MLP has shown to obtain impressive results. The MLP has advantage to learn from the incomplete data, it is fault tolerant, are less complex, easy to design, easy to maintain, fast and speedy in terms of computation, highly responsive to noisy data and are capable of learning and generalizing the accumulated knowledge. Another DL model which is exploited in our work is CNN. One of the advantage of CNN is that it can detect the important features without human support automatically. CNN has been used for the detection of DDoS attacks in [188]–[190]. In CNN each output value hangs on a small number of input values, this is known as sparsity of connections. The sparsity of connections reduces overfitting during training and retain the size of the network substantially small at the same time. RNN are useful because they are not limited by the length of an input and can use temporal context to improved forecast meaning. The basic objective of LSTM is to attain vanishing gradient descent which is an optimization algorithm that calculate neural network weights to avoid long term dependency problems. The deep learning model's performance is compared with machine learning algorithms [191]. The models are tested on various hyperparameter values such as the learning rate, and those provided the best results are discussed in this chapter. For all the models the last layer is dense with sigmoid activation function as our data has two classes as normal and attack. Another DL technique we have implemented is LSTM that capture long term dependencies.

5.3.1 MLP deep learning model

Figure 5.4 shows the flow chart of the MLP model implemented for this work. Input shape for the MLP model is 2d data; the dataset employed is in the form of the matrix, so there is no need to change the shape of the dataset for this model. The proposed model consists of the first input layer, followed by three dense layers. The output from each layer becomes the input to

the next layer. One dropout layer is added to save the system from heating. The output from the dropout layer is fed to the fully connected layer, which then provides input for the dense layer with sigmoid function.

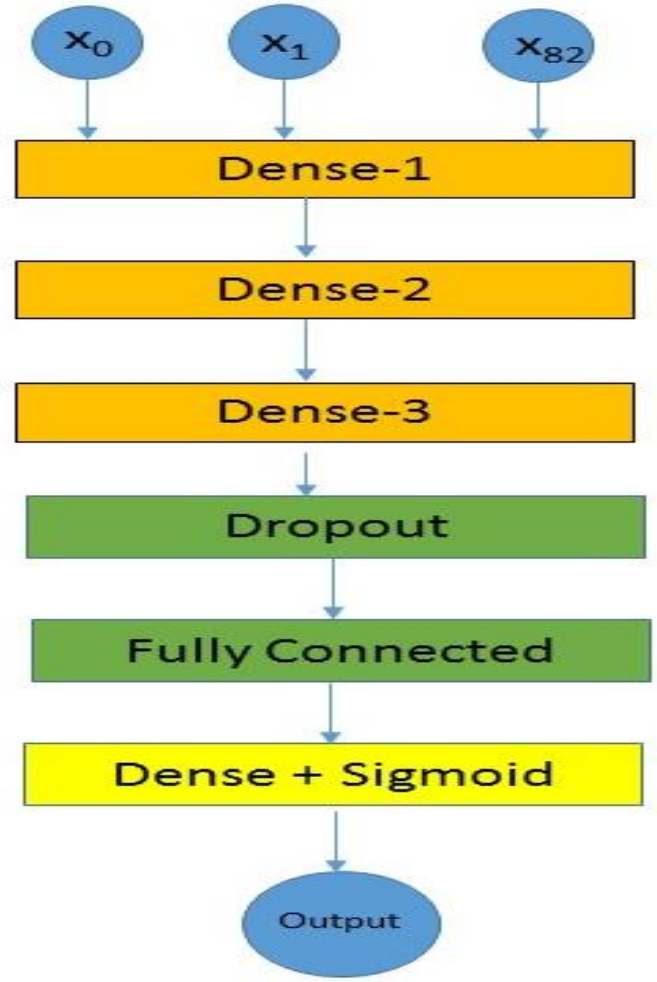

Figure 5.4 MLP deep learning model

5.3.2 CNN deep learning model

Figure 5.5 presents the architecture of the CNN model employed. In any CCN model, there are three types of the main layer as a convolutional layer which, pooling layer and dense layer. 1d-CNN accepts input shape of data in the 3d form as (batch, steps, channels) the dataset is converted to a 3d shape accordingly. Dataset used total has 83 attributes, including last label attribute so we converted data using reshape function as {data.shape(0), data.shape(1), 1} and fed input shape as {81,1} and used relu as activation function. Max pooling layer is added discard features with a low score and keeps only features with the highest score. The last layer is dense with a sigmoid activation function.

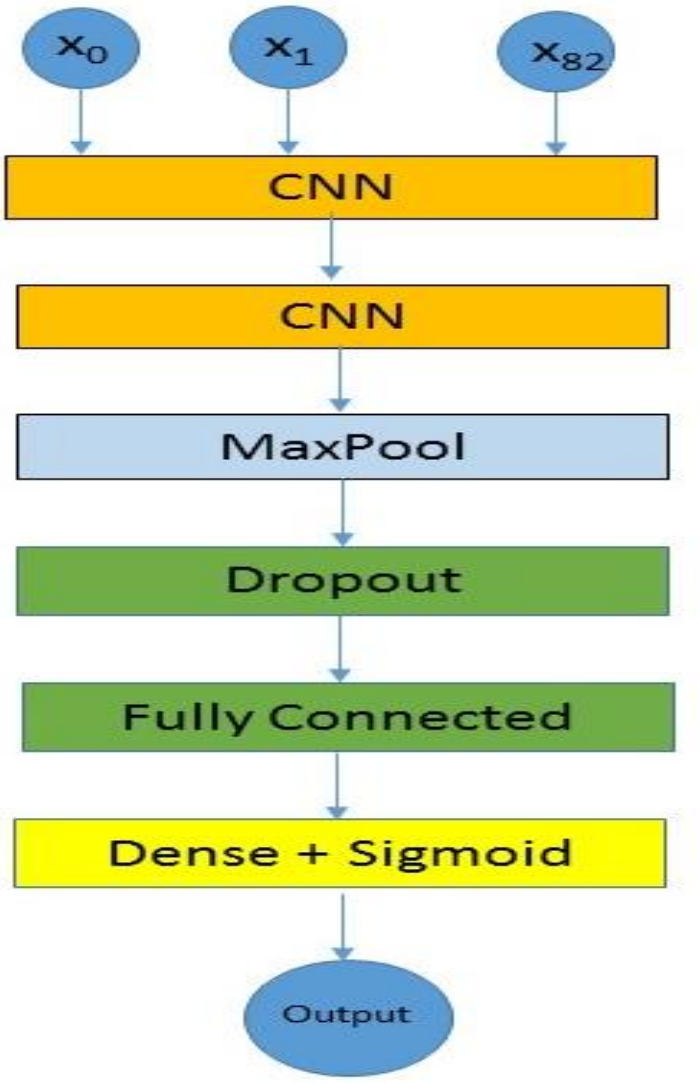

Figure 5.5 CNN deep learning model

5.3.3 LSTM deep learning model

LSTM is a type of RNNs in that nodes are connected to other nodes in the same layer to improve learning by removing and remembering specific information. The flow graph of the LSTM model is presented in Figure 5.6, and it accepts input shape of data in the 3d form as (batch, steps, channels); the dataset is converted to a 3d shape accordingly. This model consists of the first LSTM layer with 128 kernels using adam activation function followed by a dropout layer with a rate of 0.5. The output from the dropout layer is connected to a fully connected layer which provides input to a dense layer with sigmoid function to classify attack and normal data.

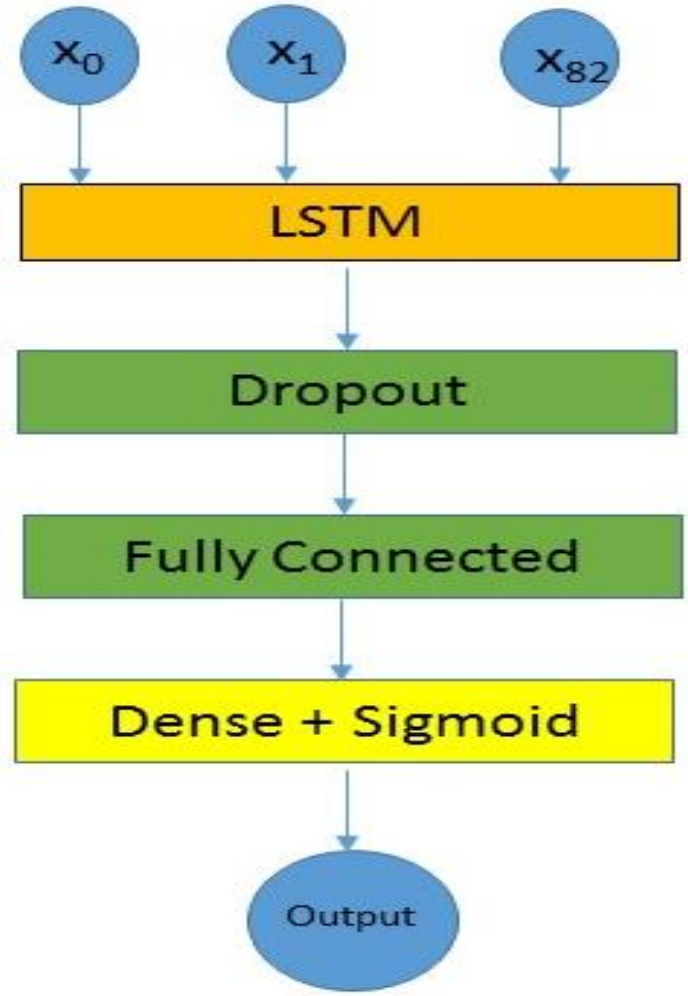

Figure 5.6 LSTM deep learning model

5.3.4 CNN+LSTM deep learning model

A hybrid CNN with LSTM model is implemented, Figure 5.7illustrates the architecture of this proposed model. This model has a first 1-dCNN layer with relu activation function, which is followed by an LSTM layer with adam activation function. Both CNN and LSTM accepts data in the 3d form as (batch, steps, channels) the dataset is converted to a 3d shape accordingly. The output from CNN is in 3d shape so there is no need to reshape the data and the output from CNN can be directly fed to the LSTM model without any processing. The rest of the parameters are the same as used in CNN and LSTM models.

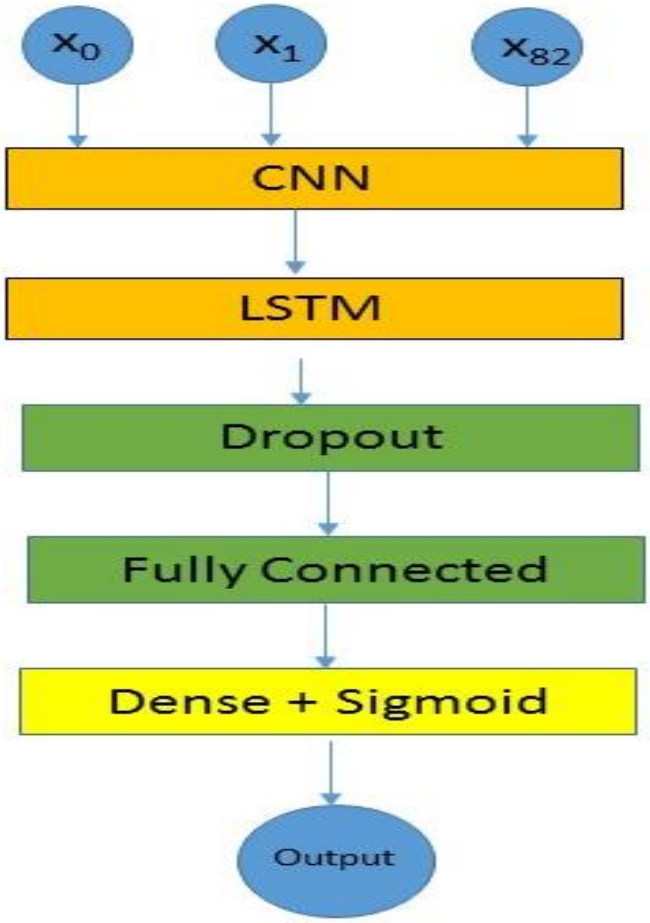

Figure 5.7 CNN + LSTM deep learning model

Apart from the mentioned deep learning models we have evaluated other popular machine learning algorithm for comparison. SVM is a popular algorithm extensively used for supervised classification and regression problems. SVM finds a decision boundary known as hyperplane that can separate n dimensional space into classes and put the data point in correct category. SVM selects the extreme vector points that create the hyperplane, and these cases are known as support vectors. Another classifier we have evaluated is Bayes classifier which is based on Bayes theorem that is used for supervised classification. The advantage of Bayes classifier is that it is a fast machine learning algorithm and make fast prediction. This is based on the probability of object for predicting the output. Random forest is based on the fundamental of combing multiple classifiers to solve a complex problem known as ensemble learning. The subsets of dataset are fed many decision trees to improve the predictive accuracy of that dataset. The prediction from each decision tree is calculated and based on majority votes of predictions the output is predicted.

5.4 Experimentation, Result, and Discussion

5.4.1 Employed environment for conducting the experiment

For the evaluation of the proposed work in this chapter, the CICIDS2017 datasets are exploited, as already explained in detail in Chapter 3. This dataset contains the most recent up to date network data with and without attack, which is very close to the real work network data. This dataset is unbalanced, so we have balanced this dataset by duplicating the method as it seriously affects the training of the deep learning method and hence the testing. Half of the dataset is used for the evaluation of this proposed work because training on full data on the computer used is not feasible; this problem is tackled in the next chapter by employing a high performance computer. For presented work, the Keras on Tensorflow package is employed for deep learning technique on 64-bit Intel Core-i7 CPU, NVIDIA GeForce GTX950M GPU with 16 GB RAM in Windows 10 environment. The machine learning algorithm is executed in MATLAB 2017a on the same computer.

Table 5.1 Parameters of deep learning models

Parameter Name	Value
DL platform	Keras on Tensorflow
Number of Features	83
Optimization Function	ADAM
Loss Function	Binary-Cross entropy
Activation Function in the Output Layer	Sigmoid
Activation Function in the Hidden Layer	Relu
Learning Rate	0.01
Dropout Rate	0.5
No. of epochs	100

5.4.2 Results

The deep learning models are implemented as discussed in the above section. All the models are evaluated on a balanced CICIDS2017 dataset. The dataset is divided into training and testing data with a learning rate of 0.01 and the maximum number of the epoch as 100; after this, the models show no improvement in terms of accuracy. Figure 5.8 shows the comparison of

accuracy value obtained on test data from deep learning models employed along with SVM, Bayes, and Random forest machine learning algorithms. The parameters and attributes of deep learning models are detailed in Table 5.1. Accuracy obtained with 1d-CNN model is 95.14 %, with MLP is 86.34%, with LSTM is 96.24% and with CNN+LSTM is 97.16 %. As it is clear from the figure, the highest accuracy we have obtained is with the CNN+LSTN model while the lowest is with the MLP layer.

Figure 5.8 also illustrates the accuracy obtained by employing machine learning methods on the same dataset. The accuracy obtained with SVM (Support Vector Machine), is 95.5%, with Bayes is 95.19%, and the random forest is 94.64%. LSTM and CNN+LSTM perform better than machine learning algorithms while 1d-CNN is almost the same, but MLP accuracy is much lower by around 9.00% than the machine learning algorithm.

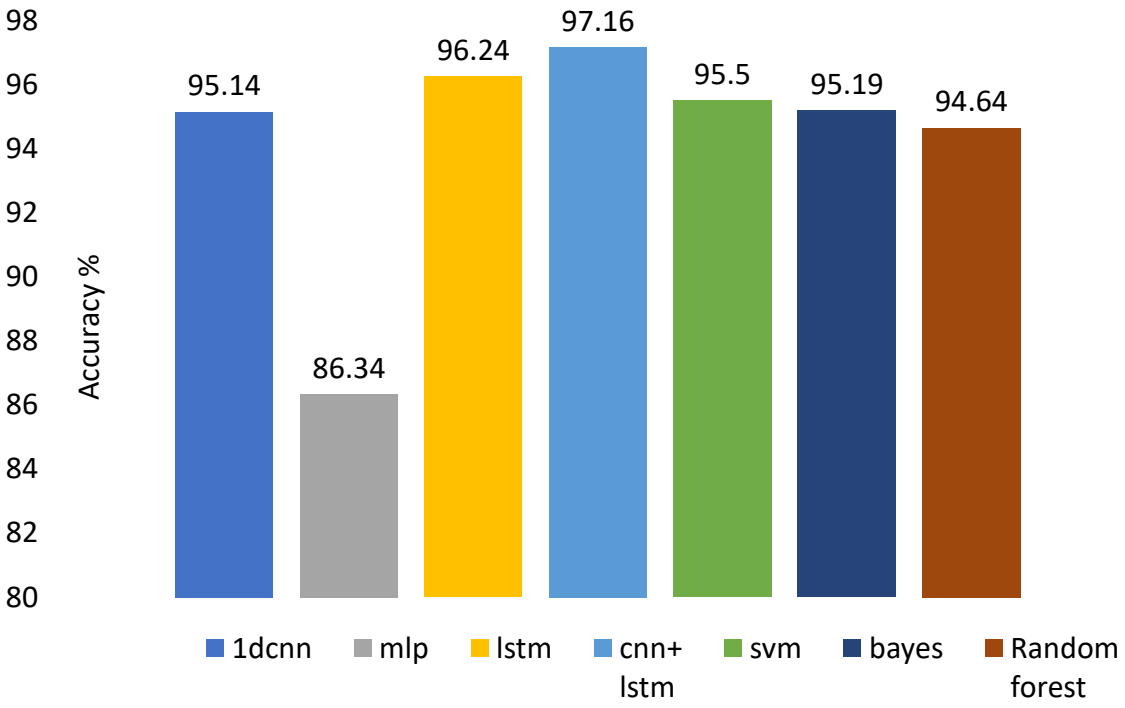

Figure 5.8 Comparison of accuracy of proposed deep models and machine learning methods

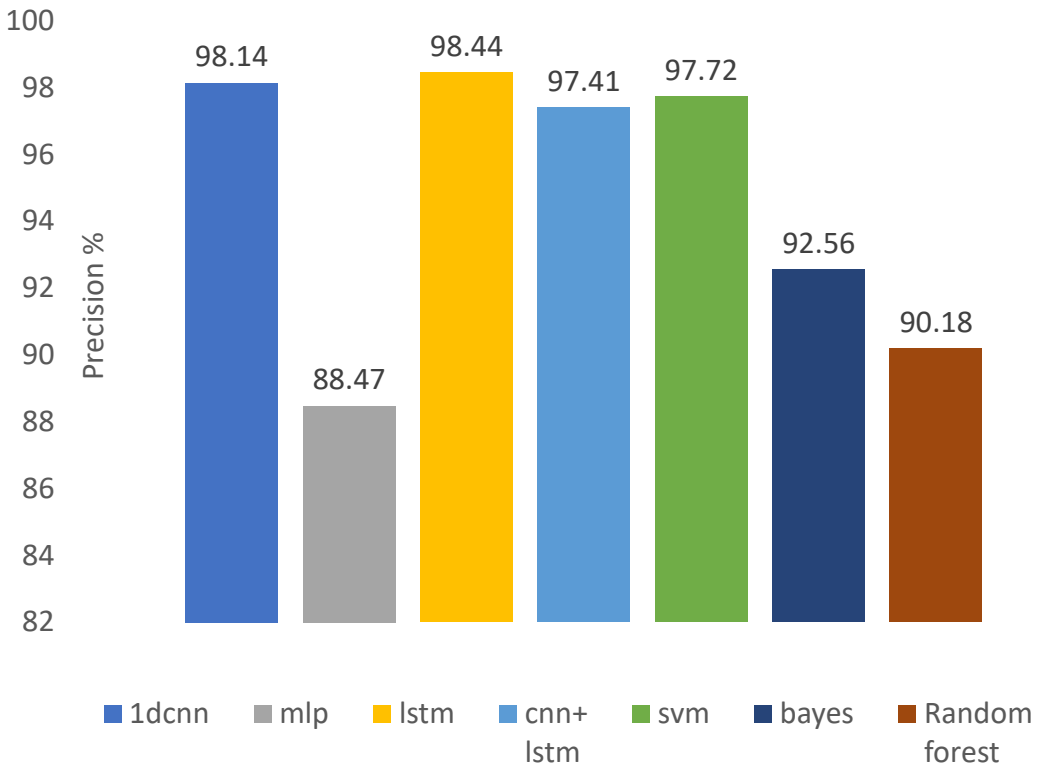

Figure 5.9 Comparison of Precision of proposed deep models and machine learning methods

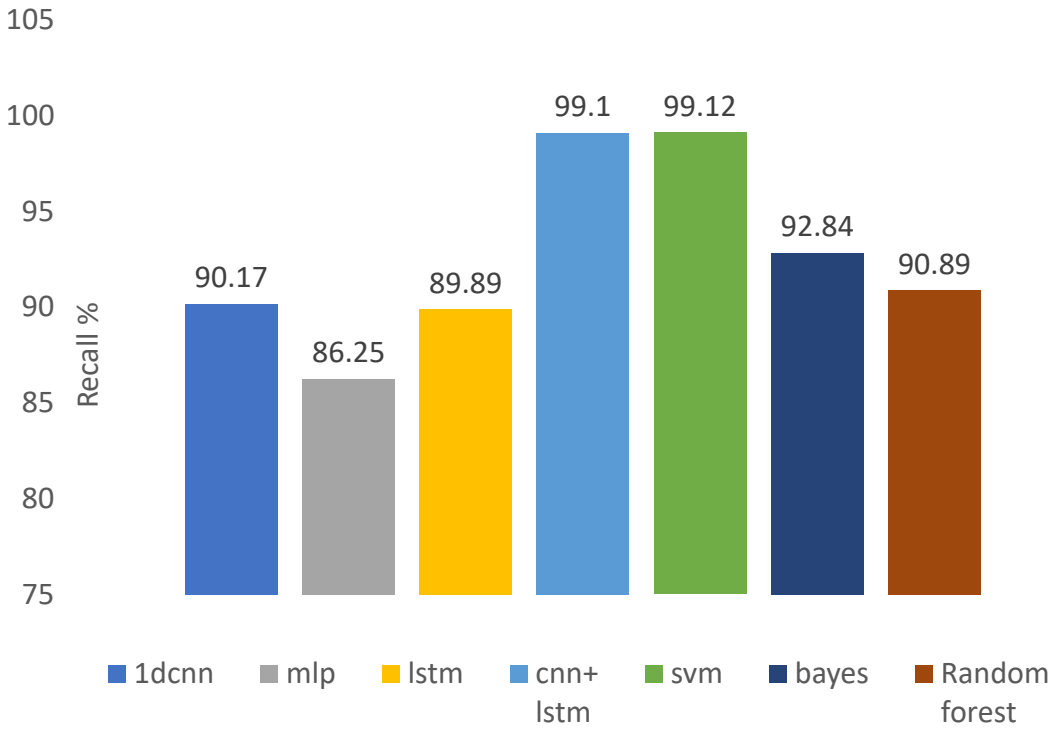

Figure 5.10 Comparison of Recall of proposed deep models and machine learning methods

Figure 5.9 presents the comparison of precision obtained by deep learning models and machine learning methods. The parameter settings are the same as those employed to obtain accuracy value. The precision value with 1d-CNN model is 98.14%, with MLP is 88.47%, with LSTM is 98.44% and with CNN+LSTM model is 97.41%. The highest precision value we have obtained is with the LSTM model, which outperforms the MLP model by around 10.00%. The LSTM model precision is more by 10.00% as compared to that of MLP. It is interesting to see that the precision of the hybrid model is lower by 1.03% as compared with LSMT alone. Figure 5.9 illustrates the comparison of deep learning models with machine learning algorithms. It can be seen from the figure that precision obtained with SVM is 97.72%, with Bayes is 92.56% and with the random forest is 90.18%. The precision obtained with the MLP model is lower than machine learning algorithms, while other models precision is better than machine learning algorithms. The precision of LSMT outperforms the SVM by around 1.20%.

The comparison of recall matric is summarized in Figure 5.10, it can be seen that the value of recall with 1d-CNN is 90.17%, with MLP is 86.25%, with LSTM is 89.89% and with CNN+LSTM is 99.1%. The precision of CNN+LSMT is higher by at least around 9.20% as compared with other models, but accuracy and recall are higher. It is interesting to see that MLP performance is still the lowest in Figure 5.10. This can be concluded based on the results obtained that the CNN+LSTM is performing better than other deep learning models and machine learning algorithms for the detection of DDoS attacks.

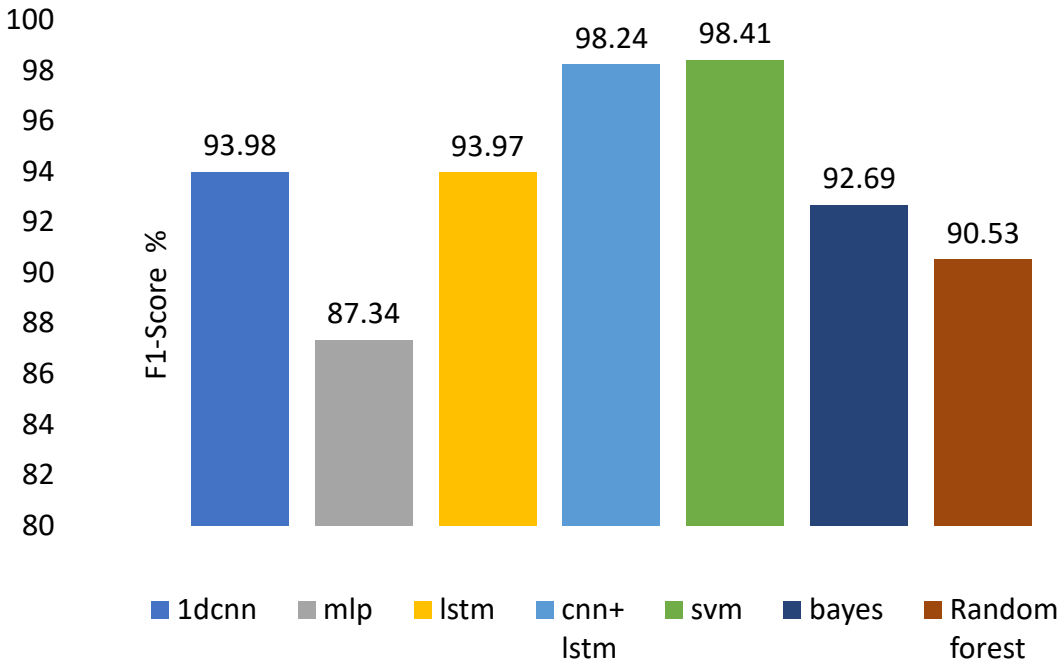

Figure 5.11 Comparison of F1-Score values of proposed models and machine learning models

F1-score of the proposed deep learning models and machine learning algorithms are presented in Figure 5.11. The F1-score value of 1d-CNN achieved is 93.98%, which is good but not the best; it is restricted to this value because the value of recall obtained in low although the precision value was high. The F1-score of MLP is obtained as 87.34 %, which is again the lowest score as compared to all the models. In the case of LSTM, it is 93.97 % which is almost near to the 1d-CNN model. The last deep learning model CNN+LSMT can achieve a hight F1-score of 98.41 %, which proves that this model has an extremely low false negative and false positive rates in comparison with other models and machine learning models. The machine learning models SVM, Bayes and Random forest have obtained F1-Score of 98.41 % and 92.69 % and 90.53 % respectively. It is interesting to note that the F1- Score of SVM machine learning algorithm is nearly equal to that of CNN+LSMT deep learning model, this could be the reason that SVM has been extensity used as a classifier in various applications before the deep learning technique begins to work into practice in research.

5.5 Chapter Summary

In this chapter, the four different deep learning models are proposed and compared with machine learning algorithms for the detection of DDoS attacks. It is found that the hybrid CNN+LSTM model performs better than the rest of the deep learning models and machine learning algorithms with an accuracy of 97.16 % and a high F1-Score of 98.24 %. Another interesting result obtained is in the case of the MLP deep learning model that performed as the worst deep learning model on the dataset employed; this is because of the problem of overfitting in MLP. It is found that except for MLP, the accuracy obtained by the other three deep learning methods is more than 95.00 % and is performing better than the machine learning algorithms SV|M, Bayes, and Random forest. The deep learning-based method does not require feature selection to be performed before the classification learning and testing, but with a large number of attributes in the datasets, the training time could be a challenging issue. In the next Chapter 6, the discovered DL model is discovered in this chapter is combined with the method proposed in Chapter 4 to form a complete IDS.

Chapter 6. Intrusion Detection System against DDoS Attack in IoT Networks

In Chapter 4, an efficient feature selection method based on six objectives and in Chapter 5, deep learning methods for the detection of DDoS attacks in IoT networks have been proposed. In this chapter, an IDS for the detection of DDoS attacks in IoT networks is proposed. The proposed IDS is based on the hybridization of the proposed methods in Chapter 4 and Chapter 5. High-performance computers have been used to evaluate the performance of the proposed method. The results obtained after the experimentation are quite satisfactory and impressive and are also compared with other proposed methods and machine learning algorithms.

6.1 Introduction

In this paper, a novel IDS based on the hybridization of the deep learning method and multi-objective optimization method for the detection of DDoS attacks in IoT networks is proposed. In chapter 4, the multi-objective-based feature selection method based on six objectives was proposed and evaluated, which provided quite satisfactory results with very high accuracy values. The ELM classifier was employed as a binary classifier, which is a machine learning algorithm. However, with the more and more sophisticated cyber-attacks being developed, the machine learning methods such as SVM, Naïve Bayes are not enough for the detection of those attacks on the networks. For example, the Imperva's client was attacked with DDoS attacks in 2019 [40]. This was not the first time the Imperva security service provider became the victim of DDoS attacks. In the year 2016 also, the Company's client came under the attack of DDoS attacks [44]. Although after the company was first attached in 2016, they developed new security mechanisms and tools for the detection of DDoS attacks because the nature and sensitivity of DDoS attacks have evolved with time; the attacks were not detected for 13 days which is huge in terms of the duration of under attack. A few more similar examples are discussed in Chapter 2 in literature review section.

There is a need for and advanced IDS for the detection of sophisticated massive DDoS in IoT networks. With the advancement in GPUs and CPUs it provides the opportunity to use the advantage of DL technique that automatically learn from the data. The best deep learning model feasible for cybersecurity is revealed in Chapter 5, which provided high performance in terms of accuracy, precision, recall and F1_Score. This is chapter the novel IDS by combining the feature selection algorithm for reducing the dimensionality of the data that is proposed in

Chapter 4, and deep learning classification model for the detection of the attacks is proposed [192].

6.2 Proposed IDS Methodology

In this section, the proposed methodology for the development of the IDS is elaborated. Figure 6.1 presents the system diagram of our proposed method; in the subsection, the algorithms employed in this method are described. The CICIDS2017 datasets are employed that are elaborated in Chapter 3, for the evaluation of the proposed IDS. The network data with and without DDoS attacks are collected, pre-processed, and normalized in range {0,1}. This normalized data is fed to the NSGA-II-aJG algorithm module of the proposed system, for the feature selection to reduce the dimensionality of the data, which have been proved in the later results section, has improved the performance of the proposed method dramatically. The six most essential objective functions, as discussed in Chapter 4, are used for the implementation of the NSGA-II-aJG algorithm, which is discussed in detail in the subsection. The reduced data as output from the previous step, become the input to the deep learning model module of the proposed system. The deep learning model consists of a CNN layer with a Relu activation function followed by a max-pooling layer. Another LSTM layer with Relu activation is followed by dropout layer is included. The proposed model classifies the attack as normal or abnormal using a sigmoid function with binary cross-entropy.

6.3 Experimentation Environment

The proposed work has been evaluated on GPU enabled High-Performance Computer (HPC) facility provided at Newcastle University. The training and validation are carried out on Keras on Tensorflow in the background on GPU NVIDIA Tesla V100 GPUs, having 16 GB VRAM with 256 GB on 10 number of nodes in HPC. Deep Learning accepts data in the 3D form as (batch, steps, channels), so the shape of the data, which is in 2D form array is changed to 3D shape accordingly. The dataset has 83 attributes in total, including the label attribute, after all the steps of data pre-processing performed, so the dataset is converted using reshape function as {data.shape(0), data.shape(1), 1}. Relu activation function has been employed for both CNN and LSTM layers, and Adam optimization function with 128 hidden neurons with 0.2 dropout probability, is employed with learning rate 0.01 on 256 batch size for 100 epochs. The output from CNN is fed to the LSTM layer; no reshaping of data is required at this stage.

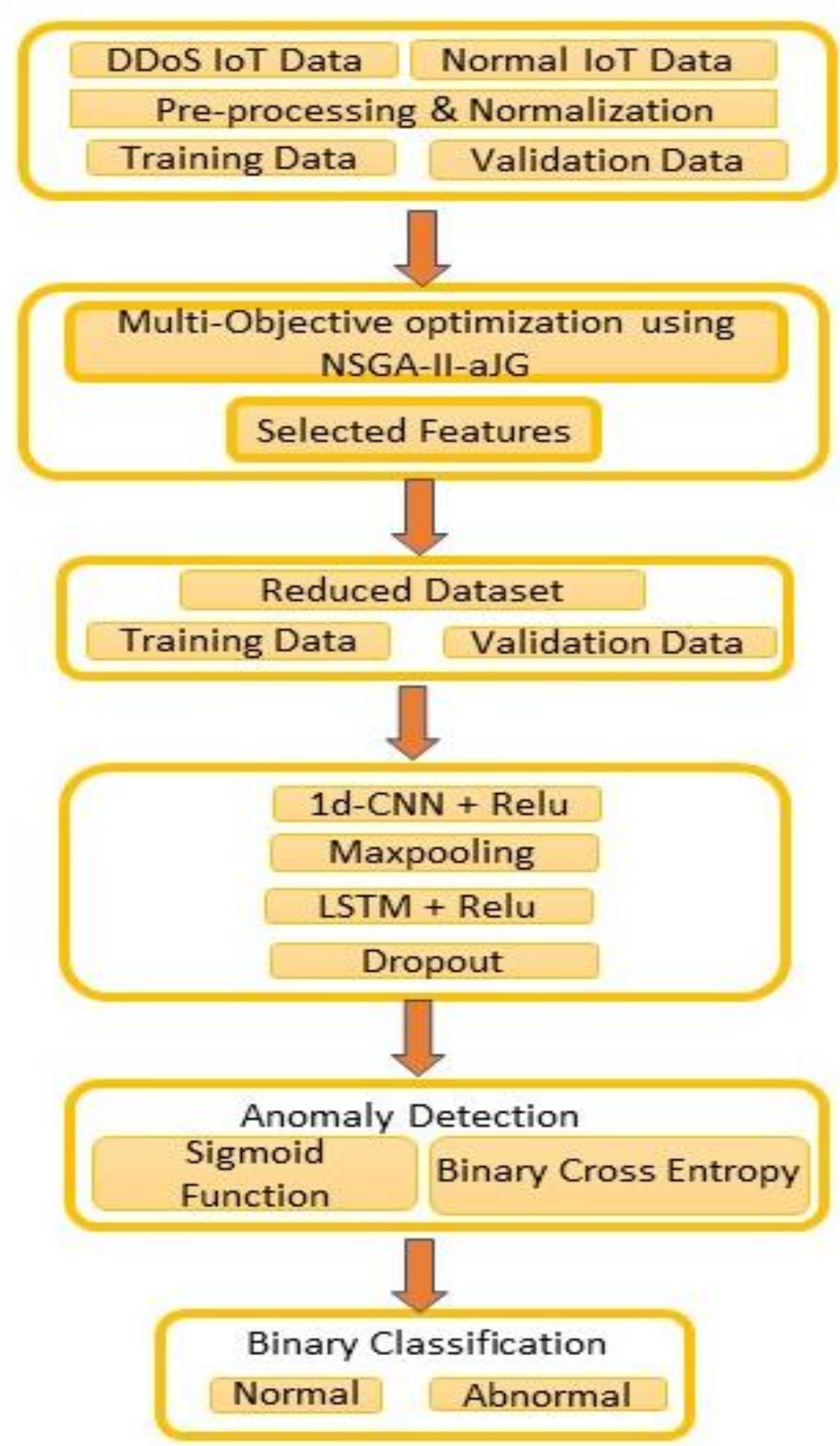

Figure 6.1 Flowgraph of proposed IDS for DDoS attack

The dataset is split between training and test sample as 90:10, which means 90.0% of the data is used for training, and 10% is used for validation. The proposed method is also compared with machine learning algorithms. It is to be noticed during the development of the IDS varying deep learning parameters were exploited, and the IDS was evaluated accordingly. The parameters mentioned are best-suited values found after many experiments and are the final parameters employed in this chapter. The results on the different parameters are irrelevant in the context of the aim of this chapter and so are not included.

6.4 Results and Discussion

This section presents the results obtained by the experimentation on the proposed method. The initial procedure in the proposed method is to reduce the dimensionality of the data by doing feature selection using NSGA-ii-aJG. Many solutions were obtained in the form of Pareto-front; the minimum number of features obtained is six, nine, twelve, and fifteen. Figure 6.2 presents the comparison of different numbers of features in terms of accuracy, precision, recall, and F1-Score. It is interesting to note that the set of fifteen number of features obtained has achieved the best value of accuracy as 98.78 %, precision as 99.03 %, recall as 99.35, and F1-Score as 98.48% on the proposed algorithm. The feature set with 12 number of features obtained F1-Score of 98.23 %, recall of 98.1 % , precision of 97.35 % and accuracy of 96.89 %. The feature set with 9 number of features has achieved F1-Score of 92.12 %, recall value of 92.89 %, precision of 92.99 % and accuracy of 92.24 %. The lowest classification values obtained is with six numbers of features as accuracy is 91.15%, precision is 92.23%, recall is 92.30%, and F1-Score as 91.1%. So, fifteen features set have been used as input to the next step in the proposed method.

Figure 6.3 presents the accuracy of our proposed method and comparison with MLP, which is another most frequently used deep learning modal; the figure also compares the obtained result with other machine learning algorithms, SVM, Bayes, Random Forest, which are commonly used for cyber-attack detection. It is clear from the graph our proposed method outperforms other methods, achieving 99.03 % accuracy, whereas MLP has achieved only 88.74% accuracy, which is lower than the machine learning algorithms. SVM has obtained 94.50 %, Bayes has achieved 94.19%, and Random Forest has achieved 93.64 %. So it can be concluded that MLP is not very efficient in learning and detecting an attack on our employed dataset, whereas SVM has worked second best.

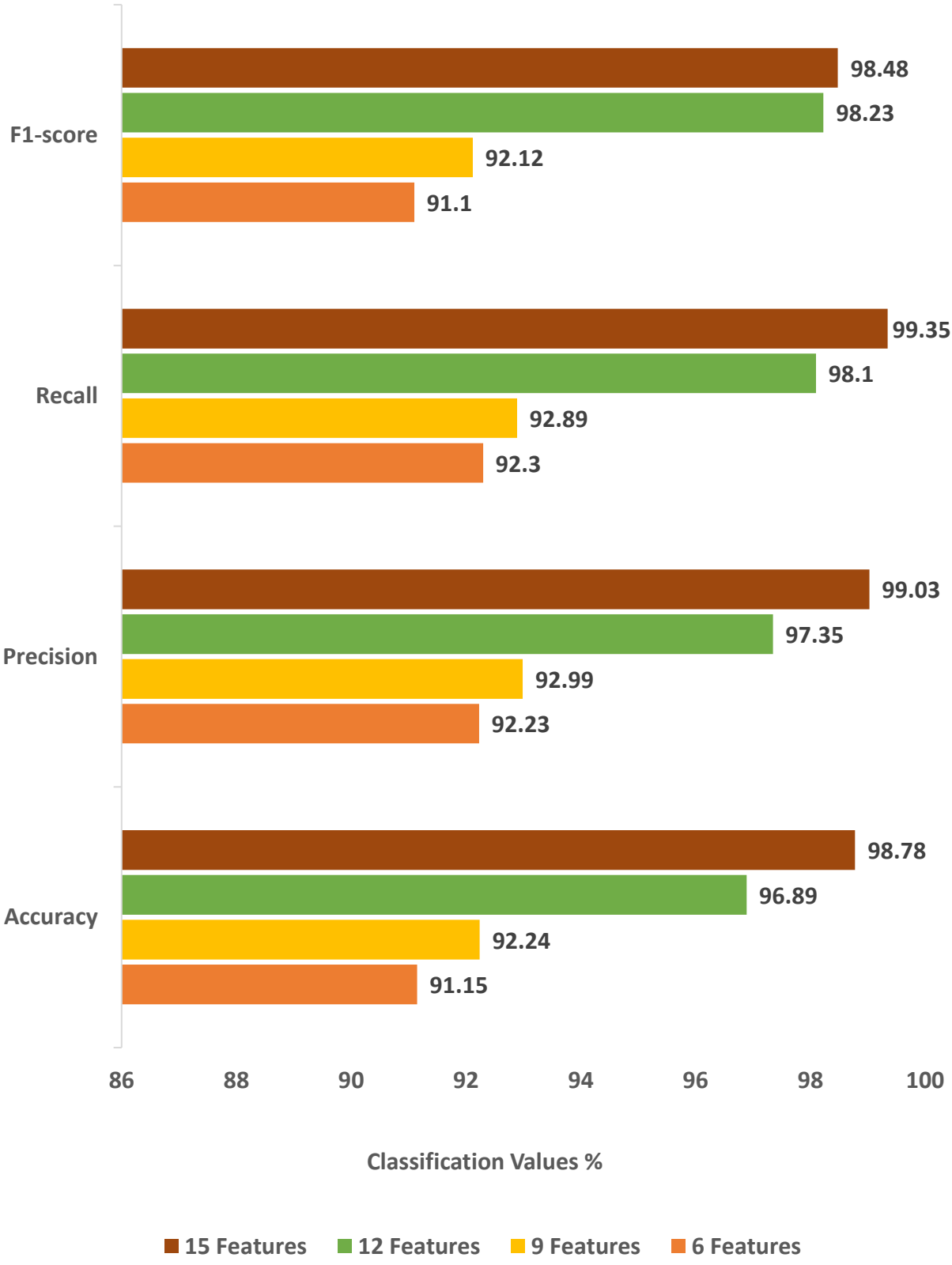

Figure 6.2 Comparison of the proposed method with machine learning methods

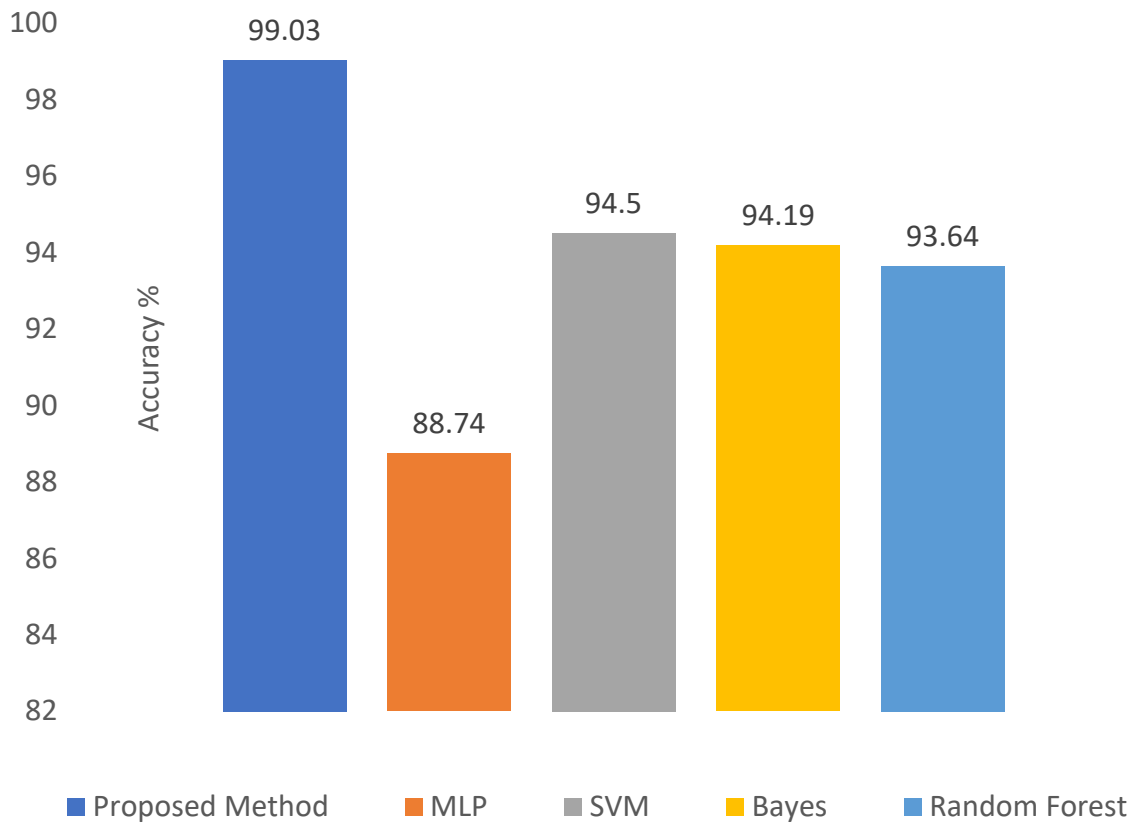

Figure 6.3 Comparison of Accuracy values of Proposed IDS with other methods

The comparison of the precision value of the proposed method, MLP, and other machine learning algorithms is presented in Figure 6.4. The precision value of the proposed work obtained is 99.26%, while MLP is the lowest at 88.57 %. Bayes has obtained 91.56 % precision; Random forest has achieved precision of 89.99 % and SVM algorithm obtained a precision value of 96.72% which is higher than the other two algorithms. It is exciting to observe that the precision value of the proposed DDoS attack detection method outperforms the rest of the methods.

Figure 6.5 summarizes the classification recall value of the proposed method, MLP, SVM, Bayes, and Random Forest Method. The proposed method has obtained the highest recall value as 99.35%, whereas the MLP has obtained the lowest recall value of 86.31% which is lowest than the machine learning algorithms. Bayes has obtained 91.85 % recall value, Random Forest has obtained 89.68 % and SVM has performed best in terms of recall value among the machine learning algorithms achieving 98.10 % recall value.

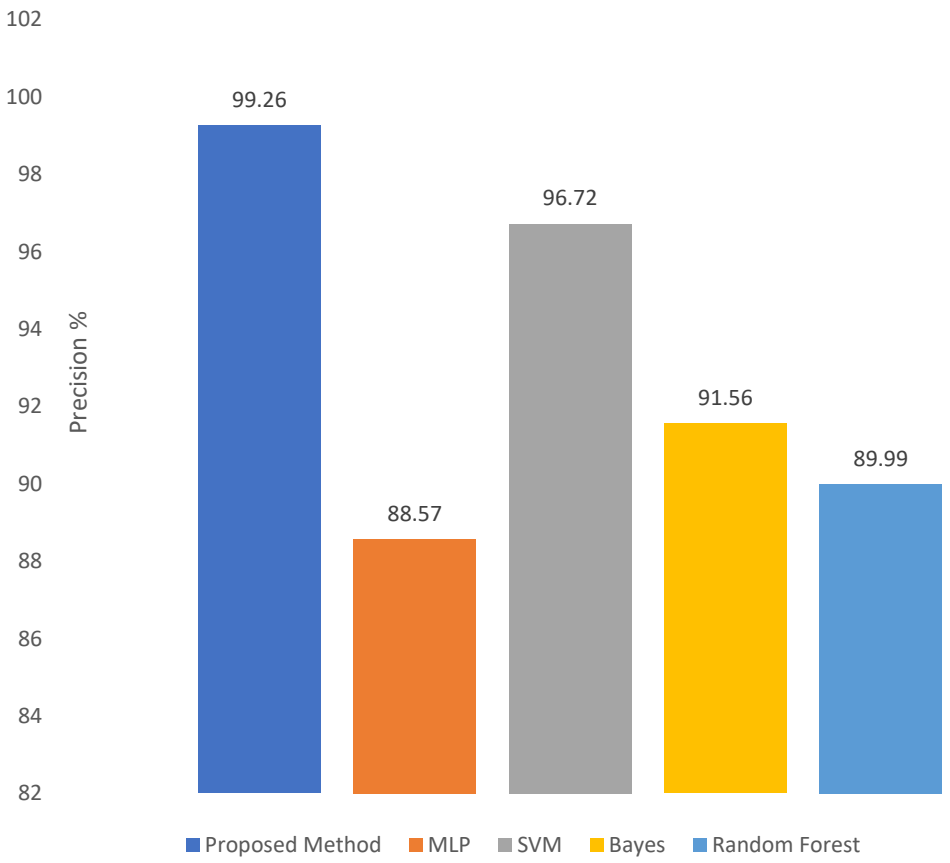

Figure 6.4 Comparison of Precision values of Proposed IDS with other methods

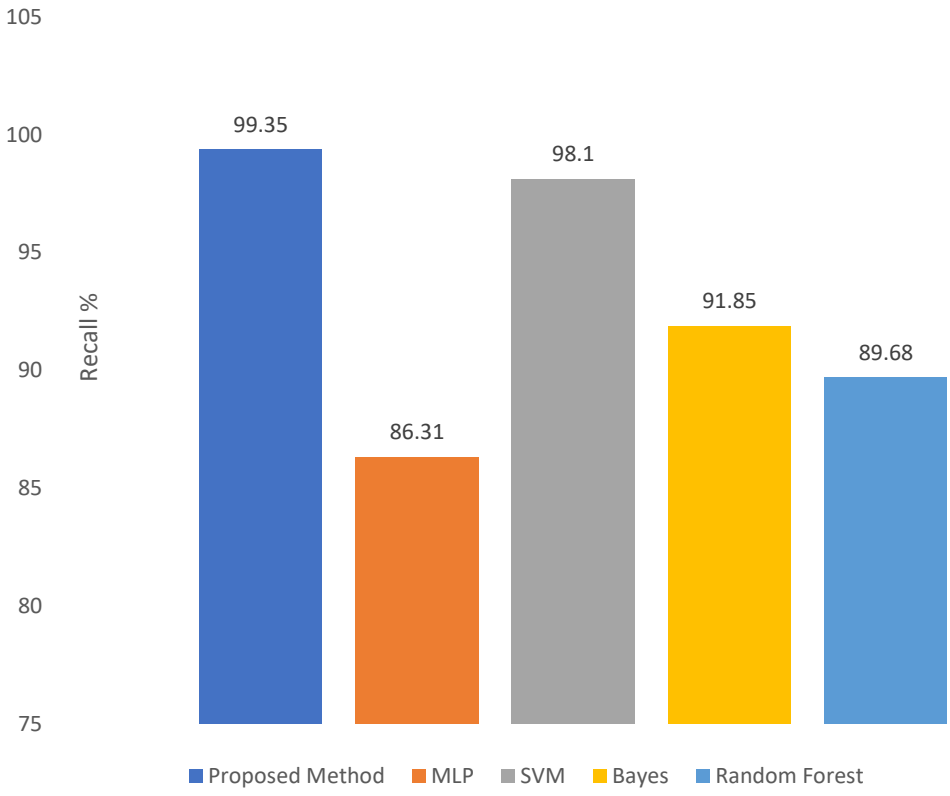

Figure 6.5 Comparison of Recall values of Proposed IDS with other methods

F1-Score is a vital evaluation metric in determining the overall performance of the method applied if accuracies are comparable. F1-Score values obtained from our experimentation are illustrated in Figure 6.6. It is interesting to note that the F1-Score of SVM is highest as compared to other machine learning methods, although accuracy value is nearly the same. SVM has achieved F1-score of 97.4 %, Bayes has obtained 91.7 % and Random Forest has achieved 89.83 % value of F1-score. F1-Score of MLP is lowest is 87.43%, whereas the proposed method has achieved a 99.36% F1-Score value.

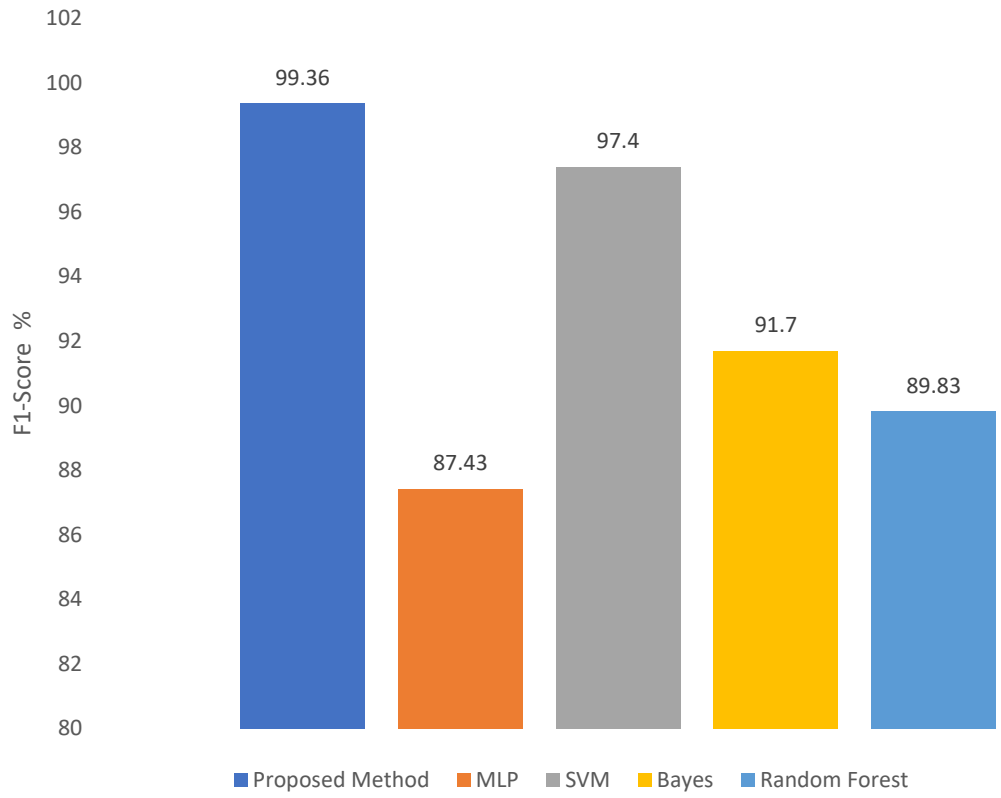

Figure 6.6 Comparison of F1-Score values of Proposed IDS with other methods

FPR is another very important performance evaluation parameter. Figure 6.7 presents the comparison of FPR value of proposed method with other methods. It can be noted from the figure that the proposed method has obtained the lowest FPR value of 0.71 %, SVM has obtained 3.5 % FPR value, Bayes has obtained 8.5 %, Random forest has obtained 9.89 % and MLP has obtained the highest value of 11.6 %. Another critical performance is TPR which is presented in Figure 6.8. The purported IDS has obtained 99.3 % TPR value which is quite impressive. The obtained value of TPR is MLP is 88.7 %, SVM has obtained 96.8 %, Bayes has obtained 91.8% and Random forest has achieved 90.1 %. In this case also SVM has achieved second highest value and MLP achieved lowest value.

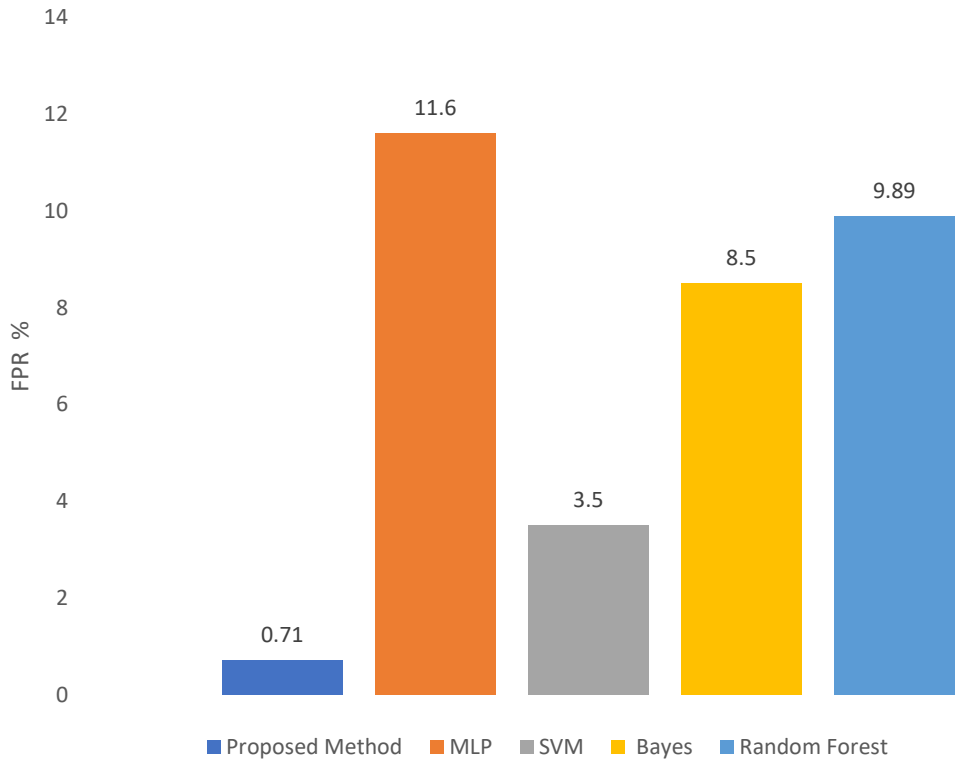

Figure 6.7 Comparison of FPR value of proposed method with other methods

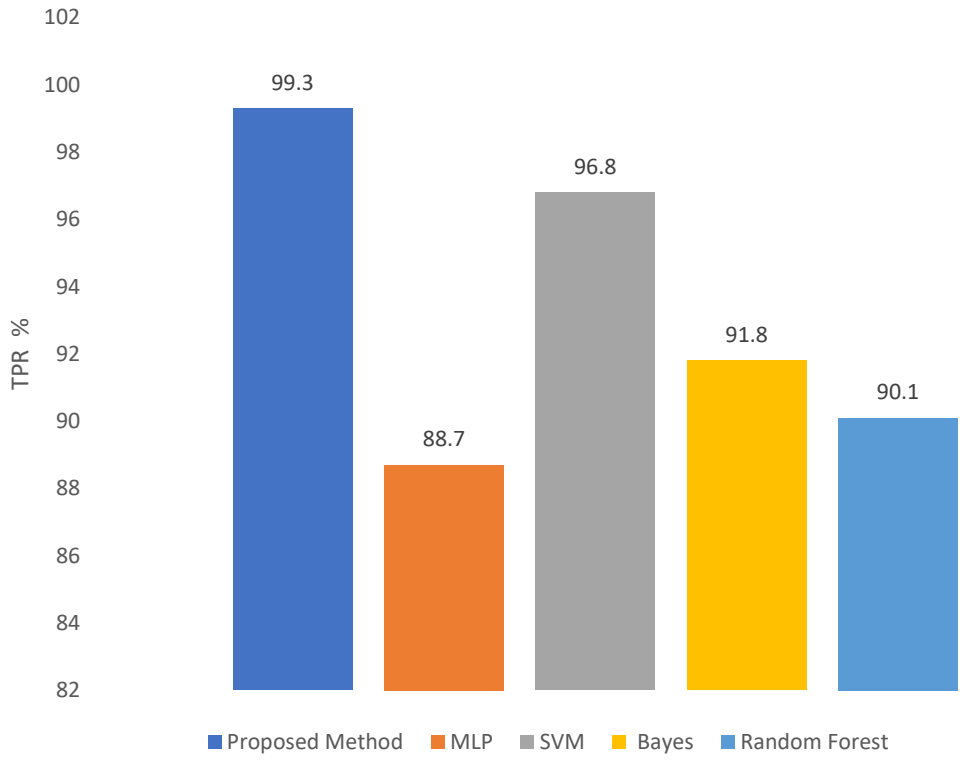

Figure 6.8 Comparison of TPR value of proposed method with other methods

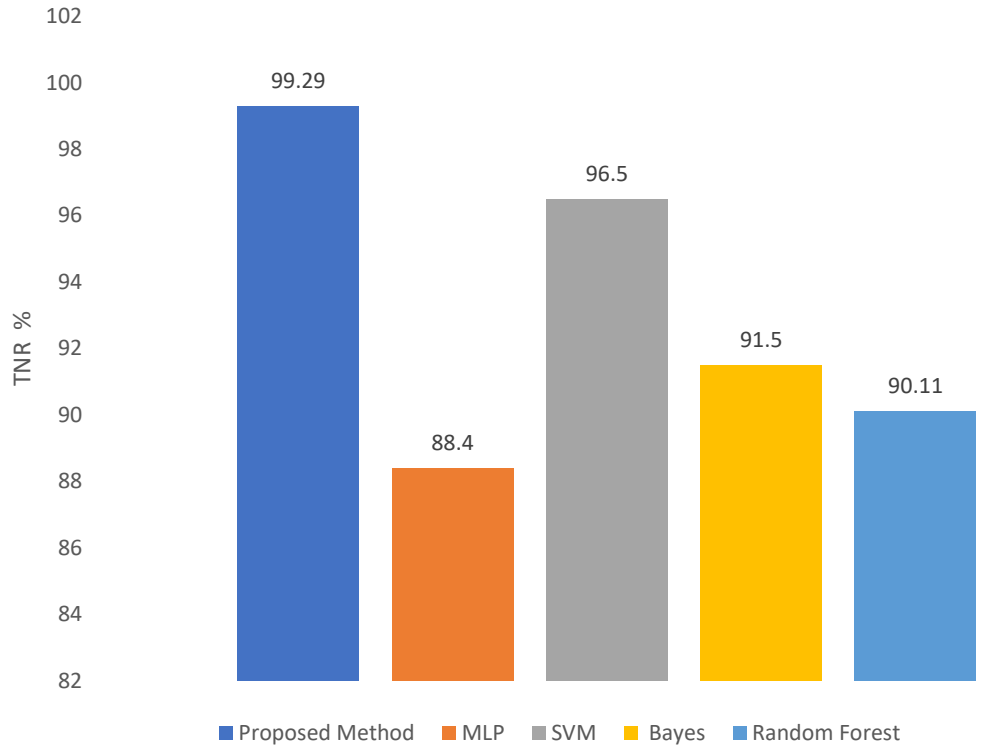

Figure 6.9 Comparison of TNR value of proposed method with other method

TNR is also a critical performance parameter which is expected to be of very high value in term of percentage. Figure 6.9 illustrate the value of TNR in percentage of proposed method and other methods. The TNR value of the proposed method is 99.29 %, which is pretty good, whereas the MLP has obtained 88.4 %, SVM has achieved 96.5 %, Bayes obtained 91.5 % and Random forest has obtained 90.11 % TNR. Figure 6.10 presents the FNR value which is also critical measure of evaluation an IDS and is expected to be lowest in term of percentage. The FNR value of MLP is 11.3 %, SVM has achieved 3.2 %, Bayes has obtained 8.2 %, Random forest has achieved 9.9 %, whereas the proposed IDS has achieved very low value of 0.7 % which is proves the performance of the proposed IDS method.

Another fascinating result in terms of training time of the proposed method is presented in Figure 6.11. The proposed method is compared with a deep learning method consisting of CNN and LSTM on same CICIDS2017 datasets, without any feature selection. The training time has reduced drastically to 15313.10 seconds, which as 11 times lower in the case of deep learning method. This proves that the proposed method is very efficient in real-time IoT cyber-attack detection as the training time has reduced to 5- folds.

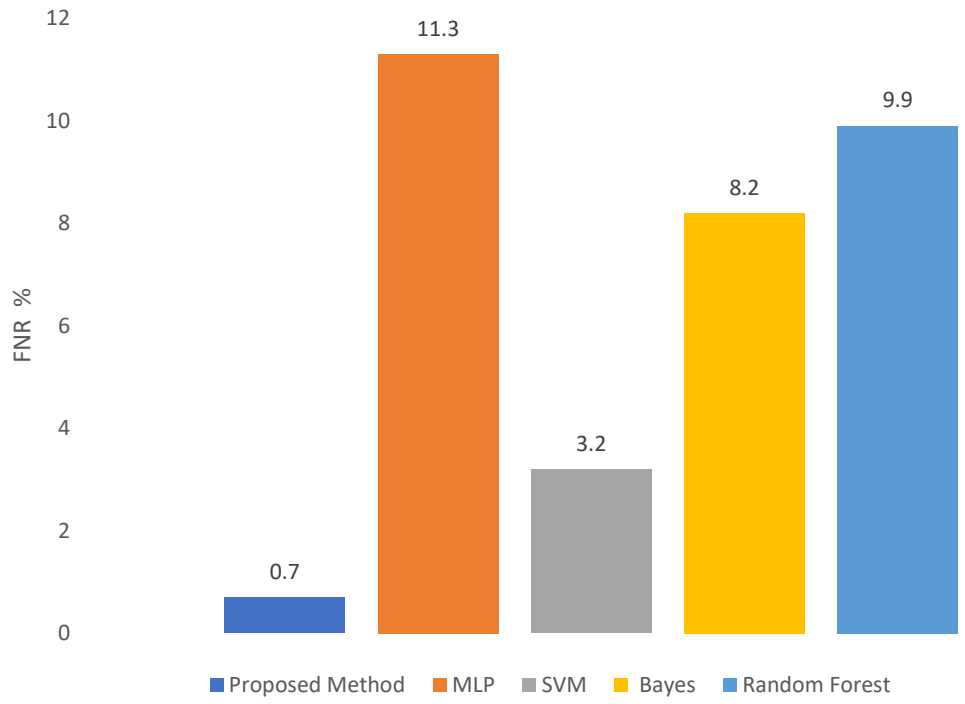

Figure 6.10 Comparison of FNR value of proposed method with other methods

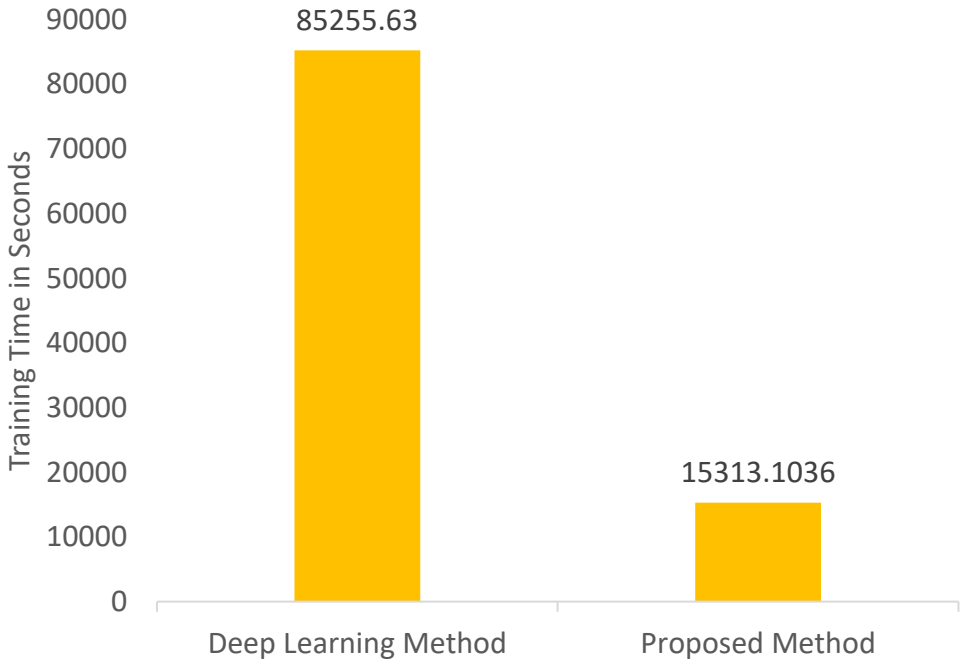

Figure 6.11 Comparison of Training time of proposed IDS with Deep Learning Method

The proposed model is furture evaluated by modifying the number of CNN and LSTM layers in the proposed IDS. We have experimented 6 different model topology with varying number of layers in deep learning module of proposed IDS with compare with the proposed IDS.

Topology 1: CNN 2 layer followed by 1 LSTM

Topology 2: CNN 3 Layer followed by 1 LSTM

Topology 3: CNN1 layer followed by 2 LSTM

Topology 4: CNN 1 layer followed by 3 LSTM

Topology 5: CNN 2 layer followed by 2 LSTM

Topology 6: CNN 3 layer followed by 3 LSTM

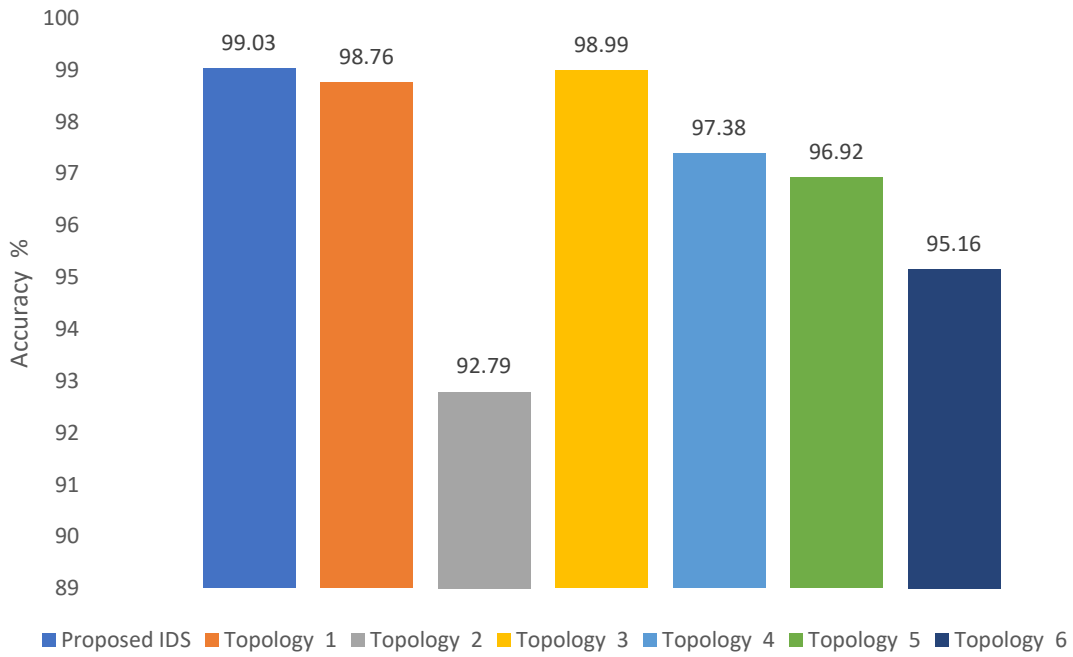

Figure 6.12 Comparison of accuracy of proposed IDS with other Topologies

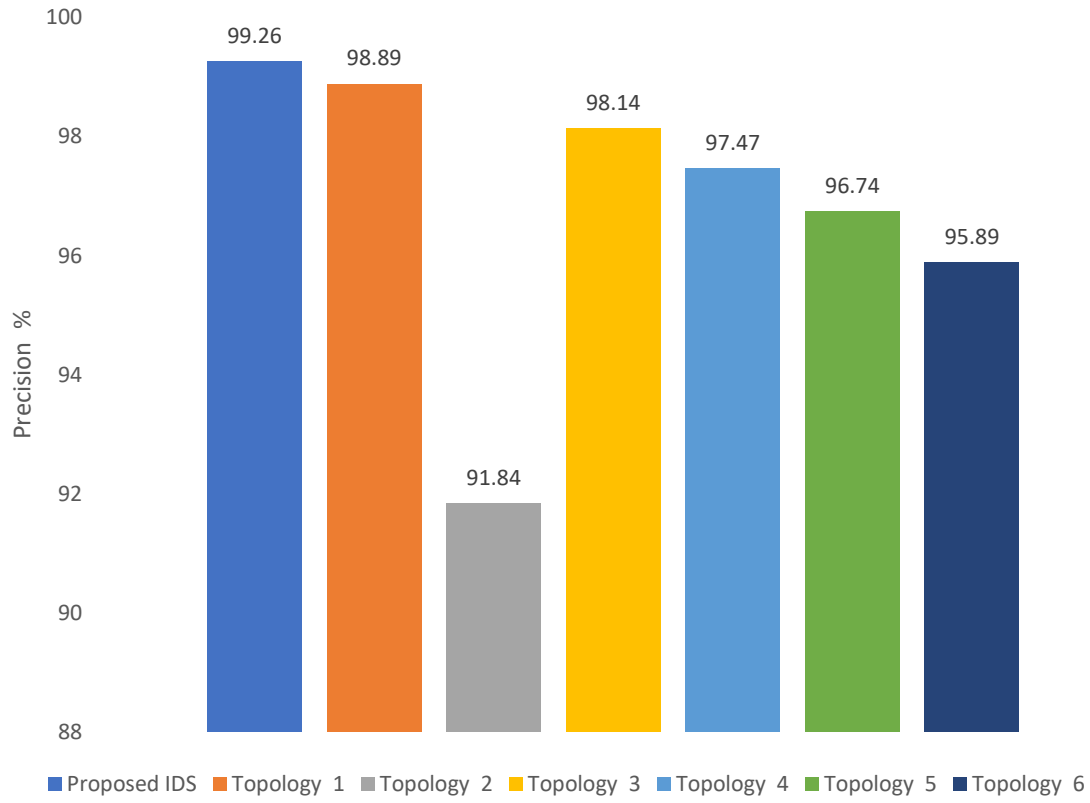

Figure 6.13 Comparison of precision of proposed IDS with other Topologie

Figure 6.12 presents the comparison of proposed IDS with other above-mentioned topologies of deep learning module in proposed IDS. Topology 1 has obtained accuracy of 98.76 % and topology 2 has achieved the lowest accuracy of 92.79 %.The obtained accuracy value of topology 3 is 98.99 %, of topology 4 is 97.38 %, of topology 5 is 96.92 % and of topology 6 is 95.16 %. Our proposed method has obtained highest accuracy value of 99.03 %. Figure 6.13 illustrate the precision values of proposed IDS and other topologies. Topology 1 has obtained 98.89 %, topology 3 has obtained 98.14 %, topology 4 obtained 97.4 % and topology 5 has obtained 96.74 %. Topology 2 and 6 has obtained lowest precision values, of 91.84 % and 95.89 % respectively. The proposed IDS has obtained highest precision value of 99.26 %. The comparison of recall values of recall of proposed IDS and other topologies is presented in Figure 6.14. Topology 2 has obtained lowest recall value of 92.24 %, second lowest recall is obtained by topology 6. Topology1 has obtained 98.21 %, topology 3 has obtained 98.26 %, topology 4 has obtained 97.1 % and topology 5 has obtained 96.98 %. Our proposed method has obtained highest recall value of 99.35 %.

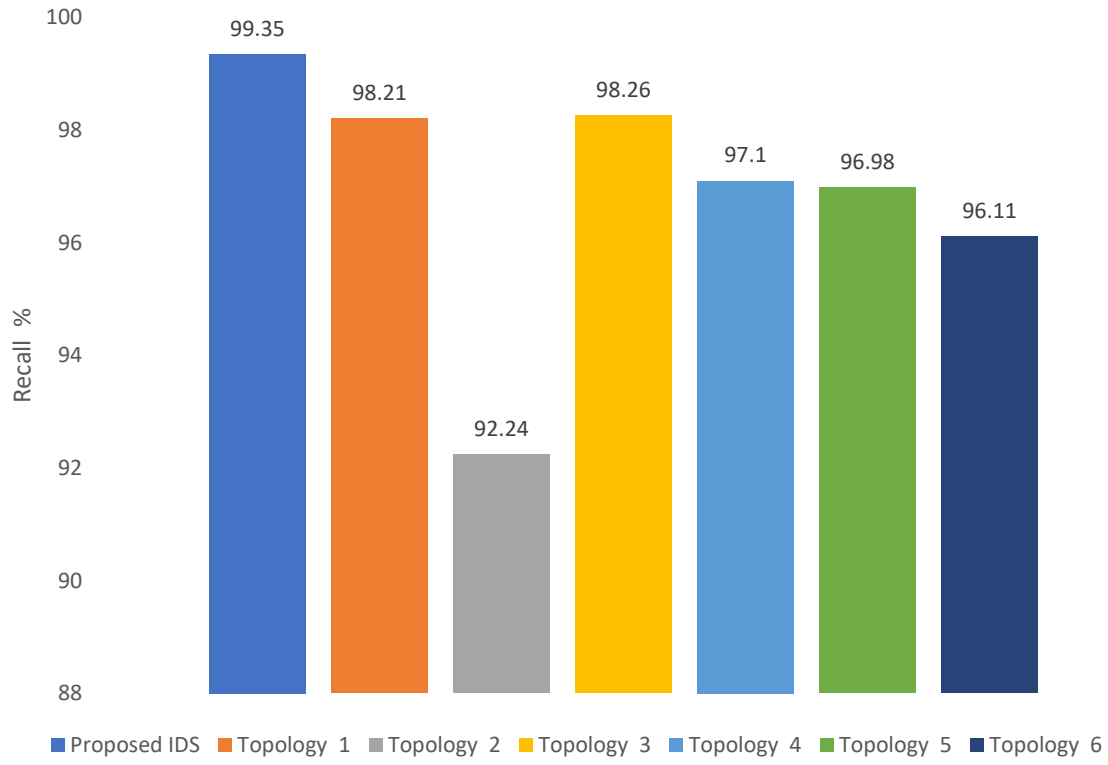

Figure 6.14 Comparison of recall of proposed IDS with other Topologies

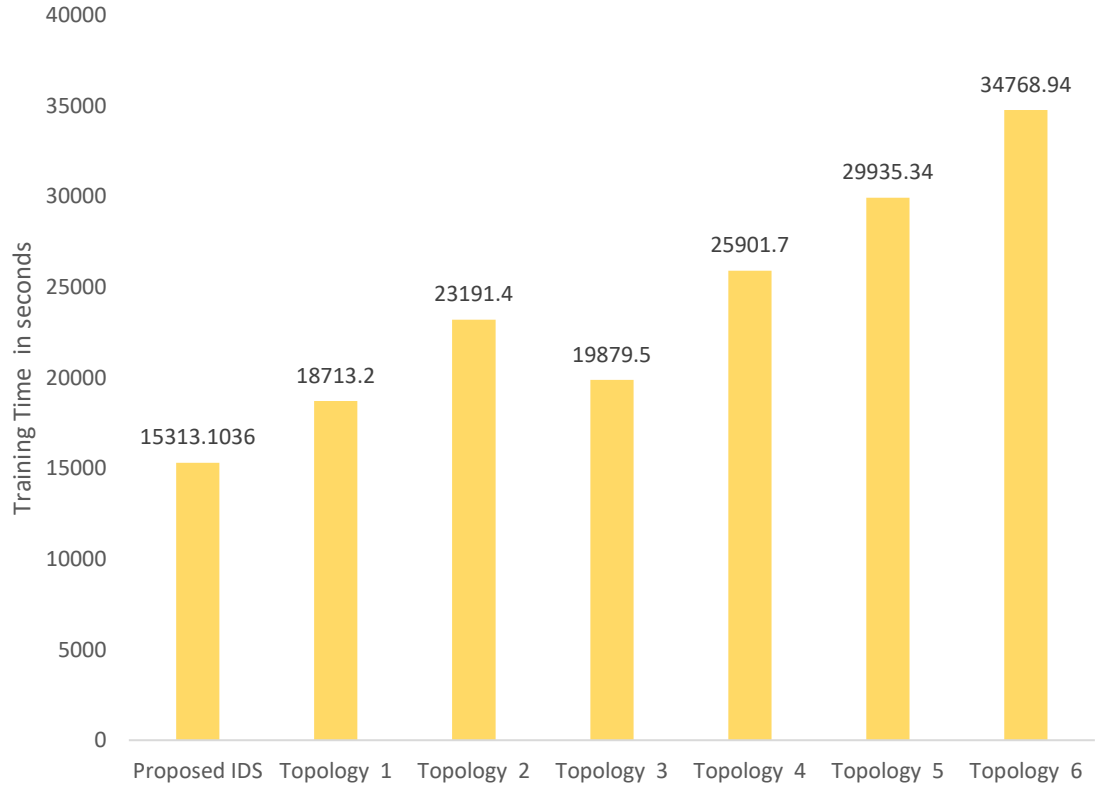

Figure 6.15 Comparison of training time of proposed IDS with other Topologies

Figure 6.15 illustrates the comparison of training time consumed by proposed IDS and other topologies. The training time taken by topology 6 has come to be highest as 34768.94 seconds, whereas proposed IDS consumed 15313.10 seconds for training on same dataset. Topology 6 has taken more than twice the time taken by proposed IDS. Training time in case of topology is 18713.2 seconds, topology 2 is 23191.4 seconds, topology 3 is 19879.5 seconds, and topology 4 is 25901.7 seconds and in topology 5 is 29935.34 seconds. Training time taken by an IDS in real time is critical especially in case of IoT networks, where the new data is generated at very high-level on daily basis. It can be seen from the data obtained from the experimentation that adding more layer does not help in achieving better performance of the IDS instead the performance has decreased by adding layers.

Generally, it is thought that adding layers will help in achieving higher performance but practically it has not come out true in out experimentation. One of the reasons for this is the problem of 'overfitting' of data. Adding layers in deep learning model will extract more features, but after a limit instead of extracting features, the model is overfitting with data that can lead to false positives. To understand it better suppose a model is trained for detecting dog and its detection performance is satisfactory. Suppose we add more layers to it, the model might learn the belt of the dog it is wearing has part of the dog. So, the trained model with more layers might not detect a dog without belt as dog instead it might detect another animal with belt as dog. We have done feature selection before applying the deep learning technique in our IDS, adding layers is causing overfitting, that is the reason adding layers is decreasing its performance. Further the proposed IDS is evaluated on partial labelled data. Figure 6.16 presents the performance of proposed IDS on partial labelled data. The CICIDS2017 dataset is modified by removeing 30 % of the label value from the data. The accuracy obtained 73.84 %, the recall valye is 71.32 %, the precision is 70.64 % and F1-score is 70.97%. It can be can be seen from the results that the proposed IDS is not very efficient in detection of attacked if the training is done on partial lablled data.

Table 6.1 presents a comparison of the proposed method with other state-of-the-art methods for DDoS attack detection. It can be concluded that the proposed method is good enough to outperform the proposed work for the detection of a DDoS attack. In [82], authors have proposed a method based on deep neural network algorithms for the detection of unforeseen and unpredictable cyberattacks. Authors have evaluated their proposed method on KDDCup99, CICIDS 2017, and NSL-KDD datasets, the experiment has run for 1000 epochs with varying

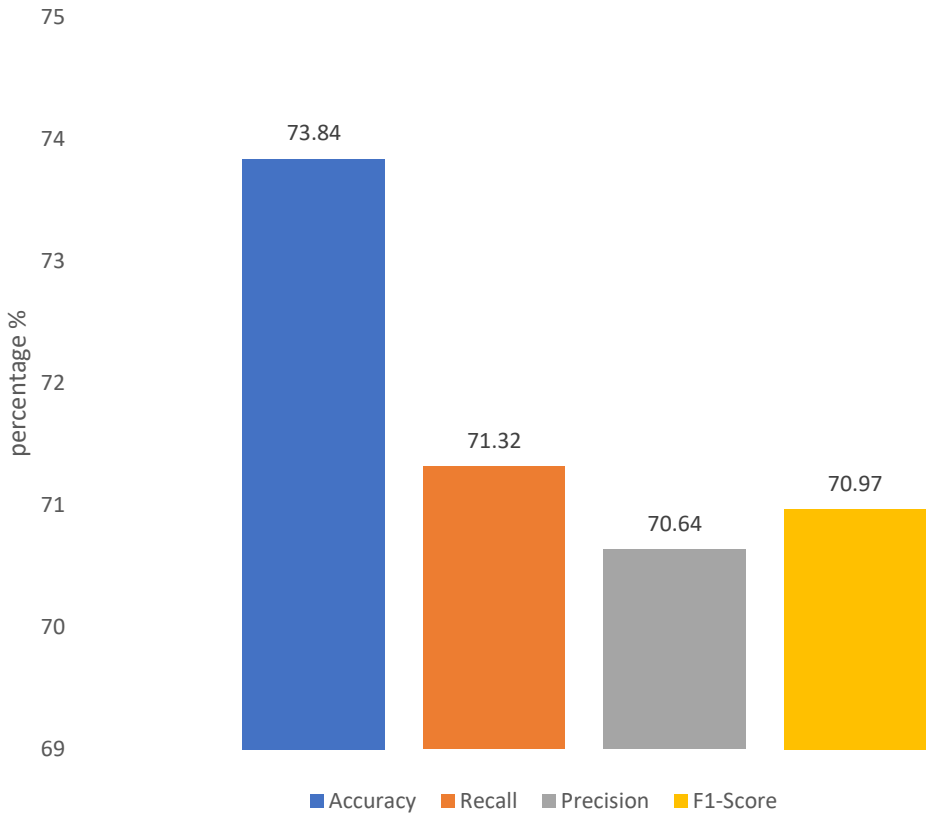

Figure 6.16 Performance of proposed IDS on partial labelled data

Table 6.1 Comparison of proposed IDS with other state-of-the-art methods

Method	Dataset	Attack	Accuracy
Deep Learning [82]	CICIDS 2017	DDoS	85.55%
	NSL-KDD	DoS	91.5%
	KDDCup99	DoS	95.55%
Entropy and PSO-BP neural network[193]	SDN Collected Data	DDoS	97.27%
Authoencoder + MKL[194]	UNB ISCX 2012	DDoS	97%
CS_DDoS [195]	SDN Collected Data	DDoS	97%
Proposed Work	CICIDS2017	DDoS	99.03%

learning rates range. The best accuracy achieved by this method is 95.55% on the KDDCup99 dataset for the detection of the DoS attack. A Software-Defined Networking based method for the detection of DDoS attack is proposed in [193]. This method exploited the combination of information entropy and backpropagation neural network. The evaluation of the proposed

method is conducted on a Java-based Floodlight and Mininet simulation software and has achieved an accuracy of 97.27%. Another algorithm for the detection of DDoS attacks based on multilevel autoencoder and multiple kernel learning (MKL) is proposed in [194]. The authors have compared their proposed method with machine learning algorithms. This method has achieved a 97% accuracy on the UNBISCX2012 dataset. Another efficient method based on a multilayer perceptron, named CS_DDoS, is proposed in [195] for the detection of DDoS attacks in the cloud environment. The proposed algorithm scans the incoming packets and classify them as normal or originates from an attacker, in the prevention phase, the malicious classified packets are denied access to the cloud. The author has also compared their proposed method with the machine learning algorithm and has achieved 97% accuracy.

6.5 Chapter Summary

In this chapter, a novel advanced Intrusion Detection System for the detection of DDoS attacks in IoT networks has been proposed. The multi-objective optimization has been adopted as an initial stage of the proposed method for feature selection to reduce the dimensionality of the dataset based on six critical objectives. Deep Learning models Convolutional Neural Network combined with LSTM has been employed for the classification of the attack. Extensive experiments have been performed using high-performance computer enabled with GPU on the latest CICIDS2017 dataset. The dataset is pre-processed and normalized to make data compatible with the proposed method. Since the feature selection method was applied before the classification of the attack on data, the training time reduced by 5-fold. The proposed method has achieved a very impressive high accuracy of 99.03% along with an F1-score value of 99.36 %. We have done extensive evaluation on proposed IDS by adding more layers in the deep learning module of the IDS. The results obtained concludes that doing feature selection before applying the deep learning technique on datasets avoids the requirement of putting more deep learning layers. Instead adding layers degrade the performance of IDS because of overfitting. The proposed method is compared with other state-of-the-art methods. It concludes that the proposed method outperforms other work, which demonstrates the robustness of the proposed method.

Chapter 7. Conclusion and Future Work

In this chapter, the work presented in this thesis has been concluded, and the significant contributions of the presented work are listed. The researched questions stated in Chapter 1 that gave the underlying motivation for conducting the presented work are answered. In the last section, the future works to extend and improve the presented work are listed.

7.1 Thesis Summary

This thesis presents three major contributions as detailed in Chapters 4, 5 and 6. Given below are significant outcomes of the thesis.

(1). Comprehensive Survey on the existing cybersecurity vulnerabilities such as inadequate authorization of IoT devices, unprotected web interfaces and in IoT networks. are presented as part of the literature review. The various types of cyber-attacks in the context of the IoT networks are detailed including the description of the DDoS attacks. The classification of the DDoS attacks in the perspective of IoT networks is given. The survey on the effect and launch of DDoS attack on IoT networks that took place in recent time is presented. This is important for understanding the seriousness of this attack and to imagine its consequences in the future if this attack is not addressed.

(2). Another significant contribution of this thesis is the review on the available and proposed IDSs for the IoT networks are discussed including their limitations that would help in the development of the advanced IDSs. The review on the proposed feature selection methods for IoT networks is detailed. Survey on the proposed IDSs based on DL methodology is presented. Furthermore, open research issues and challenges in the context of the DDoS attacks on the IoT networks are detailed.

(3). The first major contribution of this thesis is proposing and implementing a multi-objective optimization method for performing feature selection and satisfying conflicting objectives for extracting optimal attributes from the datasets for the detection of the DDoS attack. The proposed method incorporated the Jumping Gene adapted NSGA algorithm for the optimized feature selection, considering six important objectives, namely maximize relevance, minimize redundancy, minimize the number of features, maximize classifier accuracy, maximize recall, and maximize precision.

(4). The proposed method obtained feature subsets as Pareto-front, that facilitate the user with choice in selecting the feature set. The extensive evaluation employing the latest CICIDS2017 dataset using standard performance metrics and comparison of the performance of the presented method with state-of-the-art algorithms is presented. The proposed method reduced the number of features from 81 to 6 and achieved a very high 99.90% accuracy. The results on other performance metrices is also quite impressive. The obtained value of recall is 100, precision is 99.90% and redundancy is 0.20 %.

(5).The second major contribution of this thesis is to propose and compare various deep learning models for the detection and classification of DDoS attacks in IoT networks. The four deep learning models feasible in the application for the cybersecurity in IoT networks is presented. The comparison of the proposed deep learning models to discover the best model in terms of performance metrics is conducted.

(6). All the proposed models are extensively evaluated on CICIDS2017 datasets using the standard performance parameters. The performance of the proposed deep learning models is compared with other machine learning algorithms in the context of DDoS attack detection in IoT networks. The proposed model has achieved accuracy of 97.16 % and F1-Score of 98.24 %.

(7). The third major contribution of this thesis is to propose a novel intrusion detection system against the DDoS attack in IoT networks. The Proposed IDS integrate multi-objective based feature selection and the deep learning methodology for the classification of the DDoS attack.

(8). The extensive evaluation of the proposed IDS on the high-performance computer over several standard assessment metrics to analyse the proficiency of the proposed method is conducted. The proposed model has obtained high accuracy of 99.03 % and F1-Score value of 99.36 %. The proposed work is compared with state-of-the-art-algorithms and machine learning methods, which have been used to a great extent in the field of cybersecurity.

7.2 Research Questions answered in this thesis

Question 1. Why cybersecurity, especially the DDoS attack, is a big problem in IoT networks?

The detailed literature is review is conducted in the context of the cybersecurity in IoT networks in Chapter 2. Reviewing the research literature published concludes that there exist many vulnerabilities in the existing IoT networks, that make it easy for the attacker to compromise the IoT devices with ease. The security vulnerabilities are detailed in the chapter

such as inadequate authorization of IoT devices, unprotected web interfaces, vulnerable network services, no proper transport encryption and verification method, poor security configuration, vulnerable cloud interface and lack of physical level security. The launch of recent DDoS attacks that took place on IoT networks has revealed that the lack of proper defence mechanism and security protocols not followed in IoT networks have resulted in the DDoS attacks. The IoT networks have, in reality, aided hackers to initiate and spread the biggest DDoS attacks that have ever taken place. Based on the challenges and problems found in the review in this thesis, it can be concluded that the security vulnerabilities present in IoT networks make it easy for the attackers to compromise thousands of IoT devices across the globe connected with Internet. Another reason is the enhanced sophisticating DDoS attacks, for example, the Mirai botnet attack, and other attacks that were improved version of Mirai attack as discussed in the literature review.

Question 2. How have DDoS attacks adversely affected IoT networks, and what method is used for their launch on the network? The efficiency in terms of detection of cyberattacks of currently deployed IDSs in the detection of DDoS attacks should be studied.

This question is very well addressed in Chapter 2 that includes the survey on the recent DDoS attacks in the context of IoT networks. The development of the IoT networks have aided the attacker for launching the DDoS attacks, so the statistical analysis of the adversities of the DDoS attacks is necessary. Another important aspect is the method of the launch of the attacks into the system, that would help find the present loopholes and would aid in the development of the advanced IDSs is also covered in the literature review. The IDS are used by all the webservers for the detection and mitigation of the cyber-attacks; however, the modern cybersecurity dense mechanisms that are mostly Machine Learning based have failed and resulted in huge economic losses and some DDoS attacks even lasted for many days without being discovered.

Question 3. What is the possible solution for developing an advanced IDS that can learn on its own for defence against the new more sophisticated cyber-attacks?

In the literature review, the study on the recent attacks on DDoS attacks on IoT networks, their mode of action, including the launch of the attacks and the deployed security mechanism against cyber-attacks are detailed. Machine learning-based IDSs have failed in the detection of the attacks. The scope of the DL in the field of cybersecurity with some proposed IDSs is detailed. DL has a unique advantage for learning new things without human interaction. That

actually makes is an idle choice to tackle the problem of the new and more sophisticated cyber-attacks. For example, the DDoS attacks on Imperva's Clients in the year 2019 [10]. The company improved and developed the security systems based on machine learning techniques [41] after the significant DDoS attacks were encountered in 2016 on one of their clients. However, as the DDoS attacks advanced with time, the detection system could not detect the 2019 DDoS attack. This question is answered in this thesis by proposing the DL based IDS primarily evaluated for DDoS attacks in IoT networks in Chapter 6.

Question 4. How to improve the applicability of DL in the field of cybersecurity in IoT networks?

The present interest of the researcher in the field of cybersecurity has moved towards the DL for developing the advanced IDSs that will be capable of detecting the advanced, sophisticated cyber-attacks. This question is addressed in Chapter 4 of this thesis by proposing a multi-objective based feature selection method fulfilling six crucial objectives. The FS method has reduced the training time in deep learning model by five folds as shown in Chapter 6.

7.3 Future work

Some issues are found during the study of the proposed method, that should be addressed in the future. In this section, we have discussed the concerns in the context of proposed work and a potential solution that can tackle those issues.

- The latest CICIDS2017 datasets have been employed for conducting the performance evaluations of the three contributions of this thesis. Although CICIDS2017 are the latest and contain the closest real-world data, this dataset is highly unbalanced. The performance of the IDSs directly gets affected by the data being used for training the system, and this dataset contains only 19 % attack data, that was modified to 39 % by duplicating the data, in this work. Duplicating the data somewhere creates the problem of redundancy in the datasets, so a class balanced real-world dataset is demanded.
- The focus of this thesis is the DDoS attacks because of its adverse effect on IoT networks, not only in terms of services available but huge financial loses it caused to industries. However, other cyber-attacks also affect the IoT networks, that are not broader and bigger as the DDoS attack is but still affect building trust on the IoT networks. For example, Man-In-The-Middle attack, Data and Identity theft, Denial of

Service are the other widespread attacks on IoT networks. So, the dataset containing these attacks as well is needed.

- The proposed model in the future can be exploited in a fog-to-node environment as illustrated in Figure 2.8, for performing distributing computing on several fog nodes for the detection of DDoS attacks. Each fog node should be able to detect the DDoS in their local IoT networks. Proposed IDS could be employed as distributed IDS for resembling the real world IoT network. The experimentation could be conducted on Apache Spark, which is an open source distributed general purpose cluster computing framework by employing several worker nodes in the network. The proposed IDS can be evaluated for the detection of other cyber-attacks by using different datasets that include the data on different cyber-attacks other than the DDoS.
- The deployment of IoT is application-specific, such as Healthcare, Smart Home, Smart City and Industrial IoT. Every type of IoT systems produce different data and build on different communication protocols and infrastructures. The advanced IDSs could focus on particular IoT system and developed according to the individual system requirements.

List of Figures

List of Tables

Appendices: A **[123]**

	Feature Name	**Description**
1	Source IP	Source IP address
2	Source Port	Source port number
3	Destination IP	Destination IP address
4	Destination Port	Destination port number
5	Protocol	Duration of the flow in Microsecond
6	Flow duration	Duration of the flow in Microsecond
7	total FWwd Packet	Total packets in the forward direction
8	total Bwd packets	Total packets in the backward direction
9	total Length of Fwd Packet	Total size of packet in forward direction
10	total Length of Bwd Packet	Total size of packet in backward direction
11	Fwd Packet Length Min	Minimum size of packet in forward direction
12	Fwd Packet Length Max	Maximum size of packet in forward direction
13	Fwd Packet Length Mean	Mean size of packet in forward direction
14	Fwd Packet Length Std	Standard deviation size of packet in forward direction
15	Bwd Packet Length Min	Minimum size of packet in backward direction
16	Bwd Packet Length Max	Maximum size of packet in backward direction
17	Bwd Packet Length Mean	Mean size of packet in backward direction
18	Bwd Packet Length Std	Standard deviation size of packet in backward direction
19	Flow Byte/s	Number of flow packets per second
20	Flow Packets/s	Number of flow bytes per second
21	Flow IAT Mean	Mean time between two packets sent in the flow
22	Flow IAT Std	Standard deviation time between two packets sent in the flow
23	Flow IAT Max	Maximum time between two packets sent in the flow

24	Flow IAT Min	Minimum time between two packets sent in the flow
25	Fwd IAT Min	Minimum time between two packets sent in the forward direction
26	Fwd IAT Max	Maximum time between two packets sent in the forward direction
27	Fwd IAT Mean	Mean time between two packets sent in the forward direction
28	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
29	Fwd IAT Total	Total time between two packets sent in the forward direction
30	Bwd IAT Min	Minimum time between two packets sent in the backward direction
31	Bwd IAT Max	Maximum time between two packets sent in the backward direction
32	Bwd IAT Mean	Mean time between two packets sent in the backward direction
33	Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
34	Bwd IAT Total	Total time between two packets sent in the backward direction
35	Fwd PSH flag	Frequency of PSH flag was set in packets travelling in the forward direction (0 for UDP)
36	Bwd PSH Flag	Frequency of PSH flag was set in packets travelling in the backward direction (0 for UDP)
37	Fwd URG Flag	Frequency of URG flag was set in packets travelling in the forward direction (0 for UDP)
38	Bwd URG Flag	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)

39	Fwd Header Length	Total bytes used for headers in the forward direction
40	Bwd Header Length	Total bytes used for headers in the backward direction
41	FWD Packets/s	Number of forward packets per second
42	Bwd Packets/s	Number of backward packets per second
43	Min Packet Length	Minimum length of a packet
44	Max Packet Length	Maximum length of a packet
45	Packet Length Mean	Mean length of a packet
46	Packet Length Std	Standard deviation length of a packet
47	Packet Length Variance	Variance length of a packet
48	FIN Flag Count	Number of packets with FIN
49	SYN Flag Count	Number of packets with SYN
50	RST Flag Count	Number of packets with RST
51	PSH Flag Count	Number of packets with PUSH
52	ACK Flag Count	Number of packets with ACK
53	URG Flag Count	Number of packets with URG
54	CWE Flag Count	Number of packets with CWE
55	ECE Flag Count	Number of packets with ECE
56	down/Up Ratio	Download and upload ratio
57	Average Packet Size	Average size of packet
58	Avg Fwd Segment Size	Average size observed in the forward direction
59	AVG Bwd Segment Size	Average number of bytes bulk rate in the forward direction
60	Fwd Header Length	Length of header for forward packet
61	Fwd Avg Bytes/Bulk	Average number of bytes bulk rate in the forward direction
62	Fwd AVG Packet/Bulk	Average number of packets bulk rate in the forward direction
63	Fwd AVG Bulk Rate	Average number of bulk rate in the forward direction
64	Bwd Avg Bytes/Bulk	Average number of bytes bulk rate in the backward direction

65	Bwd AVG Packet/Bulk	Average number of packets bulk rate in the backward direction
66	Bwd AVG Bulk Rate	Average number of bulk rate in the backward direction
67	Subflow Fwd Packets	The average number of packets in a sub flow in the forward direction
68	Subflow Fwd Bytes	The average number of bytes in a sub flow in the forward direction
69	Subflow Bwd Packets	The average number of packets in a sub flow in the backward direction
70	Subflow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
71	Init_Win_bytes_forward	The total number of bytes sent in initial window in the forward direction
72	Init_Win_bytes_backward	The total number of bytes sent in initial window in the backward direction
73	Act_data_pkt_forward	Count of packets with at least 1 byte of TCP data payload in the forward direction
74	min_seg_size_forward	Minimum segment size observed in the forward direction
75	Active Min	Minimum time a flow was active before becoming idle
76	Active Mean	Mean time a flow was active before becoming idle
77	Active Max	Maximum time a flow was active before becoming idle
78	Active Std	Standard deviation time a flow was active before becoming idle
79	Idle Min	Minimum time a flow was idle before becoming active
80	Idle Max	Maximum time a flow was idle before becoming active
81	Idle Std	Standard deviation time a flow was idle before becoming active

82	Idle mean	Mean time a flow was idle before becoming active
83	Label	

References

[1] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, 'Twenty security considerations for cloud-supported Internet of Things', *IEEE Internet of things Journal*, vol. 3, no. 3, pp. 269–284, 2015.

[2] I. H. S. Statista, 'Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)', 2018.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, 'DDoS in the IoT: Mirai and other botnets', *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] A. Lohachab and B. Karambir, 'Critical Analysis of DDoS—An Emerging Security Threat over IoT Networks', *J. Commun. Inf. Netw.*, vol. 3, no. 3, pp. 57–78, Sep. 2018, doi: 10.1007/s41650-018-0022-5.

[5] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, 'REATO: REActing TO Denial of Service attacks in the Internet of Things', *Computer Networks*, vol. 137, pp. 37–48, 2018.

[6] L. Deng, 'A tutorial survey of architectures, algorithms, and applications for deep learning', *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[7] M. binti Mohamad Noor and W. H. Hassan, 'Current research on Internet of Things (IoT) security: A survey', *Computer Networks*, vol. 148, pp. 283–294, Jan. 2019, doi: 10.1016/j.comnet.2018.11.025.

[8] Casey Crane, 'The 15 Top DDoS Statistics You Should Know In 2020', Nov. 16, 2019. https://cybersecurityventures.com/the-15-top-ddos-statistics-you-should-know-in-2020/.

[9] Oleg Kupreev, Ekaterina Badovskaya, Alexander Gutnikov, 'DDoS attacks in Q2 2019', Aug. 05, 2019. https://securelist.com/ddos-report-q2-2019/91934/.

[10] Vitaly Simonovich, 'Imperva Blocks Our Largest DDoS L7/Brute Force Attack Ever (Peaking at 292,000 RPS)', Jul. 24, 2019. https://www.imperva.com/blog/imperva-blocks-our-largest-ddos-l7-brute-force-attack-ever-peaking-at-292000-rps/.

[11] Oli Pinson-Roxburgh, 'BULLETPROOF ANNUAL CYBER SECURITY REPORT 2019'. https://www.bulletproof.co.uk/industry-reports/2019.pdf.

[12] Akamai, 'Financial Services Attack Economy'. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-financial-services-attack-economy-executive-summary-2019.pdf.

[13] Charles DeBeck,Joshua Chung, Dave McMillen, *I Can't Believe Mirais: Tracking the Infamous IoT Malware*. 2019.

[14] MarketsandMarkets, *DDoS Protection and Mitigation Market*. 2019.

[15] A. A. Diro and N. Chilamkurti, ‘Distributed attack detection scheme using deep learning approach for Internet of Things’, *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[16] R. Vishwakarma and A. K. Jain, ‘A survey of DDoS attacking techniques and defence mechanisms in the IoT network’, *Telecommun Syst*, vol. 73, no. 1, pp. 3–25, Jan. 2020, doi: 10.1007/s11235-019-00599-z.

[17] ‘Top IoT Vulnerabilities.’ https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.

[18] M. M. Ahemd, M. A. Shah, and A. Wahid, ‘IoT security: A layered approach for attacks & defenses’, in *2017 International Conference on Communication Technologies (ComTech)*, 2017, pp. 104–110.

[19] M. A. Khan and K. Salah, ‘IoT security: Review, blockchain solutions, and open challenges’, *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[20] J. Sengupta, S. Ruj, and S. Das Bit, ‘A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT’, *Journal of Network and Computer Applications*, vol. 149, p. 102481, Jan. 2020, doi: 10.1016/j.jnca.2019.102481.

[21] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, ‘Internet of Things: Security vulnerabilities and challenges’, in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, pp. 180–187.

[22] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, ‘Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home’, *arXiv preprint arXiv:1805.01525*, 2018.

[23] Z. Ling, K. Liu, Y. Xu, Y. Jin, and X. Fu, ‘An end-to-end view of iot security and privacy’, in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[24] A. Mosenia and N. K. Jha, ‘A comprehensive study of security of internet-of-things’, *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2016.

[25] P. Varga, S. Plosz, G. Soos, and C. Hegedus, ‘Security threats and issues in automation IoT’, in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, 2017, pp. 1–6.

[26] M. Kumar and C. Guria, ‘The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization’, *Information sciences*, vol. 382, pp. 15–37, 2017.

[27] P. Shukla, ‘Ml-ids: A machine learning approach to detect wormhole attacks in internet of things’, in *2017 Intelligent Systems Conference (IntelliSys)*, 2017, pp. 234–240.

[28] D. Airehrour, J. A. Gutierrez, and S. K. Ray, ‘SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things’, *Future Generation Computer Systems*, vol. 93, pp. 860–876, 2019.

[29] D. Yin, L. Zhang, and K. Yang, ‘A DDoS attack detection and mitigation with software-defined Internet of Things framework’, *IEEE Access*, vol. 6, pp. 24694–24705, 2018.

[30] Y. Zou, B. Champagne, W. Zhu, and L. Hanzo, ‘Relay-Selection Improves the Security-Reliability Trade-Off in Cognitive Radio Systems’, *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 215–228, Jan. 2015, doi: 10.1109/TCOMM.2014.2377239.

[31] G. Writer, ‘The 5 worst examples of IoT hacking and vulnerabilities in recorded history’, *Internet: https://www. iotforall. com/5-worst-iot-hacking-vulnerabilities*, 2017.

[32] N. Agrawal and S. Tapaswi, ‘Defense Mechanisms against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges’, *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019, doi: 10.1109/COMST.2019.2934468.

[33] P. Gope and B. Sikdar, ‘Lightweight and privacy-preserving two-factor authentication scheme for IoT devices’, *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[34] A. Prakash, M. Satish, T. S. S. Bhargav, and N. Bhalaji, ‘Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture’, *Procedia Computer Science*, vol. 87, pp. 275–280, Jan. 2016, doi: 10.1016/j.procs.2016.05.161.

[35] J. J. Kponyo, J. O. Agyemang, G. S. Klogo, and J. O. Boateng, ‘Lightweight and host-based denial of service (DoS) detection and defense mechanism for resource-constrained IoT devices’, *Internet of Things*, vol. 12, p. 100319, Dec. 2020, doi: 10.1016/j.iot.2020.100319.

[36] A. Roohi, M. Adeel, and M. A. Shah, ‘DDoS in IoT: A Roadmap Towards Security Countermeasures’, in *2019 25th International Conference on Automation and Computing (ICAC)*, Sep. 2019, pp. 1–6, doi: 10.23919/IConAC.2019.8895034.

[37] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, ‘A Survey on Security and Privacy Issues in Internet-of-Things’, *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017, doi: 10.1109/JIOT.2017.2694844.

[38] S. S. Kumar and K. Kulothungan, ‘An Anomaly Behavior based Detection and Prevention of DoS Attack in IoT Environment’, in *2017 Ninth International Conference on Advanced Computing (ICoAC)*, 2017, pp. 287–292.

[39] E. Alomari, S. Manickam, B. B. Gupta, S. Karuppayah, and R. Alfaris, 'Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art', *arXiv preprint arXiv:1208.0403*, 2012.

[40] Akshaya Asokan, 'Massive Botnet Attack Used More Than 400,000 IoT Devices', Jul. 26, 2019. https://www.bankinfosecurity.com/massive-botnet-attack-used-more-than-400000-iot-devices-a-12841.

[41] *Imperva CounterBreach 2.0 Introduces New Machine Learning Algorithm to Protect Data Against Insider Threats*. 2017.

[42] M. Pelloso, A. Vergutz, A. Santos, and M. Nogueira, 'A Self-Adaptable System for DDoS Attack Prediction Based on the Metastability Theory', in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[43] Leah Alger., 'DDoS attackers increasingly abuse public cloud services', Sep. 11, 2018. https://www.devopsonline.co.uk/ddos-attackers-increasingly-abuse-public-cloud-services/.

[44] L. Jin, S. Hao, H. Wang, and C. Cotton, 'Your remnant tells secret: residual resolution in DDoS protection services', in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 362–373.

[45] L. Constantin, 'Thousands of hacked CCTV devices used in DDoS attacks.', Jun. 28, 2016. http://www.pcworld.com/article/3089346/security/thousands-of-hacked-cctv-devices-used-in-ddos-attacks.html.

[46] D. CIDESPANOLPORTUGUES, *Large CCTV Botnet Leveraged in DDoS Attacks*. 2016.

[47] A. G. Eustis, 'The Mirai Botnet and the Importance of IoT Device Security', in *16th International Conference on Information Technology-New Generations (ITNG 2019)*, 2019, pp. 85–89.

[48] M. Goncharov, 'Criminal hideouts for lease: Bulletproof hosting services'.

[49] N. Woolf, 'DDoS attack that disrupted internet was largest of its kind in history, experts say.(2016)', *URL https://www. theguardian. com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet*, 2016.

[50] M. Antonakakis *et al.*, 'Understanding the mirai botnet', in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.

[51] G. MEZZOFIORE, 'A university was attacked by its lightbulbs, vending machines and lamp posts', Feb. 13, 2017. /18/only-earth-can-see-total-solar-eclipse/.

[52] C. Dietz *et al.*, 'IoT-Botnet Detection and Isolation by Access Routers', 2019.

[53] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, ‘Intrusion detection systems for IoT-based smart environments: a survey’, *J Cloud Comp*, vol. 7, no. 1, p. 21, Dec. 2018, doi: 10.1186/s13677-018-0123-6.

[54] S. Anwar *et al.*, ‘From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions’, *Algorithms*, vol. 10, no. 2, p. 39, Jun. 2017, doi: 10.3390/a10020039.

[55] W. Bul’ajoul, A. James, and M. Pannu, ‘Improving network intrusion detection system performance through quality of service configuration and parallel technology’, *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 981–999, 2015.

[56] H. Bostani and M. Sheikhan, ‘Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach’, *Computer Communications*, vol. 98, pp. 52–71, 2017.

[57] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, ‘Mobile network intrusion detection for IoT system based on transfer learning algorithm’, *Cluster Computing*, vol. 22, no. 4, pp. 9889–9904, 2019.

[58] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. U. Rehman, and K. Salah, ‘COLIDE: a collaborative intrusion detection framework for Internet of Things’, *IET Networks*, vol. 8, no. 1, pp. 3–14, 2018.

[59] N. V. Abhishek, T. J. Lim, B. Sikdar, and A. Tandon, ‘An intrusion detection system for detecting compromised gateways in clustered IoT networks’, in *2018 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2018, pp. 1–6.

[60] L. Liu, B. Xu, X. Zhang, and X. Wu, ‘An intrusion detection method for internet of things based on suppressed fuzzy clustering’, *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 113, 2018.

[61] Z. A. Khan and P. Herrmann, ‘A trust based distributed intrusion detection mechanism for internet of things’, in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 1169–1176.

[62] E. Anthi, L. Williams, and P. Burnap, ‘Pulse: an adaptive intrusion detection for the internet of things’, 2018.

[63] A. Amouri, V. T. Alaparthy, and S. D. Morgera, ‘Cross layer-based intrusion detection based on network behavior for IoT’, in *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)*, 2018, pp. 1–4.

[64] K. Yang, J. Ren, Y. Zhu, and W. Zhang, ‘Active learning for wireless IoT intrusion detection’, *IEEE Wireless Communications*, vol. 25, no. 6, pp. 19–25, 2018.

[65] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, ‘An automata based intrusion detection method for internet of things’, *Mobile Information Systems*, vol. 2017, 2017.

[66] B. Arrington, L. Barnett, R. Rufus, and A. Esterline, ‘Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms’, in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, 2016, pp. 1–6.

[67] J. M. R. Danda and C. Hota, ‘Attack identification framework for IoT devices’, in *Information Systems Design and Intelligent Applications*, Springer, 2016, pp. 505–513.

[68] Q. D. La, T. Q. Quek, J. Lee, S. Jin, and H. Zhu, ‘Deceptive attack and defense game in honeypot-enabled networks for the internet of things’, *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1025–1035, 2016.

[69] P. Satam, ‘A Methodology to Design Intrusion Detection Systems (IDS) for IoT/Networking Protocols’, 2019.

[70] J. J. Villalobos, I. Rodero, and M. Parashar, ‘An Unsupervised Approach for Online Detection and Mitigation of High-Rate DDoS Attacks Based on an In-Memory Distributed Graph Using Streaming Data and Analytics’, in *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT ’17*, Austin, Texas, USA, 2017, pp. 103–112, doi: 10.1145/3148055.3148077.

[71] I. Ko, D. Chambers, and E. Barrett, ‘Unsupervised learning with hierarchical feature selection for DDoS mitigation within the ISP domain’, *ETRI Journal*, vol. 41, no. 5, pp. 574–584, 2019, doi: https://doi.org/10.4218/etrij.2019-0109.

[72] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, ‘Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps’, *Knowledge-Based Systems*, vol. 71, pp. 322–338, 2014.

[73] Y.-L. Wan, J.-C. Chang, R.-J. Chen, and S.-J. Wang, ‘Feature-selection-based ransomware detection with machine learning of data analysis’, in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, 2018, pp. 85–88.

[74] J. Yin, T. Tao, and J. Xu, ‘A multi-label feature selection algorithm based on multi-objective optimization’, in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[75] U. Singh and S. N. Singh, ‘Optimal feature selection via NSGA-II for power quality disturbances classification’, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 2994–3002, 2017.

[76] W. Wang, Y. He, J. Liu, and S. Gombault, ‘Constructing important features from massive network traffic for lightweight intrusion detection’, *IET Information Security*, vol. 9, no. 6, pp. 374–379, 2015.

[77] O. Y. Al-Jarrah, A. Siddiqui, M. Elsalamouny, P. D. Yoo, S. Muhaidat, and K. Kim, ‘Machine-learning-based feature selection techniques for large-scale network intrusion detection’, in *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2014, pp. 177–181.

[78] Y. Zhu, J. Liang, J. Chen, and Z. Ming, ‘An improved NSGA-III algorithm for feature selection used in intrusion detection’, *Knowledge-Based Systems*, vol. 116, pp. 74–85, 2017.

[79] S. Sharma, S. R. Nabavi, and G. P. Rangaiah, ‘Jumping gene adaptations of NSGA-II with altruism approach: performance comparison and application to Williams–Otto process’, in *Applications of Metaheuristics in Process Engineering*, Springer, 2014, pp. 395–421.

[80] S. Rathore and J. H. Park, ‘Semi-supervised learning based distributed attack detection framework for IoT’, *Applied Soft Computing*, vol. 72, pp. 79–89, 2018.

[81] A. Abeshu and N. Chilamkurti, ‘Deep learning: The frontier for distributed attack detection in fog-to-things computing’, *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018.

[82] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, ‘Deep Learning Approach for Intelligent Intrusion Detection System’, *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[83] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, ‘Deep learning approach combining sparse autoencoder with SVM for network intrusion detection’, *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[84] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, ‘A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection’, *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[85] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, ‘Deep learning approach for network intrusion detection in software defined networking’, in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 258–263.

[86] Z. Wang, ‘The applications of deep learning on traffic identification (2015)’, *KU Leuven*, vol. 2017, 2016.

[87] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ‘A deep learning approach to network intrusion detection’, *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[88] Y. Li, R. Ma, and R. Jiao, ‘A hybrid malicious code detection method based on deep learning’, *International Journal of Security and Its Applications*, vol. 9, no. 5, pp. 205–216, 2015.

[89] M.-J. Kang and J.-W. Kang, ‘A novel intrusion detection method using deep neural network for in-vehicle network security’, in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–5.

[90] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, ‘A deep learning approach for network intrusion detection system’, in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 21–26.

[91] C. Wu, Y. Guo, and Y. Ma, ‘Adaptive anomalies detection with deep network’, 2015.

[92] S. Venkatraman, M. Alazab, and R. Vinayakumar, ‘A hybrid deep learning image-based analysis for effective malware detection’, *Journal of Information Security and Applications*, vol. 47, pp. 377–389, 2019.

[93] S. Yajamanam, V. R. S. Selvin, F. Di Troia, and M. Stamp, ‘Deep Learning versus Gist Descriptors for Image-based Malware Classification.’, in *ICISSP*, 2018, pp. 553–561.

[94] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, and J. Chen, ‘Detection of malicious code variants based on deep learning’, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[95] S. Ni, Q. Qian, and R. Zhang, ‘Malware identification using visualization images and deep learning’, *Computers & Security*, vol. 77, pp. 871–885, 2018.

[96] W. Shanks, *Enhancing Intrusion Analysis through Data Visualisation*. SANS Institute, Inc, 2015.

[97] W. Aigner *et al.*, ‘Visual analytics: Foundations and experiences in malware analysis’, in *Empirical Research for Software Security*, CRC Press, 2017, pp. 159–192.

[98] K. R. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, ‘Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security’, in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1–6.

[99] C. D. McDermott, F. Majdani, and A. V. Petrovski, ‘Botnet detection in the internet of things using deep learning approaches’, in *2018 international joint conference on neural networks (IJCNN)*, 2018, pp. 1–8.

[100] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, ‘Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning’, *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 88–95, 2018.

[101] G. Thamilarasu and S. Chawla, ‘Towards deep-learning-driven intrusion detection for the Internet of Things’, *Sensors*, vol. 19, no. 9, p. 1977, 2019.

[102] Y. Xin *et al.*, ‘Machine learning and deep learning methods for cybersecurity’, *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[103] C. Yin, Y. Zhu, J. Fei, and X. He, ‘A deep learning approach for intrusion detection using recurrent neural networks’, *Ieee Access*, vol. 5, pp. 21954–21961, 2017.

[104] Y. Zeng, H. Gu, W. Wei, and Y. Guo, ‘$ Deep-Full-Range $: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework’, *IEEE Access*, vol. 7, pp. 45182–45190, 2019.

[105] X. Yuan, C. Li, and X. Li, ‘DeepDefense: identifying DDoS attack via deep learning’, in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–8.

[106] A. Diro and N. Chilamkurti, ‘Leveraging LSTM networks for attack detection in fog-to-things communications’, *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124–130, 2018.

[107] F. Meng, Y. Fu, F. Lou, and Z. Chen, ‘An effective network attack detection method based on kernel PCA and LSTM-RNN’, in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, 2017, pp. 568–572.

[108] W. F. Kindel, E. D. Christensen, and J. Zylberberg, ‘Using deep learning to probe the neural code for images in primary visual cortex’, *Journal of Vision*, vol. 19, no. 4, pp. 29–29, Apr. 2019, doi: 10.1167/19.4.29.

[109] Y. Kim, S. Kim, N. Kang, T. Kim, and H. Kim, ‘Estimation of frequency based snowfall depth considering climate change using neural network’, *Journal of Korean Society of Hazard Mitigation*, vol. 14, no. 1, pp. 93–107, 2014.

[110] I. Stojmenovic and S. Wen, ‘The fog computing paradigm: Scenarios and security issues’, in *2014 federated conference on computer science and information systems*, 2014, pp. 1–8.

[111] S. Yi, Z. Qin, and Q. Li, ‘Security and privacy issues of fog computing: A survey’, in *International conference on wireless algorithms, systems, and applications*, 2015, pp. 685–695.

[112] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, 'Fog computing and its role in the internet of things', in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[113] L. M. Vaquero and L. Rodero-Merino, 'Finding your way in the fog: Towards a comprehensive definition of fog computing', *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[114] W. Dou, Q. Chen, and J. Chen, 'A confidence-based filtering method for DDoS attack defense in cloud environment', *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1838–1850, 2013.

[115] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, 'From today's intranet of things to a future internet of things: a wireless-and mobility-related view', *IEEE Wireless communications*, vol. 17, no. 6, pp. 44–51, 2010.

[116] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, 'DDoS attack protection in the era of cloud computing and software-defined networking', *Computer Networks*, vol. 81, pp. 308–319, 2015.

[117] J. François, I. Aib, and R. Boutaba, 'FireCol: a collaborative protection network for the detection of flooding DDoS attacks', *IEEE/ACM Transactions on networking*, vol. 20, no. 6, pp. 1828–1841, 2012.

[118] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, 'The impact of class imbalance in classification performance metrics based on the binary confusion matrix', *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

[119] Z. Chiba, N. Abghour, K. Moussaid, A. El omri, and M. Rida, 'Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms', *Computers & Security*, vol. 86, pp. 291–317, Sep. 2019, doi: 10.1016/j.cose.2019.06.013.

[120] R. Vijayanand, D. Devaraj, and B. Kannapiran, 'Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection', *Computers & Security*, vol. 77, pp. 304–314, 2018.

[121] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, 'Toward developing a systematic approach to generate benchmark datasets for intrusion detection', *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012, doi: 10.1016/j.cose.2011.12.012.

[122] 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.

[123] ‘Network Traffic Flow analyzer’. http://netflowmeter.ca/netflowmeter.html (accessed Jan. 23, 2020).

[124] W. Goralski, ‘Chapter 5 - IPv4 and IPv6 Addressing’, in *The Illustrated Network (Second Edition)*, W. Goralski, Ed. Boston: Morgan Kaufmann, 2017, pp. 139–173.

[125] ‘Convert ip address to decimal | ip2country.net’. http://www.ip2country.net/ip2country/ip_number.html (accessed Nov. 25, 2020).

[126] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, ‘RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks’, *Future Internet*, vol. 12, no. 3, Art. no. 3, Mar. 2020, doi: 10.3390/fi12030044.

[127] J. Jang and S. Yoon, ‘Feature Concentration for Supervised and Semi-supervised Learning with Unbalanced Datasets in Visual Inspection’, *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2020, doi: 10.1109/TIE.2020.3003622.

[128] X. Xiaolong, C. Wen, and S. Yanfei, ‘Over-sampling algorithm for imbalanced data classification’, *Journal of Systems Engineering and Electronics*, vol. 30, no. 6, pp. 1182–1191, Dec. 2019, doi: 10.21629/JSEE.2019.06.12.

[129] A. Hanskunatai, ‘A New Hybrid Sampling Approach for Classification of Imbalanced Datasets’, in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, Apr. 2018, pp. 67–71, doi: 10.1109/CCOMS.2018.8463228.

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

[131] Y. Gu, K. Li, Z. Guo, and Y. Wang, ‘Semi-Supervised K-Means DDoS Detection Method Using Hybrid Feature Selection Algorithm’, *IEEE Access*, vol. 7, pp. 64351–64365, 2019, doi: 10.1109/ACCESS.2019.2917532.

[132] ‘7 Techniques to Handle Imbalanced Data’, *KDnuggets*. https://www.kdnuggets.com/7-techniques-to-handle-imbalanced-data.html/ (accessed Nov. 25, 2020).

[133] K. J. Singh, K. Thongam, and T. De, ‘Detection and differentiation of application layer DDoS attack from flash events using fuzzy-GA computation’, *IET Information Security*, vol. 12, no. 6, pp. 502–512, Apr. 2018, doi: 10.1049/iet-ifs.2017.0500.

[134] G. Chandrashekar and F. Sahin, ‘A survey on feature selection methods’, *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[135] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, ‘Feature selection for high-dimensional data’, *Prog Artif Intell*, vol. 5, no. 2, pp. 65–75, May 2016, doi: 10.1007/s13748-015-0080-y.

[136] I. Zamani, ‘Optimal distributed generation planning based on NSGA-II and MATPOWER’, Brunel University London, 2015.

[137] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ‘A fast and elitist multiobjective genetic algorithm: NSGA-II’, *IEEE Trans. Evol. Computat.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[138] M. Ramteke and S. K. Gupta, ‘Biomimicking altruistic behavior of honey bees in multi-objective genetic algorithm’, *Industrial & Engineering Chemistry Research*, vol. 48, no. 21, pp. 9671–9685, 2009.

[139] V. De Buck, C. A. M. López, P. Nimmegeers, I. Hashem, and J. Van Impe, ‘Multi-objective optimisation of chemical processes via improved genetic algorithms: A novel trade-off and termination criterion’, in *Computer Aided Chemical Engineering*, vol. 46, Elsevier, 2019, pp. 613–618.

[140] J. Valadi and P. Siarry, *Applications of metaheuristics in process engineering*. Springer, vol. 31, no. 1, pp. 76–77, 2014.

[141] ‘Enhancing the performance of MOEAs: an experimental presentation of a new fitness guided mutation operator: Journal of Experimental & Theoretical Artificial Intelligence: Vol 29, No 1’. https://www.tandfonline.com/doi/abs/10.1080/0952813X.2015.1132260 (accessed Dec. 28, 2019).

[142] B. Sankararao and S. K. Gupta, ‘Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using two jumping gene adaptations of simulated annealing’, *Computers & Chemical Engineering*, vol. 31, no. 11, pp. 1496–1515, 2007.

[143] B. McClintock, *The discovery of characterization of transposable elements: the collected papers of Barbara McClintock*. Garland Pub., 1987.

[144] N. Safaeian, M. Miri, and M. Rashki, ‘New simulation-based frameworks for multi-objective reliability-based design optimization of structures’, *Applied Mathematical Modelling*, vol. 62, May 2018, doi: 10.1016/j.apm.2018.05.015.

[145] H. Ren, Y. Lu, Q. Wu, X. Yang, and A. Zhou, ‘Multi-objective optimization of a hybrid distributed energy system using NSGA-II algorithm’, *Frontiers in Energy*, vol. 12, no. 4, pp. 518–528, 2018.

[146] M. Roopak, G. Tian and J. Chambers, ‘Multi-Objective based Feature Selection for DDoS Attack Detection in IoT Network’, *IET Networks*, 2020.

[147] H. Hao, M. Wang, and Y. Tang, ‘Feature selection based on improved maximal relevance and minimal redundancy’, in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 1426–1429.

[148] G. Karakaya, S. Galelli, S. D. Ahipaşaoğlu, and R. Taormina, ‘Identifying (quasi) equally informative subsets in feature selection problems for classification: A max-relevance min-redundancy approach’, *IEEE transactions on cybernetics*, vol. 46, no. 6, pp. 1424–1437, 2015.

[149] J. Xu, B. Tang, H. He, and H. Man, ‘Semisupervised feature selection based on relevance and redundancy criteria’, *IEEE transactions on neural networks and learning systems*, vol. 28, no. 9, pp. 1974–1984, 2016.

[150] P. Tang, H. Fang, and H. Si, ‘Maximal relevance feature selection for human activity recognition in smart home’, in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 4264–4268.

[151] I. H. Witten and E. Frank, ‘Data mining: practical machine learning tools and techniques with Java implementations’, *Acm Sigmod Record*, vol. 31, no. 1, pp. 76–77, 2002.

[152] R. Qian, Y. Yue, F. Coenen, and B. Zhang, ‘Visual attribute classification using feature selection and convolutional neural network’, in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, 2016, pp. 649–653.

[153] S. Li, H. Yu, and L. Yuan, ‘A novel approach to remote sensing image retrieval with multi-feature VP-tree indexing and online feature selection’, in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, 2016, pp. 133–136.

[154] B. Jiaramaneepinit and C. Nuthong, ‘Extended Extreme Learning Machine: A Novel Framework for Neural Network’, in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2018, pp. 1629–1634, doi: 10.1109/SMC.2018.00282.

[155] J. Keuper and F.-J. Pfreundt, ‘Distributed Training of Deep Neural Networks: Theoretical and Practical Limits of Parallel Scalability’, *arXiv:1609.06870 [cs]*, Dec. 2016, Accessed: Nov. 26, 2020. [Online]. Available: http://arxiv.org/abs/1609.06870.

[156] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, ‘Extreme learning machine: a new learning scheme of feedforward neural networks’, in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, Jul. 2004, vol. 2, pp. 985–990 vol.2, doi: 10.1109/IJCNN.2004.1380068.

[157] A. Budiman and M. I. Fanany, ‘Pose-based 3D human motion analysis using Extreme Learning Machine’, in *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, Oct. 2013, pp. 3–7, doi: 10.1109/GCCE.2013.6664834.

[158] G. Huang, H. Zhou, X. Ding, and R. Zhang, ‘Extreme Learning Machine for Regression and Multiclass Classification’, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, Apr. 2012, doi: 10.1109/TSMCB.2011.2168604.

[159] Q. Zhang, D. Jian, R. Xu, W. Dai, and Y. Liu, 'Integrating heterogeneous data sources for traffic flow prediction through extreme learning machine', in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 4189–4194, doi: 10.1109/BigData.2017.8258443.

[160] T. Hussain, S. M. Siniscalchi, C.-C. Lee, S.-S. Wang, Y. Tsao, and W.-H. Liao, 'Experimental study on extreme learning machine applications for speech enhancement', *IEEE Access*, vol. 5, pp. 25542–25554, 2017.

[161] Y.-P. Zhao, G. Huang, Q.-K. Hu, J.-F. Tan, J.-J. Wang, and Z. Yang, 'Soft extreme learning machine for fault detection of aircraft engine', *Aerospace Science and Technology*, vol. 91, pp. 70–81, 2019.

[162] A. EL Bakri, M. Koumir, and I. Boumhidi, 'Extreme learning machine for fault detection and isolation in wind turbine', in *2016 International Conference on Electrical and Information Technologies (ICEIT)*, May 2016, pp. 174–179, doi: 10.1109/EITech.2016.7519584.

[163] M. A. Salam, H. M. Zawbaa, E. Emary, K. K. A. Ghany, and B. Parv, 'A hybrid dragonfly algorithm with extreme learning machine for prediction', in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2016, pp. 1–6, doi: 10.1109/INISTA.2016.7571839.

[164] X. Wu, X. Feng, E. Boutellaa, and A. Hadid, 'Kinship Verification using Color Features and Extreme Learning Machine', in *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, Jul. 2018, pp. 187–191, doi: 10.1109/SIPROCESS.2018.8600423.

[165] H. Erişti, Ö. Yıldırım, B. Erişti, and Y. Demir, 'Optimal feature selection for classification of the power quality events using wavelet transform and least squares support vector machines', *International Journal of Electrical Power & Energy Systems*, vol. 49, pp. 95–103, Jul. 2013, doi: 10.1016/j.ijepes.2012.12.018.

[166] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, 'Decision tree based light weight intrusion detection using a wrapper approach', *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, Jan. 2012, doi: 10.1016/j.eswa.2011.06.013.

[167] 'KDD Cup 1999 Data'. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed Nov. 26, 2020).

[168] A. R. Yusof, N. I. Udzir, A. Selamat, H. Hamdan, and M. T. Abdullah, 'Adaptive feature selection for denial of services (DoS) attack', in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, Nov. 2017, pp. 81–84, doi: 10.1109/AINS.2017.8270429.

[169] 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*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[170] E. Balkanli, A. N. Zincir-Heywood, and M. I. Heywood, ‘Feature selection for robust backscatter DDoS detection’, in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, Oct. 2015, pp. 611–618, doi: 10.1109/LCNW.2015.7365905.

[171] M. Stevanovic and J. M. Pedersen, ‘On the use of machine learning for identifying botnet network traffic’, *Journal of Cyber Security and Mobility*, vol. 4, no. 2, pp. 1–32, Jan. 2016, doi: 10.13052/jcsm2245-1439.421.

[172] F. K. Wai, Z. Lilei, W. K. Wai, S. Le, and V. L. L. Thing, ‘Automated Botnet Traffic Detection via Machine Learning’, in *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South), Oct. 2018, pp. 0038–0043, doi: 10.1109/TENCON.2018.8650466.

[173] G. Kirubavathi and R. Anitha, ‘Structural analysis and detection of android botnets using machine learning techniques’, *Int. J. Inf. Secur.*, vol. 17, no. 2, pp. 153–167, Apr. 2018, doi: 10.1007/s10207-017-0363-3.

[174] G. Guo and N. Zhang, ‘A survey on deep learning based face recognition’, *Computer Vision and Image Understanding*, vol. 189, p. 102805, 2019, doi: https://doi.org/10.1016/j.cviu.2019.102805.

[175] B. Alshemali and J. Kalita, ‘Improving the reliability of deep neural networks in NLP: A review’, *Knowledge-Based Systems*, 2019, doi: https://doi.org/10.1016/j.knosys.2019.105210.

[176] G. E. Hinton, ‘Deep belief networks’, *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[177] Q. Wang and G. Guo, ‘Benchmarking deep learning techniques for face recognition’, *Journal of Visual Communication and Image Representation*, vol. 65, p. 102663, 2019, doi: https://doi.org/10.1016/j.jvcir.2019.102663.

[178] B. Hu, Z. Lu, H. Li, and Q. Chen, ‘Convolutional Neural Network Architectures for Matching Natural Language Sentences’, in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2042–2050.

[179] Y. LeCun, Y. Bengio, and G. Hinton, ‘Deep learning. nature 521’, 2015.

[180] D. H. Hubel and T. N. Wiesel, ‘Ferrier lecture-Functional architecture of macaque monkey visual cortex’, *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 198, no. 1130, pp. 1–59, 1977.

[181] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ‘Gradient-based learning applied to document recognition’, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[182] ‘An example of convolutional neural network’. https://www.researchgate.net/figure/An-example-of-convolutional-neural-network-A-typical-CNN-model-has-three-main-parts_fig3_331442449.

[183] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, ‘Long short term memory networks for anomaly detection in time series’, in *Proceedings*, 2015, p. 89.

[184] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin Heidelberg: Springer-Verlag, 2012.

[185] K. J. Singh and T. De, ‘MLP-GA based algorithm to detect application layer DDoS attack’, *Journal of Information Security and Applications*, vol. 36, pp. 145–153, Oct. 2017, doi: 10.1016/j.jisa.2017.09.004.

[186] M. Wang, Y. Lu, and J. Qin, ‘A dynamic MLP-based DDoS attack detection method using feature selection and feedback’, *Computers & Security*, vol. 88, p. 101645, Jan. 2020, doi: 10.1016/j.cose.2019.101645.

[187] D. Başkaya and R. Samet, ‘DDoS Attacks Detection by Using Machine Learning Methods on Online Systems’, in *2020 5th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2020, pp. 52–57, doi: 10.1109/UBMK50275.2020.9219476.

[188] S. Haider *et al.*, ‘A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks’, *IEEE Access*, vol. 8, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.

[189] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón, and D. Siracusa, ‘Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection’, *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, Jun. 2020, doi: 10.1109/TNSM.2020.2971776.

[190] R. Hwang, M. Peng, C. Huang, P. Lin, and V. Nguyen, ‘An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection’, *IEEE Access*, vol. 8, pp. 30387–30399, 2020, doi: 10.1109/ACCESS.2020.2973023.

[191] M. Roopak, G. Y. Tian, and J. Chambers, ‘Deep Learning Models for Cyber Security in IoT Networks’, in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0452–0457.

[192] M. Roopak, G. Y. Tian, and J. Chambers, ‘An Intrusion Detection System Against DDoS Attacks in IoT Networks’’, presented at the IEEE 10th Annual Computing and Communication Workshop and Conference (CCWC), USA, 2020.

[193] Z. Liu, Y. He, W. Wang, and B. Zhang, ‘DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN’, *China Communications*, vol. 16, no. 7, pp. 144–155, 2019.

[194] S. Ali and Y. Li, ‘Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network’, *IEEE Access*, vol. 7, pp. 108647–108659, 2019.

[195] A. Sahi, D. Lai, Y. Li, and M. Diykh, ‘An efficient DDoS TCP flood attack detection and prevention system in a cloud environment’, *IEEE Access*, vol. 5, pp. 6036–6048, 2017.