

2021-10-19

MACHINE LEARNING BASED EFFECTIVE INTRUSION DETECTION FRAMEWORK FOR SDN ORCHESTRATED INTERNET OF THINGS

Esubalew, Mulat

<http://ir.bdu.edu.et/handle/123456789/12802>

Downloaded from DSpace Repository, DSpace Institution's institutional repository



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF ELECTRICAL AND COMPUTER
ENGINEERING

MACHINE LEARNING BASED EFFECTIVE INTRUSION
DETECTION FRAMEWORK FOR SDN ORCHESTRATED
INTERNET OF THINGS

By
Esubalew Mulat

A Thesis Submitted to the School of Graduate Studies of Bahir
Dar Institute of Technology in Partial fulfillment of the
Requirements for the Degree of

Master of Science
in Computer Engineering

July 2012 E.C.

Bahir Dar, Ethiopia

Bahir Dar University

Bahir Dar institute of Technology

School of Graduate Studies

Faculty of Electrical and Computer Engineering

Machine Learning Based Effective Intrusion Detection Framework
for SDN Orchestrated Internet of Things

By

Esubalew Mulat

Advisor:

Henock Mulugeta (PhD)

Co-advisor:

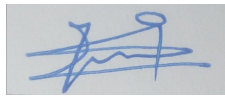
Fikreselam Gared (Assoc. Prof.)

Declaration

I declare that this thesis titled, 'Machine Learning Based Effective Intrusion Detection Framework for SDN Orchestrated Internet of Things' and the work presented in it are my own, and all sources of materials used for the thesis has been clearly stated and attributed.

Student Name: Esubalew Mulat

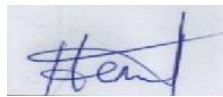
Signature:



This thesis has been submitted for examination with my approval as a university advisor.

Advisor Name: Henock Mulugeta (PhD)

Advisor's Signature:



Co-advisor Name: Fikreselam Gared (Assoc. Prof.)

Co-advisor's Signature:



Bahir Dar University
Bahir Dar Institute of Technology
School of Graduate Studies
Faculty of Electrical and Computer Engineering
Thesis Approval Sheet

Student

Esubalew Mulat [Signature] 09/11/12 E.C.
Name Signature Date

The following graduate faculty members certify that this student has successfully presented the necessary written final thesis and oral presentation for partial fulfillment of the thesis requirements for the Degree of Master of Science in Computer Engineering.

Approved by:

Advisor:

Henock Mulugeta (PhD) [Signature] 09/11/2012 E.C.
Name Signature Date

External Examiner:

Abebe Tesfahun (PhD) [Signature] 09/11/2012 E.C.
Name Signature Date

Internal Examiner:

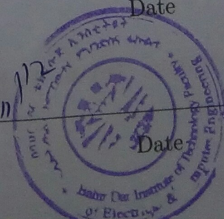
Tenbit Admassu [Signature] 10/11/2012 E.C.
Name Signature Date

Chair Holder:

Aduana Nene [Signature] 13/11/12 E.C.
Name Signature Date

Faculty Dean:

Tewodros Gera Workneh [Signature] 13/11/12 E.C.
Name Faculty Dean Signature Date



Abstract

Internet of Things (IoT) is an extension of the Internet. It extends the human to human interconnection and intercommunication of the Internet by including things, shifting the current Internet scenario to bring anytime, anywhere, and anything communication. A discipline in networking evolving in parallel with IoT is Software Defined Networking (SDN). It is a significant technology and can solve the different problems existing in the traditional network systems. It provides a new home to address the different challenges existing in different network based systems including IoT. Yet, the issue of security of SDIoT environments is still a headache for security researchers since the vulnerability space of IoT systems are intensified by the incorporation of SDN having its own security limitations. One important security challenge prevailing in such systems is a guarantee of service availability. The ever increasing denial of service (DoS) attacks are responsible to such service denials.

A centralized signature based intrusion detection systems (IDS) is proposed and developed in this work. Three machine learning (ML) algorithms, decision tree (DTree), Random Forest (RF) and Support Vector Machine (SVM) are used in the experiments. A very popular and recent benchmark dataset, CICIDS2017, has been used for training and validating the ML models. An accuracy result of 99.967% has been achieved by using only eleven features on the Wednesday's release of the dataset. This result is higher than the achieved accuracy results of related works considering the original CICIDS2017 dataset. A maximum cross validated accuracy result of 99.728% has been achieved on the same release of the dataset. These developed models meet the basic requirement of a supervised IDS systems developed for smart environments and can effectively be used in different IoT service scenarios.

Key Words: CICIDS2017, DoS, Intrusion detection, IoT, SDIoT, SDN, Security

Acknowledgement

Forget about thinking now and then to bring a solution for an existing problem or rehearsing here and there to organize, beautify, and finalize a research work, it is really a blessing to sit on a chair and keep calm for five minutes. I would like to thank God for every bit of good he has done for me and specially for all the blessings I have but I don't know and I don't feel. My next acknowledgement goes to the reason of my survival, the reason of my pleasures and the reason of my hopes, for her for St. Mary.

Henock Mulugeta (PhD) is a kind of teacher that you want to be like. During my research works, I have got it him more humble and caring. Another man of honor is Fikreselam Gared (Assoc. Prof.). I have said it earlier and I repeat it once again that he is a kind of person that this world really lacks. His dedications and professional ethics worth a copy. I would like to pass my gratitude for both Henock Mulugeta (PhD) and Fikreselam Gared (Assoc. Prof.) for their support, understanding, care and invaluable comments in my work starting from the commencing periods of the research to the end.

Contents

Declaration	ii
Approval	iii
Abstract	iv
Acknowledgements	v
List of Figures	ix
List of Tables	xi
Abbreviations	xii
Symbols	xiv
1 Introduction	1
1.1 Introduction	1
1.2 Motivations	5
1.3 Statement of the Problem	5
1.4 Objective	8
1.4.1 General Objective	8
1.4.2 Specific Objectives	8
1.5 Scope of the Research	8
1.6 Significance of the Study	9
1.7 Contributions of the Work	10
1.8 Thesis Organization	10
2 Background	11
2.1 IoT	11
2.1.1 Architecture of IoT	12
2.1.2 IoT Applications	13
2.2 SDN	15
2.2.1 Architecture of SDN	16
2.2.2 Advantages of SDN	16
2.3 DoS Attacks and Defense Mechanisms	17

2.3.1	Motivations for and Trends of DoS Attacks	18
2.3.2	DoS Classification	18
2.4	IDS Systems	20
2.4.1	Requirements of IDS Systems Designed for IoT	21
3	Literature Review	22
3.1	IoT Security	22
3.2	Component IoT System Security	23
3.3	DoS Attack Mitigation	24
3.4	More on IDS Systems in IoT	25
3.5	IDS Systems Involving SDN and IoT	29
4	System Development	32
4.1	Dataset Selection	32
4.2	Hardware and Software Requirements	35
4.2.1	Hardware Resources	35
4.2.2	Software Requirements	35
4.3	Devising SDIoT Architecture	37
4.3.1	Design Considerations	37
4.3.1.1	IoT Infrastructure Layer Components	37
4.3.1.2	Placement of SDN Switches and IoT Gateways	38
4.3.1.3	6LoWPAN Gateway Implementation Design	38
4.4	ML Steps Followed in IDS Development	40
4.4.1	Data Cleaning and Preprocessing	40
4.4.2	Train Test Data Split	43
4.4.3	Training	44
4.4.4	Testing	44
4.5	Principles Considered in the Work	44
4.5.1	Parameter Tuning	44
4.5.2	Feature Selection	45
4.5.3	Overfitting	46
4.5.4	Efficiency Enhancement	47
4.6	IDS Architecture Design	47
4.7	Evaluation	50
4.7.1	Evaluation Techniques	50
4.7.2	Evaluation Measures	51

5	Results and Discussion	53
5.1	Results and Discussions	53
5.2	Session 1: Simple Default Experiments	54
5.3	Session 2: Experiments with Parameter Tuning and Feature Selection	54
5.3.1	Parameter Tuning with DTree	55
5.3.2	Parameter Tuning with RF	58
5.3.3	Parameter Tuning with SVM	60
5.4	Results with CV	62
5.4.1	CV with DTree	63
5.4.2	CV with RF estimator	64
5.5	Comparison with Other Related Works	66
6	Conclusion and Future Works	68
6.1	Conclusion	68
6.2	Future Works	69
	Bibliography	71
A	List of features in CICIDS2017 dataset	79
B	Description of features selected by the best RF estimator	82
C	Steps to Determine the Eleven Aggregated Features	83

List of Figures

Figure 2.1	Common architecture of the traditional IoT system	13
Figure 2.2	Architecture of the SDN paradigm	17
Figure 4.1	Architecture of the proposed SDIoT system	40
Figure 4.2	ML development steps followed	41
Figure 4.3	Architecture of the proposed IDS system	48
Figure 5.1	Performance results of tuned DTree estimators on CICIDS2017-Wed1 dataset	56
Figure 5.2	Performance results of the best performing DTree estimator on CICIDS2017-Wed1 dataset	57
Figure 5.3	Performance results of tuned DTree estimators on CICIDS2017-Wed2 dataset	58
Figure 5.4	Performance results of tuned RF estimators on CICIDS2017-Wed1 dataset	59
Figure 5.5	Performance results of tuned RF estimators with 17, 19, 20, 21 and 23 trees on CICIDS2017-Wed1 dataset	60
Figure 5.6	Performance results of tuned RF estimators on CICIDS2017-Wed1 dataset	61
Figure 5.7	Performance results of tuned RF estimators on CICIDS2017-Wed2 dataset	61
Figure 5.8	Performance results of tuned DTree estimators on CICIDS2017-Wed2 dataset	63
Figure 5.9	Performance results of the best performing DTree estimator on CICIDS2017-Wed1 dataset using cross validation	64
Figure 5.10	Cross validated experiments on CICIDS2017-Wed2 dataset using RFECV and DTree	64
Figure 5.11	RF estimators with CV and RFE filter selection module on CICIDS2017-wed1	65

Figure 5.12 RF estimators with CV and RFECV filter selection module on CICIDS2017-wed1	65
Figure 5.13 Performance of RF estimators with CV and RFE filter se- lection module on CICIDS2017-Wed2 dataset	66

List of Tables

Table 4.1	Total number of DoS/DDoS attacks in CICIDS2017 dataset . . .	34
Table 4.2	Description of features to be collected from the switches	49
Table 5.1	Accuracy results of the three estimators with their default parameter values	54
Table 5.2	Summary of results of the tuned decision tree estimators	56
Table 5.3	Features selected by the best performing RF estimator	62
Table 5.4	Accuracy results of the SVM estimator with SelectKBest fea- ture selector	62
Table 5.5	Performance comparison of works targetting DoS attacks on the CICIDS2017 dataset	67
Table 5.6	Performance comparison of works considering all attacks on the CICIDS2017 dataset	67
Table B.1	Description of selected features	82

Abbreviations

6BG	IPv6 Border Router
6LoWPan	IPv6 over Low power Wireless Personal Area Network
API	Application Program Interface
AI	Artificial Intelligence
Avg	Average
BWD	Backward
CAPEX	Capital Expenditure
CIC	Canadian Institute for Cyber security
CICIDS2017	CIC Intrusion Detection System 2017 Dataset
CPS	Cyber Physical Systems
CPU	Central Processing Unit
CV	Cross Validation
DNS	Domain Name System
DoS	Denial of Service
DR	Detection Rate
DTree	Decision Tree
FAR	False Alarm Rate
FN	False Negative
FP	False Positive
FWD	Forward
HDD	Hard Disk Drive
HIDS	Host based Intrusion Detection System
HTTP	Hyper Text Transport Protocol
ICMP	Internet Control Message Protocol
ICN	Information Centric Network
IDS	Intrusion Detection Ssystem
IG	Informatio Gain

IoT	I nternet of T hings
IP	I nternet P rotocol
IPv6	I P version 6
LAN	L ocal A rea N etworking
MAC	M edium A ccess C ontrol
ML	M achine L earning
NAN	N ot a N umber
NFC	N ear F ield C ommunication
NFV	N etwork F unction V irtualization
NIDS	N etwork I ntrusion D etection S ystem
NTP	N etwork T ime P rotocol
OOP	O bject O riented P rogramming
OF	O pen F low
OPEX	O perational E xpenditure
PL	P rogramming L anguage
QoS	Q uality of S ervice
RF	R andom F orest
RFE	R ecursive F eature S election
RFECV	R ecursive F eature S elecetion with C ross V alidation
RFID	R adio F requency I Dentification
s6BR	simplified I Pv 6 B order R outer
SDIoT	S oftware D efined I oT
SDN	S oftware D efined N etworking
SVM	S upport V ector M achine
SYN	S ynchronize
TCP	T ransport C ontrol P rotocol
TN	T rue N egative
TP	T rue P ositive
UDP	U ser D ata G ram
WAN	W ide A rea N etworking
WSN	W ireless S ensor N etwork

Symbols

<i>Tbs</i>	Tera bits per second
<i>GHz</i>	Giga Hertz
<i>GB</i>	Giga Bytes

Chapter 1

Introduction

1.1 Introduction

Internet of Things (IoT) is an extension of the Internet. People use the Internet to get different kinds of benefits. If human beings use the Internet to get several benefits in life, then why not objects are permitted to the communication area to add extra other benefits for human beings? IoT aims to answer this question by extending the scenario in Internet, the interconnection and intercommunication of people to people, by including things. Thus in IoT, an interconnection and intercommunication of things among themselves and with different people will be possible. By things, it is meant any physical object in the world, equipped with sensors and/or actuators, communication capability and processing units. Things can also be virtual objects like objects in an Object Oriented Programming (OOP), processes, database and other related entities found in the computer science world [1].

IoT is envisioned to make the life of human beings smarter and better than have been ever before. Even if it is at its infancy, IoT has already seen some initial deployments in different sectors. Health Care, mobile asset tracking, intelligent fleet management, smart grid, environmental hazard detection and protection, home automation, smart agriculture and industrial service are only some of the applications expected to be acquired from provision of IoT systems [2, 3]. The potential applicability of IoT is immense and it is too difficult to anticipate what

it can bring in the future.

An equally evolving discipline in networking is Software Defined Networking (SDN). SDN is just an intelligent networking paradigm which is characterized by its focus on the separation of the control plane and the data plane. In the traditional networking paradigm, routing, network management, and other network related decisions are undertaken by routers and switches which at the same time are responsible for forwarding of data to the intended interface. But such network configurations impose scalability, network management, flexibility, interoperability and other problems on the underlying network system [4, 5]. Thus, the traditional network paradigm hinders proper service functioning and guarantying of various Quality of Service (QoS) requirements.

SDN is a promising paradigm in solving the aforementioned problems in which the traditional networking paradigm can hardly do. It is found promising for different technologies like cloud computing, data center, and future technologies (Information Centric Network (ICN), 5G, IoT, and others) [4]. It is highly catching the eye of industries, academia and governmental organizations for the enormous advantages it brings to different systems. Cox et al. in [4] added that SDN has already been deployed in over 100 famous companies starting from its infancy including Google, China Mobile, AT&T, T-Mobil, Telefonica. The general consensus is that SDN is the heart of future networks.

IoT is one of the technologies that can highly benefit from SDN. Significant role is expected to be rendered by SDN in different aspects of IoT systems and it is one of the prominent technologies, along with network virtualization, network function virtualization (NFV) and others, believed to be key enablers of IoT. Some ambiguity or uncertainty issues arise on the level of contribution SDN provides to different technologies including IoT.

As has been stated earlier, the traditional network paradigm is not capable of addressing the problems incurred in IoT implementation. This is mainly because components of IoT and the overall IoT systems are heterogeneous, constrained and highly scalable in nature. All these and other factors make realization of IoT systems much difficult. The dynamic and agile nature of SDN, however, can address the aforementioned challenges in IoT paving the way for more secure, highly scalable and interoperable IoT systems. SDN is considered revolutionary network

technology and is supporting heterogeneous networking with rapid evolution and dynamism using programmable planes. Tayyaba et al. in [6] stated that the SDN and IoT integration can meet the expectations of control and management in diverse scenarios.

Even if IoT is envisioned to provide immense opportunities for the human being, its vision will not come with ease and comfort. Rather a milliard of challenges has to be addressed and lots of vulnerabilities shall be filled up. There are different open issues which make realization of IoT systems very difficult. One prime challenge that faces these systems is the issue of security. Several security works are out there to defend these systems against different security breaches. Yet, the issue of security is at its infancy and its lowest levels for IoT systems. And as it is stated in [7], recent security attacks have revealed the ubiquitous-ness of security loopholes in IoT.

Moreover, even if SDN is hoped for providing conducive environment for novel security works in IoT, there are extra additional unique vulnerabilities imposed by SDN itself. These inherent vulnerabilities of SDN arise due its architecture, and implementations in controller's Application Program Interface (API), memory and other units [4, 8, 9]. Due to such accompanied vulnerabilities in SDN, additional attacks are imposed upon SDN orchestrated IoT systems (SDIoT) in addition to those imposed by the nature of IoT.

Accounting to the extended vulnerability space in SDIoT systems, enormous types of attacks are launched towards them. One class of such attacks target the availability of network systems and devices. In [10] availability was mentioned to be one of the three main requirements of IoT alongside confidentiality and integrity. The attacks that target on the availability of system services are collectively referred to as Denial of Service (DoS) attacks. As its is put in [11] put, these attacks are widely used cyber-attacks. These DoS attacks are ever increasing. This rise is mainly accounted for the enormous integration of poorly secured IoT devices to the Internet, which in turn are recruited and used for botnet attacks. The vulnerabilities found within devices, communication links and communication protocols can be used by these attacks to deny service to the intended users.

These DoS attacks might sometimes end up causing only minor inconvenience to users. But in many circumstances they are very much costly and have devastating

effects. This catastrophe is expected to further escalate in IoT systems. It is due to the fact that many IoT systems are deployed in environments, like health systems and vehicular traffic services, where a minimal service unavailability might result in disastrous consequences which can include loss of money and even human life [10, 12]. Mendez et al. in [13] stated that DoS attacks are listed as one of the key challenges to be addressed in the IoT. The stealth nature of DoS attacks, the significant loss it incurs with a minor attack operation, the very constrained environment provided by IoT systems, and additional vulnerable space granted by SDN systems, among others, make SDIoT systems a conducive area for DoS attacks.

This research work aims to come up with solutions that defend SDIoT systems from a number of DoS attacks launched against them. More specifically the work focuses on the development of an Intrusion detection (IDS) system which provide second line defense to network systems. IDS are very crucial elements in defense of IoT systems alongside with other protection and mitigation measures. Several works have been carried out to provide IDS systems which detect DoS attacks targeting SDIoT systems. Yet, still a lot remains in such regards.

The IDS solutions implemented thus far have used various techniques, and have different architecture and scopes. In this research work, a machine learning (ML) based centralized network IDS systems using CICIDS2017 dataset is developed. The CICIDS2017 dataset has been prepared by the Canadian institute for Cyber Security (CIC) in response to the demand for latest and relevant dataset for intrusion detection works. It has grasped modern attacks including various DoS attacks and traffic behaviors and has attracted the eye of researchers concerned in security works.

Several experiments have been conducted using Decision Tree (DTree), Random Forest (RF) and Support Vector Machine (SVM) estimators in the work. These estimators are trained upon the aforementioned dataset. Dimensionality reduction, parameter tuning and the principle of cross validation have been incorporated to have better efficiency, better detection performance, and also to avoid overfitting. High accuracy results have been recorded in the various experiments. Moreover, a new high detection accuracy of 99.967% have been achieved using an RF estimator with twenty ensemble trees.

1.2 Motivations

The human being has benefited and has also lost from use of different computer technologies. There have often been intense arguments between the supporters and the opponents on the adoption of various technologies. Issues like loss of memory, depression, radio frequency radiations and other health problems are often mentioned to be caused or intensified by the use of digital computing and communication devices. These issues can exist in IoT in a more complex fashion if several issues are not handled well.

It is upto future researchers including the researcher of this work to have a firm understanding of the impacts of a technology at first and then determine, in a better way, to bring the best of a technology. In addition, leaving out the possible negative impacts, IoT is really an awesome technology to be part of. This is the main aspiration of the researcher behind his focus on IoT systems. The focus on security work mainly accounts for the inclination of the researcher to the Cyber security competitions and the seriousness of the issue of security in IoT and other network based systems.

1.3 Statement of the Problem

With IoT is expected a smarteneing of the human lives from different perspectives. Yet this vision of IoT will not come with ease and comfort. Rather a milliard of challenges have to be addressed and lots of vulnerabilities shall be filled up. There are different open issues that make realization of IoT systems very difficult. One prime challenge that faces these systems is the issue of security. Security challenges of these systems are coming more prominent now than that were in traditional Internet system [3]. This issue of security is an extremely sensitive requirement in IoT. The authors in [14, 15] noted the danger of using IoT systems before addressing the existing security vulnerabilities. Several security works are out there to defend these systems against various kinds of security breaches. Yet what has been done so far is at a grass root level for these systems [16].

SDN provides lots of conducive opportunities for IoT systems. And the integration of SDN and IoT is expected to be a potential feasible solution to enhance the management and control functionalities of the IoT network [7]. Many other advantages are expected to be leveraged from this network paradigm including support for security. However, SDN itself comes up with a number of vulnerabilities. These vulnerabilities coupled with those in IoT systems, escalate the attack surface of SDIoT systems for adversaries to launch different attacks. One class of such attacks are the DoS attacks which are concerned with denial of availability of an IoT service to its intended use.

As it is put in [11], these attacks are widely used cyber-attacks. The attacks are ever increasing [17, 18]. This rise is mainly accounted for the enormous integration of poorly secured IoT devices to the Internet, which in turn are recruited and used for botnet attacks [15]. The vulnerabilities found within devices, communication links and communication protocols can be used by these attacks to deny service to the intended users.

Starting from seldom inconvenience to users, lots of devastating consequences can be imposed by these attacks. They can take the capitals, resources and even lives of human being [12]. The fatal consequences might be easily encountered in environments like health services and vehicular traffic services, where a minimal service unavailability is highly risky. The stealth nature of DoS attacks, the significant loss it incurs with a minor attack operation, the very constrained environment of IoT systems, and an extended vulnerable space granted by SDN systems, among others, make SDIoT systems a conducive area for DoS attacks.

Several works have been done to address the losses incurred due to a launch of one or more DoS attacks in these SDIoT systems. Yet, as has been stated earlier, a lot remains in such regards. This research work aims to come up with a solution that defend SDIoT systems from a number of DoS attacks. Several IDS systems have been developed for IoT systems.

One technique of IDS development is the use of ML algorithms that use a reference dataset for training and validation. Different datasets are out there for use for ML based IDS systems. KDD'99 and its revised version NSL-KDD, CAIDA, DARPA, LBNL, ICSI, MNIST, CIFAR-10 and other datasets have been used in the literature for DoS experimentation [19, 20].

The developed works have achieved various detection results which in some circumstances can reach a detection rate (DR) of 100% [19]. However, even if the results achieved are very high, the solutions are ineffective when deployed for real cases. It is mainly because most of them use the outdated and expired datasets [20, 21]. Consistent performance evaluation challenges are encountered by IDS systems caused by these unreliable datasets. This unreliability of the datasets emanates from the fact that the datasets don't represent the current actual attack traffic behavior. Works that are based on realistic and recent dataset are deemed very important.

Considering the insufficiency of the legacy dataset for developing IDS systems, CIC has developed new datasets that fill this gap. The CIC-DOS, CICIDS2017, and CICIDS2018 developed by the institute have attracted many recent security researchers [22]. This research work focuses on the CICIDS2017 dataset for its emerging popularity. The dataset expresses a more realistic network scenario, which includes normal traffic mixed with high-volume and low-volume malicious traffic with sneaky behavior, such as slow application layer attacks [21]. This dataset contains relatively new attack types [23]. Several IDS works are implemented using this dataset.

However, even if works that use the CICIDS2017 dataset exist, many of them are not specially designed to provide effective and efficient IDS systems against DoS attacks. Most of the research works conducted using this dataset are driven by other motivations which can be providing online detection, [24], enhancing the performance of some ML classifier, [25–27], checking the efficiency of some feature selection algorithms or meeting other related objectives [22, 23]. Some others aimed to provide anomaly detection [28].

Due to such reasons most of the works have achieved lower efficiency and detection accuracy in many circumstances. This work specifically aims to design and develop an ML based detection framework having high detection accuracy. In addition to detection performance, a faster response is badly needed in many IoT service scenarios. And thus, this work is also concerned in addressing this timing performance requirement by reducing the number of features using some kind of feature selection mechanism.

1.4 Objective

1.4.1 General Objective

This research work aims to bring an effective security solution that is used to defend various kinds of DoS attacks against SDIoT systems. More specifically, it aims to develop an effective logically centralized signature based IDS system. A performance increase of the IDS solution in terms of detection accuracy, processing speed and processing overhead is the main focus of the work.

1.4.2 Specific Objectives

Starting from analysis of the problem domain to the final solution development and verification tasks, there are a number of activities to be executed. Various kinds of analysis, design, implementation, optimization and other related tasks expected of a research work focusing on IDS development are executed. This work specifically aims:

- to analyze and design an SDIoT architecture
- to analyze, collect and customize a dataset
- to design an IDS system architecture
- to select, train and test an ML estimator for classification
- to analyze and optimize the performance of the ML model built

1.5 Scope of the Research

IoT is a very generic term. Unlike many of the networking technologies, it doesn't refer to a single technology or a certain specified functionality. Rather, it is a collective term used to refer the massive integration of smart devices to the Internet to provide anytime, anywhere, and anything networking paradigm. Thus, no single

work can defeat the hassles of the whole IoT system and provide a comprehensive solution for it. The scope of a work shall be delimited well using some kind of criterion. In this work, the most common and generic IoT architecture which bases on a gateway or resourceful devices is selected for study. Such a condition has also been used under [10] and by other works when making a survey of IDS systems designed for IoT. Under such a condition, securing the IoT system is reduced to securing the gateway of the IoT network.

It is not only the concept of IoT which is broad to study in this research work. The scope of a DoS attack is also very wide since there are a number of DoS attacks having very unique characteristics requiring special focused approach. And it is difficult to secure systems from the whole DoS attacks in a single work. A limited set of DoS attacks shall be selected for study. Such selection of DoS can be made based on the protocol stack the attack bases on, the behavior of the attacks, the severity level of the attacks, the DoS attacks supported in a specified dataset or other measures.

The DoS attacks that have been considered in this work are the one that are included in the CICIDS2017 dataset. These attacks are the DoS Hulk, Slowhttptest, Slowlories, DoS GoldenEye attacks, Heartbleed attacks and DDoS Loit attacks.

1.6 Significance of the Study

High availability of service delivery is badly required in many IoT scenarios. It is a pleasure for attackers to make these systems down. In addition, the rise of IoT has significantly benefited attackers to launch DoS attacks since a large number of less secured IoT devices create an ideal place for use of a system as a botnet. This research work provides an IDS system that detects a set of DoS attacks that emanate from adversaries with different kinds of interest. This security work keeps IoT systems safe from certain kinds of DoS attacks directed towards them or the otherwise emanating from them. And it can have a big significance in a move towards realizing IoT systems as the vision ahead. The other benefits that this research work can provide to the broader dimensions of the human life can be anticipated by extrapolating what a secure IoT system can bring.

Even if this study is focused on enhancing security of SDIoT systems, the solution developed can have a broader impact on different network based systems. The schemes utilized can have a contribution in securing non IoT based SDN systems as well. Cyber Physical Systems (CPS) based application scenarios, governmental and non-governmental institution networks, the mass media, and others can benefit from such a work in one way or the other.

1.7 Contributions of the Work

The following contributions are provided by this research work:

- the work provides an effective IDS system with a high detection performance particularly suited to defend the tremendously increasing DoS attacks.
- the work provides an efficient systems in terms of memory use and processing time which is essential in real time IoT application scenarios where real-time processing is badly needed.
- unlike many other related works, this work provides an in depth parameter tuning and feature selection works and shows how effective performance improvement can be provided using parameter tuning and feature selection.
- it shows the need and opportunities granted by SDN systems especially for IoT and other related emerging technologies.

1.8 Thesis Organization

This research document is organized as follows: Chapter two gives background information about IoT, SDN, DoS attacks and IDS systems. Chapter three presents a review of related works. Chapter four presents detailed methods and procedures used in the development of the research work. In chapter five is presented the results acquired from the experiments and their discussions. Finally, the conclusion of this research work and tips pointing some research gaps related to this work are presented in chapter six.

Chapter 2

Background

This chapter presents background information related to this research work. More specifically, basic background information about IoT and its applicable areas, SDN and its advantages, DoS attacks and their classification, and finally IDS systems and their classifications is presented.

2.1 IoT

IoT is an extension of the Internet. It can easily be defined as simply the time by which the number of people connected to the Internet are exceeded by the things or objects [6]. In the normal Internet the people all over the world interconnect and intercommunicate to get various kinds of advantages. The advantages that one can earn from the Internet might be getting certain information, accessing and attending tutorials from abroad, making online payments, chatting with peers, watching and accessing movies and other videos, getting remote health assistance and obviously a lot more. The idea behind the IoT system is, if people can interconnect and intercommunicate to leverage different kinds of advantages over the web, then why not objects are incorporated to the current Internet system to provide more services and help to the human being?

In IoT, the human to human interconnection and intercommunication in the conventional Internet is extended to include things or objects. Thus, in IoT, in addition to the human being, things can communicate with each other and with the human being to render more services than that can be delivered by the normal Internet. The term “things” that is used in IoT can refer to any real world objects that are faced in our day to day lives which can be the televisions and refrigerators in our homes, our clothes and shoes, our cars, the covers of different medical or industrial products, our electrical switches and bulbs, the ovens, thermometer and other devices used in weather applications, the mechanical ventilation and other devices used in health institutions, the surveillance camera used in highways or the camera that are thrown deep into oceans and the sea for documentary works, the traffic light in the town and many other physical objects. In addition to these physical devices, the things in IoT can also include virtual objects like software processes, objects in OOP based programs, database, processes and other related entities found in our computer systems [1].

IoT is a very broad and generic term. It refers to a collection of diverse range of devices, enabling technologies and systems working together to provide the goal of providing network connectivity from anywhere, anytime and anything. It encompasses enabling technologies like Local Area Networking (LAN), Wide Area Networking (WAN), Cloud systems, Radio Frequency Identification (RFID), Near Field Communication (NFC), CPS, Adhoc Networks, mobile computing, IPv6 over Low Power Wireless Personal Area Networks (6LoWPan), and others. These and many other technologies, service contexts, and applications shall integrate to realize IoT as is expected.

2.1.1 Architecture of IoT

A three-layer architecture is followed in a traditional IoT network [16]. The first layer is the physical layer which is also known as the perception layer. It is responsible for collecting data from the environment with the help of sensors. This layer is also responsible for changing the mood of the environment with help of actuators. The second layer, the network layer, comprises the network elements and protocols essential for transmitting data from one IoT device to another through a network communication link. On top of this network layer is



Figure 2.1: Common architecture of the traditional IoT system

the application layer which is responsible to carry out user's demand by using the underlying network and perception layers [16]. Figure 2.1 shows this common architecture of IoT systems.

2.1.2 IoT Applications

IoT is envisioned to make the life of human beings smarter and better than have been ever before. And its application area encompasses a diverse range. Starting from our bodies or homes, to any place we go lots of IoT services can be and will be encountered if everything goes as the vision set. Even if IoT is still at its infancy, it has already seen some kind of deployment in different sectors. Some of the application areas in which IoT is seeing a deployment include smart home applications, health care, smart grid systems, smart transportation and other smart city services, smart industry applications and others. An overview of the tasks that can be rendered in these application areas is presented below.

Home Automation: This service of IoT is concerned with automating and smartening of household activities like the lighting system, the microwave or heat system, energy saving works and remote control of household appliances like the TVs, refrigerator and others.

Health Care: Chen et al. in [3] stated the applications of IoT in smart health care as intelligent drug/medicine control, hospital management, collection and analysis of human physiology and medicine parameters, and remote medical service for family and community. These and many other services can be executed utilizing the advantages earned from IoT.

Smart Transportation: The smart transportation system aims to smartening the transportation system in a milliard of ways. Traffic information collection and dissemination, traffic guidance and control, vehicle monitoring and highway coordination, remote vehicle monitoring and control are some of the services involved in smart transportation system.

Smart City: Smart city application consists of a set of services that makes a city smart. These services include smart environmental protection and monitoring which deal with pollution source tracing, water, air and sound pollution monitoring, and other environment related activities. It also includes activities like social security works, emergency protection, monitoring and notification. Food traceability and smart agriculture works can also be included in smart city applications [3].

Smart Grid: As Chen et al. in [3] has stated the services under this category include monitoring of power facilities, smart substations, automatic power dispatch, smart power, smart scheduling and remote meter reading services. These and many other functionalities can be rendered by utilizing an IoT system to smarten the power grid system.

Industrial Applications: Some functionalities of IoT in industries can be smartening of the production lines found in the industry, smartening the process control, waste treatment, product life-cycle monitoring, and other related activities.

The aforementioned application areas are only some exemplary service scenarios. In IoT, the service dimensions and case scenarios to be addressed are very wide and almost limitless.

2.2 SDN

In parallel with the IoT, an equally evolving discipline in networking is SDN. SDN is a networking paradigm which is characterized by its focus on the separation of the control plane from the data plane [6]. A broad range of functionalities are rendered and executed in a network system. From a broader perspective, these functionalities are grouped under two main categories. The first category contains the forwarding of packets or data from one point or node on a network to another. This is what is called the data plane service. The other functionality is concerned with the control of such forwarding of data. This is what is normally called the control plane functionality in networking.

The control plane service is concerned with the execution of a number of tasks which include determination of the route to be followed by packets, formulation of flow action rules, updating route information, execution of network configuration and management tasks, and others. In the traditional networking the data plane and control plane functionalities are collocated in a single device. Thus networking devices like routers and switches are responsible to forward data from one port to another and at the same time to execute certain kinds of control functionalities. Due to such collocation of the control plane and data plane functionalities, a number of challenges and problems arise in the traditional networks.

Some of the problems of collocation of the two planes include scalability and interoperability issues, difficulty in network configuration and management, less flexibility, security issues and others [4, 5]. SDN is expected to solve the above issues by separating the two network functionalities. Thus, in SDN, the two planes are found separated from one other and there are devices dedicated to render data plane functionality known as SDN switches and also there is a dedicated unit called the SDN control unit which is responsible for all control functionalities. The functionalities of the intermediate network elements like routers and switches are simplified to mere forwarding of data. Programmability is a key principle in SDN systems and the network is made programmable by using some kind of programming language (PL).

2.2.1 Architecture of SDN

The architecture of SDN system is also composed of three layers [6]. This architecture is shown in Figure 2.2. The bottom layer is the data plane layer which consists of SDN devices that are responsible for forwarding of data from one port to another. OpenFlow (OF) switches are the main components here in the data plane. The switches forward data as per the instruction received from the control unit.

The second layer is the control plane. It is the intermediate layer of the SDN system. In this layer is found the SDN controller which is also known as the “the network operating system”. As has been discussed in earlier, the controller is responsible to formulate various control rules and pass these flow rules to the underlying switches in the data plane. The controller may request flow statistics information from the switches or in the other direction switches might request the controller for a flow control decision to be made and may send statistical information when certain conditions prevail. This intercommunication between the two layers is accomplished via certain protocols or APIs [6]. The OF protocol is the dominant protocol utilized for data plane and the control plane communications.

On top of the control plane layer is the application layer. The application layer is responsible to render services or customer requirements. These applications are communicated with the controller using some kind of North-bound interfaces like the RESTfull API [6]. These applications are just normal software programs which can be developed using any of the languages supported by the controller.

2.2.2 Advantages of SDN

As has been mentioned earlier, SDN provides different advantages for various kinds of systems that depend on networking. These systems can be cloud systems, data centers, campus networks, IoT systems, Wireless Sensor Network (WSN), and many more. Over one hundred of major companies have already deployed SDN based components in their systems [4]. And these companies mention that they have got various advantages by leveraging SDN in their networking systems. But, the report on the types and levels of advantages earned from SDN vary from one

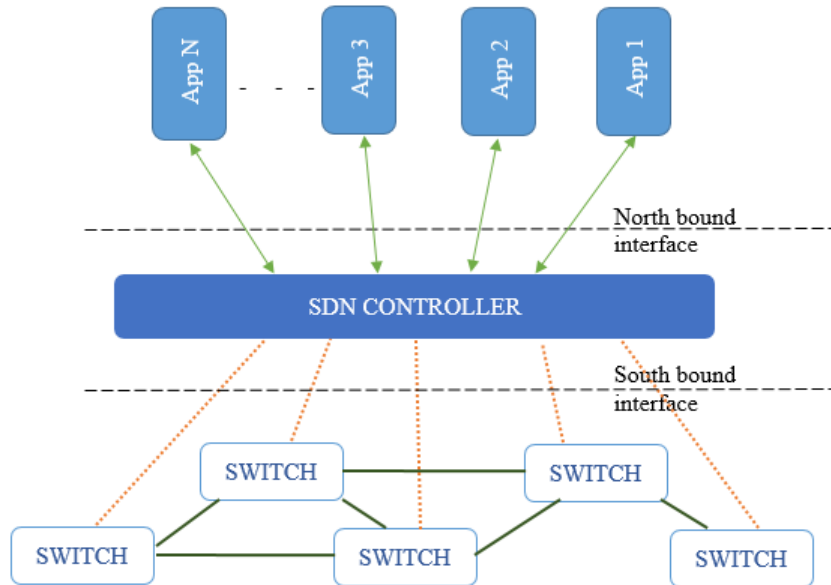


Figure 2.2: Architecture of the SDN paradigm

company to another. In general, the interest of companies and organizations for SDN lies on earning one or more of the following advantages [4, 8].

- Flexibility
- Inter-operability
- Easier network configuration and management
- Scalability
- Reduction of CAPEX and OPEX costs
- Quality of services
- Security

2.3 DoS Attacks and Defense Mechanisms

DoS attacks refer to a collection of attacks that make network devices or overall network systems unavailable to their intended users. These attacks achieve this disruption or complete denial of service by taking advantages of different vulnerabilities. These vulnerabilities can be found in the implementations of the various

protocols and technologies of the TCP/IP protocol stack or by creating deformed packets that are intended to crush servers and other units which are not able to process such deformed packets. These attacks might also use extremely enormous floods of “legal traffic” to meet their destructive objectives.

2.3.1 Motivations for and Trends of DoS Attacks

DoS attacks target different kinds of entities ranging from personal home users to different governmental institutions [29]. The target entities might be gaming and stock exchange sites, political organizations, mass media, banks, transport and other organizations. And as can be projected from the range of the targets, the motivations behind these DoS attacks vary. According to [29], there are five main reasons that are behind the launch of such attacks. These are financial or economic benefit, revenge, ideological beliefs, intellectual challenge and Cyber warfare.

Starting from the late 90’s the effects of DDoS attacks have come to be significant. And no matter what the intentions are behind, the trends show that these DoS attacks are increasing from time to time [17, 18]. A very catastrophic network attack that targets the Dyn domain name system (DNS) servers has been recorded in October 2016. It had very devastating consequences on many major organizations which include the New York Times, Amazon, Twitter and Netflix [30]. The attack is a DoS attack which utilizes IoT devices as botnets resulting in flood traffic of magnitude 1.2 Tbs.

2.3.2 DoS Classification

As has been stated earlier, there are a number of attacks that are responsible to deteriorate or to completely deny the normal functioning of network devices and/or overall network systems. To understand such a vast concept consisting of enormous and somewhat complex components, the familiar and recommended approach of the divide and conquer principle shall be used.

Various criteria can be taken to classify DoS attacks. And DoS attacks might be classified as simple DoS attacks or DDoS attack by considering the number of

devices employed in the attack. In simple DoS attacks, the attacks are launched from a single device whereby in DDoS attacks a number of devices cooperate to cause a successful service denial. DoS attacks might also be classified based on the TCP/IP protocol stack layer in target. Taking such a criterion, the attacks can be referred to as application layer DoS attacks, transport or network layer DoS attacks, mac layer DoS attacks and physical layer DoS attacks. They can also be classified by considering the network functionalities in target. And some more metrics can be used in classification of DoS attacks.

A fully covered and a simpler taxonomy of DoS attacks is made by [29]. In this survey work DoS attacks are classified in to four categories by considering the impact or goals of the attacks on the victim's network resources. These categories are also further divided into different other sub-categories by considering the mechanisms utilized to achieve the attacks.

The first category of DoS attacks put in this taxonomy is resource depletion attack. The aim of these resource depletion attacks is to exhaust the capability of network devices by overflowing the main resources used like the memory, processing unit and the like. The second category of DoS attacks is the bandwidth depletion attacks. Such attacks aim to overflow the bandwidth, an essential resource of a network, and then eventually disrupt the normal functioning of the network system. Infrastructure attacks are a collection of attacks found in the third DoS attack category. They are the most catastrophic type of DoS attacks [29]. These attacks aim to destroy the functioning of core elements of the Internet like the DNS servers. Such attacks eventually destroy as many network services as possible. DNS flooding attack is a famous infrastructure DoS attack. The fourth type of DoS attack category is the so called zero-day attack. It is a collection of attacks with no knowhow of the vulnerabilities and the attack strategy to be followed prior to their initial launch, that are called zero day attacks. According to the 2017 Symantec Internet Security Threat Report, more than three billion zero-day attacks were reported in 2016 [31]. Detail information about such attacks can be found from the survey work mentioned earlier [29].

2.4 IDS Systems

IDS systems are second line defense mechanisms used in defending network based systems from various security breaches. They are used for detecting the existence of any intrusions or anomalies within a system. The detection results are further used by some kind of response mechanisms for complete security. IDS systems are considered indispensable and mandatory components in system security [32].

However, development of an IDS system is a challenging task since flash events have to be told apart from flooding attacks and stealth malicious flows must be identified from normal traffic flows well. Though the main aim behind IDS systems is detecting attacks, a variety of IDS systems exist based on the problem on hand or the approaches utilized to develop IDS systems. A bit elaborate discussion of IDS system categories is made below to have a better understanding of them.

To start with, IDS systems can be categorized into three as signature based, anomaly based or hybrid systems by considering how IDS systems are developed. In signature based systems, prior known attack traces are used as a reference to the detection of an attack. In this technique, a matching of attack patterns is made. A current traffic flow pattern is compared against a previously configured attack signature stored in a database to check whether a match exists or not [33]. This approach results in a high detection accuracy and a low false alarm rate. But, such a technique can only detect attacks of which their signatures are known and stored.

Unlike signature based systems, anomaly systems make their bases on identification of deviations from normal network activities [28, 34]. Any deviations from what is said to be normal is considered anomaly or malicious. These anomaly based systems are effective in detecting novel attacks whose signature is not yet known. However, such systems have high false alarm rates (FAR) and lower detection rates (DR) when they are compared with the signature based approaches [33]. The third IDS system, hybrid systems, merges the benefits of the two systems and hence results in better optimized results.

IDS systems can also be categorized as host based IDS (HIDS) or network based IDS (NIDS) on the basis of the location from which these systems get data for

traffic classification. The HIDS make detection based on data collected from a host device [31]. Auditing logs from the OS, server logs and various application logs can be used for such a purpose [10, 31]. Unlike the HIDS, NIDS systems have a wider scope and are based on data from the whole or a certain portion of the network traffic. Packets traversing a network can be sniffed at border gateways or other intermediate points selected for traffic capturing.

IDS systems can also be centralized or distributed based on the way the IDS system makes data collection and detection. They can also be grouped as passive vs active, offline vs online [35]. Still other classification of IDS systems might be possible. In this research work it is aimed to development a logically centralized signature based NIDS detection system.

2.4.1 Requirements of IDS Systems Designed for IoT

Normally conventional IDS systems might be functional in IoT environments. More specifically, in ML based IDS systems, similar methods and datasets can be used for development of IDS systems for IoT [20]. However, these IDS systems designed for conventional networks might not be fully suited to smart IoT environments. This is due to the reason that somewhat strict requirements are expected of conventional IDS system to be fully suitable for use in IoT. Elrawy et al. in [10] has emphasized that higher detection accuracy, low false positive rate, low energy consumption, fast processing, low performance overheads are badly desired to be satisfied by IDS systems proposed for these smart environments.

Chapter 3

Literature Review

A comprehensive study of the literature concerned in the interest areas of this research work has been conducted. The literature review is organized as follows. It starts with a high level discussion of security issues prevailing in IoT systems. Then security issues of some component IoT systems and the reasons why such works can't fit to the larger IoT system is made. Such a discussion is followed by a review of some selected IoT related security solutions which emphasize on defending DoS attacks aiming IoT and SDN systems and their integration. This work being an ML based IDS solution, more focus has been given on the review of ML based detection works especially the one using the CICIDS2017 dataset for training and validation.

3.1 IoT Security

As has been stated in the introductory sections, the enormous significance IoT is expected to bring is highly challenged by the security issues prevailing in it. The heterogeneous, constrained and highly scalable nature of IoT systems have challenged and frustrated IoT security researchers. And there are arguments on whether leaving the idea of IoT altogether or spend the best effort possible and realize secured IoT systems. Many researchers sort out for the latter option despite the existing challenges are despairing. The authors in [14–16, 36–40] made a

survey on IoT security. These works were accomplished with an objective of identifying various security vulnerabilities in IoT systems, exploring existing security research works and eliciting the gaps required to be filled. Availability, confidentiality, integrity, authentication and other security issues found in the normal Internet systems also prevail in such systems, but this time rather intensified by the aforementioned nature of IoT.

Deogirikar et al. in [14] have given a comprehensive survey of the different attack vectors that exist in IoT and discussed the solutions implemented to secure these systems. Oracevic et al. in [36] presented a survey of security works related to IoT. It discussed up-to-date IoT security solutions. Different techniques have been followed by the solution works surveyed. Some of them are cryptography based mechanisms which utilize methods like ICMetric, ECC and RSA. Some others are architecture based working on some aspects of the IoT architecture. Hardware based security primitives used to secure IoT systems are also discussed in this survey. Some of the solutions are motivated by some extra principles like the human immune system. But, the work was not detailed enough and also it didn't systematically organize the solutions discussed to render a comprehensive insight. The authors in [15, 16, 38–40] have shown the different challenges and prospective measures used in securing IoT systems.

3.2 Component IoT System Security

IoT is a generic term that indicates a massive interconnection and intercommunication of possibly any devices to the Internet. To bring the anything, anytime, and anywhere interconnection scenario of IoT, several technologies shall be incorporated. Securing IoT systems is in one ways or the other related in securing such component systems and devices. Yet, the security solutions utilized for the component systems doesn't escalate for the more generic IoT system for different reasons [5, 41]. In the next subsections of this section, a discussion of security works developed for two IoT component systems and the reason why IoT demands a broader spectrum is presented.

As has been stated earlier IoT is not an independent and a standalone technology. WSNs are one prime components in IoT systems. And lots of works have been

carried out in securing these systems [5]. However, the security works developed for WSN can't be used in IoT. This is mainly due to the fact that WSN are confined to a certain LAN and the challenges of the broader Internet is not its issue. Therefore, solutions aimed for WSN are only suited to one section of an IoT network.

Another component system in IoT are the CPS. These component systems are networks of various sensor and actuators with a computer equipped networked control system. There are a number of research gaps in the development of these systems. Like the other network based systems, security is one of the research gaps in need of effective addressing in these systems [42]. CPS systems arise from the control system but the wide incorporation of IP enabled devices within such systems is making the difference between an IoT system and a CPS system blurred [41]. Even if different similarities and intersections exist with IoT, such systems lack the characteristics of IoT. One thing is it's not a must for CPS devices or system to interconnect to the larger Internet which is a prime requirement in IoT system. And as its bases and focus is on control systems, works in CPS might not be directly applied or might not have a big relevance for IoT security which is more focused on connectivity and communication [41].

Such an intersection with IoT is not limited for WSN and CPS systems, rather there are other systems and technologies which follow a similar argument with it. And a special focus shall be made on securing IoT systems by considering several issues that exist within its environment.

3.3 DoS Attack Mitigation

In IoT systems, lots of attacks are launched to deprive privacy, integrity, availability and other security requirements. These security issues have to be addressed to bring a sound IoT system. One such a hot challenge in the security of these systems is ensuring high availability of the systems. Some security researchers have focused on defending attacks aimed to deprive availability of IoT systems. This focus on availability, is accounted mainly for the ever increasing trend and possible devastating consequences of DoS attacks.

Several researches have been made to defend IoT systems from DoS attacks [19, 30, 43–47]. The researchers used several techniques in developing the solutions. For better understanding of the solutions used to defend systems from DoS attacks the overall techniques utilized can be categorized and studied into two groups. These categories are protection measures and mitigation measures. A discussion of these categories and the techniques that can be used within them is discussed below.

Protection measures aim to protect systems from different kinds of DoS attacks altogether. Techniques such as load balancing, use of secure overlays, use of honeypots, awareness based mechanisms and filters are used as means of protection [29]. One such protection measure was proposed by [45]. In this work a mobile edge computing based protection framework which deploys smart filters at the edge of attack-source/destination networks using the power earned from the edge devices was designed and developed. An emulation network with SDN and NFV capabilities was constructed for experimentation and found an improved accuracy and detection rate. Different other protection based approaches were also proposed and developed [48].

Like the other protection measures used in various engineering problems, it is costly and very difficult to protect network systems from DoS attacks. That is why there are mitigation measures utilized in defense of systems from such attacks. The mitigation mechanisms aim to lessen the effects of the attacks after they prevail in the network. It has attracted a huge number of researchers. Some exemplary mitigation works can be get in [19, 20, 30, 33, 43, 44, 46, 47, 49]. The mitigation mechanisms can further be categorized as detection measures, response measures and tolerance measures. Since this research work focuses on detection, an extended discussion of IDS systems is given below. The discussion of the other mitigation mechanisms is not provided in this work.

3.4 More on IDS Systems in IoT

IDS works are one part of mitigation solutions. Several works developed IDS systems that are expected to run on a range of IoT scenarios [12, 43, 44, 46, 50]. These works considered a generic network structure and didn't specifically consider a certain service scenario. Different other works targeted specific IoT application

scenarios like smart home applications [44] smart city services [51] smart industrial systems [52] and other specific cases. The specifically developed IoT solutions are significant in many respects since they provide more emphasis on the basic requirements and network traffic characteristics of such systems. However, many of the works consider customized small networks with self-generated attack traffics. This issue makes it difficult to anticipate how the system performs to real attacks since they are not validated with benchmark datasets. One IDS system specifically developed for a specific IoT scenario was presented in [44].

Anthi et al. in [44] provided an interesting IDS system that detects several DoS and other Cyber-attacks that target a smart home IoT application. It implemented a test-bed that consists 8 IoT devices commonly found in a smart home system like Smart TV, NetCam Camera, Smart Lamp and other related devices and sensors. A 3 weeks long benign data collection and a 2 weeks long attack traffic data collection was made for use in the IDS development. The number and type of devices are selected based on the statistics from Cisco's VNI report. The results achieved have higher performance with 96.2%, 90.0% and 98.2% F-score. Though a higher DR was achieved, the results would have been increased using various techniques like the use of recursive feature extraction techniques, exhaustive parameter tuning or related mechanisms from ML rather than just using simple entropy and information gain (IG). Moreover, the dilemma on whether these and other related security systems will adequately defend real Cyber-attacks is a question since the dataset prepared lacks conformity with the criteria required by a reliable dataset [32].

In addition to the application scope of the IDS systems, the techniques used in implementing detection systems also vary. ML, statistical approaches or other knowledge based techniques were utilized throughout the literature to implement such systems [31]. The statistical approaches are based on statistical measures of various packet and flow parameters. The mean, median, mode and standard deviation measures can be used for such a purpose [31]. A univariate approach which focuses on a single feature, a multivariate approach which focuses on statistical measures of a combination of features, or time series approach which makes observations in a given time interval can be used.

One statistics based detection system was implemented in [53]. In this work a novel method of statistics collection based on sFlow was proposed for eliminating

the OF related constraint of difficulty in having aggregated flow statistics. The data collected using the sFlow flow monitoring mechanism was then periodically sent to the anomaly detection module at 30s interval to provide a near real time detection. The anomaly detection algorithm was implemented as a NOX application conducting periodical entropy-based calculations. An evaluation of the IDS using a production traffic collected from a university campus was made. A 100% detection accuracy and 23%, 39.3% False Positive (FP) rate with the native OF implementation while a similar detection accuracy with a FP of 50% was achieved with sFlow accounting the loss due to sampling. Portscan and Worm propagation attacks were handled in addition to DDoS attacks in this work.

Another statistic detection system is found in [54]. This work provided entropy based measure of intrusion detection. A windows size of 50 packets was chosen and the destination IP address of an incoming packet was selected for entropy calculation which in turn was used for detecting anomaly. A DR of 96% was achieved in this work. Though the work provided a simple and flexible detection system using only the randomness of the destination IP address, more combinations of attributes can be made and used to get higher DR values.

ML based works are being widely used in IDS system development [26]. Both supervised and unsupervised approaches can be used. One ML based work is found in [19]. In this work, a two stage Artificial Intelligence (AI) based IDS system empowered by the global view of SDN technology was proposed and implemented targeting IoT networks. Selection of important features was carried out by leveraging the Bat algorithm with Swarm division and differential mutation algorithms. The work has achieved a higher convergence over other Swarm intelligence algorithms. In addition, the work has got a higher detection performance in different attack classes reaching a DR performance of 100% for DoS attacks. However, it is based on KDD dataset which is outdated and expired for use as an evaluation dataset for IDS works [21]. In addition, testing didn't consider cross validation (CV) and/or other related principles that are significant in avoiding overfitting.

Several other ML based IDS solutions are developed so far. Examples of them can be found in [20, 23, 44, 55]. When working with ML based solutions, the selection of a dataset is an important issue to be addressed well. It has been investigated that real datasets have been used extensively for evaluation and validation of DoS related research recently [56]. A real dataset is a data collected from a real network

system running real operating systems, real or more realistic network applications, and other related platforms. Accounting their appropriate nature for validation and also their simplicity to handle, these real systems alongside with simulation based datasets are shifting the attentions of researchers [56].

Different datasets are out there for use in an ML based IDS system. KDD'99 and its revised version NSL-KDD, CAIDA, DARPA, LBNL, ICSI, MNIST, CIFAR-10 and other datasets have been used in the literature for DoS experimentation [19]. The developed works have achieved various detection results reaching to a DR of 100% in some circumstances [19]. However, even if the results achieved are very high, the solutions are ineffective when deployed for real cases since most of the datasets mentioned above are outdated and unreliable for use [21].

Considering the insufficiency of the legacy dataset for developing IDS systems, CIC has developed new datasets that fill this gap. The CIC-DOS, CICIDS2017, and CICIDS2018 developed by the institute have attracted many security researchers. This work focuses on the CICIDS2017 dataset. The dataset expresses a more realistic network scenario, which includes normal traffic mixed with high-volume and low-volume malicious traffic with sneaky behavior, such as slow application layer attacks, [21]. This dataset contains relatively new attack types [23]. Lots of IDS works were implemented using this dataset.

One very important work which used the CICIDS2017 and the other recent datasets from the CIC was provided by [21]. The work designed and implemented an on-line DoS/DDoS attack detection using RF, AdaBoost, DTree, stochastic gradient and other ML algorithms. The detection work used the three well known datasets CICIDS2017, CIC-DoS, and CSE-CIC-IDS2018 and also had prepared its own customized dataset. It achieved results reaching up to 99.93% accuracy for a customized dataset. Its DR and PREC values for the CICIDS2017 dataset are 80% and 99.2% respectively. The work also had succeeded in reducing the number of features to twenty-eight with a reasonable accuracy using RF and then succeeded in reducing the features further to twenty features using its own new algorithm for feature selection. Several additional experiments were carried out in this work to calibrate and evaluate the system by adjusting the sampling rate, minimum flow table length and maximum flow table length parameters. However, even if a very high result were achieved and CV was considered, several techniques could be leveraged and experiments performed to improve the accuracy even more. In

addition, for IoT related works it is very important to decrease the number of features the smallest possible and lower number of features than the achieved number of twenty could be selected with a reasonable accuracy.

Ahmin et al. in [55] proposed and implemented a hierarchical IDS system combining DTree and various rule based algorithms using the CICIDS2017 dataset. It used the entire CICIDS2017 dataset and had collected DR values for five attack types namely DoS Hulk 96.782%, DoS slowloris with 97.758%, DoS Slowhttptest with 93.841%, DoS GoldenEye with 67.571%, Heartbleed 100%. Even if it had used a new hierarchical approach its DR was not that high.

A new detection model based on the LeNet-5 and LSTM neural network algorithms using the CICIDS2017 and the CTU dataset was proposed and implemented in [23]. The LeNet algorithm is widely used in image processing and it was used in this work to extract spatial features while LSTM is widely used in sound processing and it was used in this work to extract temporal features. The work has applied these algorithms for network detection systems and has succeeded in having estimators having an accuracy of 99.91%. The authors in [27, 28, 35, 57–60] have also developed IDS system using this dataset.

As has been stated earlier, CICIDS2017 has attracted many researches. And several detection systems have achieved high accuracy or DR values. But, many of them lack one or more of the following issues; failure to provide an optimum dimensionality reduction, limited parameter tuning steps made which, the otherwise, can enhance detection performance, failure to provide hierarchical or other kinds of estimator combinations, among others are seen in many of the works. Moreover, contrary to the objective of this research work, most of them are generic and doesn't specifically consider to efficiently and effectively circumvent the effects of the ever increasing and costly DoS attacks.

3.5 IDS Systems Involving SDN and IoT

Securing traditional IoT networks are difficult in many circumstances. Elrawy et al. in [10] stated that conventional IDS systems are not suited for IoT smart environments. SDN provides a convenient opportunity to secure IoT systems

in a new way [6, 7]. Contrasting to this advantage, however, a number of new attack vectors are brought by SDN itself. One of such problems arise due to the centralized nature of SDN based networks. This centralized nature coupled with other vulnerabilities attracts adversaries to launch DoS attacks since compromising the controller brings the entire system down.

Accounting this issue and the enormous advantages leveraged from SDN, a couple of detection works were carried out to defend SDIoT systems. Yet, security works considering the integration of SDN and IoT seem missing in many cases. Kalkan et al. in [61] stated that only a few works have leveraged SDN for strengthening IoT security. Kalkan et al. in [61] added that providing security is an important priority for the heterogeneous SDIoT environment.

To develop detection systems in an SDN context, various novel traffic anomaly detection algorithms like threshold random walk with rate-limiting, credit based rate limiting (TRW-CB), maximum entropy detector, and NETAD were implemented [30]. In [46] a novel security algorithm that detects and mitigates DoS attacks aiming to free the constrained IoT environment from such attacks was designed and implemented. The work was based on the payload size of packets and stream of packets to detect an attack. The proposed algorithm's performance was evaluated using a simulation work conducted using the Cooja simulator of the Contiki operating system. Benign packet delivery ratio and malicious packet drop ratio were used as performance evaluation metrics. A malicious packet delivery ratio of about 6 per 100, 16 per 250 attacks, 25 per 300 attacks and other closer results have been achieved. Such and other results show better performance than previously accomplished related works.

Another work focusing on SDN systems was discussed under [62]. In this work a light weight flow based IDS system was developed which resulted in a high DR of 98% and relatively low FAR. Periodic collection of statistical information about the flows was made using SDN switches. Afterwards traffic classification was made using feature extraction and aggregation techniques. The work prepared and used its own dataset using the advantage of flow statistics collection by the OF switches. However, a very simple network was used to collect attack traffic which might result in certain deviations from the ground real traffic flow.

Another work that focuses on the IoT and SDN integration was implemented in [30]. This work tried to exploit the various opportunities provided by SDN network paradigm to detect and mitigate DDoS attacks and provide truly inline anomaly detection. It discussed the importance of the West-East interface of the SDN controllers for inter-domain security, early detection of attacks using source based attack mitigation principle and other related opportunities. Yet, no kind of design and implementation was made in this work.

Summing up, several works have been undertaken to provide second line security mechanisms. And these works have utilized different techniques to make SDIoT systems more secure. However, a due emphasis was not given on the study and utilization of virtualization, parallel computing, extensive parameter tuning and dimensionality reduction in making the security systems more efficient and have higher performance. Moreover, despite many research works were implemented using ML algorithms, most of them used outdated and unreliable datasets. This work focuses on filling some of the existing gaps, more specifically on parameter tuning, dimensionality reduction and cross validation using an up-to-date and popular dataset, CICIDS2017.

Chapter 4

System Development

Like almost all research works, this work has been started by conducting a literature review about the focus area. A research gap has been identified from the review conducted. Afterwards, some kind of analysis about possible solutions that can be used have been done. A detection system is proposed to be developed in this work on account of the enormous significance it has in the defense of a network based system including IoT.

In order to develop the proposed IDS solution, several materials have been collected and lots of experiments have been conducted. ML principles and operations like feature selection, parameter tuning, and cross validation have been considered in the experiments. In the following sections of the chapter, the materials required, the overall steps conducted and the associated reasons behind the selection of these resources and other related principles incorporated in the work are discussed.

4.1 Dataset Selection

The issue of having relevant dataset for high performance IDS system development is a critical challenge in many circumstances. As Sharafaldin et al. in [32] has noted, most of the datasets that have been widely used throughout the literature are outdated and unreliable for IDS development. The CICIDS2017 dataset is suitable in many circumstances. Basically eleven metrics have been set for

evaluating the performance of datasets. These measures are complete network configuration, complete traffic, labelled dataset, complete interaction, complete capture, available protocols, attack diversity, heterogeneity, feature set and availability of meta data. The CICIDS2017 meets all of them [32]. In addition to its conformity to the evaluation standards, the CICIDS2017 dataset is believed to have the characteristics expected of a real-time network traffic [26].

In the preparation of the whole CICIDS2017 dataset, a traffic capture was made for five days starting from Monday to Friday. Twelve attacks of various type are supported within the dataset. DoS attack traffic collection have been mainly carried out on the Wednesday's traffic capture session. Four DoS attack families namely DoS Hulk, DoS Slowhttptest, Slowloris, and DoS GoldenEye attacks have been supported in this session. This session has also a traffic capture for Heartbleed attack. Additional DDoS attacks are launched and the corresponding traffic capture has been made on the Friday's release.

In this research work, this recent and very popular dataset is used as a benchmark for training and validating the experiments conducted. More specifically the whole Wednesday's traffic capture has been selected since it is the main session concerned with DoS attacks. In addition, the DDoS attack that has been incorporated in the Friday's capture has also been considered. A brief description of the six DoS/DDoS attacks families collected during the two sessions is presented below. Table 4.1 shows the class distribution of DoS/DDoS attacks prevailing in the CICIDS2017 dataset.

DoS Hulk: HULK is a short for HTTP Unbearable Load King. It is a DoS attack that targets web servers and which achieve its objectives by flooding the servers with uniquely designed HTTP requests. A single attacker can launch this attack to disrupt a less secured web server entirely in just a couple of minutes.

DoS Golden Eye: it is application level DoS attack tool used to bring websites down. Socket smashing is done until all the available sockets are consumed [63].

DoS Slowloris: it is a simple yet very effective low volume DoS attack. It opens and maintains multiple open connections to eventually break normal connections to the target.

Table 4.1: Total number of DoS/DDoS attacks in CICIDS2017 dataset

No	Type	Total number
1	Benign	440031
2	DoS Golden eye	10293
3	DoS Hulk	231073
4	DoS Slowhttptest	5499
5	DoS Slowloris	5796
6	Heartbleed	11
7	DDoS Loit	128027

DoS Slowhttptest: it is a DoS attack tool used to generate low bandwidth application level DoS attacks. It achieves its goal by using partial connections and slow requests. Slowloris, Slow Http Post and Slow Read attacks are supported by this tool.

Heartbleed: it is an attack that uses the advantage of the vulnerability found in the implementation of the OpenSSL library while implementing the Heartbeat protocol. Though heartbleed attack is not DoS attack, it is a vulnerability that can be comprised to render DoS attacks in the future [64, 65]. Accounting this issue and the presence of only small number of heartbleed attacks in the dataset, detection of this attack has been considered in this work. Moreover, Filho et al. in [21] stated that this attack can be assumed to be a DDoS. It has been considered and labeled as a DoS attack in both [21] and [55].

DDoS Loit: it is another DDoS attack used in the dataset. Little details are found for this attack throughout the literature.

Two customized datasets are prepared from the CICIDS2017 dataset. These datasets are labeled CICIDS2017-Wed1 and CICIDS2017-Wed2 datasets. The CICIDS2017-Wed1 contains the entire Wednesday release of the CICIDS2017 dataset. It has considered the Heartbleed attack for the reason mentioned in a previous discussion of this section.

Even if the CICIDS2017-Wed1 customized dataset has considered detection of Heartbleed attack, it might deviate from the objective of the research since no known cases which make the Heartbeat vulnerability, upon which the Heartbleed attack depends on, is a source for DoS attacks has been found yet. In addition,

even if the DoS attacks are collected on the Wednesday's capture, the traffic capture carried out on Friday has one additional DDoS attack namely the DDoS Loit. The CICIDS2017-Wed2 customized dataset contains the traffic samples of Wednesday's session without the samples containing Heartbleed attack. On top of this part of the Wednesday's release the DDoS Loit attack release of Friday are added to make the CICIDS2017-Wed2 customized dataset.

Hereafter whenever the term DoS is used it refers to the five attacks collected in the Wednesday's release. And whenever a DDoS is used it refers to the DDoS attack launched and collected in the Friday's release. This labeling is as per what's put in the original research paper generating the dataset [32], which is also used in different other works.

4.2 Hardware and Software Requirements

4.2.1 Hardware Resources

The entire work is accomplished on a single computer with the following specifications: Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz processor, 8GB main memory, L1-L3 cache memory, 1TB HDD running a x64 Ubuntu 18.04 operating system.

4.2.2 Software Requirements

In this work Jupyter Notebook and different other software products have been used to develop the IDS model. A discussion of the development software, ML framework and the various data analysis tools or packages used in this work is given below.

I. PL Choice

Several PLs can be used in the development of ML based systems. Python, C++, JavaScript, Java, C#, Julia, Shell, R, TypeScript, and Scala are the top most programming languages used for machine learning development. Among them

Python is the most commonly used PL for ML works and it is selected in this work for development of the detection application.

II. Development IDE

The Jupyter Notebook is a client-server based web application supporting interactive data science program development and presentation. Documents can be created and shared using Jupyter notebook. Data analysis and transformations, graph visualization, numerical simulations, parallel presentation of code units, multimedia documents and explanatory texts, and other related functionalities are supported by Jupyter Notebook. In this work it's used to execute the entire ML tasks and data visualization works. It is selected in this work for its simplicity, and user friendliness.

II. ML Development Framework

To facilitate development of ML works, several frameworks have been developed. Scikit-learn, Tensor flow, Weka, MLib (Spark), are some of the most popular ML development frameworks. These frameworks provide an implementation of a number of well-prepared ML algorithms used for classification, regression, clustering, parameter tuning, dimensionality reduction and other related tasks. Scikit-learn is a Python library built upon Numpy and SciPy. And it is used for development of ML programs. It is popular for its speed and well prepared documentation, high tasks coverage. In this work a recent version of Scikit-learn, version 0.22.1, is used.

IV. Numpy

Numpy is a Python library used for mathematical computing. It supports large, multidimensional array objects and matrices, along with a collection of linear algebra, Fourier transform, and a large collection of other mathematical operations to operate on these arrays.

V. Pandas

Pandas is an open-source data analysis and manipulation tool built to provide a fast and easy to use functionalities. It is developed for Python PL and offers powerful data structures and operations. It is used in this work to provide different

data analysis and extraction tasks.

VI. Matplotlib

It is a Python library used to create and visualize publication quality plots of various types. It is used in this work to visualize the results of the experiments conducted.

4.3 Devising SDIoT Architecture

SDN and IoT can be integrated in different ways. And coming up with a feasible and efficient architecture for SDIoT systems is still an open issue for the researchers. Elrawy et al. in [10] has surveyed and discussed the various proposed architectures for IoT systems and the need to design new architectures. A very common implementation of SDIoT systems is the one that uses gateway devices. In this SDIoT implementation, all the interconnections and communications of the things in the IoT are carried out using a gateway. The role of the IoT things is reduced to sensing and forwarding the data to the gateway or the other way to actuating the environment based on the instructions received via this gateway. In addition to providing Internet connectivity, a gateway device is also used for securing the individual devices and the overall system from different kinds of Cyber-attacks.

4.3.1 Design Considerations

4.3.1.1 IoT Infrastructure Layer Components

There can be different components in an IoT infrastructure layer. The prime one which has been hoped in the future of IoT systems are the elements of a 6LoWPAN network. Basically these elements are the 6LoWPAN IoT devices. The 6LoWPAN devices are sensor nodes with IPv6 communication capability which is made possible by using compressed IPv6 headers. Yet, these devices have very constrained resources to independently make internet-working. The 6BR are routers used to help the 6LoWPAN devices connect with the rest of the Internet by rendering

protocol translation, header compression and decompression, packet fragmentation and de-fragmentation, routing, and related tasks.

Even if they are expected to incur a challenge on future advancement of IoT systems, proprietary IoT devices and gateways using technologies like ZigBee and Bluetooth can be part of the IoT. These devices have their own communication protocols and network architectures.

These technologies can take part in the proposed SDIoT environment by connecting the corresponding gateways to the SDN switches. Self capable devices like IoT nodes are also parts of an IoT infrastructure and can be a part in the same way the proprietary gateways are linked to the SDN.

4.3.1.2 Placement of SDN Switches and IoT Gateways

An important decision in making the architecture of SDIoT systems is determining the placement of the SDN switches and IoT gateways. More specifically, the issue of merging the SDN switch with the IoT gateway or using disparate switches and gateways is somewhat challenging. Some works have used a single unit rendering both gateway and switching functionalities together. While others have used disparate units on which both of them assume similar responsibilities as they used to do in traditional networking systems. The later option have been followed in this research work for the sake of preserving the intent and hence the advantages of an SDN network paradigm. However, slight design modifications of the devices have been made to create more flexible and scalable IoT systems. These changes are made upon the implementations of the 6BR. These design modifications are discussed in the following subsection. Pure non-intelligent SDN switches are used in this work despite slight performance overheads can be an issue.

4.3.1.3 6LoWPAN Gateway Implementation Design

In order to make scalable IoT systems, one prime issue that shall be addressed is the implementations of the IoT gateways. In addition to the scalability issues, there comes a challenge of single point of failure. A distributed implementation of border routers for 6LoWPAN has been made in different works. Ge et al.

in [66] has brought an interesting multiple border gateway architecture which provides revised neighbor discovery, and new updated router selection and routing algorithms. In this implementation of distributed border router, devices are able to efficiently select a better border router for routing. Better route finding for an intranet and internetworking communications is designed and implemented in the work. Detail designs of the work can be found in the original work [66].

The problem with this implementation in the future of IoT is the existence of extended intelligence in the border gateways which results in increased CAPEX costs, increased processing burden and high energy consumption around the nodes near the border gateways. These issues are handled better by taking the advantages of SDN and/or NFV. In this research work, this 6LoWPAN architecture utilizing multiple border gateways is used with slight modifications.

The current implementations of the border gateways is split into two parts as basic gateway functionality and extended gateway functionalities. The basic functionality contains functionalities like neighbor discovery and protocol translation or adaptation. These basic functionalities are the one that are essential for a 6LoWPAN node to reach to an SDN network. These functions are normally the one provided within the lower layers of the 6BR protocol stack including the MAC and adaptation layers. The adaptation layer is an additional layer in the TCP/IP protocol stack of 6BRs which is placed between the MAC layer and network layer and is responsible for packet header compression and decompression, packet fragmentation and defragmentation tasks. A simple 6LoWPAN border router (s6BR) is responsible for such basic gateways services.

The extended border gateway functionalities are the one that can be executed after a 6LoWPAN node has made an access to the SDN switch. These functions are mainly provided within the higher layers of the border gateways. These functions will be taken from the care of border gateways to the SDN to provide more scalable and energy efficient implementations. These extended functions of a gateway are implemented as NFV functions within the SDN network. In such an implementation, IoT systems can get flexibility, resource efficiency and other possible advantages of NFV. The general structure of the proposed SDIoT system is shown in Figure 4.1. Only a high level design of the architecture has been made to provide simpler system picture of the proposed SDIoT system.

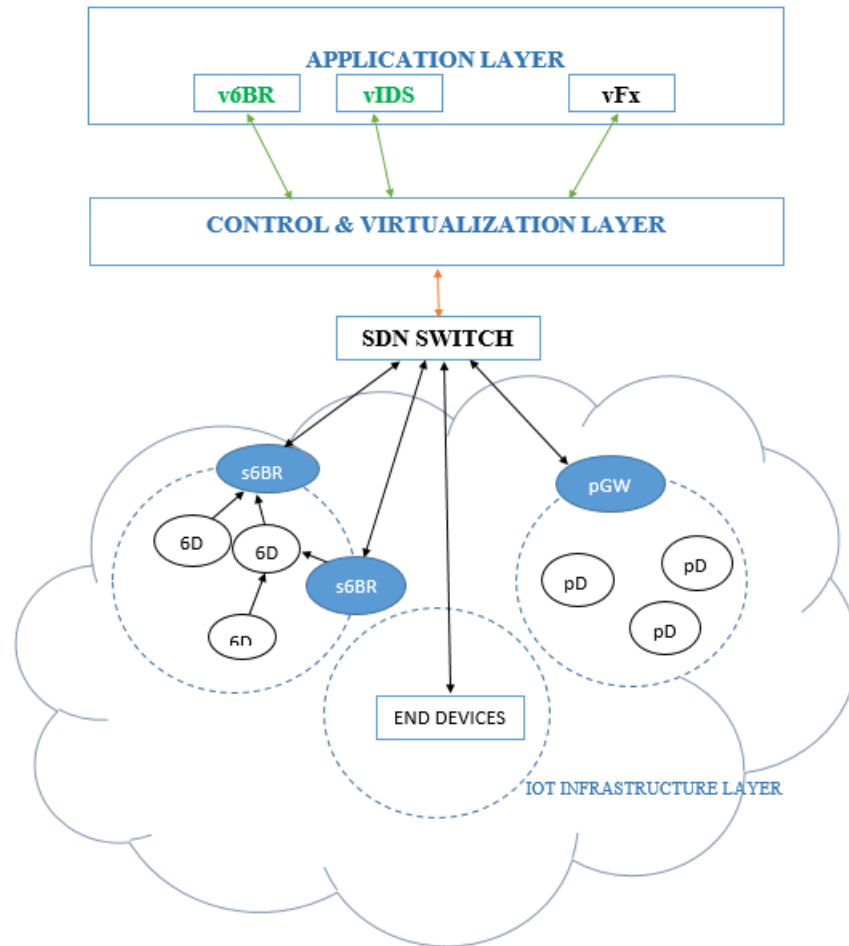


Figure 4.1: Architecture of the proposed SDIoT system

6D refers to a 6LoWPAN device, **pD**: refers to a proprietary device, **v6BR**: refers to virtualized 6BR function, **vIDS**: refers to a virtual implementation of the proposed IDS system, **vFx**: refers to other virtualized functions

4.4 ML Steps Followed in IDS Development

Several steps are followed when developing an IDS system. As an ML based IDS system, some kind of data cleaning and preprocessing, model training and model testing works have been carried out in this work. Underneath is presented a discussion of the steps involved and the tasks executed.

4.4.1 Data Cleaning and Preprocessing

A dataset on hand mightn't be processed as it is by an ML algorithm. Missing values, irrelevant features, categorical data or other issues which inhibit processing

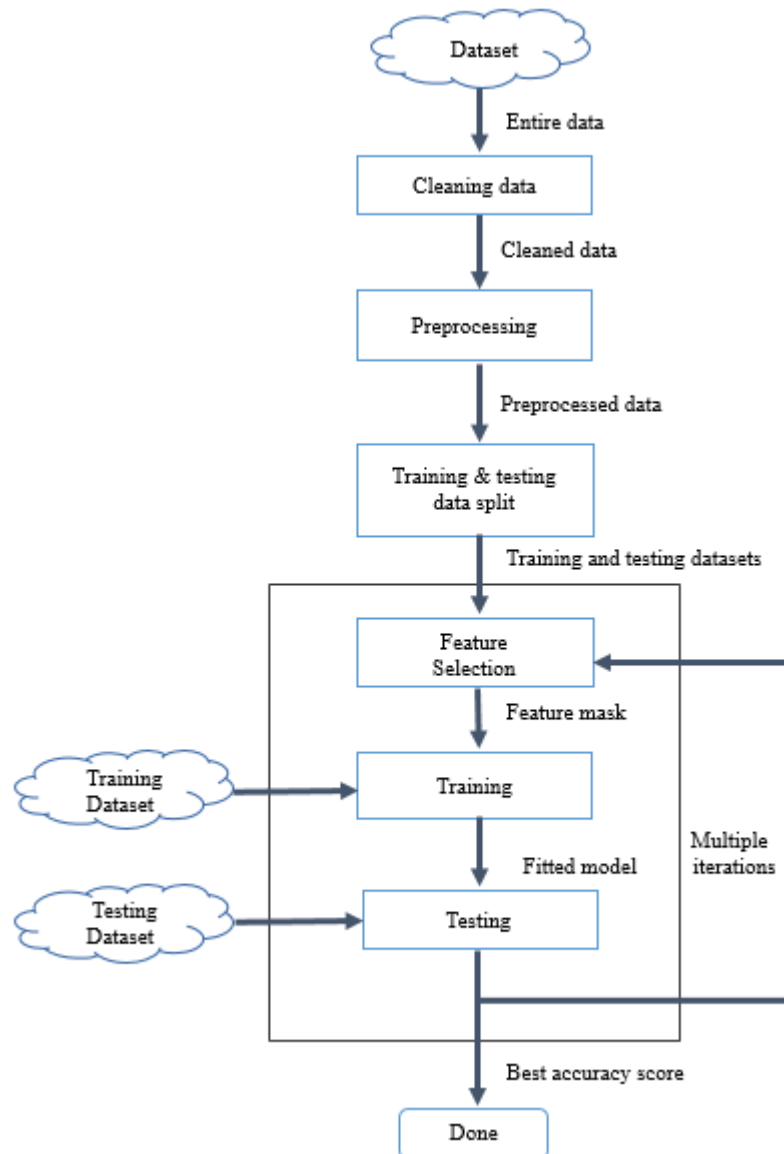


Figure 4.2: ML development steps followed

of a dataset by an ML algorithm might exist. Standardization, data normalization and other issues might prevail in some circumstances. Missing values, irrelevant features and the issue of categorical column are present in the CICIDS2017-Wed1 and CICIDS2017-Wed2 customized datasets. The following data cleaning and preprocessing steps have been undertaken in this work.

I – Data Cleaning

A couple of rows/columns having not a number (NaN) value or having infinite values exist in the customized datasets. A total of 1297 and 1299 values are

not available for use in the CICIDS2017-Wed1 and CICIDS2017-Wed2 datasets respectively. Further operations can't be carried out without fixing these values.

Several measures can be utilized to fix the issue of NaN values. One is dropping out rows or columns having a certain threshold of NaN values, while the other is imputing a missing value by another value which can be the mean, median, mode or other statistical measure of a column, a row, or a certain set of data. The decision shall be made intelligently based on the knowledge one has about the dataset on hand. In this work, since the missing values are mostly from the benign category and we have a sufficient number of benign samples, the rows associated with the NaNs and infinities have been just dropped. A similar operation is performed in [55] on the same dataset.

II- Removing Constant Features

Some columns contain no kind of information required for the classification of normal or attack traffic. Ten columns in both of the customized datasets have constant values. These columns are Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, Bwd Avg Bulk Rate. These constant columns are irrelevant for any kind of detection works. Thus they have been removed since they, the otherwise, might result in performance decrease and unwanted complexities.

III- Categorical Data Processing

Scikit-learn is based on Numpy and Scipy and assumes the data on hand is of a numeric type. It mightn't support or process categorical data as it is. In both the CICIDS2017-Wed1 and CICIDS2017-Wed2 customized datasets, the only column having categorical information is the 'target' column which contains information about the type of traffic. A categorical encoding action has been carried out in this work to convert the categorical target column to a numeric one using the available module for such a purpose, Ordinal Encoder preprocessing functionality.

4.4.2 Train Test Data Split

Another ML operation is executed after the dataset has been cleaned and preprocessed. This operation is splitting of the dataset for training and testing. These two portions of the datasets are required for training the estimator and then testing the performance of the corresponding model. Two common techniques are used to generate these training and test datasets. The techniques are the percent split and K-fold cross validation.

i. Percent Split

In this technique, a certain portion of the overall dataset is selected for training the estimator. The rest part is reserved for testing the model built. A 70%-30% percent split scheme has been utilized in this work. In this scheme 70% of the overall dataset is dedicated for training the estimator and the rest is reserved for testing the model. This scheme is used in many research works.

ii. K-fold Cross Validation

One of the challenges that occur in ML system development is the issue of overfitting [22, 26, 44, 67]. Some more details of this phenomenon is described under subsection 4.5.3. One of the mechanisms used in addressing overfitting is to use a strategy called K-fold cross validation [67].

In K-fold cross validation, the entire dataset is split into K folds. One of the folds is used to test the model while the rest part is used for training. In such a way, K number of trainings are made. Each time a fold which has not previously been used for testing will be kept for testing while the rest (K-1) folds are used for training. The prediction performance of the model will then be the average performance of the models in each of the K experiments. In this work a very common cross validation scheme, a 10-fold cross validation, is used. The data is split into 10 folds. 10 experiments will be made each time using 9 of the folds for training and the single fold left will be used for testing.

4.4.3 Training

At the heart of an ML based work is building of the model which is used for classification or other related tasks. That is what is accomplished in the training phase. An ML algorithm is subject to train on a portion of the overall dataset, the training dataset, which has been prepared previously in the data split section. After training an algorithm, it results in a model that has learned from the data. A number of estimators are out there for classification. In this work DTree, RF, and SVM estimators have been used. These estimators have been selected for their simplicity, wider usage throughout the literature and specially for their high performance in related work [21, 59, 60].

4.4.4 Testing

Testing and/or validation works are very crucial and probably the final steps in an ML based work. After the model is trained, its prediction capability shall be checked before it is deployed for a specific security use. This is made by providing the model to make predictions on the other set of the dataset which is normally called the test dataset.

4.5 Principles Considered in the Work

4.5.1 Parameter Tuning

ML training is all about finding the best solution from the broad set of possible solutions. A hyper hypothesis space is a term used to refer to the super set containing all the possible solutions as per the given inductive bias of the estimator. Parameter tuning is a search through such a hypothesis space to get the best optimized and high performing model. A rigorous experimentation is usually conducted to find the best hyper parameter values since it is difficult to anticipate the effects of the changes in parameter values [68]. Criterion, `min_samples_leaf`, `min_samples_split`, `max_depth`, `max_features` can be used in tuning a DTree, while the number of estimators (`n_estimators`) and the aforementioned parameters used

for tuning a DTree estimator can be used for tuning an RF estimator. Kernel, gamma, C and other parameters are used for tuning an SVM algorithm. It is difficult and time consuming to make a search through the entire hypothesis space. Only a restricted set of hypothesis can be considered in an experiment. Such restriction is made by selecting the parameters to tune and fixing the values that these parameters can take. Due to such a reason only limited parameters and values have been used in this work.

4.5.2 Feature Selection

One very important task in ML related works is feature selection. Feature selection is one part of dimensionality reduction. Not all features available within a dataset are important or equally important for detection of attacks. In many circumstances, increasing the number of features above a certain level doesn't have a noticeable significance on the classification performance. It rather adds up complexity and performance delays. It is not only that, it might also result in overfitting and classification performance decrease [27]. Thus, whenever possible it is desired to search for a small set of features that can adequately classify the traffics in a dataset.

Three types of feature selection techniques are utilized for such a purpose. These are filter based feature selection, wrapper techniques and embedded methods. Filter based techniques are based on statistical approaches. These techniques utilize the correlation of the individual input variables against the target variable. Based on the score of the correlation values, features that pass a certain threshold value are selected. These techniques are independent of the ML algorithm used and are completed before any training is made. They provide a very fast feature selection but since they don't consider the detection performance of a combination of a set of features, these techniques might not select features that provide higher accuracy.

On the other hand, the wrapper techniques try to rigorously find the best combination of features which provide higher model performance in terms of accuracy or other metrics. They generate a subset of the overall feature set, train the model using the subset, determine the performance of the subset. These steps are repeated

again and again till the best subset is determined. The subset selected is the best subset that results in a certain estimator to have a higher performance. Such subset is algorithm dependent and may vary from one estimator to another. Unlike filter methods, wrapper methods can always provide the best subset of features which can have the highest performance. In this work, the wrapper method, more specifically Recursive Feature Selection (RFE) which is supported in Scikit-learn is used.

4.5.3 Overfitting

Overfitting is one of the prime challenges to be addressed in an ML based research work. This phenomenon is a situation in which the trained model has a very high performance when testing the model with the data from which the model is trained but the model performs poorly when testing with other unseen data.

In order to address this problem, several techniques have been used throughout the literature [67]. Some of them like weight decay and post pruning are confined to certain algorithms or set of algorithms. Some other techniques are generic and can avoid overfitting when training algorithms of a broad range. One of such generic techniques used to lessen overfitting is to use separate training, testing and validation datasets. This scheme of using separate testing and validation steps is effectively achieved by the use of cross validation mentioned under subsection 4.4.2.

Another generic technique used to lessen overfitting is to use small number of features when building the model. Training an estimator using the whole feature set is usually a naive option. Using lots of features in learning an ML algorithm causes what is known as “the curse of dimensionality” [27]. Smaller number of features are preferred to have higher detection performance and to avoid overfitting. The idea behind this technique is a phenomenon told in simpler terms is better and preferable than told in a more complex fashion. Such a concept is commonly known as Ocam’s razor [67]. Feature selection and other dimensionality reduction techniques like principal component analysis are used for such a purpose. Some more details of this principle and the the various approaches that are available for such a purpose have been discussed under subsection 4.5.2.

4.5.4 Efficiency Enhancement

One other issue given a focus by this research work is the issue of efficiency. By efficiency it is meant to render high system performance by using limited resources the best possible. Efficiency can be judged in terms of the time required, memory use and other related factors. The main resource given a consideration in this research work is the classification time required by the classifiers. This consideration of efficiency is made on account that IoT systems are meant to be deployed in environments where a real time performance is required in many circumstances.

Several techniques can be applied here in order to enhance efficiency like the use of smaller number of features in the model and also to use faster algorithms. A focus is given to the prior technique in this work. Once again feature selection is considered in making efficient models in addition to the use of it for suppressing overfitting. An explicit measure of timing is not used in this work rather the timing performance is analyzed with respect to the number of features selected in the models. The smaller the number of features is the faster the classifier is to respond to an incoming traffic characteristics.

4.6 IDS Architecture Design

After devising the architecture of the SDIoT system and building of the model, what is left then is setting the placement of the IDS. In other words, the modules required, their interconnections, and the place the modules reside in shall be determined. Decisions like the way to sniff the traffic shall also be considered when designing the architecture.

Different implementations of an IDS can be made for the proposed SDIoT systems. It can be implemented as an SDN application. It can also be implemented as part of the internal architecture of a controller by adding extra functionality to the controller. A separate dedicated hardware that senses the traffic flow in the network and then perform detection might also be used.

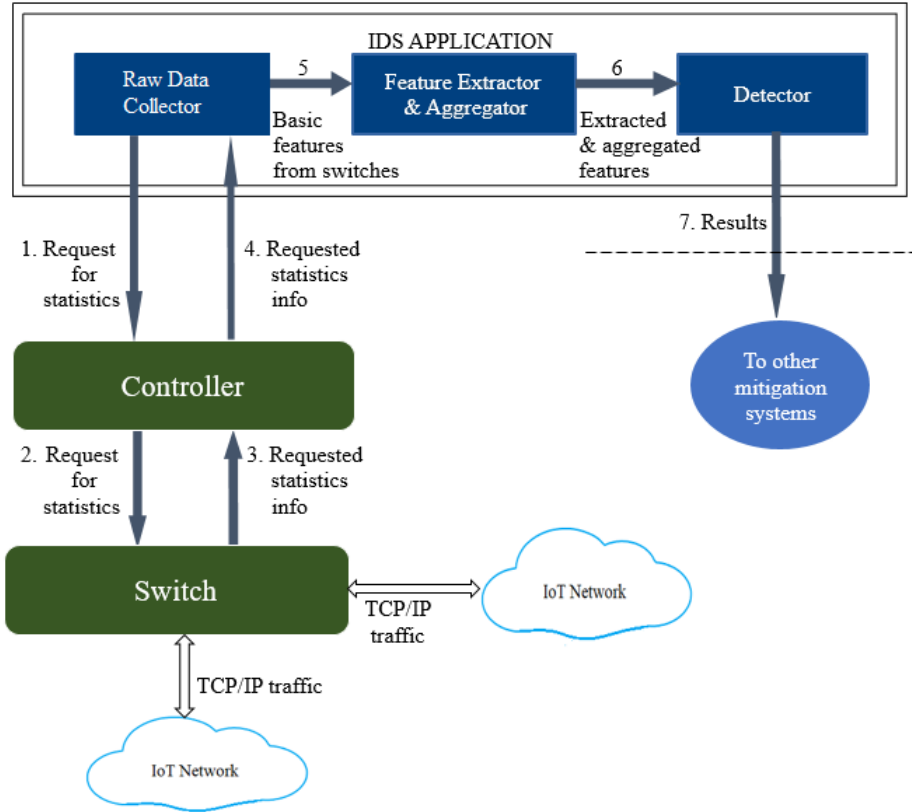


Figure 4.3: Architecture of the proposed IDS system

In this work, the IDS is implemented as a separate SDN application. A modular implementation of the IDS system is proposed and the whole system is split into three modules. These modules are a statistics data collector module, a feature extractor and aggregator module and a detector module. The placement of the modules and the data flow within the modules is seen in Figure 4.3. Virtualization related processes are once again removed for simplicity. The details of the tasks accomplished in the modules is given below.

I. Raw Data Collection

The first task of the IDS is collection of traffic statistics and related basic information about a flow. This is what is done by the statistical data collector module. Basic information about the flows can be obtained by parsing the `FLOW_REMOVED` and `FLOW_STATS` messages of an SDN network [62]. The `FLOW_REMOVED` message is sent to a controller when there is no match to a certain traffic flow or there is no packet flow for a specified period of time,

Table 4.2: Description of features to be collected from the switches

No	Extracted Feature	Description
1	Destination port	Port number of the destination host
2	Time interval	The duration between consecutive packets of a flow
3	Packet count	Number of packets in a flow
4	Byte count	Number of bytes in a packet

idle_timeout. The FLOW_STATS is a periodic message sent to a controller periodically in response to a statistics information requested by a controller. More specific messages can be sent to reduce the communication overhead.

In this work, the flow statistics collection messages are used to collect the required statistics information about a flow at different depths. This implementation of the statistics collection process doesn't require an implementation of extra novel capabilities for such a purpose beyond what can be got from the controller. In addition, it doesn't depend or have no effect on the internal implementation of a controller. The features collected are then forwarded to the feature aggregation phase to derive the necessary features required for the IDS module. Table 4.2 shows the summary of the features required from the switches.

II. Traffic Aggregation

The raw statistic collected from the switches can't be used for detection. After statistic collection is made, the important features shall be selected and aggregated. The features that shall be selected and aggregated are the one that are selected by the best performing estimator mentioned in section 5.3.2. These features selection and aggregation works are done upon the features collected by the statistics collector module explained above. The flow tables of SDN switches contain the required base features which can intern be used to derive the required features. Extra mathematical computations shall be carried out to get the additional aggregated features.

III. Traffic Classification

Traffic classification is performed by the detector module. This detector module is the third module used in the proposed IDS system. It contains an implementation

of the model developed during the ML training phase which contains the rules and parameters used for classification. This module accepts the aggregated traffic features from the feature extractor and aggregator module. It is this module that finally determines whether there are any anomalies within the traffic flow or not. And if there are any, this module determines the type of attack on hand.

4.7 Evaluation

After the design and implementation of the system, a performance measurement shall be made to assess how it is performing on classifying attacks. Underneath is discussed the evaluation techniques and evaluation measures used in the research work.

4.7.1 Evaluation Techniques

Starting from the most theoretical measure of mathematical modeling, different kinds of evaluation techniques have been proposed and used to evaluate how the various security solutions are performing. One such a technique is the use of mathematical models. A mathematical model is the most theoretical evaluation technique. Mathematical symbols are used to represent systems or components and validation is done by using such mathematical symbols. Simulation and emulation works are also very significant ways of making system evaluations. They provide an easy and cost effective way of making network experimentations. NS2, OPNET 14, Qulanet3 and OMNET++, Qualnet3 are some exemplary simulation environments while Mininet, NS3 and Emulab are some of the emulation tools.

One other technique used in evaluation of DoS research works is the use of real systems. In real systems the OS, devices, software and other components used in the experimentation are realistic. Though it is considered a best technique, the difficulty in making or altering system configurations, security matters, costs and other related issues are big hindrances for its wider use. Real or simulation/emulation based datasets are also being popular evaluation mechanisms in our days. They provide a benchmark to compare the performance of one solution against the

others. In this work a performance evaluation is done with the help of a benchmark dataset. Customized datasets from the CICIDS2017 dataset has been used both for training and finally evaluating the performance of the work.

4.7.2 Evaluation Measures

Once again, the performance of security systems can be evaluated from lots of perspectives. The detection rate, timing performance, energy consumption, resource consumption and other metrics can be used for such a purpose. In this research four detection performance metrics and a timing performance in terms of the number of features is used in evaluating how the system is performing in relation to the other systems.

The four detection performance measures are:

Accuracy: accuracy measures how often instances subject to an ML classifier are told apart correctly. It is the ratio of accurately classified instances to the total number of instances subject to classification. It is the main measure used in the measure of the performance of this system. Mathematically it is computed as:

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

Sensitivity: it's is also known as Recall, True Positive Rate (TPR) or DR [69]. Sensitivity determines the measure of the true cases identified in terms of the total true cases. In the case of this work it refers to the ratio of the number of attacks detected by the IDS model as an attack to the total number of attack traffics supplied to it.

$$DR = \frac{TP}{TP + FN}$$

Precision: refers to the ratio of positively detected attacks to the total number of attacks labeled as an attack regardless of whether they are attacks or not. Mathematically, it is described as follows:

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

F1-score F1-score provides a harmonic avg measure of two of the aforementioned metrics, the precision and sensitivity, of an estimator [21]. It is calculated as follows:

$$F1 - score = \frac{2 * DR * Precision}{DR + Precision}$$

Chapter 5

Results and Discussion

5.1 Results and Discussions

A total of 2,104 experiments have been carried out in this research work. Starting from a simple training with estimators having default parameter values, several experiments involving some kind of ML operation have been carried out. In these experiments, high accuracy results have been obtained. Moreover, in the second sessions an experiment which uses an RF estimator with 20 trees, a 99.967% accuracy have been achieved which is higher than the results scored by any previous related works.

The whole experiments conducted have been grouped under three sessions in order to have a better viable organization of the results. In the first session four experiments have been conducted. These experiments have been carried out using estimators with their default values. A simple percent split strategy has been used and no additional parameter tuning or feature selection works are taken here. In the second session, a set of experiments involving feature selection and parameter tuning steps have been carried out. Once again a percent split strategy has been used without consideration of cross validation. Multiple experiments involving cross validation alongside a feature selection operation have been carried out in the third session. The results achieved and the corresponding discussions have been presented in the following sections.

Table 5.1: Accuracy results of the three estimators with their default parameter values

No	Evaluation measure	Classifier	CICIDS2017 Wed1	CICIDS2017 Wed2
1	Accuracy (%)	DTree	99.942	99.946
		RF	99.939	99.936
2	Weighted avg precision (%)	DTree	99.942	99.946
		RF	99.939	99.936
3	Weighted avg recall (%)	DTree	99.942	99.946
		RF	99.939	99.936
4	Weighted avg f1-score (%)	DTree	99.942	99.946
		RF	99.939	99.936

5.2 Session 1: Simple Default Experiments

The experiments in this session have been carried out on the cleaned dataset with 69 features by considering only default parameter values. No feature selection and parameter tuning operations are done. Three ML algorithms, DTree and RF are used in the experiments. A total of four experiments which account learning of the DTree and RF the estimators upon the two datasets are conducted in this session. Experiments involving SVM are not conducted just for very long training time required. The results acquired are shown in Table 5.1.

As has been shown in the table the three estimators have a higher performance on the given datasets even if there are no feature selection and parameter tuning procedures. Yet, the model shall be enhanced to have a higher detection performance and also optimized to be efficient interms of memory use and classification speed.

5.3 Session 2: Experiments with Parameter Tuning and Feature Selection

Three kinds of feature selection operations mentioned in section 4.5.2 can be used for such a purpose. Feature selection using an effective RFE feature selection

algorithm have been used in this work. Alongside to this feature selection action is accomplished a parameter tuning task.

Min_samples_leaf is selected for tuning a DTree, n_estimators is selected for tuning an RF. Unlike these two estimators, SVM take long period of time for training. And parameter tuning is not done with SVM classifier. Instead training the estimator was made with only some feature selection experiments based on filter based feature selection mechanism.

5.3.1 Parameter Tuning with DTree

A) On CICIDS2017-Wed1 dataset

The first experiment of this session is made using a DTree estimator. This session is a continuation of the DTree based experimentation carried out in the first phase. In this session, RFE is used to make feature selection. This feature selection is made sixty-eight times corresponding to the number of features to be selected which range from one to sixty-eight. The DTree estimator is tuned by setting the values of the min_samples_leaf parameter to 1, 2, 3 and then 4.

As it can be seen from Figure 5.1, DTree estimator gives high accuracy results reaching slightly greater than 99.96% accuracy in two of the estimators. In addition, the performance of the DTree estimator varies as it is tuned from min_samples_leaf=1 to min_samples_leaf=4. It is the DTree estimator having min_samples_leaf value of 2 which has the highest performance. The estimator has achieved the highest accuracy value with only ten features. The maximum accuracy results achieved and the corresponding number of features selected by each of the estimators is summarized in Table 5.2.

The best DTree estimator:

The graph in Figure 5.2 shows the performance of the best DTree estimator. As can be seen from the figure, the estimator reaches a very high accuracy value of over 99% using only three features. And it immediately reaches a maximum accuracy value of 99.961% using only ten features. Then after, it shows certain small ups and downs but never reaches a value greater than the maximum achieved. These

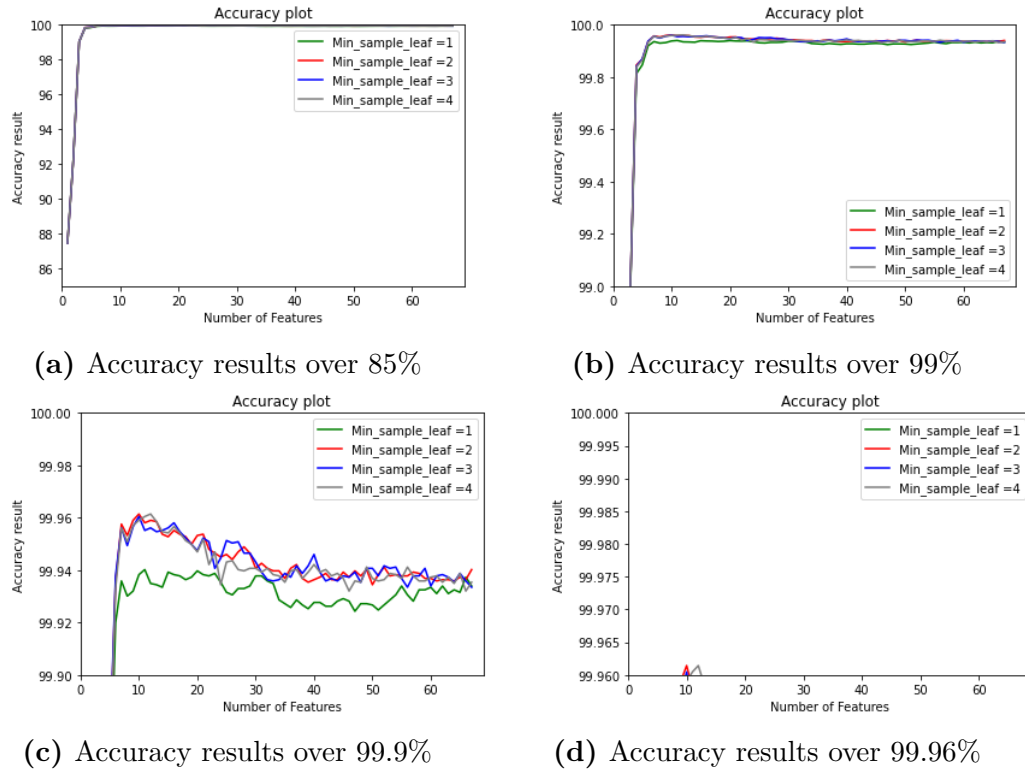


Figure 5.1: Performance results of tuned DTree estimators on CICIDS2017-Wed1 dataset

Table 5.2: Summary of results of the tuned decision tree estimators

No	min_samples.leaf	Best accuracy (%)	No. of features resulting best accuracy
1	1	99.940	11
2	2	99.961	10
3	3	99.960	10
4	4	99.961	12

ups and downs seen after the estimator fits well using the optimum features might not be uncommon. Such an issue can happen due to a loss in accuracy encountered when fitting the estimator with less relevant features.

B) On CICIDS2017-Wed2 dataset

Four tuned DTree estimators with min_samples.leaf of 10, 20, 60 and 100 have been applied on the CICIDS2017-Wed2 dataset. Feature selection is once again

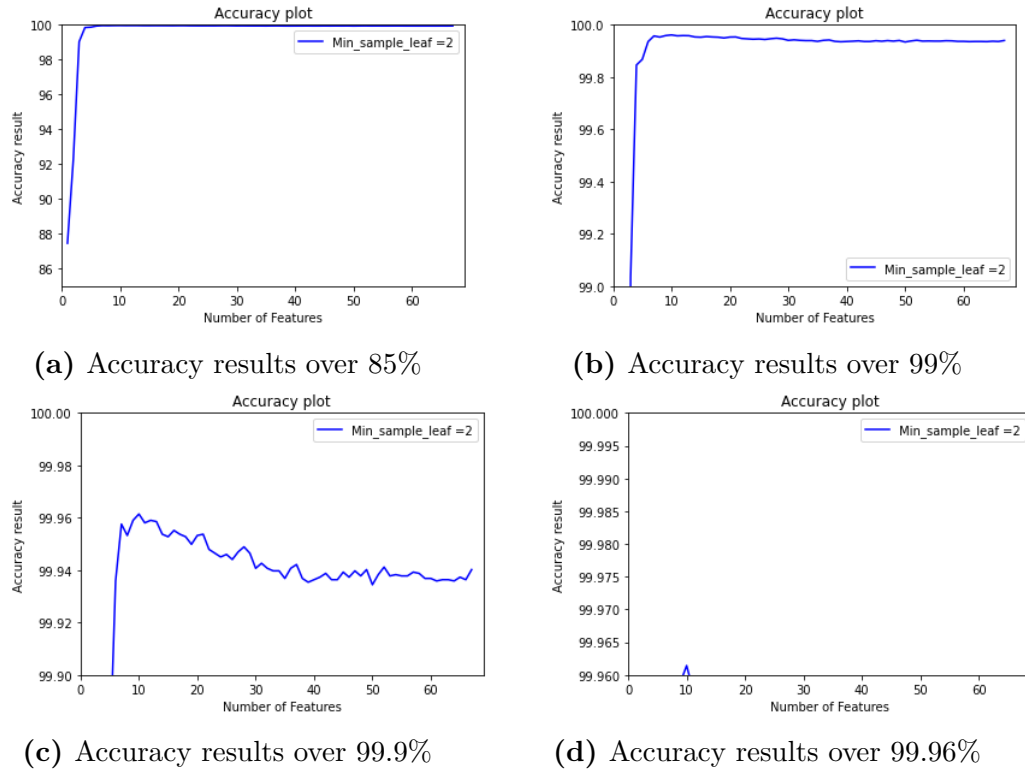


Figure 5.2: Performance results of the best performing DTree estimator on CICIDS2017-Wed1 dataset

performed in these experiments. However, since the tuned DTree estimators have relatively higher performance for feature numbers within a range of ten and twenty, a complete sixty-eight feature selection iterations are not carried out in this session. Rather, the number of features to be selected are made to be in the range of one to twenty-five. This decision is made by assuming that the estimator's performance on this dataset will have only a few deviations from the results achieved by using CICIDSS2017-Wed1 dataset since the DDoS has similar characteristics as that of the rest DoS attacks [26]. In addition, to make highly efficient IDS systems suitable for IoT system scenarios, small number of features are desired which is in compromise with our assumptions. A maximum accuracy result of 99.961% has been got in this research work with the use of only ten features. Figure 5.3 shows the accuracy result of the estimators.

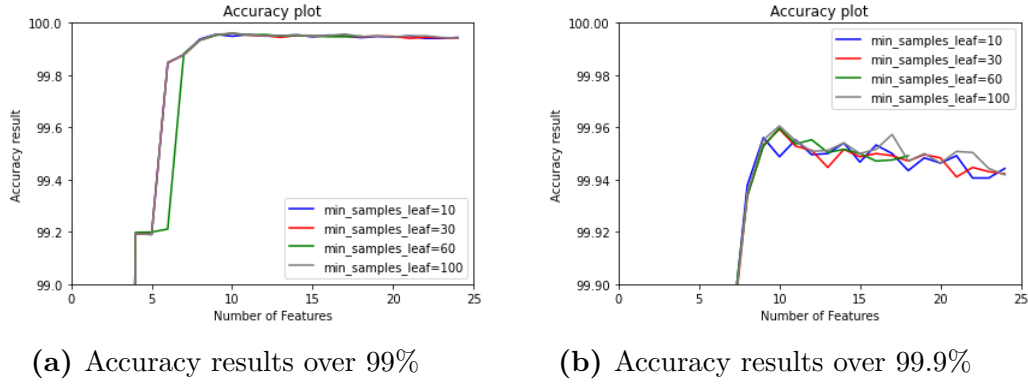


Figure 5.3: Performance results of tuned DTree estimators on CICIDS2017-Wed2 dataset

5.3.2 Parameter Tuning with RF

A) On CICIDS2017-Wed1 dataset

After completing the parameter tuning and feature selection experiments using DTree classifier, an RF estimator is proposed and trained in this session. One prime parameter which can be used for tuning in RF is the number of ensemble trees used to construct the model. At first a set of experiments have been performed using RF estimators having 10, 20, 30 and 100 ensemble trees. The corresponding accuracy results are shown in Figure 5.4. Since the one with 20 ensemble trees has higher performance than any of the other RF estimators including the previously trained DTree estimators, further experiments have been carried out using RF classifiers with 17, 19, 21 and 23 trees.

As has been shown in Figure 5.5 a similar pattern is observed in the additional four RF estimators with 17, 19, 21 and 23 trees with that of the previous four RF estimators. A reasonably closer accuracy results have been seen using the estimators. Yet, none has succeeded to get an accuracy value equaling or beating the highest observed by the RF estimator with 20 trees.

High performing RF estimator:

In Figure 5.6 is shown the performance of the highest performing RF estimator, the one with 20 trees. In this figure a more or less similar pattern is observed as

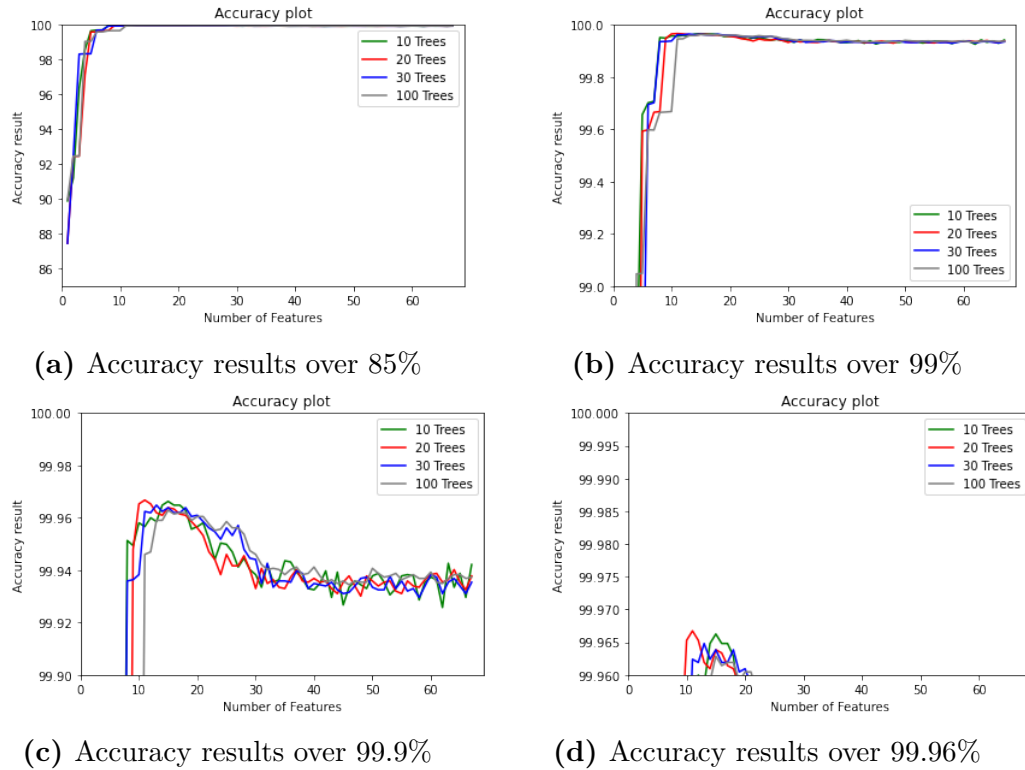


Figure 5.4: Performance results of tuned RF estimators on CICIDS2017-Wed1 dataset

the one achieved by the high performing DTree estimator seen in Figure 5.2. Yet some differences are visible in the results. The first thing observed is, in Figure 5.6 it has taken more number of features for the corresponding estimator to reach a reasonably high accuracy values. In addition, the estimator has achieved a new maximum accuracy than have been achieved even by other related works. It has achieved a new accuracy measure of 99.967%.

In order to achieve this result, the estimator has used only eleven features. The description of these features is shown in Table 5.3. A filter based feature selection is made to check if the corresponding features are also what can be expected from statistical distribution of the data. The SelectKBest function of a Scikit-learn is used for the filter based feature selection using `f_classif` scoring function.

B) On CICIDS2017-Wed2 dataset

Similar parameter tuning and feature selection steps that have been taken in the previous RF parameter tuning experiment was taken on the CICIDS2017-Wed2

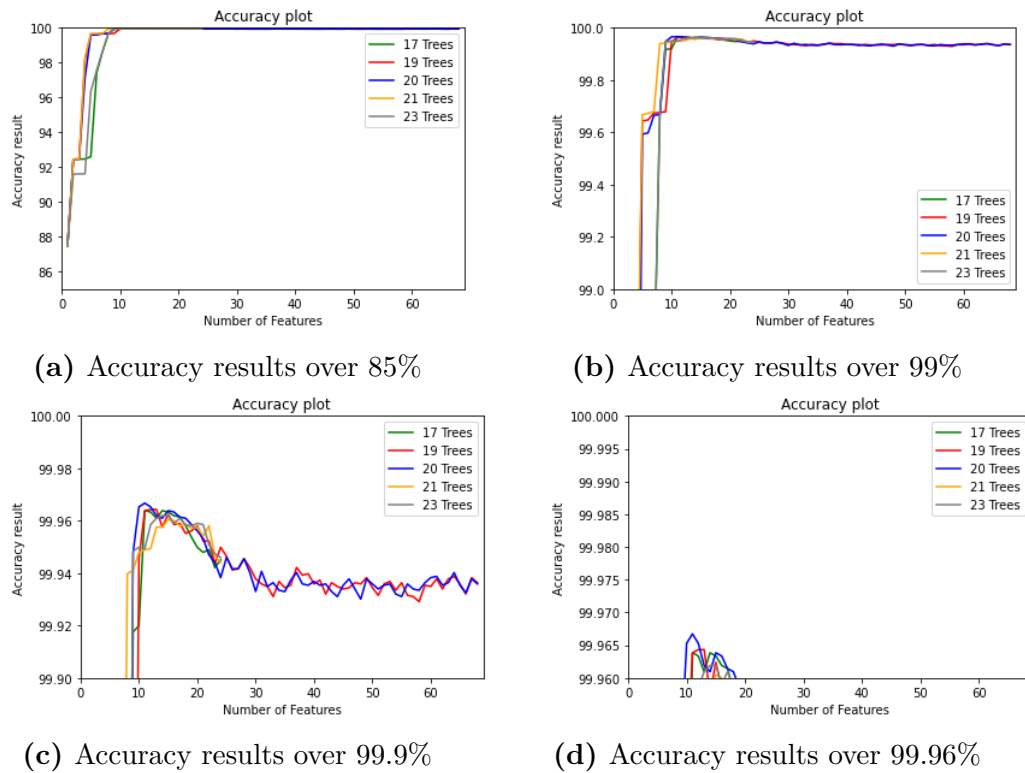


Figure 5.5: Performance results of tuned RF estimators with 17, 19, 20, 21 and 23 trees on CICIDS2017-Wed1 dataset

dataset. But, this time feature selection is made only to twenty-five starting from one. In addition to that, the selected number of estimators in this experiment are 12, 20, 40 and 70. Figure 5.7 shows the accuracy result achieved.

5.3.3 Parameter Tuning with SVM

Unlike trainings made using DTree and RF algorithms, SVM doesn't expose features for use by wrapper techniques. In addition, SVM takes a much longer time to train on the two customized datasets. Due to such factors, an intensive feature selection steps can't be carried out when using SVM estimator. On the other hand, training the estimator using the whole sixty-eight features will not give an efficient solution since this is a relatively high number for use. Two options are left to select the required features.

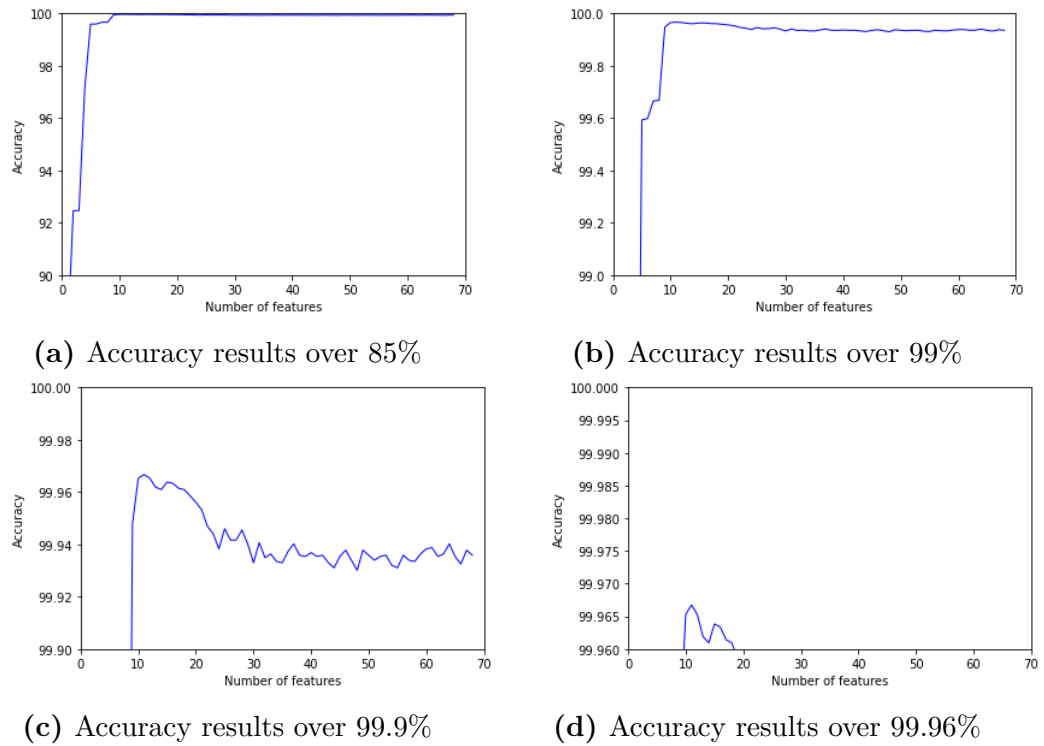


Figure 5.6: Performance results of tuned RF estimators on CICIDS2017-Wed1 dataset

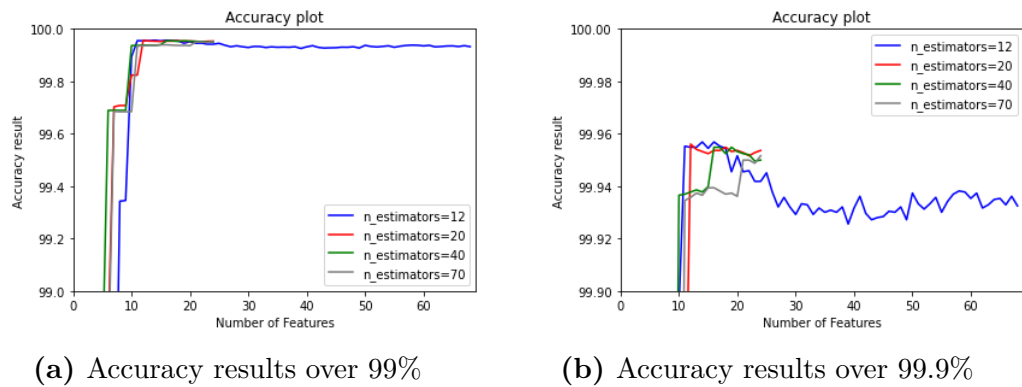


Figure 5.7: Performance results of tuned RF estimators on CICIDS2017-Wed2 dataset

One is the use of filter based feature selection methods, while the other is just to assume the features selected by DTree and RF algorithms are also the best selections of an SVM. The first one is chosen is this work.

Six feature sets are selected for training an SVM estimator. These sets are the one considered the best nine, ten, eleven, twelve, thirteen, and fourteen number

Table 5.3: Features selected by the best performing RF estimator

No	Feature Name	Is selected by the best DTree	Is selected in best 11 SelectKBest filter with f_classif
1	Destination Port	Yes	No
2	Bwd Packet Length Mean	No	Yes
3	Bwd Packet Length Std	Yes	Yes
4	Flow IAT Mean	Yes	No
5	Bwd Packets/s	No	No
6	Max Packet Length	Yes	Yes
7	Packet Length Mean	Yes	No
8	Packet Length Std	No	Yes
9	Avg Bwd Segment Size	No	Yes
10	Fwd Header Length	Yes	No
11	Init_Win_bytes_forward	Yes	No

Table 5.4: Accuracy results of the SVM estimator with SelectKBest feature selector

No	Number of features	Accuracy (%)
1	9	88.516
2	10	91.637
3	11	90.26
4	12	90.03
5	13	88.642
6	14	75.559

of features by the filter using f_classif scoring function. The results is shown in Figure 5.4. But, due to a very slow training and verification time required by the SVM and also due to the poor detection accuracy results achieved, further experimentations have not been conducted with it, neither with cross validation nor to operations made using the CICIDS2017-Wed2 dataset.

5.4 Results with CV

In this work an experiment involving cross validation with CV=10 have been made. The experiments have been conducted with the help of a Recursive Feature Elimination with Cross Validation (RFECV) based feature selection. In addition, RFE is also utilized in some of the experiments. The use cases and rationale for

use of the two feature selection functions has been given alongside the discussion of the corresponding experiments.

5.4.1 CV with DTree

A) On CICIDS2017-Wed1 dataset

The first experiment involving CV is made using similar decision tree estimators used in the second session. A wrapper based feature selection is made to enhance the performance of the model by searching the best combination of features. RFECV function supported by scikit-learn framework is used. High accuracy results have been scored in this experiment, but when compared to the previous experiments of this research work the accuracy achieved is lower. The achieved scores have been shown in the Figure 5.8.

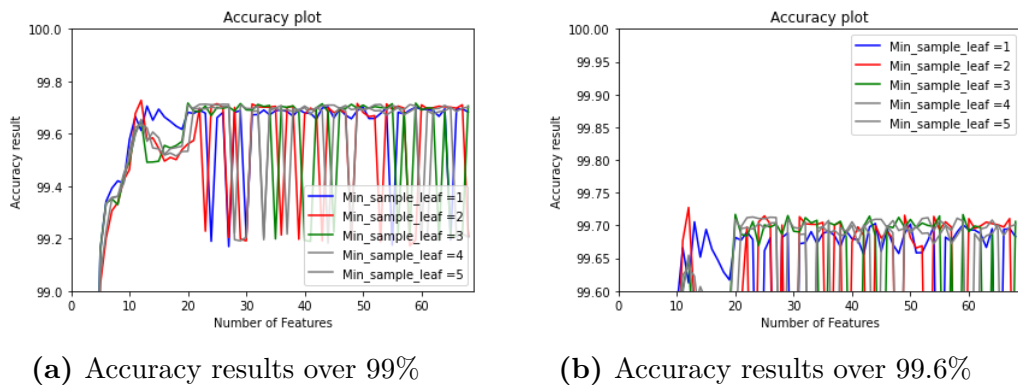


Figure 5.8: Performance results of tuned DTree estimators on CICIDS2017-Wed2 dataset

Results of best DTree estimator: In Figure 5.9 is shown the result of the best performing DTree estimator trained using cross validation.

The results follow a more or less similar pattern to that of Figure 5.2 and Figure 5.6. But, slight differences are observed in this figure. The first is the graph converges before reaching 99.8% accuracy. Though there are rises and falls, the graph is increasing. Especially, the big decline in accuracy seen after a peak value has been achieved is not observed in this experiment. The third thing that is observed is the valleys created at multiple points. The reason behind such valleys is not clear for the researcher.

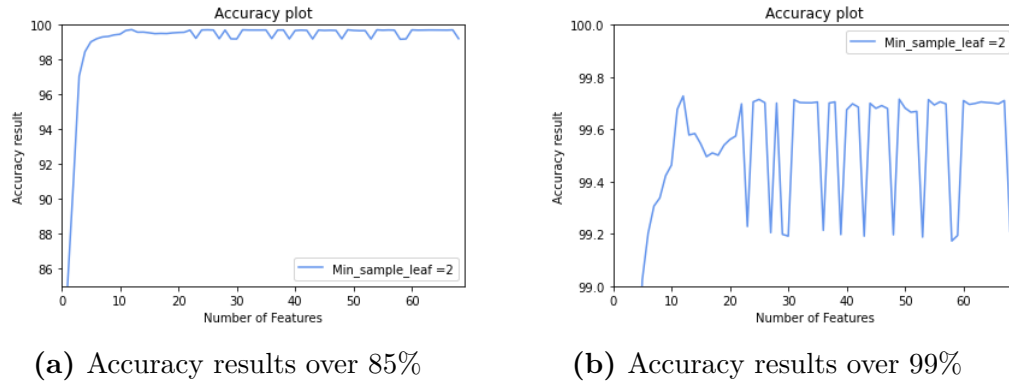


Figure 5.9: Performance results of the best performing DTree estimator on CICIDS2017-Wed1 dataset using cross validation

B) On CICIDS2017-Wed2 dataset

Three cross validation experiments have been conducted on the CICIDS2017-Wed2 dataset. Min_samples_leaf values of two, twenty and seventy have been used in this work. A maximum accuracy result of 99.489% has been achieved using sixteen features. Figure 5.10 shows the results achieved. These values are altered just to increase the hypothesis set by including new other hypotheses.

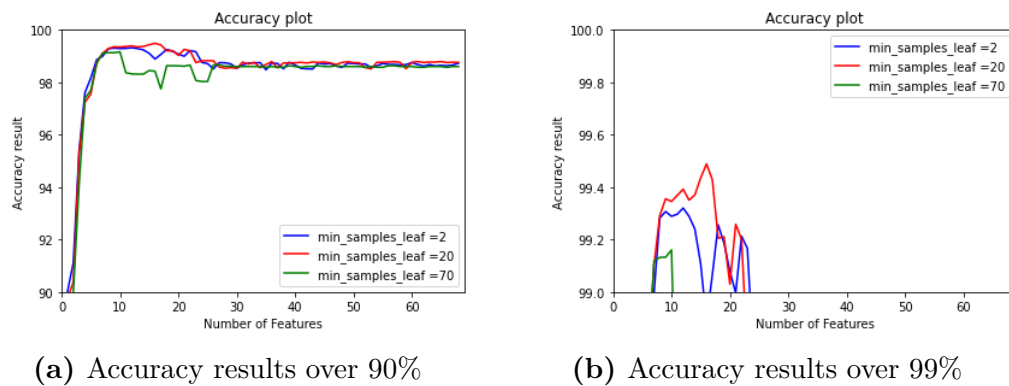


Figure 5.10: Cross validated experiments on CICIDS2017-Wed2 dataset using RFECV and DTree

5.4.2 CV with RF estimator

A) On CICIDS2017-Wed1 dataset

Additional cross validated experimentations have been carried out using RF estimators having 10, 20, 30 and 100 ensemble trees. In the experiments the RFECV

is not used. Rather, the dimensionally reduced data from the second session utilizing the four RF estimators with 10, 20, 30 and 100 trees has been used. This selection is made just for simplicity and fast response and with an assumption that RFE can perform well even with cross validated experiments. The results achieved are shown in Figure 5.11. A maximum accuracy result of 99.72% has been got in this work only with ten features.

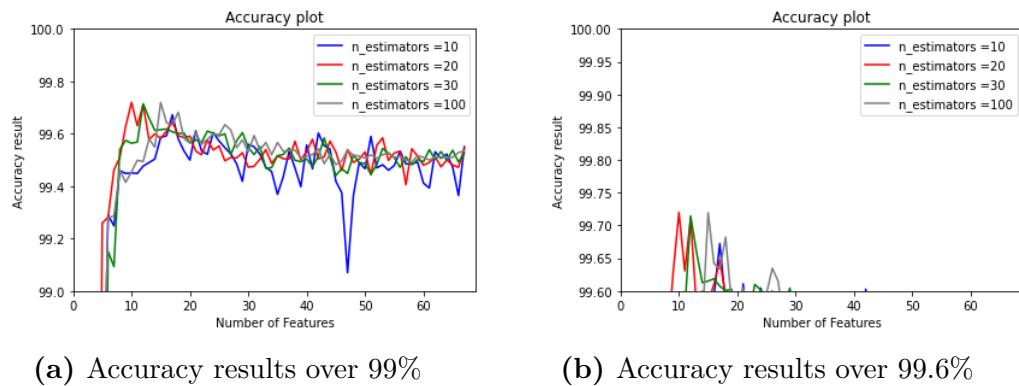


Figure 5.11: RF estimators with CV and RFE filter selection module on CICIDS2017-wed1

Additional experimentations which involve RFECV have been made with five RF estimators having 5, 15, 20, 35 and 60 trees. The corresponding results are shown in Figure 5.12. And a maximum accuracy value of 99.67% has been acquired in this work using only twelve features.

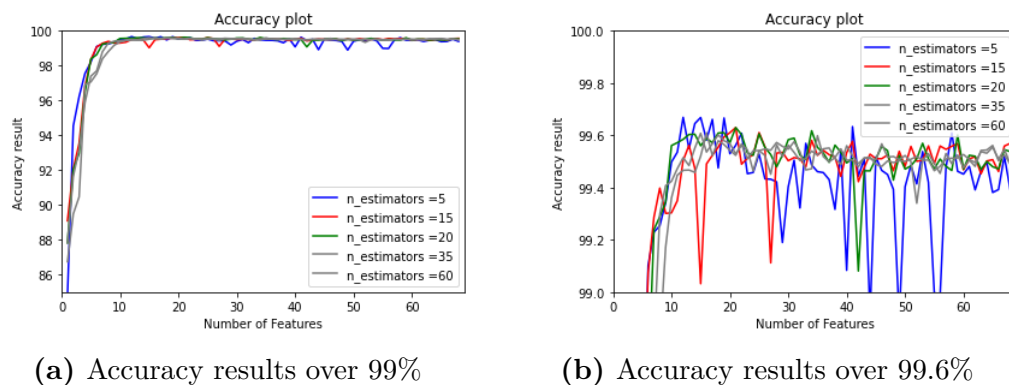


Figure 5.12: RF estimators with CV and RFECV filter selection module on CICIDS2017-wed1

B) On CICIDS2017-Wed2 dataset

Additional experimentations involving cross validation have been conducted using the CICIDS2017-Wed2 dataset. At first experiments have been tried to be

undertaken using RFECV. Yet, due to successive kernel failures, successful experimentations couldn't be undertaken. An approximation using RFE is used as a sole means of conducting dimensionally reduced cross validated experimentations on the given dataset. The features selected by the four RF estimators used to train the CICIDS2017-Wed2 dataset in the second session are used. The results achieved have been summarized in Figure 5.13. A maximum accuracy of only 98.886% has been achieved which is less than the corresponding result scored by using the DTree estimator.

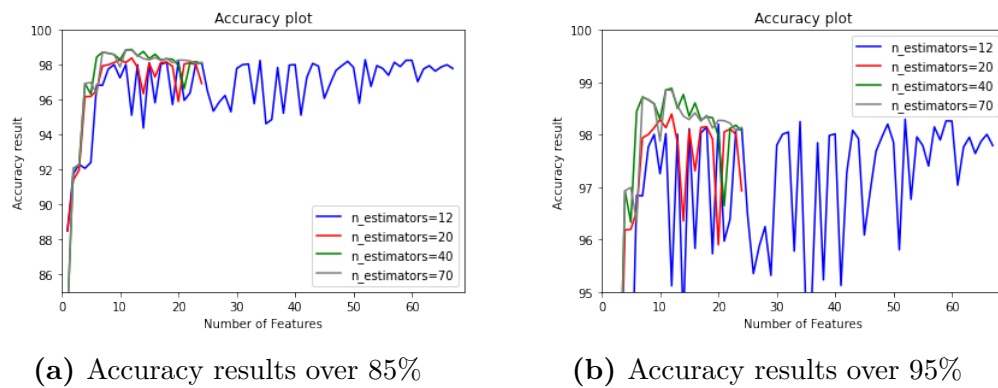


Figure 5.13: Performance of RF estimators with CV and RFE filter selection module on CICIDS2017-Wed2 dataset

5.5 Comparison with Other Related Works

Most of the research works conducted using this dataset are driven by other motivations. Due to these reasons comparing this research work against many CICIDS2017 based IDS systems is difficult. Gu et al. in [27] stated that results for DDoS detection performance on the CICIDS2017 dataset are missing. Despite it states that no experimental results are available, there are a few of them. In this work is presented a comparison of works conducted using CICIDS2017 dataset and which are directly or indirectly concerned with DoS/DDoS attacks.

Even if it is not aimed neither for SDN nor IoT systems, the work undertaken by [59] is very competitive. It even has slightly higher detection accuracy results and slightly better number of feature use than this research work in a certain case. However, the work is beaten in detection accuracy for the original unsampled CICIDS2017 dataset. But, more importantly, it is PCA which has been used for

dimensionality reduction. This has somewhat significant performance overheads for use in IoT systems which rather require faster performances and lower processing overheads. In addition, the work is prone to overfitting since no kind of cross validation works have been conducted.

Table 5.5: Performance comparison of works targetting DoS attacks on the CIDS2017 dataset

No	Work	Best accuracy (%)	Best precision (%)	Number of features	SDN Based	Remark
1	[This]	99.97	-	11	Yes	DDoS & DDoS
2	[21]	-	99.2	-	No	DoS
3	[57]	99.844	-	8	No	DoS
		99.981	-	8	No	Only DDoS
4	[26]	99.9	-	-	No	Only DDoS
5	[27]	-	98.86 (DR)	-	No	DoS
6	[58]	-	100	-	No	Only HTTP slowheader DoS

Table 5.6: Performance comparison of works considering all attacks on the CIDS2017 dataset

No	Work	Best accuracy (%)	Precision (%)	No of features	SDN Based	Remark
1	[59]	99.6	98.8 (DR)	10	No	For all attacks
		-	98.323	10	No	DoS & DDoS
		100	99.999	10	No	DoS & DDoS (Class balanced dataset)
2	[60]	99.16	-	-	No	Overall result
		-	98.647	-	No	DoS
3	[35]	95.53	98.94	-	No	Result for DDoS
4	[55]	99.665	94.457 (DR)	-	No	Overall result
		-	97.972 (DR)	-	No	DoS & DDoS
5	[23]	99.912	99.85	-	No	Overall result
		99.778	99.942	-	No	Overall result
6	[28]	-	92.85 (DR)	-	No	DDoS
		-	60(DR)	-	No	DoS

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Different kinds of attacks are launched to breach the privacy and security of IoT systems. One class of such attacks targets the availability of network systems and devices which are collectively referred to as DoS attack. This attack on availability is listed as one of the main issues that shall be addressed for the IoT. The stealth nature of DoS attacks, the significant loss it incurs with an easy attack operation, the very constrained environment provided by IoT systems, and an extended vulnerability space granted by SDN systems, among others, make SDIoT systems a conducive area for DoS attacks.

In defending systems against DoS attacks, various protection and mitigation measures have been developed. Yet, a comprehensive solution that addresses multiple security issues is not at place. In this work, an effective and efficient IDS system is proposed and developed to defend SDN orchestrated IoT systems from DoS attacks. The CICIDS2017 dataset has been used to train three ML algorithms, DTree, RF, and SVM in developing the detection system.

Several experiments have been undertaken and high accuracy results have been achieved. A new high accuracy result of 99.967% has been achieved on the CICIDS2017-Wed1 dataset using only eleven features. Feature selection and parameter tuning have been used to enhance detection performance and efficiency

of the estimators. `Min_samples_tree` and `n_estimators` have been used for tuning DTree and RF estimators respectively. Only feature selection has been made on the SVM estimator. In addition to the reduction of the number of features, cross validation has been used to reduce overfitting. High cross validated accuracy results of 99.728% has been achieved using twelve features with the use of a DTree estimator.

The high detection accuracy collected with the use of only eleven features accompanied with the use of a relatively faster ML algorithm make the collected result suitable for use in smart environments. This model meets the basic requirement of supervised IDS systems developed for smart environments. It can effectively be used with other related IDS systems supporting other Cyber attacks and providing anomaly detection to provide a sound detection system. However, even if it is this model being repeatedly labelled to be an effective model in this work, it is not the sole independent choice for practical use. The high performing cross validated model developed using a DTree estimator is a competent choice and it is not actually possible to say this model has a lower detection accuracy than the previously mentioned model unless a verification is done using another set of verification data. Moreover, the later one might rather be preferred for its efficiency since it is a DTree model used to develop it which is even faster than an RF estimator with twenty trees.

6.2 Future Works

A number of techniques and approaches have been proposed and implemented in different research works so far to defend DoS attacks. Due to the extended vulnerability space exposed by the SDIoT systems and the enormous significance IoT and SDN systems are expected to bring in the future, much researches targeting DoS defense mechanisms for these SDIoT systems will be required. Security works which involve any kind of protection or mitigation mechanism is yet a task left for future researchers.

This work bases on the advantage earned by ML. And after all ML work is all about searching for the best solution from whole hyper parameter space. As has been explained in the experiments, various parameter tuning steps have been conducted

in this research work. Yet, only a few parameters have been considered for tuning in this work. A lot remains in such regards. And it is up to the researchers to conduct extended parameter tuning to enhance detection performance of IDS systems especially with a focus on development of overfitting prone models.

One other research gap is devising of technology and gateway independent architecture for SDIoT systems. This architecture is what is highlighted and advocated by the Internet Protocol for Smart Objects (IPSO) organization [70]. This organization mentions the various disadvantages of the traditional IoT architecture, which is based on the use of gateways. Chen et al. in [3] has also noted the importance of open, flexible and standardized architecture for IoT and calls for a new kind of IoT architecture for IoT systems. Solutions developed for such kind of new environments might have new traffic characteristics and hence require new datasets and new solutions. This point is also another research gap to be addressed in future works.

One of the main challenges in security works aiming DoS attacks is the difficulty in defending zero day attacks. Various anomaly based solutions are devised in such regards. Yet, the performance of these systems is far from what is expected or shall be met. It is up to future researchers to bring anomaly based detection system having high detection accuracy and low FAR.

Bibliography

- [1] C. Vandana, “Security improvement in iot based on software defined networking (sdn),” *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 5, no. 1, pp. 2327–4662, 2016.
- [2] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, “How can heterogeneous internet of things build our future: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2011–2027, 2018.
- [3] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, “A vision of iot: Applications, challenges, and opportunities with china perspective,” *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.
- [4] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, and H. L. Owen, “Advancing software-defined networks: A survey,” *IEEE Access*, vol. 5, pp. 25487–25526, 2017.
- [5] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, “A survey on software-defined wireless sensor networks: Challenges and design requirements,” *IEEE access*, vol. 5, pp. 1872–1899, 2017.
- [6] S. K. Tayyaba, M. A. Shah, O. A. Khan, and A. W. Ahmed, “Software defined network (sdn) based internet of things (iot) a road ahead,” in *Proceedings of the International Conference on Future Networks and Distributed Systems*, pp. 1–8, 2017.
- [7] 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.

- [8] Q. Yan, F. R. Yu, Q. Gong, and J. Li, “Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges,” *IEEE communications surveys & tutorials*, vol. 18, no. 1, pp. 602–622, 2015.
- [9] D. Kreutz, F. M. Ramos, and P. Verissimo, “Towards secure and dependable software-defined networks,” in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pp. 55–60, 2013.
- [10] M. F. Elrawy, A. I. Awad, and H. F. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” *Journal of Cloud Computing*, vol. 7, no. 1, p. 21, 2018.
- [11] L. Liang, K. Zheng, Q. Sheng, and X. Huang, “A denial of service attack method for an iot system,” in *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 360–364, IEEE, 2016.
- [12] 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.
- [13] D. M. Mendez, I. Papapanagiotou, and B. Yang, “Internet of things: Survey on security and privacy,” *arXiv preprint arXiv:1707.01879*, 2017.
- [14] J. Deogirikar and A. Vidhate, “Security attacks in iot: A survey,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 32–37, IEEE, 2017.
- [15] 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, 2017.
- [16] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of things (iot) security: Current status, challenges and prospective measures,” in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336–341, IEEE, 2015.
- [17] “Denial of service attacks increasing threat to cybersecurity.” <https://www.floridatechonline.com/blog/information-technology/>

- denial-of-service-attacks-increasing-threat-to-cybersecurity/.
(Accessed on 05/29/2020).
- [18] “Major ddos attacks increased 967% this year - techrepublic.” <https://www.techrepublic.com/article/major-ddos-attacks-increased-967-this-year/>. (Accessed on 05/29/2020).
- [19] J. Li, Z. Zhao, R. Li, and H. Zhang, “Ai-based two-stage intrusion detection for software defined iot networks,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2093–2102, 2018.
- [20] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, “Internet of things: A survey on machine learning-based intrusion detection approaches,” *Computer Networks*, vol. 151, pp. 147–157, 2019.
- [21] F. S. d. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, “Smart detection: An online approach for dos/ddos attack detection using machine learning,” *Security and Communication Networks*, vol. 2019, 2019.
- [22] B. A. Tama, L. Nkenyereye, S. R. Islam, and K.-S. Kwak, “An enhanced anomaly detection in web traffic using a stack of classifier ensemble,” *IEEE Access*, vol. 8, pp. 24120–24134, 2020.
- [23] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, “Network intrusion detection: Based on deep hierarchical network and original flow data,” *IEEE Access*, vol. 7, pp. 37004–37016, 2019.
- [24] G. Loganathan, “Real-time intrusion detection using multidimensional sequence-to-sequence machine learning and adaptive stream processing,” 2018.
- [25] A. Yulianto, P. Sukarno, and N. A. Suwastika, “Improving adaboost-based intrusion detection system (ids) performance on cic ids 2017 dataset,” in *Journal of Physics: Conference Series*, vol. 1192, p. 012018, IOP Publishing, 2019.
- [26] R. Vinayakumar, M. Alazab, K. 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.

- [27] 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.
- [28] P. A. A. Resende and A. C. Drummond, "Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling," *Security and Privacy*, vol. 1, no. 4, p. e36, 2018.
- [29] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, p. 1550147717741463, 2017.
- [30] M. E. Ahmed and H. Kim, "Ddos attack mitigation in internet of things using software defined networking," in *2017 IEEE third international conference on big data computing service and applications (BigDataService)*, pp. 271–276, IEEE, 2017.
- [31] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, 2019.
- [32] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," in *ICISSP*, pp. 108–116, 2018.
- [33] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.
- [34] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proceedings of the 2019 ACM Southeast Conference*, pp. 86–93, 2019.
- [35] A. Janagam and S. Hossen, "Analysis of network intrusion detection system with machine learning algorithms (deep reinforcement learning algorithm)," 2018.
- [36] A. Oracevic, S. Dilek, and S. Ozdemir, "Security in internet of things: A survey," in *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, IEEE, 2017.

- [37] E. Tabane and T. Zuva, “Is there a room for security and privacy in iot?,” in *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pp. 260–264, IEEE, 2016.
- [38] L. Chen, S. Thombre, K. Järvinen, E. S. Lohan, A. Alén-Savikko, H. Leppäkoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala, *et al.*, “Robustness, security and privacy in location-based services for future iot: A survey,” *IEEE Access*, vol. 5, pp. 8956–8977, 2017.
- [39] D. M. Mendez, I. Papapanagiotou, and B. Yang, “Internet of things: Survey on security and privacy,” *arXiv preprint arXiv:1707.01879*, 2017.
- [40] J. Granjal, E. Monteiro, and J. S. Silva, “Security for the internet of things: a survey of existing protocols and open research issues,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.
- [41] A. Burg, A. Chattopadhyay, and K.-Y. Lam, “Wireless communication and security issues for cyber–physical systems and the internet-of-things,” *Proceedings of the IEEE*, vol. 106, no. 1, pp. 38–60, 2017.
- [42] R. Mitchell and I.-R. Chen, “A survey of intrusion detection techniques for cyber-physical systems,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–29, 2014.
- [43] O. Anthony, J. Odeyabinya, and S. Emmanuel, “Intrusion detection in internet of things (iot),” *International Journal of Advanced Research in Computer Science*, vol. 9, no. 1, 2018.
- [44] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, “A supervised intrusion detection system for smart home iot devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.
- [45] N.-N. Dao, T. V. Phan, J. Kim, T. Bauschert, S. Cho, *et al.*, “Securing heterogeneous iot with intelligent ddos attack behavior learning,” *arXiv preprint arXiv:1711.06041*, 2017.
- [46] S. Kajwadkar and V. K. Jain, “A novel algorithm for dos and ddos attack detection in internet of things,” in *2018 Conference on Information and Communication Technology (CICT)*, pp. 1–4, IEEE, 2018.

- [47] B.-C. Chifor, I. Bica, and V.-V. Patriciu, "Mitigating dos attacks in publish-subscribe iot networks," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6, IEEE, 2017.
- [48] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.
- [49] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.
- [50] 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.
- [51] A. Elsaedy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "A machine learning approach for intrusion detection in smart cities," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–5, IEEE, 2019.
- [52] M. Niedermaier, M. Striegel, F. Sauer, D. Merli, and G. Sigl, "Efficient intrusion detection on low-performance industrial iot edge node devices," *arXiv preprint arXiv:1908.03964*, 2019.
- [53] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Computer Networks*, vol. 62, pp. 122–136, 2014.
- [54] S. M. Mousavi and M. St-Hilaire, "Early detection of ddos attacks against sdn controllers," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, pp. 77–81, IEEE, 2015.
- [55] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 228–233, IEEE, 2019.
- [56] S. Behal and K. Kumar, "Trends in validation of ddos research," *Procedia Computer Science*, vol. 85, pp. 7–15, 2016.

- [57] S. Singh Panwar, Y. Raiwani, and L. S. Panwar, "Evaluation of network intrusion detection with features selection and machine learning algorithms on cicids-2017 dataset," *Available at SSRN 3394103*, 2019.
- [58] G. Loganathan, J. Samarabandu, and X. Wang, "Real-time intrusion detection in network traffic using adaptive and auto-scaling stream processor," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2018.
- [59] R. Abdulhammed, H. Musafar, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.
- [60] V. Gustavsson, "Machine learning for a network-based intrusion detection system: An application using zeek and the cicids2017 dataset," 2019.
- [61] K. Kalkan and S. Zeadally, "Securing internet of things with software defined networking," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 186–192, 2017.
- [62] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for sdn," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 787–793, IEEE, 2017.
- [63] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 318–322, IEEE, 2018.
- [64] "Can heartbleed be used in ddos attacks? — network world." <https://www.networkworld.com/article/2176153/can-heartbleed-be-used-in-ddos-attacks-.html>. (Accessed on 05/29/2020).
- [65] "High-severity vulnerability in openssl allows dos attacks — securityweek.com." <https://www.securityweek.com/high-severity-vulnerability-openssl-allows-dos-attacks>. (Accessed on 05/29/2020).
- [66] W. Ge, L. Zheng, P. Luo, and Z. Liu, "Implementation of multiple border routers for 6lowpan with contikios," 2015.

-
- [67] T. M. Mitchell *et al.*, “Machine learning,” 1997.
- [68] “How to tune a decision tree? - towards data science.” <https://towardsdatascience.com/how-to-tune-a-decision-tree>. (Accessed on 05/29/2020).
- [69] “Precision and recall definition — deepai.” <https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall>. (Accessed on 05/29/2020).
- [70] A. Dunkels, “Ipsos smart objects - oma specworks.” <https://omaspecworks.org/develop-with-oma-specworks/ipso-smart-objects/>. (Accessed on 05/29/2020).

Appendix A

List of features in CICIDS2017 dataset

- | | |
|--------------------------------|----------------------------|
| 1. Destination Port | 13. Bwd Packet Length Mean |
| 2. Flow Duration | 14. Bwd Packet Length Std |
| 3. Total Fwd Packets | 15. Flow Bytes/s |
| 4. Total Backward Packets | 16. Flow Packets/s |
| 5. Total Length of Fwd Packets | 17. Flow IAT Mean |
| 6. Total Length of Bwd Packets | 18. Flow IAT Std |
| 7. Fwd Packet Length Max | 19. Flow IAT Max |
| 8. Fwd Packet Length Min | 20. Flow IAT Min |
| 9. Fwd Packet Length Mean | 21. Fwd IAT Total |
| 10. Fwd Packet Length Std | 22. Fwd IAT Mean |
| 11. Bwd Packet Length Max | 23. Fwd IAT Std |
| 12. Bwd Packet Length Min | 24. Fwd IAT Max |

-
- | | |
|----------------------------|-------------------------|
| 25. Fwd IAT Min | 46. RST Flag Count |
| 26. Bwd IAT Total | 47. PSH Flag Count |
| 27. Bwd IAT Mean | 48. ACK Flag Count |
| 28. Bwd IAT Std | 49. URG Flag Count |
| 29. Bwd IAT Max | 50. CWE Flag Count |
| 30. Bwd IAT Min | 51. ECE Flag Count |
| 31. Fwd PSH Flags | 52. Down/Up Ratio |
| 32. Bwd PSH Flags | 53. Average Packet Size |
| 33. Fwd URG Flags | 54. AvgFwd Segment Size |
| 34. Bwd URG Flags | 55. AvgBwd Segment Size |
| 35. Fwd Header Len | 56. Fwd Header Length |
| 36. Bwd Header Length | 57. FwdAvg Bytes/Bulk |
| 37. Fwd Packets/s | 58. FwdAvg Packets/Bulk |
| 38. Bwd Packets/s | 59. FwdAvg Bulk Rate |
| 39. Min Packet Length | 60. BwdAvg Bytes/Bulk |
| 40. Max Packet Length | 61. BwdAvg Packets/Bulk |
| 41. Packet Length Mean | 62. BwdAvg Bulk Rate |
| 42. Packet Length Std | 63. SubflowFwd Packets |
| 43. Packet Length Variance | 64. SubflowFwd Bytes |
| 44. FIN Flag Count | 65. SubflowBwd Packets |
| 45. SYN Flag Count | 66. SubflowBwd Bytes |

67. Init_Win_bytes_forward	74. Active Min
68. Init_Win_bytes_backward	75. Idle Mean
69. act_data_pkt_fwd	76. Idle Std
70. min_seg_size_forward	77. Idle Max
71. Active Mean	78. Idle Min
72. Active Std	79. Label
73. Active Max	

Appendix B

Description of features selected by the best RF estimator

Table B.1: Description of selected features

No	Feature Name	Description
1	Destination Port	Port address of the destination host
2	Bwd Packet Length Mean	Mean size of packet in backward direction
3	Bwd Packet Length Std	Standard deviation size of packet in backward direction
4	Flow IAT Mean	Average time between two packets sent in the flow
5	Bwd Packets/s	Number of backward packets per second
6	Max Packet Length	Maximum length of a packet
7	Packet Length Mean	Mean length of a packet
8	Packet Length Std	Standard deviation length of a packet
9	AVG Bwd Segment Size	Average segment size observed in the backward direction
10	Fwd Headr Length	Total bytes used for headers in the forward direction
11	Init_Win_bytes_forward	The total number of bytes sent in initial window in the forward direction

Appendix C

Steps to Determine the Eleven Aggregated Features

1. Destination Port

1. Return a destination Port

2. Bwd Packet Length Mean

1. Filter packets in background direction
2. Get size of each packets using BYTE_COUNT
3. Calculate the average of the size of the packets using

3. Bwd Packet Length Std

1. Filter packets in background direction
2. Get size of each packets using BYTE_COUNT
3. Calculate standard deviation of the size of the packets

4. Flow IAT Mean

1. Get the arrival time of each packet
2. Calculate the inter arrival time of two consecutive packets
3. Calculate the average of the inter arrival times found in step two

5. Bwd Packets/s

1. Filter packets in background direction
2. Get the total number of packets using PACKET_COUNT
3. Get the duration of the arrival of packets
4. Divide the total number of packets to the given duration

6. Max Packet Length

1. Filter all packets
2. Get size of each packets using BYTE_COUNT
3. Determine the maximum size of the packets

7. Packet Length Mean

1. Filter all packets
2. Get size of each packets using BYTE_COUNT
3. Calculate the average size of the packets

8. Packet Length Std

1. Filter all packets
2. Get size of each packets using BYTE_COUNT
3. Calculate the standard deviation of the size of the packets

9. AVG Bwd Segment Size

1. Filter packets in backward direction
2. Get size of each packet's segment
3. Calculate the average size of the segments

10. Fwd Headr Length

1. Filer packets in forward directions
2. Get header length of the packets
3. Calculate the total number of bytes in the headers

11. Init Win bytes forward

1. Filter all packets in forward directions
2. Filter packets sent in the initial windows
3. Get the number of each bytes in these packets using `BYTE_COUNT`
4. Calculate the total number of bytes