Attack Detection and Mitigation for IoT Ecosystems: Adaptive, Scalable, and Lightweight Approaches

Pradeepkumar Gajendra Bhale

Attack Detection and Mitigation for IoT Ecosystems: Adaptive, Scalable, and Lightweight Approaches

Thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Pradeepkumar Gajendra Bhale

Under the supervision of

Prof. Sukumar Nandi and Prof. Santosh Biswas

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

April 2024

This thesis is dedicated to Parents, family and friends !

For their unconditional love, patience, sacrifices and continued support during my successful journey

People are often unreasonable, illogical, and self-centered.
Forgive them anyway.

If you are kind,
people may accuse you of selfish ulterior motives.
Be kind anyway.

If you are successful,
you will win some false friends and some true enemies.
Succeed anyway.

If you are honest and frank,
people may cheat you.
Be honest and frank anyway.

What you spend years building,
someone could destroy overnight.
Build anyway.

If you find serenity and happiness,
they may be jealous.
Be happy anyway.

The good you do today,
people will often forget tomorrow.
Do good anyway.

Give the world the best you have,
and it may never be enough.
Give the best you've got anyway.

You see, in the final analysis it is between you and God;
It was never between you and them anyway.

- Mother Teresa

Declaration

I hereby certify that

a. The work contained in this thesis is original and has been done by myself and the general supervision of my supervisor(s).

b. The work reported herein has not been submitted to any other Institute for any degree or diploma.

c. Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.

d. I also affirm that no part of this thesis contains plagiarised contents to the best of my knowledge and I understand and take complete responsibility if any complaint arises.

e. I am fully aware that my thesis supervisor(s) are not in a position to check for any possible instance of plagiarism within this submitted work.

Date : ____/_____/_____ **Pradeepkumar Gajendra Bhale**

Place: Guwahati, India

Thesis Certificate

This is to certify that the thesis entitled **"Attack Detection and Mitigation for IoT Ecosystems: Adaptive, Scalable, and Lightweight Approaches"** being submitted by **Pradeepkumar Gajendra Bhale** to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a record of bonafide research work carried out by him under our supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.

To the best of our knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Prof. Sukumar Nandi
Professor
Department of Computer Science and Engineering
IIT Guwahati, Assam, India
Email: sukumar@iitg.ac.in

Prof. Santosh Biswas
Professor
Department of Computer Science and Engineering
IIT Bhilai, Chhattisgarh, India
Email: santosh@iitbhilai.ac.in

Date : ____/_____/_____

Place: Guwahati, India

Acknowledgments

This thesis is the product of the combined efforts of numerous individuals who have supported me directly or indirectly throughout my exhilarating PhD journey.

First and foremost, I would like to express my profound gratitude to my supervisors, Prof. Sukumar Nandi and Prof. Santosh Biswas. I feel incredibly fortunate to have Prof. Sukumar Nandi as my supervisor. He has granted me the autonomy and time to think independently, never imposing strict deadlines but always reminding me of the time that passes. His greatest attribute is his ability to let students develop according to their potential. His exceptional patience and perseverance are truly admirable. As the saying goes, "In pursuit of perfection, you achieve excellence." Throughout my PhD journey, I occasionally felt as though I had reached a dead end and that my efforts might be fruitless. During these moments, I shared my anxieties with him, and he consistently offered new directions and a glimmer of hope. His guidance allowed me to emerge from negativity with renewed determination to explore the paths he suggested. In addition, he has significantly improved my technical writing by providing invaluable advice on manuscript editing, paragraph flow, and grammar.

I will forever be grateful to Prof. Santosh Biswas for welcoming me as his PhD student and providing assistance throughout my journey, especially during the initial, stressful days when I struggled to find my research direction. He consistently offered support, motivation, and new research perspectives, even when the journal revision process became monotonous. His energy, enthusiasm, and readiness to discuss research over the phone were invaluable. Alongside our technical conversations, we shared many candid discussions on various topics that I will always cherish. His guidance, including invaluable advice and feedback on my journal paper reviews, has significantly impacted my PhD experience.

I am truly delighted to have Prof. Jatindra Kumar Deka, Prof. Diganta Goswami, and Prof. Partha Sarathi Mandal as esteemed members of my doctoral thesis committee. I express my sincere gratitude for their valuable time, insightful evaluations, and constructive advice for improving the quality of my research work. I am proud to be a student at IIT Guwahati and thankful for the dedicated efforts of everyone who has created a conducive environment for research and academic growth. My appreciation extends to Prof. Diganta Goswami (former Head, Dept. of CSE), Prof. S. V. Rao (former Head, Dept. of CSE), Prof. Jatindra Kumar Deka (current Head, Dept. of CSE), Prof. Gautam Biswas (former Director), Prof. T. G. Sitharam (former Director), and Parameswar K. Iyer (present officiating Director), as well as all Deans and administrative staff. I also extend my heartfelt thanks to the faculty and staff of the Department of CSE for their unwavering support and assistance.

Also, I thank Ministry of Education (MoE), the Government of India, for providing financial assistance throughout the Ph.D. programme. Thanks to various funding agencies and esteemed conferences that awarded travel grants and fellowships encouraging participation in various research venues, I am extremely grateful to funding organisations for providing the necessary computing resources used during the work, including the Information Security Research and Development Centre (ISRDC) under the Information Security Education and Awareness (ISEA) Project (Phase-II), ICPS, Department of Science and Technology (DST), Govt. of India, and Dept. of CSE, Indian Institute of Technology Guwahati.

I would like to extend my gratitude to the friendly and supportive office staff at the Department of CSE, including Mr. Monojit Bhattacharjee, Ms. Gauri, Mr. Gourish Mazumder, and Mr. Prabin Bharali, for diligently handling all office-related issues. I also acknowledge the assistance provided by the scientific officers, Mr. Nanu Alan Kachari, Mr. Bhriguraj Borah, Mr. Nava Kumar Boro, Mr. Raktajit Pathak, and Mr. Pranjit Talukdar. They have always strived very hard to ensure that we do not face any issues related to Internet connectivity, computer peripherals, printer cartridges, etc. On a more personal note, I feel privileged to have picked up so much knowledge from them, and I want to express my gratitude for that.

I would like to extend a special thanks to all security guards, janitors, housekeeping staff, mess and canteen employees, food court workers, hostel caretakers, wardens, medical personnel, and drivers at IIT for ensuring our comfort and convenience during our stay. These individuals are the unsung heroes and lifelines of the institute. Spending an extended period on a residential campus means that friends and colleagues become like family, as we share our joys, sorrows, celebrations, special moments, and birthdays with them. This acknowledgement would be incomplete without expressing gratitude to these cherished individuals who have been an integral part of my journey.

I hold immense gratitude for the teachers who have guided me throughout my educational journey, from my school days to my college life. It is their dedication and hard work that have paved the way for me to stand where I am today. I appreciate them all, but Mr. V. V. Kulkarni, my math teacher, and Mr. Sanjiv Kamble, my Hindi teacher, are exceptional. Their inspiration and direction helped to shape my path. Through his teachings, Mr. Kulkarni provided me with a solid foundation in mathematics that was instrumental in my journey towards engineering. His encouragement drove me to study computer science and engineering. In a similar way, Mr. Kamble's inspirational words encouraged me to set high goals and believe in my abilities. With their unwavering support, I became the first student from my village to pursue engineering and, later, the first to be admitted to IIT for a Ph.D. This acknowledgment is a tribute to the remarkable influence that my teachers have had on my life, and it is my honour to recognise them here. Their guidance not only imparted knowledge but also fueled my determination. I will always be grateful to them for

believing in me and pushing me to go beyond my own limits. I would also like to thank Prof. Shashikala Tapaswi, my M.Tech. supervisor at Atal Bihari Vajpayee-Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior, for his encouragement and guidance in my decision to pursue a PhD degree.

Among other notable mentions are Dipojjwal, Arijit, Ujjwal, Pawan, Akshay, Sumita, Sikha, Debanjan, Manoj, Saurav, and Saurav Kumar, whom I need to acknowledge for being a part of this wonderful journey of doctoral research. And thank you to all the wonderful people I did not mention here who assisted me in my pursuit. I sincerely acknowledge my seniors, Niladri Sett Sir, Subhrendu Sir, Sukanta Dey, Pranav Sir, Bikramjit Sir, and Madhurima Madam, for all their help and support. I would like to thank the anonymous Contiki Cooja and FIT-IoT-Lab users, bloggers, and forum moderators whose responses to various questions helped me when I got stuck. I am obligated to acknowledge my bicycle and cell phone. Both of them have been used so roughly, yet they have always remained my faithful partners. During my time at IITG, I have been grateful to a few special people. I would like to thank Debabrata Sir, Sasmita Madam, Ujjwal Sir Madam, and Swayamprava Madam for all of their love and support, as well as for thinking of me and my wife as members of their family. I am lucky to have them in my life. I thank the doctors team and medical staff at IITG for their healthcare services. A special thanks goes to the library and computer centre staff for their support and making all the resources available to us for learning.

Without having an adequate support from your family members, traveling the doctoral journey could only remain a dream. I want to express my gratitude to my sister, Dr. Pranita Bhale, and my brother-in-law, Dr. Arun Kamble, for inspiring me to pursue doctoral study. I would have missed the amazing voyage of PhD research without their motivation. My younger brother (Pravin) has always been caring and supportive. His advice on numerous personal problems was quite helpful to me. A special thanks goes to my cousins, Pari, Prathibha, Prakash, Priti, and Priya. I have admired their presence in my life. I appreciate Mrs. Arti, my sister-in-law, for always making me feel welcome and taking good care of me whenever I visit my hometown. Thank you, Arti!. The sweetness and hilarity of my two nephews, Harsh and Aayush, have always made me smile☺. Thank you, Aayush and Harsh. I would also like to thank Mr. Rajendra Bhale and Sunil Bhale, my uncles, for their love and affection!.

I am grateful to my sweet daughter Anami and my kind wife Pooja for all of the sacrifices they have made and the patience they have shown throughout this PhD. I am extremely thankful that you have faith in me and have been there for me throughout this arduous and difficult journey. In the midst of all those challenging times, their smiles and embraces served as a boost and braced me to get back to work. Thank you for the special prayers that you made for me to complete this work. I count myself extremely fortunate to

have such a warm and supportive family standing by my side, offering their unwavering love and encouragement.

The utmost level of gratitude is extended to my parents. Their efforts assured me that no hurdle could slow my progress. Thank you, Mother (Aai) (Mrs. Uma Bhale), Father (Papa) (Mr. Gajendra Makaji Bhale) for being so kind, supportive, concerned, and loving. This thesis is a culmination of your blessings! Last but not least, I want to express my gratitude to God for his mercy and favour, which allowed me to successfully complete my PhD journey.

Guwahati, November 2023 **Pradeepkumar Gajendra Bhale**

Abstract

The constant proliferation of Internet of Things (IoT) technology is evidenced by its significant integration into various sectors like the automotive industry, manufacturing, agriculture/agrotech, healthcare, transportation, and smart homes. Despite its usefulness in streamlining and automating processes, IoT technology also exposes numerous security vulnerabilities. To deal with these threats, the literature has proposed intrusion detection systems (IDSs). However, implementing these IDS solutions in IoT ecosystems is not ideal due to the distinct limitations of IoT devices, such as their restricted processing power, memory, and energy resources. Consequently, conventional IDS solutions cause high energy consumption, latency, and network congestion, reducing their effectiveness in IoT environments. The contributions of the thesis revolve around designing IDS solutions that are adaptive, scalable, and lightweight.

In this thesis, we first present the details (IoT network, architecture, applications, operating system, IoT simulators, IoT real testbed, IoT attack, IoT datasets, machine learning, and deep learning methods) of an IoT domain and highlight the significant challenges. The first contribution is an edge-based machine learning (ML) method that makes it easier for IDS to find distributed denial-of-service (DDoS) attacks in IoT networks, such as BashLite and Mirai botnets. The first contribution involves the combination of edge computing techniques with a naïve Bayes (NB) classifier to detect DDoS attacks along with botnet attacks. The proposed IDS model is evaluated using metrics such as accuracy, precision, recall, and F-measure. It demonstrates scalability and achieves comparable memory utilisation in terms of ROM/RAM, energy usage, and response time.

The second contribution of the thesis presents a distributed, lightweight, and energy-efficient Packet Inspection Agent (PIA) designed for the IoT ecosystem. By leveraging the Total Variation Metric (TVM) and Packet Flow Count (PFC), the PIA effectively detects and mitigates low-rate DDoS (LRDDoS) attacks within IoT networks. Moreover, the proposed PIA employs lightweight technology to minimize both the false negative rate (FNR) and false positive rate (FPR), while significantly reducing the time required for LRDDoS attack detection in IoT networks.

In the third contribution of this thesis, we present an IDS specifically designed for detecting mixed-rate DDoS (MRDDoS) attacks in IoT networks. This contribution focuses on a lightweight and transparent IDS with an optimal placement strategy. The placement problem is formulated as a weighted minimum vertex cover problem of a K-uniform hypergraph and solved using an approximation algorithm. The IDS module is based on a Long Short-Term Memory (LSTM) model, where a novel offline training method for LSTM is proposed using WGAN-generated artificial flows. Extensive experiments conducted on

the Contiki and FIT IoT-LAB testbeds provided a comprehensive performance analysis of the proposed schemes. The results demonstrate that our method outperforms the current state-of-the-art approaches by effectively detecting attacks while minimising energy consumption.

Finally, in this thesis, we introduce roaming IDSs that are capable of detecting and mitigating multiple mixed attacks in IoT networks. These attacks include DDoS, rank, and sinkhole attacks. In our approach, a lightweight shadow honeypot is deployed and easily moved to a different node or device within the IoT network, where it is placed in a strategic location that is most likely to be targeted by potential attackers. Markov chain analysis (MCA) is employed to determine the most probable node or device to be attacked based on the current IoT network profile. This analytical model strengthens the validity of the RENO model and enhances its effectiveness in identifying potential attacks. The potential benefits of the RENO solution can be observed when its performance is compared to similar approaches in terms of throughput, energy use, memory use, FPR, FNR, accuracy, and time to detect an attack. In summary, this thesis provides valuable insights into the creation of adaptive, scalable, and lightweight security solutions for various attacks on IoT ecosystems.

Keywords: *Internet of Things (IoT), Intrusion Detection Systems (IDS), Edge Computing, Machine Learning (ML), Naive Bayes Classifier, Nested One-Class Support Vector Machine (NOC-SVM), Distributed Denial-of-Service (DDoS) Attack, Packet Inspection Agent (PIA), Low-Rate DDoS (LRDDoS) Attacks, Mixed-Rate DDoS (MRDDoS) Attacks, LSTM-based IDS, Wasserstein Generative Adversarial Network (WGAN), Rank attacks, Sinkhole attacks, Buffer overflow (BOF) attack, Roaming IDS, Markov Chain Analysis (MCA).*

Table of Contents

List of Figures

Page

List of Tables

Page

List of Algorithms

List of Acronyms

Acronym	Expansion
IoT	Internet of Things
CII	Critical Information Infrastructure
IoHT	Internet of Healthcare Things
VIoT	Vehicular IoT
UAV	Unmanned Aerial Vehicles
SIoT	Satellite IoT
IIoT	Industrial IoT
IDS	Intrusion Detection System
DoS	Denial of Service
MITM	Man-in-the-Middle
RBAC	Role-Based Access Control
TLS	Transport Layer Security
DTLS	Datagram Transport Security
OTA	Over-the-Air
AI	Artificial Intelligence
ML	Machine Learning
AML	Adversarial Machine Learning
LRDDoS	Low-rate DDoS
MRDDoS	Mixed-rate DDoS
PIA	Packet Inspection Agent
FNR	False Negative Rate
FPR	False Positive Rate
PSD	Power Spectral Density

WGAN	Wasserstein Generative Adversarial Network
SHA	Sinkhole Attack
RFID	Radio-Frequency Identification
LPWANs	Low-Power Wide-Area Networks
OS	Operating System
WSNs	Wireless Sensor Networks
IPv6	Internet Protocol version 6
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
CoAP	Constrained Application Protocol
RPL	Routing Protocol for Low-Power and Lossy Networks
RAM	Random Access Memory
ROM	Read-Only Memory
RF	Radio Frequency
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
LR	Logistic Regression
ANN	Artifcial Neural Network
EL	Ensembles Learning
RBM	Restricted Boltzmann Machine
DBN	Deep Brief Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
FE	Feature Extractor
PCA	Principle Component Analysis
MEMU	Memory utilization
EBNB	Edge-Based Naive Bayes
TVM	Total Variation Metric
PFC	Packet Flow Count
PD	Probability Distribution
VM	Variation Metric
RT	Response Time
CAIDA	Cooperative Association for Internet Data Analysis

LLN	Low Power Lossy Network
BR	Border Router
HIDS	Host-based Intrusion Detection System
NIDS	Network-based Intrusion Detection System
DODAG	Destination-Oriented Directed Acyclic Graph
SDN	Software-Defined Networking
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network
6BR	6LoWPAN Border Router
BOF	Buffer Overflow
ADS	Anomaly Detection System
AS	Attack Score
ASM	Attack Score Module
LWSHP	Lightweight Shadow Honeypot
MCM	Markov Chain Model
PFBG	Packet Flow Behavior Graph
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
HTTP	Hypertext Transfer Protocol
DNS	Domain Name System
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
HTTPD	HTTP Daemon
MM	Multiple Mix (used in the context of attack types)
ADT	Attack Detection Time (a performance metric)
THP	Throughput (a performance metric)
MEMC	Memory Consumption (a performance metric)
ACC	Accuracy (a performance metric)
TCP/TLS	Transmission Control Protocol/Transport Layer Security
UDP/DTLS	User Datagram Protocol/Datagram Transport Layer Security

List of Symbols

Symbol	Description
b	Class labels representing "normal" and "abnormal" data
A	Feature vector containing features such as source IP address, port number, packet size
a_i	Individual features within the feature vector A
$P(b/A)$	Conditional probability of class b given the feature vector A
$P(A/b)$	Conditional probability of feature vector A given class b
$P(b)$	Prior probability of class b
$P(a_i/b)$	Conditional probability of feature a_i given class b
$P(b/a_1, \ldots, a_n)$	Conditional probability of class b given individual features $a_1, \ldots, a_n$
α	Proportional to symbol
MAP	Maximum a posteriori probability
$\arg\max$	Argmax operator, returns the argument (class b) that maximizes the expression
N_{6BR}	Number of 6LoWPAN Border Routers
N_{PIA}	Number of Packet Inspecting Agents
M_{GTV}	Generalized Total Variation Metric
ϕ_1 *and* ϕ_2	Discrete distributions across distinct random values
k	Number of distinct random values
A_x *and* A_y	Random unique samples
μ	A parameter in the M_{GTV} equation
M_{TV}	Total Variation Metric
$P(x)$	Probability distribution
I_i	Packet frequency
A_{PIAT}	Packet frequency

χ	Intermediate variable
VM	Variation metric
$P(EX_i)$	External traffic probability
ET_S	Sample time window for external traffic
A_r A_v A_h	Different scenarios for LRDDoS attack detection
T_S	Sample time window
IX	Internal traffic
$\delta_{traff.}$	Threshold for internal traffic
$\Delta_{ipcount}$	Threshold for IP address flow
IP_i	IP Address of an attacking node
$INFO_{GET}$	Information received for LRDDoS attack detection
X_{Block}	Traffic flow blocking rule
K	Parameter representing the number of hops
$\mathbb{H}$	Hypergraph
$\mathbb{V}$	Vertex set
$\mathbb{E}$	Edge set (hyperedges)
VC	Vertex Cover
Z	Optimal value of linear program
$\omega(\mathbb{V}_i)$	Weight assigned to vertex $\mathbb{V}_i$
$x(\mathbb{V}_i)$	Decision variable for vertex $\mathbb{V}_i$
S^*	Optimal solution for the minimum weighted vertex cover
$Generator\ (\zeta)$	Neural network component in GAN that generates fake data
$Discriminator\ (\vartheta)$	Neural network component in GAN that discriminate fake data
D_{loss}	Discriminator loss in WGAN
G_{loss}	Generator loss in WGAN
$\sigma\,(\cdot)$	Sigmoid function
i_t, f_t, O_t, $\tilde{C}$	Variables in LSTM cell equations
W_i, W_f, W_o	Weight matrices in the LSTM cell
X_{real}, X_{norm}	Real and normalized values in feature normalization
A_F	Activation Function

b_i, b_f, b_o, b_c	Biases in the LSTM cell.
x_t, h_t, t	Input, output, and time in the LSTM cell.
X_{min}, X_{max}	Minimum and maximum values in feature normalization.
P	Matrix of transition probabilities.
A_p	Attack probability.
t	Time duration.
$t_i\left(e_j^{\beta}\right)$	Attack score for the ith object after the jth event of attack β.
I	Set of objects (nodes) under consideration.
$I^*\left(e_j^{\beta}\right)$	Maximum attack rating of an object in set I for event e_j^{β}.
$Node_{ID}$	Identification of IoT nodes.
$Node_{rank}$	Rank of an IoT node.
$Timestamp$	Timestamp associated with an alert.
n	Number of alerts generated by ASM during every time interval t.
P_{ij}^t	Transition probability from state i to state j at timestamp t in the Markov chain.
$Q_t = i$	The system is in class i at timestamp t in the Markov chain.
$Q_{t+1} = j$	The system is in class j at timestamp $t+1$ in the Markov chain.
$Totalevent\,(Q_t = i)$	The total number of events where the system was in class i.
$Alert_{score}$	The calculated attack score for a set of alerts.

"The Internet of Things is not just a concept; it is a paradigm shift that requires a fundamental rethinking of security."

- Steve Grobman

CHAPTER

1

Introduction

The Internet of Things (IoT) is a system made up of interconnected physical devices, vehicles, buildings, and other objects embedded with sensors, software, actuators, and network connectivity, which enables them to collect and exchange data over the Internet [6, 7]. IoT devices, such as smart wearable devices, smart home appliances, smart fire alarms, smart vehicles, smart bicycles, and smart camera systems frequently collect personal data and provide a range of features to automate and assist our day-to-day lives. Hence, these devices have grown in popularity in recent years. This is due to advancements in sensor technology, cost-effective implementation of IoT solutions, improved efficiency and productivity in industries, new business models, increased connectivity, and greater awareness and understanding of IoT benefits [8, 9].

The proliferation of IoT devices is not restricted to the domestic environment; it also enables smart operations in Critical Information Infrastructure (CII)[1] sectors, as depicted in Figure 1.1. These sectors are essential systems and services that underpin a country's economy, security, and social wellbeing. These sectors include energy, transportation, water and sewage, health, emergency and financial services, and digital communication networks [11]. In the last few years, the market has seen a number of noteworthy IoT projects that have gained a lot of attention. Figure 1.2 depicts a few examples of notable IoT projects that have attained significant market penetration. This figure depicts the global distribution

[1] Critical Information Infrastructure is the country's most important system, network, and asset. These include the energy, transportation, water supply, and communications [10]

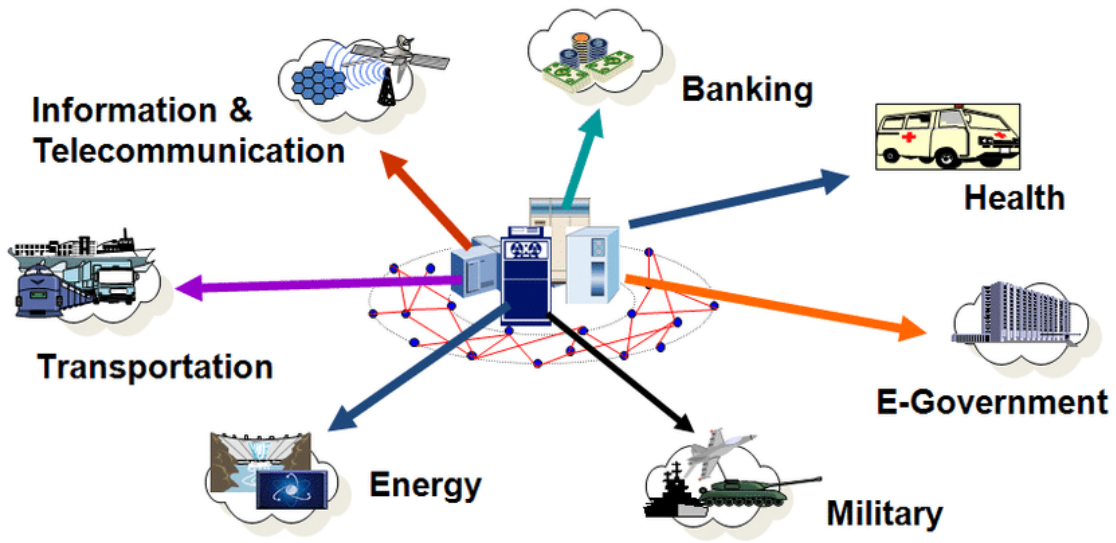


Figure 1.1: Critical Information Infrastructure

of these IoT projects throughout the Americas, Europe, and Asia/Pacific regions. The illustration shows that the Americas leads in health care and smart supply chain projects. On the other hand, Europe has demonstrated greater engagement in moving forward initiatives related to smart cities [12, 13]. Figure 1.3 shows a visual picture of how the market share of different IoT projects is spread worldwide [13]. According to the statistics, there is a noteworthy trend in which IoT projects that are centred around industries, smart cities, smart energy, and smart vehicles have acquired huge market shares, greatly outperforming other project categories.

In the near future, sixth-generation (6G) wireless communication technology will make it easier to use new IoT applications like the Internet of Healthcare Things (IoHT), Vehicular IoT (VIoT) and autonomous driving, Unmanned Aerial Vehicles (UAV), Satellite IoT (SIoT), and Industrial IoT (IIoT) [14, 15, 16]. However, IoT ecosystems [2] are vulnerable to cyberattacks due to their reliance on Information Communication Technology (ICT), growing interconnectivity, and IoT devices. The majority of IoT devices in use today were made without much or any thought to cyber security [17]. Meanwhile, security breaches in IoT pose severe consequences, including network congestion, system disruptions, data corruption, application downtime, financial loss, system blackouts, and even loss of life in

[2] IoT ecosystems refer to interconnected networks of devices, sensors, and software that collaborate and communicate seamlessly to collect, exchange, and analyze data for various applications and services.

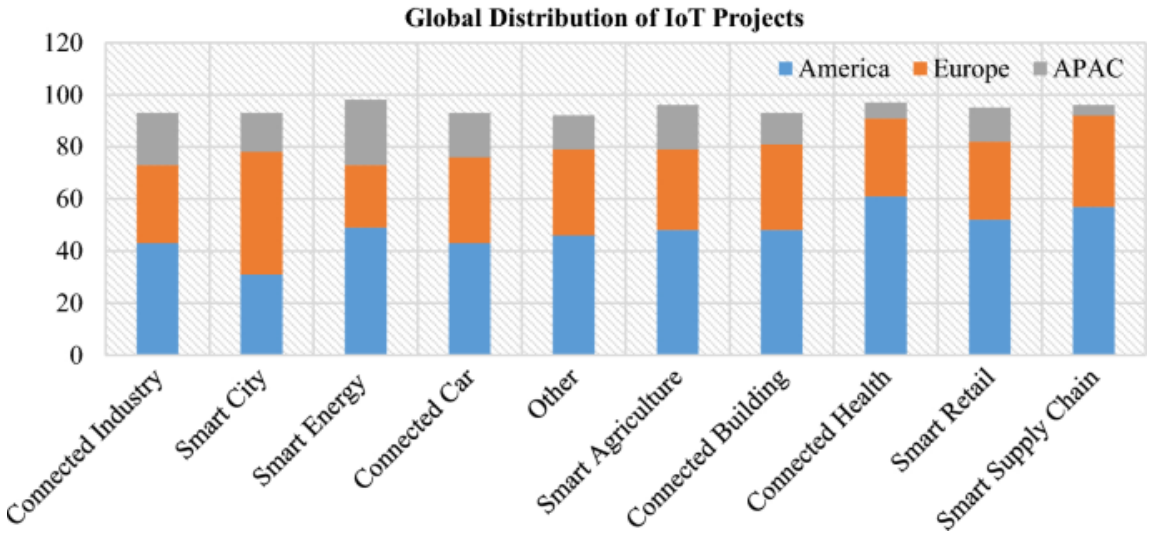


Figure 1.2: IoT Project Landscape: America, Europe, and Asia/Pacific Distribution

critical use cases [18, 19, 20]. Figure 1.4 illustrates that there are more than 112 million cyberattacks on IoT systems by 2022 [21]. This represents a significant increase compared to the approximately 32 million incidents recorded in 2018. IoT malware incidents increased by 87 percent year over year in the most recent year on record. In order to reduce attacks on the IoT ecosystem, IoT-enabled security solutions are needed.

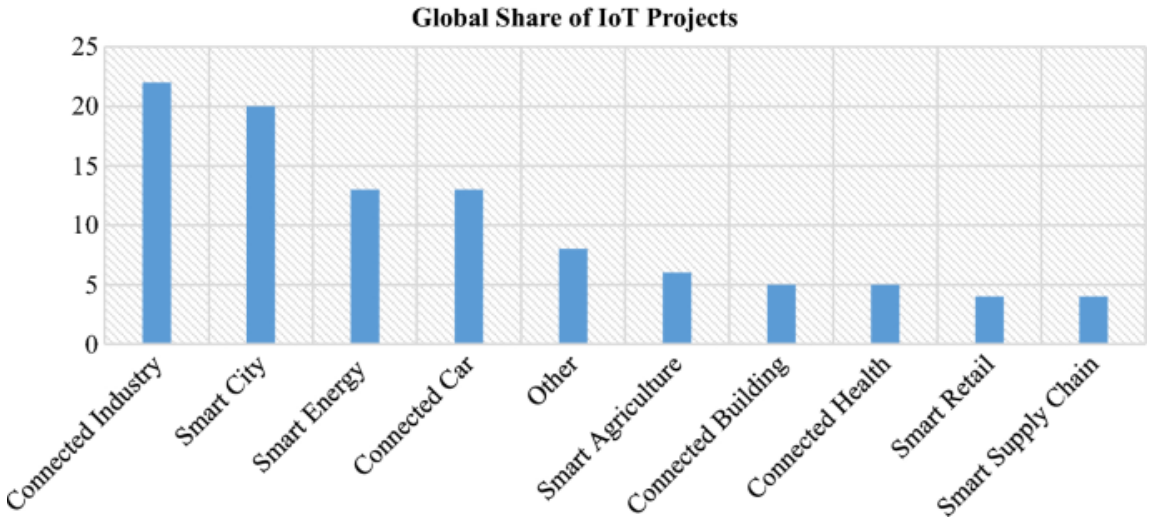


Figure 1.3: Worldwide Distribution of IoT Projects: Global Share

The main factors that make IoT ecosystems susceptible to cyber attacks are their inherent characteristics like limited computational power and their heterogeneity. Specifically, IoT devices with restricted computational power, memory, radio bandwidth, and battery

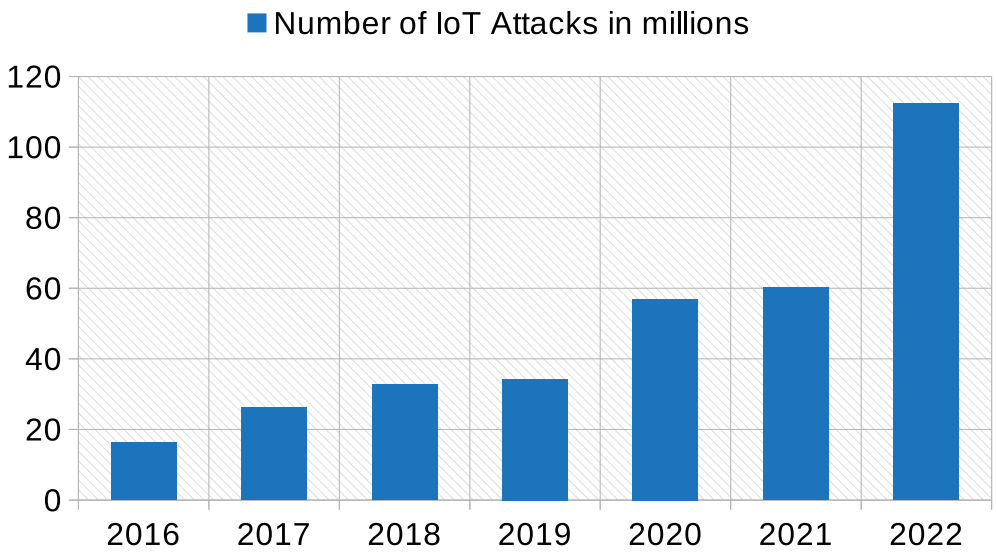


Figure 1.4: Rising Tide of IoT Cyber Attacks: Annual Statistics 2016-2022

resources cannot feasibly perform computationally demanding and latency-sensitive security tasks that produce heavy computation and communication loads. As a result, it is not possible to implement complex and robust security measures. Furthermore, the heterogeneity of IoT devices in terms of hardware, software, and protocols poses a significant challenge in developing and deploying security mechanisms that can cope with the scale and variety of devices. Therefore, it is clear that there is a significant gap between the security requirements and the security capabilities of currently available IoT ecosystems [3].

1.1 Motivation for the Research Work

The IoT is seen as the next significant technological revolution, reshaping our interaction with the physical world [22, 23]. However, these technologies are also associated with significant security flaws [24, 25, 26, 27]. Many reputable companies and research organizations have shown that the IoT ecosystem can have various security issues. These include Denial of Service (DoS) attacks, Man-in-the-Middle (MITM) attacks, Rank attacks, Sinkhole attacks, Wormhole attacks, Buffer Reservation attacks, Fragment Duplication, node capture, and IP Address Spoofing. Recently, IoT devices have been used in botnets, such as Mirai, to launch some of the biggest Distributed Denial of Service (DDoS) and spam attacks.

[3] In this thesis, the terms IoT network and IoT ecosystem are used interchangeably.

This thesis is motivated by the critical need to enhance security for IoT ecosystems, which are often seen as vulnerable targets for cyberattacks. The primary goal is to provide security measures to protect, detect, and mitigate cyber threats impacting IoT ecosystems.

As part of enhancing IoT ecosystem security, many studies have focused on specific security goals like confidentiality, integrity, availability, authentication, authorization, non-repudiation, privacy, resilience, auditing, and accountability [28, 29, 30]. However, applying these security measures uniformly across different IoT devices is challenging due to their diverse nature. Intrusion Detection Systems (IDS) design is a popular method for detecting various attacks in IoT ecosystems [31, 32, 33], and many studies have suggested using these IDS to address various attacks [34, 35]. However, the primary emphasis of these studies is on attack detection, with less attention paid to managing vast amounts of data from intelligent devices, where to execute IDS solutions, improving computational capacity, energy usage, scalability, and overall security.

1.2 Research Questions

This thesis critically examines the existing security approach employed in the IoT ecosystem and identifies several key issues. Our research is primarily based on the following questions.

RQ1: ***What are the state-of-the-art (SoTA) IDS security solutions proposed for detecting DDoS attacks, Rank attacks, Buffer Reservation attacks, sinkhole attacks, buffer overflow attacks, and botnet attacks in IoT networks? What are the current research gaps in the existing SoTA security solutions?***

SoTA IDS security solutions are intended to detect and mitigate various threats within the IoT ecosystem. To address these security issues, researchers have developed novel strategies. For DDoS attacks, machine learning-based methods utilising network traffic analysis, anomaly detection algorithms, and behaviour modelling have shown promising results. Rank attacks, which involve manipulating network metrics, can be detected using statistical analysis and anomaly detection techniques. Buffer reservation attacks can be prevented by improving buffer management algorithms and monitoring buffer use in real time. Anomaly detection and traffic analysis can discover sinkhole attack redirection patterns. Secure coding and runtime monitoring of programme execution can prevent buffer overflow

attacks. Machine learning techniques, network traffic monitoring, and behaviour modelling identify infected devices and coordinated harmful behaviours in botnet attacks.

However, there are still research gaps that need attention. A potential obstacle is the resource-constrained nature of IoT devices, which requires the development of lightweight IDS solutions that consume minimal power and computational resources. Another barrier to the implementation of unified IDS systems is the lack of standardisation and interoperability among various IoT devices and platforms. In addition, the varied and dynamic nature of IoT networks creates obstacles for the accurate and prompt identification of attacks. As threats get more sophisticated, IDS algorithms and approaches must be constantly researched and updated. Our research should focus on these gaps, exploring novel approaches to overcome resource limitations, improving interoperability, enhancing real-time detection capabilities, and developing adaptive IDS systems capable of addressing emerging threats in the constantly evolving IoT landscape.

RQ2: ***What type of security solution can be implemented into an IoT ecosystem that can be implemented on resource-constrained and heterogeneous IoT devices?***

In an IoT ecosystem characterised by heterogeneous and resource-constrained devices, the implementation of security solutions requires careful consideration. One way to do this is to use lightweight security solutions made for IoT devices with limited resources. Effective encryption and authentication may be achieved with lightweight cryptography methods while using a small amount of processing power and storage space. Secure communication protocols, such as Transport Layer Security (TLS) or Datagram Transport Security (DTLS), provide encrypted and authenticated data transfer while taking into consideration the resource constraints of IoT devices. Access control methods like role-based access control (RBAC) and lightweight identity protocols make it possible to control device access and authentication without adding too much work. In addition, behaviour-based anomaly detection algorithms can detect deviations from normal device behaviour, enabling the identification of potential security vulnerabilities without relying solely on resource-intensive signature-based methods. Furthermore, over-the-air (OTA) update systems and patch management solutions may guarantee that IoT ecosystem receive frequent updates and security fixes without putting too much load on their resources. By combining these security measures, the IoT ecosystem can balance resource limitations and security. This approach

allows for effective defence against various threats while considering the heterogeneous nature of IoT devices.

RQ3: ***How can we design an IDS that is effective at detecting attacks, while also considering the management of large amounts of data from smart devices, enhancing computational power, scalability, and overall security?***

Designing an effective IDS for IoT systems requires careful consideration of data management, computing power, scalability, and security. Data management strategies are required to manage large amounts of data. These approaches must include systems for optimal storage, processing, and retrieval. Utilising lightweight algorithms, parallel processing, distributed computing, and hardware acceleration can maximise computational capacity. Scalability is improved through scalable architectures such as microservices and containerization, use of cloud resources, and elastic scaling. Artificial intelligence (AI) and machine learning (ML) techniques improve attack detection efficiency by training IDS models on large data sets and using anomaly detection, deep learning, and ensemble methods. Advanced network traffic analysis methods, such as packet inspection and flow-based analysis, enable detection of suspicious activity and attack patterns. Real-time monitoring and threat intelligence feeds enhance detection. Security aspects include encryption, access control, secure communication protocols, and privacy techniques. IDS Designs that address these aspects can successfully detect threats, manage massive amounts of data, and improve compute performance, scalability, and overall security in IoT ecosystems to protect smart devices and overall infrastructure.

RQ4: ***How to choose a suitable device of IoT ecosystem for IDS execution?***

Considerations for selecting an appropriate device for IDS execution in an IoT environment include its processing capabilities, power requirements, network connectivity, scalability, security features, compatibility, cost-effectiveness, longevity, and manufacturer support. Evaluating the device's computational capabilities ensures that it can effectively manage IDS algorithms and data analytics. Energy efficiency is critical, especially for battery-powered devices, to ensure long operation without frequent recharging. Network connectivity options should be aligned with the IoT ecosystem and allow seamless integration

and communication with other devices. Scalability is very important to accommodate the growing size of the network. Secure boot techniques and hardware security modules improve device security. Modular architecture and compatibility with IDS software enable easy integration into the IDS infrastructure. Cost effectiveness balances device capabilities with the cost of efficient resource allocation. Longevity and vendor support ensure ongoing updates and protection against new threats. Consideration of these factors enables the selection of an appropriate device that has the necessary characteristics for efficient IDS execution, contributing to the overall security of the IoT ecosystem.

RQ5: ***How do distributed, edge-enabled, and roaming IDS contribute to improving overall security in the IoT ecosystem?***

Distributed IDS, edge-enabled IDS, and roaming IDS are all important components for enhancing IoT ecosystem security. Distributed IDS distributes IDS functionality across multiple nodes, which can improve scalability, reduce single points of failure, and enhance performance. This distribution of IDS capabilities ensures efficient detection and response to security threats, effectively accommodating the increasing number of devices present in IoT environments.

Edge-enabled IDS brings intrusion detection capabilities closer to the edge devices or gateways in the IoT network. By performing IDS tasks at the edge, closer to where data is generated, edge-enabled IDS reduces reliance on centralised processing, minimises data transmission to a central server, and enhances real-time threat detection and response. This approach is particularly useful in resource-constrained IoT environments, where limited resources or intermittent connectivity may pose challenges for centralised IDS deployment.

Roaming IDS addresses the dynamic nature of IoT devices moving across different networks or locations. In the IoT ecosystem, devices frequently change their network connections, making it difficult for traditional IDS systems to effectively track their activities. Roaming IDS adapts to these changes, ensuring continuous monitoring and detection of device behaviour as they move across networks. By allowing IDS capabilities to roam with the devices, it provides consistent security coverage and maintains threat detection regardless of the devices' location or network connectivity. Overall, distributed IDS, edge-enabled IDS, and roaming IDS all contribute to enhancing security in the IoT ecosystem. They provide

scalability, real-time threat detection, and continuous monitoring capabilities, enabling effective intrusion detection and response while considering the distributed and dynamic nature of IoT devices.

RQ6: ***Can machine learning-based IDS approaches support the automatic detection of a range of cyber attacks based on network packet features collected from a range of IoT devices?***

Machine learning (ML) based IDS systems can automatically detect various cyberattacks by analyzing network packet features collected from a diverse range of IoT devices. These systems use ML algorithms to detect patterns and anomalies in the data that can be used to identify different types of attacks. By training on labeled datasets that include both normal and malicious network traffic, the algorithms can learn the characteristics of different attacks and make accurate predictions in real time.

ML-based IDS approaches offer several advantages in the context of IoT environments. They can adapt to evolving attack techniques and detect previously unknown attacks. This flexibility allows them to detect sophisticated and emerging threats that are difficult to detect using traditional rule-based or signature-based detection methods. In addition, these approaches can handle the large-scale and dynamic nature of IoT device networks. Regular updates and continuous training are crucial to ensure the effectiveness of ML based IDS systems in detecting evolving attack vectors and protecting IoT ecosystems.

RQ7: ***Can AML techniques be used to evaluate the robustness of a ML based IDS for the IoT?***

Adversarial Machine Learning (AML) can help strengthen ML-based IDS in the IoT ecosystem. In AML, adversarial attacks are studied, and defensive measures are designed to determine how vulnerable ML models are to such attacks. ML-based IDS can be tested for adversarial attempts to evade detection or manipulate its behaviour using AML techniques.

Security researchers can use AML assessments to understand the vulnerabilities of the ML-based IDS system and develop mitigation measures. By testing and refining the ML models against adversarial attacks, the IDS can detect a wide range of cyber threats in the IoT ecosystem. With this methodology, ML-based IDS systems remain effective and trustworthy against new attack methods and protect IoT networks and devices.

RQ8: ***Can adversarial training enhance the robustness of a ML based IDS for the IoT?***

Adversarial training is a method that can be used to make an ML-based IDS more reliable for the IoT. Adversarial instances during training make the ML model more robust against real-world attacks.

An adversarial example is a carefully crafted input designed to trick the ML model into making a mistake. Small, undetectable alterations to benign inputs produce adversarial instances. The ML model learns to detect and respond to these small manipulations by being exposed to adversarial samples during training. This improves the generalization of the IDS model and its resistance to adversarial manipulations.

1.3 Thesis Contributions

In this section, we present a brief overview of the contributions of the thesis.

1.3.1 DDoS attacks including Botnet attacks detection and mitigation in IoT ecosystem (Contribution 1)

As a response to RQ1, RQ2, RQ3, RQ4, and RQ5, this contribution proposes an adaptive and energy-efficient solution for detecting DDoS attacks in IoT networks. The main objectives of Contribution 1 are:

- The proposed security solution is to protect against DDoS attacks, including BashLite and Mirai botnets. These attacks are some of the most common and well-known threats to IoT ecosystems.

- Our research involves exploring a technique to identify and choose optimal features from network traffic data.

- The suggested strategy leverages ML techniques to achieve the best performance in identifying attacks. This strategy improves the accuracy of attack detection. Appropriate hyperparameters are selected for training effective models like the Naive Bayes Classifier and the Nested One-Class Support Vector Machine (NOC-SVM) to identify botnet attacks.

- The proposed work is being evaluated using CICIDS2017, Kitsuni, BoT-IoT, IScX, KDD99 cup, N-BaIoT, and an in-house generated dataset.

1.3.2 Low rate DDoS (LRDDoS) attack detection and mitigation in IoT ecosystem (Contribution 2)

To address RQ1, RQ2, RQ3, and RQ4, this contribution includes an investigation of low-rate DDoS (LRDDoS) attack detection and mitigation in IoT ecosystems. The main objectives of Contribution 2 are:

- The proposed method introduces a distributed, lightweight, and energy-efficient Packet Inspection Agent (PIA). Multiple PIAs cooperate with each other in a distributed environment. They analyze network packets based on Total Variation Metric (TVM) and Packet Flow Count (PFC) within the IoT network.

- The method generates simulated attack and non-attack traffic using the Contiki Cooja simulator. It detects LRDDoS attacks by analysing network traffic from real and compromised nodes. The technique can also be adapted to internal and external network traffic in the IoT ecosystem to detect and mitigate LRDDoS attacks in the future.

- The method, proposed and implemented by the PIA, is capable of monitoring the traffic characteristics within the IoT ecosystem. Through detailed analysis, it precisely detects and mitigates LRDDoS attacks while minimizing the False Negative Rate (FNR) and False Positive Rate (FPR).

1.3.3 Mixed Rate DDoS attacks (HrDDoS and LrDDoS) detection in IoT ecosystem (Contribution 3)

To address the research questions RQ1, RQ2, RQ3, RQ4, RQ5, RQ6, RQ7 and RQ8, this chapter introduces two contributions. The first contribution focuses on detecting mixed-rate DDoS (MRDDoS) attacks using an edge-enabled Power Spectral Density (PSD) approach. The second contribution involves detecting and mitigating MRDDoS attacks using a lightweight and transparent IDS with an optimum placement strategy (OPTIMIST).

It also incorporates Wasserstein Generative Adversarial Network (WGAN) techniques to enhance the robustness of the IDS. The main objectives of Contribution 3 are as follows:

- Unlike existing works which focus either on high-rate or low-rate DDoS, this work provides a solution for MrDDoS attack detection, which can detect and mitigate both high and low-rate DDoS attacks.
- The first contribution of this chapter incorporates distributed, lightweight, and energy-efficient edge devices. It also uses Power Spectral Density (PSD) to detect MRDDoS attacks.
- The second contribution of this chapter proposes a novel training method to build the IDS solution. WGAN is used to generate artificial flows from public data sets as well as in-house-generated data sets to reduce the distribution bias of the data sets. The WGAN-generated flows are mixed with the public and in-house generated training data sets and used for LSTM model training.
- A novel hybrid IDS placement algorithm is proposed, which runs transparently without incurring any network overhead. The IDS node selection is optimized, which balances energy overhead and IDS coverage. The problem is formulated as the weighted minimum vertex cover problem of a K-uniform hypergraph, and an approximation solution is provided.
- Extensive experiments on Contiki and FIT IoT-LAB testbed are done for competitive performance analysis of the proposed scheme. The results show that our proposed scheme is most effective in detecting the attacks while consuming minimum energy compared with the existing benchmark protocols.

1.3.4 Multiple Mix attacks detection and mitigation in IoT ecosystem (Contribution 4)

In this contribution, several research questions, namely RQ1, RQ2, RQ3, RQ4, RQ5, and RQ8, are addressed by presenting comprehensive strategies to detect and mitigate multiple mixed attacks. These attacks include DDoS attacks, rank attacks, and sinkhole attacks. The contribution is further divided into two sub-contributions. The first sub-contribution

focuses on detecting sinkhole attacks by placing detection modules at the edge devices. In the second sub-contribution, detecting multiple mixed attacks, incorporating roaming attack detection modules. The primary objectives of Contribution 4 are as follows:

- In the first sub-contribution, we proposed an edge-assisted hybrid intrusion detection system (EaHIDS) that uses the Expectation-Maximization approach for the Gaussian Mixture HMM to detect and mitigate Sinkhole Attacks (SHAs) within the IoT landscape.
- We employed the SHAP (SHapley Additive exPlanations) methodology for feature selection. We chose this method because of its excellent feature selection capabilities coupled with a minimal computational burden.
- To ensure comprehensive result analysis, we utilized a combination of publicly available dataset and in house generated dataset. The in house generated data was produced with the assistance of our Distributed Data Collection module.
- In the second sub-contribution, we proposed a roaming IDS that can identify rank attacks, sinkhole attacks, and DDoS attacks and proposed an IDS named as RENO.
- A roaming IDS installation based on an attack score (AS) and a Markov chain analysis of IoT networks is demonstrated.
- Extensive experiments are conducted on the Contiki-Cooja and the FIT IoT-LAB testbeds to evaluate the competitive performance of the proposed RENO security solution. In comparison to current state-of-the-art techniques, the findings demonstrate that our suggested security solution is scalable and best at identifying attacks while consuming the least amount of power and memory

1.4 Thesis Organisation

The remaining part of the thesis is organised into the following chapters:

- **Chapter 2 - *Background and Literature Survey:*** This chapter provides all the basic details about IoT networks, architecture, applications, operating systems, simulators, and IoT attacks. It also explains the IoT datasets, background, and related works necessary to propose the contributions in the following chapters.

- **Chapter 3 - *DDoS attack detection and mitigation in the IoT ecosystem:*** This chapter presents novel edge-enabled ML-based approaches to detect DDoS attacks, including Botnet attacks in IoT environments. This Chapter presents contribution 1.

- **Chapter 4 - *Low-rate DDoS (LRDDoS) attack detection and mitigation in IoT ecosystem:*** In this chapter, we present a solution for detecting and mitigating Low-rate DDoS (LRDDoS) attacks within the IoT ecosystem. Our approach introduces a lightweight distributed packet inspection agent designed to identify LRDDoS attacks in IoT networks. By implementing this agent, we aim to enhance the security of IoT systems by effectively detecting and combating LRDDoS attacks. This Chapter presents contribution 2.

- **Chapter 5 - *Mixed Rate DDoS attacks (HRDDoS and LRDDoS) detection and mitigation in the IoT ecosystem:*** This chapter introduces two sub-contributions to detect mixed-rate DDoS attacks in the IoT ecosystem. The first sub-contribution uses edge-based PSD analysis. The second sub-contribution, on the other hand, combines WGAN with LSTM-based IDS. The first sub-contribution emphasises an edge-based approach, while the second emphasises distributed IDS deployment utilising a K-uniform hypergraph for attack detection. These two sub-contributions substantially identify and mitigate mixed-rate DDoS attacks in the IoT environment. This Chapter presents contribution 3.

- **Chapter 6 - *Multiple Mix attacks detection and mitigation in IoT ecosystem:*** In this chapter, we describe a complete security strategy for dealing with a variety of attacks, such as DDoS attacks, buffer reservation attacks, rank attacks, and sinkhole attacks. Our solution is based on the implementation of a roaming strategy using

Shadow Honeypots. This strategy aims to strengthen IoT ecosystem security against a variety of attacks. This Chapter presents contribution 4.

- **Chapter 7 - *Conclusions and Future Work:*** In the concluding chapter, significant conclusions are drawn based on the findings of the thesis work. Additionally, the chapter explores prospective directions for future research, highlighting areas that that demand more investigation.

"Connecting everything brings everyone together, including the hackers. Security is not an option; it is a requirement."

- Stu Sjouwerman

CHAPTER

2

Background and Literature Survey

In this thesis, we contribute to several aspects of the IoT ecosystem security. Each chapter is dedicated to a specific research problem and includes a targeted literature survey relevant to that problem, rather than providing a broad, overarching survey. This chapter outlines the background and fundamental concepts of IoT networks utilised throughout the thesis. First, the IoT reference model and the available operating systems for smart devices are described. Next, simulators and real test-bed used for IoT research are explained along with potential attacks. Finally, brief descriptions of some of the existing works are presented.

2.1 Internet of Things (IoT)

The concept of the IoT was first introduced by Kevin Ashton, who proposed the integration of RFID technology in supply chains with the Internet [6]. As technology has advanced, the definition of IoT has expanded to encompass a broader range of applications across various domains, such as smart cities, agriculture, manufacturing, energy management, transportation, home automation, environmental monitoring, and healthcare [36, 37, 38]. Currently, IoT refers to a framework of interconnected electronic devices equipped with software, actuators, sensors, and networking capabilities. This configuration enables these devices to independently gather and transmit data from their environment, eliminating the need for human intervention [39, 40].

The growth of IoT technology can be attributed to many interconnected factors syner-

gistically contributing to its widespread adoption across various industries and applications [41, 42]. The seamless connection between devices and systems is made possible by enhanced wireless networks like 5G and Low-Power Wide-Area Networks (LPWANs). The miniaturisation and cost reduction of hardware components, including sensors, actuators, microcontrollers, gateways, and user interface devices, make IoT solutions more accessible and affordable. The availability of cloud computing infrastructure supports large-scale IoT implementations since it offers processing, analytics, and storage capabilities. Data analytics and artificial intelligence (AI) advances make it possible for businesses to get useful information from the data that IoT devices generate. This helps businesses make better decisions and work more efficiently. Standardization and interoperability, coupled with an emphasis on security and privacy, further facilitating IoT adoption. Finally, the increasing consumer demand for smart, connected devices contributes to the rapid expansion of IoT technologies across various sectors.

2.2 IoT Architecture

IoT architecture refers to the overall structure and design of an IoT system, which includes its hardware, software, communication protocols, data management and analysis, and security features. There are different types of IoT architecture, but most of them follow a similar framework that consists of the following layers, as shown in Figure 2.1.

1. Perception Layer: It is the lowest layer in the IoT architecture. It comprises small devices such as RFID tags, sensors, and actuators. These devices have limited resources such as power, processing, storage, and communication interfaces. In order to reduce power consumption, they use microcontrollers with less RAM and ROM and low-power radios. The majority of IoT devices use batteries, although they can also be powered by the mains or by energy-harvesting components. The perception layer devices can be stationary or mobile, however stationary devices are more common.

2. IoT Access Network Layer: This is the second layer of the IoT architecture, which include several communication technologies, such as 6LoWPAN, Bluetooth Low Energy, LoRa, LoRaWAN, WiFi, ZigBee, RF, and Thread. Each technology has different properties in terms of range, data rate, power consumption, and scalability. Some

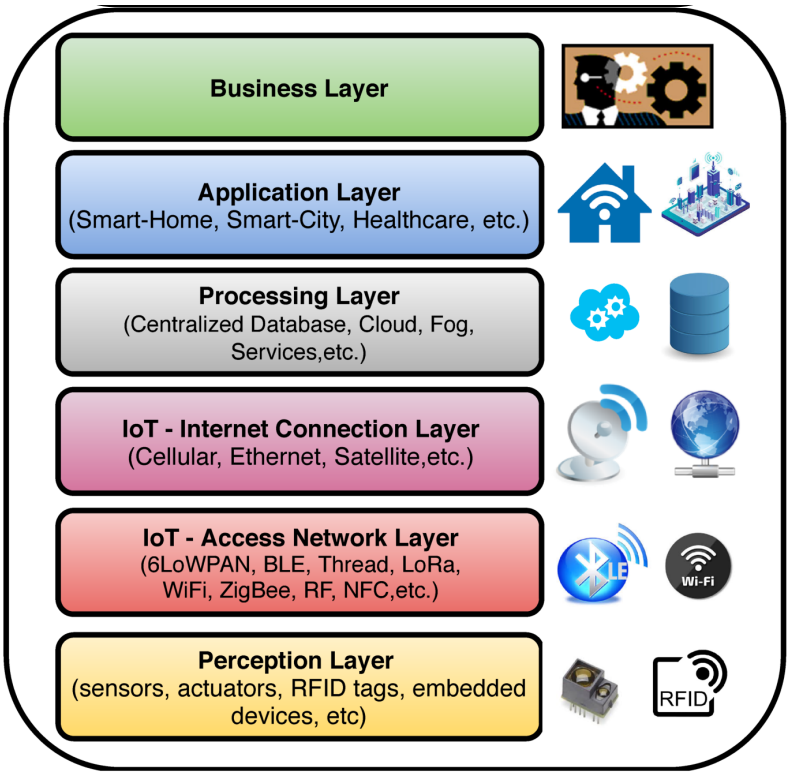


Figure 2.1: Generic IoT architecture

are designed for specific applications like Thread for smart homes. Most of these technologies require a gateway or border router to connect the IoT network nodes to the internet.

3. IoT-Internet Connection Layer: It is the third layer in the IoT architecture and it facilitates the connection between the inner IoT network and the internet using Ethernet, cellular networks, or satellite communication. Typically, a border router or gateway is required to establish the connection between the IoT network and the internet.

4. Processing Layer: This layer is responsible for processing, analyzing, and storing the data collected from the Perception Layer. Designers can choose between centralized or distributed storage and processing systems, such as cloud or fog computing environments. The processed and analyzed data is used to provide middleware services in this layer. This layer is critical in IoT design because it pulls useful information from gathered data, which might be huge in volume, variety, and validity.

5. Application Layer: This is the fifth layer of the IoT architectural design. The purpose of this layer is to provide end users with features and services. This layer facilitates communication and data exchange between IoT devices and user applications. At this layer, the IoT system can provide various services such as monitoring, control, and automation of devices, data analytics, and decision-making. The applications can be web-based, mobile-based, or stand-alone, and they are accessible from a variety of devices, including smartphones, tablets, and desktops.

6. Business Layer: This is a critical component of the Internet of Things (IoT) architecture that focuses on the business aspects of IoT deployment. It provides the necessary infrastructure for managing and monetizing IoT services and applications. The primary goal of the business layer is to create value for businesses by optimizing the use of IoT technologies.

2.3 IoT Applications

IoT applications enable various devices, objects, or machines to connect and exchange data through the Internet, streamlining processes and enhancing functionality. These multiple sectors shown in Figure 2.2 demonstrate the diverse ways IoT can be utilized across various industries:

- Smart Homes: IoT simplifies the control and automation of appliances, energy systems, security, and lighting. Examples of IoT applications in this domain include smart thermostats, locks, and voice-controlled assistants.

- Healthcare: IoT enables remote patient monitoring, tracks medical equipment, and aids with senior care. IoT plays a significant role in improving patient care and streamlining medical operations through wearables, telemedicine, and modern hospital equipment.

- Agriculture: IoT applications streamline farming by tracking soil conditions, weather patterns, and animal health. Examples include precision farming, intelligent irrigation, and livestock management.

- Manufacturing: IoT is revolutionising manufacturing by automating processes, predicting maintenance needs, and managing supply chains. Smart factories use IoT devices and sensors to boost efficiency, minimise downtime, and improve product quality.

- Transportation: IoT applications streamline traffic management, fleet monitoring, and proactive vehicle maintenance. Solutions such as connected vehicles, smart traffic signals, and real-time public transit updates contribute to this improved transportation experience.

- Energy Management: IoT helps save energy in homes, offices, and factories. Examples include smart grids, systems that adjust energy use based on demand, and tools that monitor energy usage. By using IoT, we can reduce energy waste and lower costs.

- Retail: IoT enhances the retail industry by improving customer experiences, streamlining inventory management, and increasing supply chain efficiency. Modernising retail environments and procedures is made possible by technologies like beacons, smart shelves, and intelligent shopping carts.

- Environmental Monitoring: IoT is essential for monitoring and analysing environmental aspects including pollution levels, air and water quality, and climate. IoT supports educated decision-making to save the environment and lessen the consequences of climate change by supplying real-time data.

- Smart city: IoT is a game-changer for metropolitan areas since it improves infrastructure, safety, and transportation. Smart energy-saving lighting, effective waste management systems, and cutting-edge emergency response strategies are essential elements. Real-time data and connections provided by the IoT will allow for smarter, more sustainable, and more habitable cities in the future.

- Wearable: IoT-enabled wearables, such as fitness trackers, smartwatches, and augmented reality glasses, have become integral to modern life. To assist users in making wise decisions about their well-being, these gadgets measure physical activity, check health, and offer personalised information. In addition to facilitating communication, wearables enable users to receive notifications, make payments, and even access digital

assistants. As new technologies emerge, wearables will improve user experiences and seamlessly integrate into daily activities.

Figure 2.2: Landscape of IoT Applications Across the Ecosystem

2.4 Operating System in IoT

An operating system (OS) is an important part of an IoT device because it helps the device run applications and manage resources. The OS helps the device use energy efficiently, follow instructions, and communicate with other devices. IoT networks usually use devices that have limited storage, low processing power, and operate on batteries [43, 44, 45]. To

make sure that programs can run on these devices, special IoT-based operating systems have been created. There are two types of IoT OS: Linux-based and non-Linux-based. Examples of non-Linux operating systems include ContikiOS/NG, TinyOS, RIOT, and Openthread, whereas Linux-based operating systems include LiteOS, Pyrix, and ARM Mbed. This research mostly examines non-Linux OS since it focuses on devices with little resources.

In 2003, the world was first introduced to ContikiOS, a well-known operating system that is known for being particularly created for Wireless Sensor Networks (WSNs) [46, 47]. Its modular architecture allows for the dynamic loading and unloading of programmes while an application is running. ContikiOS is a multi-threaded, event-driven operating system that was developed using the C programming language. This was made possible by the Protothread foundation. A notable feature of ContikiOS is its adaptability, providing dynamic memory allocation, a comprehensive TCP/IP stack via uIP, and compatibility with a wide range of hardware platforms. In 2017, a derivative of the original ContikiOS, known as Contiki-NG, emerged with the objective of augmenting low-power communication, implementing standard protocols such as IPv6/6LoWPAN, CoAP, and RPL, and enhancing documentation quality. Contiki-NG receives frequent updates and enhancements as a result of its vibrant community [48]. Contiki-NG uses RPL-Lite, a more condensed version of the RPL protocol, whereas Contiki-OS uses ContikiRPL, an implementation of the RPL protocol. RPL-Lite outperforms ContikiRPL in terms of performance and dramatically decreases ROM footprint by prioritising non-storing mode over storing mode and simplifies the administration of many instances. Consequently, Contiki-NG presents a compelling alternative for researchers and developers in the realm of WSNs [49].

TinyOS is another popular operating system for devices with limited resources. It is different from other systems by its monolithic architecture and fixed memory allocation [50, 51]. Notably, this architecture situates all processes within the kernel space, rendering the entire operating system susceptible to failure in the event of an application bug. Memory, on the other hand, is statically allocated, further distinguishing TinyOS from alternative OS designs. The NesC programming language is used by applications running on TinyOS, and they follow an event-driven execution architecture. These components consist of three fundamental elements, namely Commands, Events, and Tasks, all implemented in C. Commands serve as requests for component execution, while Events function as signals

indicating the completion of a Command. Unlike Commands and Events, Tasks are not executed instantaneously. Instead, newly generated Tasks join a queue, awaiting execution by the scheduler once the current program concludes.

RIOT is a versatile, real-time, open-source operating system tailored for embedded smart devices [52]. It has a microkernel architecture and multi-threading reduce context switches, kernel size, and memory requirements for hardware devices. With its developer-friendly API and support for multiple programming languages, RIOT facilitates the rapid development and deployment of IoT solutions. RIOT allows for dynamic memory allocation and ensures low memory and power consumption as only necessary modules are compiled into the system. The operating system supports a range of stacks, including RPL, 6LoWPAN, IPv6, and common IP protocols. C serves as the primary programming language for RIOT-based applications.

FreeRTOS is a popular Real-Time Operating System (RTOS) designed for embedded systems, including IoT networks [53, 54]. It is ideal for resource-constrained devices, offering low power consumption, a small memory footprint, and real-time capabilities. Principal characteristics include preemptive multitasking, portability across microcontroller platforms, efficient memory management, modularity, and extendibility via libraries and auxiliary components. Similar to the RIOT operating system, FreeRTOS is built upon a microkernel architecture and embraces a multi-threading programming model. Although it requires third-party libraries for Internet connectivity, it also supports dynamic memory allocation. Typically, applications for FreeRTOS are developed using the C programming language; however, C++ is also a viable option, granting developers flexibility in their choice of programming tools. This versatility, combined with its ease of use and adaptability, renders FreeRTOS a popular choice for a wide range of IoT applications.

Mbed OS is an open-source, real-time operating system designed for ARM Cortex-M microcontrollers, targeted at IoT devices [55]. It offers a platform for developing connected, energy-efficient applications, with built-in security features and support for various communication protocols. The modular architecture allows developers to optimise resource usage and power consumption, while advanced power management prolongs battery life. Mbed OS provides comprehensive APIs, libraries, and development tools compatible with C and C++ languages, as well as an online IDE and command-line interface. An active

community and extensive documentation further strengthen the Mbed OS ecosystem, which facilitates the development of IoT applications.

The Zephyr Project is an open-source, real-time operating system (RTOS) tailored for IoT networks and embedded systems [56, 57]. It supports multiple architectures, including ARM, x86, ARC, and RISC-V, ensuring broad compatibility across hardware platforms. Zephyr offers a small memory footprint, low-power operation, and preemptive multithreading for efficient task execution. It was created in C and C++ and offers a modular, adaptable framework that enables developers to produce customised IoT applications. The Zephyr Project offers a comprehensive platform for developing and deploying IoT applications across diverse use cases and environments.

In the course of our study, we make use of an OS to assist us in the process of designing, developing, and demonstrating the suggested solution inside a real setting. Table 2.1 shows how the different IoT OSs compare to each other. It has been demonstrated that ContikiOS-NG is preferable for IoT research since it supports both multithreading and event-driven methodologies and enables the creation of modular applications due to its modular design. It also has the benefit of being easily replicable in simulation tools like Cooja [46, 47]. Because of all of these factors, the implementation of the suggested solution in our thesis will be done using ContikiOS-NG.

Table 2.1: IoT Operating Systems Explored: Features and Differences

Operating System	**Architecture**	**Programming Language(s)**	**Memory Allocation**	**Programming Model**
Contiki	Multiple (MSP430, AVR, ARM Cortex-M, etc.)	C	Static	Event-driven, protothreads
TinyOS	Multiple (ARM, MSP430, AVR, etc.)	nesC (based on C)	Static	Event-driven, task-based
RIOT OS	Multiple (ARM, MSP430, AVR, x86, RISC-V, etc.)	C, C++	Static and Dynamic	Event-driven, multithreading
FreeRTOS	Multiple (ARM, AVR, x86, RISC-V, etc.)	C	Static and Dynamic (with Heap)	Preemptive multithreading
Mbed OS	ARM Cortex-M microcontrollers	C, C++	Static and Dynamic (with Heap)	Event-driven, multithreading
Zephyr Project	Multiple (ARM, x86, ARC, RISC-V, etc.)	C, C++	Static and Dynamic	Preemptive multithreading

2.5 Simulators for IoT Research

In the development and evaluation of novel security solutions for WSN and IoT research, a considerable number of researchers depend on simulation tools and real testbeds. In recent years, an array of open-source simulators and real testbeds has emerged to cater to the diverse needs of the research community [58]. A robust simulator and real testbed should possess the following characteristics:

- Producing Real Network Traffic: It is crucial to create and make use of authentic network traffic that closely mimics what would be seen in a real-world implementation to assess the impact of prospective attacks correctly. If a simulator does not support actual software, it may not create authentic traffic. As a result, it would construct traffic patterns based on distributions, which may not reflect network circumstances.

- Scalability: The simulator and testbed should be capable of handling many IoT devices. It allows researchers to examine the performance of IoT solutions in various network sizes and situations.

- Heterogeneity support: IoT networks may include a wide range of devices, communication protocols, and applications. The simulator and testbed should support various hardware platforms, operating systems, and communication standards, enabling researchers to model and test complex, heterogeneous IoT networks.

- Extensibility and modularity: The simulator and testbed should allow researchers to easily modify and add new models, algorithms, and protocols. A extensible and modular architecture enables the research community to perpetually adapt simulation tools to evolving requirements and technological advances in IoT.

- Support for experimentation and performance evaluation: The simulator and testbed should include automated testing, performance measurements, and analysing experiments. This allows researchers to evaluate and compare the efficacy of IoT security solutions under different conditions and scenarios.

- Open-source and community-driven: An open-source simulator and testbed encourage community contributions, ensuring the continuous improvement and adaptation of the

tools to the latest advancements in IoT research. A vibrant community also fosters collaboration, knowledge sharing, and support among researchers.

A variety of simulators, including NS-3 [59], OMNeT++ [60], and Cooja simulators [46, 47] allow researchers to build and test IoT systems while taking into consideration network traffic, device diversity, and energy usage. These tools include NS-3 and OMNeT++, which offer comprehensive support for a wide range of IoT devices and protocols, and Cooja, which is made especially for wireless sensor networks running the Contiki OS. These simulators allow researchers to fine-tune and optimise their ideas for real-world application by analysing IoT network behaviour and performance under varying conditions.

On the other hand, real testbeds are IoT-LAB [61], FIT IoT-LAB [62], WISEBED [63], and Fed4FIRE+ [64]. These testbeds give researchers access to actual devices and infrastructure, which enables the testing and assessment of IoT applications in real-world settings. These testbeds offer diverse hardware platforms and networking technologies, facilitating the examination of IoT solutions' interoperability, scalability, and robustness. Testbeds are crucial for verifying IoT solution performance in a real-world setting prior to deployment.

F-Interop is an online platform that focuses on providing remote testing and interoperability for IoT devices and applications [65]. It acts as a bridge between simulation tools and actual testbeds. It supports various IoT standards and protocols, providing tools for automated testing, performance evaluation, and debugging. F-Interop is essential in ensuring that IoT solutions are interoperable and can seamlessly interact with other devices and applications.

A comparative analysis of the aforementioned simulators and real testbeds is presented in Table 2.2. It can be observed that most simulators, such as NS-3, OMNeT++, and Cooja, do not generate real traffic by running actual applications; instead, they create dynamic traffic or use patterns like exponential distributions. In contrast, genuine network traffic is produced by real testbeds like IoT-LAB, FIT IoT-LAB, F-Interop, WISEBED, and Fed4FIRE+. Furthermore, half of the simulators lack support for real-world operating systems and IoT devices. Simulating with a genuine OS enables the generation of realistic and accurate results concerning the impact of attacks on various IoT networks. The suggested approach should also work with an OS that is simple to install on a physical hardware

Table 2.2: Comparison of IoT Simulators and Real Testbeds: Features and Capabilities

Tool	Primary Purpose	Traffic Type	Supported Devices/OS	Energy Modeling	Limitations
NS-3	Network simulation	Realistic	Wide range	Yes	Steep learning curve, may have performance issues
OMNeT++	Network simulation	Realistic	Wide range	Yes	Limited support for some IoT protocols, complex to set up
Cooja	Wireless sensor networks	Realistic	Contiki OS	Yes	Limited to Contiki OS, scalability issues
IoT-LAB	IoT application testing	Real-world	Various platforms	Yes	Limited availability, requires reservation
FIT IoT-LAB	IoT application testing	Real-world	Various platforms	Yes	Limited availability, requires reservation
F-Interop	Interoperability testing	Real-world	Various standards	N/A	Online platform, may have limited support for some protocols
WISEBED	Wireless sensor networks	Real-world	Various platforms	Yes	Limited hardware platforms, access restrictions
Fed4FIRE+	IoT, cloud, networking	Real-world	Various platforms	Yes	Complex setup, requires access to multiple testbeds

device.

In conclusion, a wide array of simulation tools and real testbeds are available to support IoT research and development. Each tool offers its own unique characteristics and capabilities, catering to different aspects of IoT experimentation. We have chosen Cooja and FIT IoT-LAB to implement our proposed security solutions in the IoT ecosystem. Contiki Cooja, a simulator integrated with the Contiki OS, enables realistic simulation and testing of IoT applications and wireless sensor networks. Concurrently, FIT IoT-LAB, a large-scale testbed, offers real-world testing and evaluation on diverse hardware platforms. By combining these tools, we can perform comprehensive assessments of our proposed security solutions, addressing both simulated environments and real-world performance, ensuring robust, reliable, and high-performing applications and protocols.

2.6 IoT Attack

In traditional cyberspace, various forms of attacks have persisted for a long time, targeting interconnected computer networks, systems, services, embedded processors, and data storage or sharing. However, the IoT domain poses a new challenge due to the vast number of connected devices and the relative ease of launching attacks. As a result, cyberattacks

affect millions of interconnected devices. This section offers a concise overview of the most prevalent attacks witnessed in an IoT ecosystem. In reference to the technical perspective, an array of attacks, depicted in Figure 2.3, can be observed across the different layers of the IoT ecosystem architecture.

2.6.1 IoT Perception Layer Vulnerabilities

1. **Node Capture/Replication Attack:**
   In wireless sensor networks (WSNs), the number of nodes can expand rapidly. Due to their low price, these nodes often have inadequate physical protection for their sensitive components including processor, memory, and communication. Thus, when unsecured sensor nodes are deployed in adversarial environments, attackers may easily compromise the sensor's internal state and capture, duplicate, or introduce a cloned node into a targeted network. The consequences of such incursions might be devastating, including complete network damage or the deactivation of a significant portion of the WSNs functionality [66].

2. **Device tampering attacks:**
   It is a cyberattack that incorporates intentionally manipulating the firmware /hardware of an IoT device. Device tampering attacks commonly aim to circumvent security protections, harvest private information, introduce malicious code, or interfere with the device's regular operation [67].

3. **Jamming Attack:**
   Jamming attacks are active attacks in the Denial-of-Service (DoS) attack category. These attacks involve creating false data streams. In a jamming attack, malicious nodes are used to disrupt communication within an IoT network through interference. These attacks have a negative impact on resource-constrained networks as they deplete the limited resources of devices, causing harm to the network. In general, jamming attacks are classified as constant, deceptive, random, and reactive. Constant jamming generates continuous noise at the same frequency as the network operates. Deceptive jamming replaces valid signals or fabricates fake signals. Random jamming disrupts the network for a specific duration, after which the jammer node remains inactive.

Reactive jamming, on the other hand, remains silent during idle channel periods but starts emitting radio signals upon sensing channel activity [68].

4. **Malicious Nodes Injection Attack:**
   In this attack, the attacker deploys malicious nodes with significant processing, storage, and transmission capabilities to collaborate and implement a coordinated attack. The attack unfolds in two phases. First, the attacker penetrates a genuine node, replicates it, and takes complete control, accessing its data. The duplicated powerful node then isolates the vulnerable node by relocating it or decreasing its energy. In the second phase, the hacked node's data is cloned onto a new malicious node. This new node, although not an exact replica, shares similar properties. Thus, two inserted malicious nodes work together to launch attacks [69].

2.6.2 6LoWPAN Layer Vulnerabilities

1. **Fragment Duplication Attack:**
   In a fragment duplication attack, an attacker duplicates a packet fragment, causing the receiving node to disregard the fragments. This attack exploits the way that the 6LoWPAN standard handles redundant fragments. The standard recommends dropping duplicate fragments to avoid the overhead and conserve resources. However, malevolent nodes may readily take advantage of this basic technique and turn it into a DoS attack [70].

2. **Buffer Reservation Attack:**
   In a buffer reservation attack, the attacker reserves the buffer space of the victim node by sending it incomplete packets. Nodes are unable to allocate additional buffer space for incomplete packets from other nodes due to their restricted resources. As a result, while the attacker occupies the buffer space, fragments from regular nodes cannot be accepted. It is crucial to note that this behaviour happens in 6LoWPAN when it is set to forward fragments using the route-over strategy, in which the receiving node reassembles all pieces of a packet before forwarding [70, 71].

3. **Malicious Worm-Virus Attack:**
   In this attack, adversaries propagate malicious code through downloads, emails, or

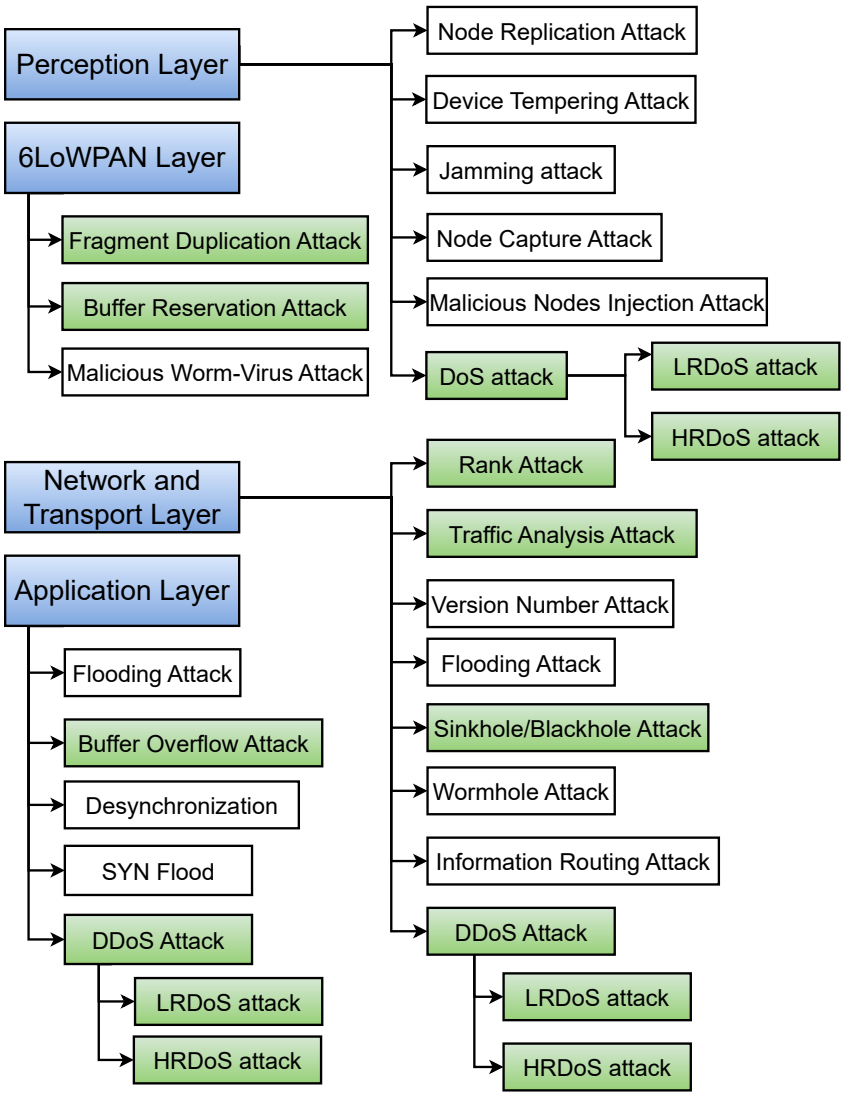


Figure 2.3: IoT Ecosystem Attack Vectors

attachments. The worm rapidly replicates within the system or network, aiming to disrupt the targeted system by consuming storage space and network bandwidth. Conversely, viruses typically aim to corrupt or modify files [72].

2.6.3 Application Layer Vulnerabilities

1. **Flooding Attack:** This kind of attack involves flooding the network with an unusually huge quantity of traffic, causing network saturation. As a consequence, the connections between nodes cease to function, disrupting normal communication. The HELLO flood attack is an example of this kind of network overload [72].

2. **Buffer Overflow Attack:** An attacker commits a buffer overflow attack when they place an excessive amount of data into a buffer, which causes the buffer to overflow and overwrite the memory locations that are contiguous to it. This vulnerability might result in unauthorised access, the execution of malicious code, and the compromise of the system. Attackers introduce harmful code into software that fails to verify or limit user input [70, 71].

3. **Desynchronization Attack:** Desynchronization attacks disrupt the coordination of communication between network entities. Attackers miscoordinate nodes by changing timing, sequencing, or synchronisation. These attacks may result in compromised data integrity, degraded performance, or a total shutdown of network services. Strong security mechanisms and durable synchronisation protocols are needed to defend against desynchronization threats [73].

4. **SYN Flood Attack:** In a SYN flood attack, the attacker sends several SYN packets to every port, frequently using a fake IP address. The server receives these ostensibly legitimate queries to establish communication, unaware of the attack. In reaction, the server sends SYN-ACK messages from each open port and waits for the final ACK packet to finish the handshake. When an attacker uses a spoofed IP address or fails to reply, the server is left waiting indefinitely for the final ACK while its resources are depleted. This kind of attack reduces the reliability of the server, which might affect the quality of service [72].

5. **DDoS Attack:** It is a malicious assault characterized by the coordinated use of multiple compromised IoT devices, forming a botnet, to inundate a targeted IoT network, device, or service with an overwhelming volume of network traffic or requests. The primary intent of an IoT DDoS attack is to exhaust the target's computational resources, network bandwidth, or memory capacity, thereby disrupting or incapacitating the IoT network's normal operations and rendering it inaccessible to legitimate users. These attacks can result in service outages, data breaches, and operational disruptions, posing a critical security threat to IoT ecosystems [72].

2.6.4 Network and Transport Layer Vulnerabilities

1. **Rank Attack:**

   A rank attack is a type of attack in routing protocols for low power and lossy networks (RPL) that aims to manipulate the internal ranking mechanism within a network. It involves an adversary intentionally altering the rankings assigned to nodes, typically in an IoT network. By exploiting vulnerabilities in the ranking algorithm, the attacker can disrupt the network's optimized topology and compromise its overall performance and efficiency. A rank attack aims to undermine the network's reliability and functionality by tampering with the hierarchical organization of nodes based on their ranks [74].

2. **Traffic Analysis Attack:**

   Traffic analysis resembles eavesdropping. However, in this kind of attack, the attacker just examines the traffic rather than gaining access to the real contents. The objective is to infer the traffic pattern, locate critical nodes, comprehend the routing topology, or discover the application's behaviour. By using this attack, you can glean information about the network without actually changing any data. There is more information about this type of attack in [72] and [75].

3. **Version Number Attack:**

   This attack targets the version number field in IPv6 Routing Protocol for Lossy Networks (RPL) routing control messages. Using the version number field, an attacker can cause routers to rebuild their routing tables by sending falsified version numbers. This allows them to track network topology changes. This can lead to increased overhead, decreased performance, and even network outages [74].

4. **Flooding Attack:**

   This attack causes network saturation by generating a massive volume of traffic, overwhelming the network. Consequently, the links between nodes become unavailable, disrupting communication. An example of such an attack is the HELLO flood attack, which illustrates this type of network saturation [72].

5. **Sinkhole/Blackhole Attack:**

   In a sinkhole attack, an intruder compromises a node in such a way as to attract all

traffic from neighboring nodes. The compromised node broadcasts falsified information to lure a significant amount of traffic. WSNs that utilize a base station, where all nodes send data, are particularly vulnerable to sinkhole attacks. This type of attack is often launched in networks using the AODV routing protocol. The malicious node waits for neighboring nodes to request a route (RREQ). Upon receiving the RREQ, the malicious node swiftly responds with a false route reply (RREP) message featuring a higher sequence number, deceiving the requesting node into believing that the route to the destination is fresh. Consequently, the requesting node ignores RREPs from other neighboring nodes and unknowingly sends packets through the malicious node. Whenever a nearby RREQ occurs, the malicious node replays a fake RREQ and diverts all routes toward it, creating what is known as a black hole [76].

6. **Wormhole Attack:**
   Wormhole attacks arise when two adversaries in different geographical networks build a communication tunnel. This tunnel can be made with a wired or wireless connection with ample range and bandwidth, working at different frequency bands. After the adversary establishes a connection through the wormhole, they may launch an attack such as man-in-the-middle to intercept and modify legitimate communications. This assault compromises communication integrity, secrecy, and authenticity, threatening network security [76].

7. **Information Routing Attack:**
   In this attack, an adversary intentionally alters the routing information to divert or intercept data packets. By tampering with routing tables, injecting false routing updates, the attacker can control the paths taken by data within the network. This allows the attacker to eavesdrop on sensitive information, modify or drop data packets, or redirect them to unauthorized destinations. Information routing attacks pose a significant threat to data confidentiality, integrity, and availability within a network [76].

8. **DDoS Attack:** The nature of this attack varies depending on its objectives, which can range from interrupting network traffic to exhausting network resources and disrupting the topology. This thesis is focused on DDoS attacks, which can be broadly

classified into three categories: High-Rate DDoS Attacks, Low-Rate DDoS Attacks, and Mixed-Rate DDoS Attacks. Low-Rate DDoS attacks are difficult to identify due to their low-rate and intermittent traffic behavior, which is similar to legitimate traffic [3]. These attacks aim to increase latency and decrease the network's throughput for genuine users rather than disrupting IoT services entirely.

2.7 Datasets for IoT Attack

In recent years, multiple datasets have emerged in the IoT security arena, each with its own set of pros and downsides. With the pervasive adoption of IoT devices, researchers are increasingly turning to IoT-related datasets to address the expanding number of unknown vulnerabilities and threats. These datasets collect information from both simulated and real-world environments in order to test the efficacy, security, and applicability of IoT devices against normal and abnormal behaviour. In order to create IDS that can be used in real-world settings, these datasets' quality is essential.

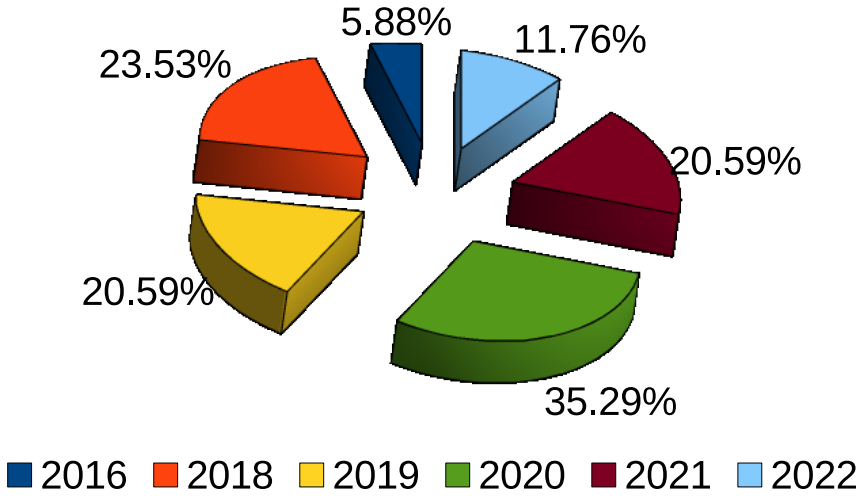


Figure 2.4: Dataset Distributation

This section provides an overview of publicly available datasets proposed for IDS in IoT environments. Figure 2.4 shows the number of IoT datasets that have been suggested over time, with a sharp rise in 2020. Moreover, Table 2.3 provides technical details and properties of different datasets, which provide general information about them. In this thesis, we used one, two, and three datasets to develop effective IDS solutions.

Table 2.3: Technical Details and Properties of Different IoT Datasets

Dataset	Year	Availability	Format	Size	No. of Features		
					Statistics	Extracted	Raw
BoT-IoT	2020	Yes - Free Download	csv, pcap	69.3GB	0	46	29
BoTIoT	2021	Yes - Free Download	csv, pcap	1.49GB		90	
TON-IoT	2020	Yes - Free Download	csv, pcap, log, txt	67GB		36	
Kitsune Network	2019	Yes - Free Download	csv, pcap	68.9GB	115		23
N-BaIoT Dataset	2018	Yes - Free Download	csv, pcap	2.1GB	115		0
BoTNeTIoT	2020	Yes - Free Download	csv, pcap	2.31GB		23	
Aposemat IoT-23	2020	Yes - Free Download	pcap	21GB		23	
DAD Dataset	2020	Yes - Free Download	csv, pcap, xml	18MB		6	
IOTID20	2020	Yes - Free Download	pcap	294.2MB		83	
IoTID	2019	Yes - Free Download	pcap	823.69MB		42	
X-IIOTID Dataset	2021	Yes - Free Download	csv, pcap	351.6MB		59	
Edge-IIoTset	2022	Yes - Free Download	csv, pcap	1.5GB	0	61	1176
IoT and DDOS dataset	2020	Yes - Available on Request	pcap	487 MB		60	
WUSTL-IIOT-2018	2018	Yes - Free Download	csv	1.27GB		7	
KNX Datasets	2022	Yes - Free Download	csv, pcap	289MB		21	
CIC IoT Dataset	2022	Yes - Free Download	pcap	1GB + 48		48	
IoTEnvironment Dataset	2021	Yes - Free Download	pcap	287 MB	N/A Pcap		
IoTKeeper	2020	Yes - Free Download	pcap	15GB		38	
IoTSentinel	2016	Yes - Available on Request	csv, pcap	27.4MB	276	0	23
WUSTL-IIOT-2021	2021	Yes - Free Download	csv	390.82MB		7	
IEEE TMC 2018 UNSW	2018	Yes - Free Download	csv, pcap	7.8GB		10	
UNSW-IoT	2019	Yes - Free Download	pcap	Unknown	N/A Pcap		

- **IoT-23 [77]:** IoT-23 is a dataset that includes network traffic from 23 IoT devices, such as routers, smart home appliances, and cameras, over the course of four weeks in

a lab setting. It consists of both regular traffic and a variety of assaults, including Brickerbot and Mirai. The dataset is divided into training and test sets and includes both encrypted and unencrypted traffic in pcap format. The dataset also contains metadata files that list the devices and different kinds of attacks. The dataset is publicly accessible for academic and research usage and can be utilised for studies on intrusion detection and prevention for IoT networks.

- **CICIDS2017 [78]:** CICIDS2017 is a comprehensive dataset that contains both normal traffic and various types of attacks, such as DoS, port scans, and brute-force attacks. It consists of several scenarios and network topologies with various traffic characteristics, including IoT traffic. The CSV-formatted dataset comprises over 80 attributes, such as packet header and payload information. Due to the size of the dataset, which is approximately 32 GB, it is one of the largest datasets that may be used for research on network security. The dataset gives researchers an option to assess the performance of different methods for attack detection and prevention. The dataset can be accessed and used without cost for scholarly and research purposes.

- **BoT-IoT [79]:** This dataset includes traffic from several IoT device-targeting botnets, such as Mirai and Gafgyt. The dataset contains both normal traffic and a variety of attack categories, including DoS and brute-force attacks. It is available in CSV format and contains more than 50 features, including packet header and payload data. Despite being modest in comparison to other datasets, the BoT-IoT dataset can be useful for research into IDS and IPS for IoT networks. The dataset can be accessed and used without cost for scholarly and research purposes.

- **Kitsune [80]:** This dataset is a public set of data that includes network activity from smart home devices, cameras, and smart plugs that were collected in a lab scenario. It includes Mirai and Satori botnet assaults and regular traffic. The dataset consists of over 12,000 pcap files, each of which contains one minute of traffic. Metadata also describes devices and attacks. The dataset, which has been divided into training and test sets, may be utilised for studies on IoT network IDS and IPS. It enables researchers to evaluate the efficacy of various algorithms and methods for detecting and mitigating intrusions on IoT devices. The Kitsune IoT Dataset is open for academic

study in pcap and CSV formats.

- **TON_IoT [81]:** This dataset is a publicly available dataset that contains network traffic gathered from IoT devices in an experimental environment. This dataset, which includes DDoS and DNS tunnelling attacks, is useful for IoT security researchers. The dataset contains more than 9,000 pcap files, each of which contains 30 minutes of traffic, and is accessible in both pcap and CSV formats. In addition, it contains metadata files containing information about the devices and types of attacks. The TON_IOT dataset may be used to assess the effectiveness of security solutions. The dataset is available at no cost for academic and research purposes.

- **KDD CUP 99 [82]:** This dataset is a collection of network traffic data captured during the DARPA Intrusion Detection Evaluation Programme. Although it is not an explicitly IoT-related dataset, it may include some traffic generated by IoT devices since these devices are part of the larger network environment. The dataset contains more than 4 million records with 42 parameters, including source and destination IP addresses, protocol type, and duration, that indicate different network connection characteristics. It is available in CSV format and is partitioned into training and test sets. Researchers can use the KDD Cup 99 dataset to evaluate the effectiveness of distinct IDS for detecting various types of attacks, such as DoS, probe, and R2L attacks. The dataset is publicly accessible for academic and research use and can aid in the development of efficient IDS systems to combat network attacks.

- **ISCX [83]:** The ISCX dataset comprises regular traffic and numerous assaults from diverse situations and network topologies. The dataset consists of training and test sets, and it has more than 16 million individual entries. Researchers may evaluate IDS and IPS for diverse networks using the ISCX dataset. The dataset is available for free download from the ISCX website for academic and research purposes.

2.8 Machine Learning /Deep Learning for IoT

Machine learning encompasses three main types: supervised, unsupervised, and hybrid /reinforcement learning. Supervised learning is based on labelled data, which provides significant information for tasks like as classification, which is often used in IDS. However,

manual data labeling is time-consuming and expensive, hindering supervised learning due to limited labeled data availability. In contrast, unsupervised learning leverages unlabeled data to extract valuable features, simplifying the acquisition of training data. Nevertheless, unsupervised learning methods generally exhibit inferior detection performance compared to supervised learning methods. Reinforcement learning involves learning from interactions with the external environment based on specific actions. Hybrid learning models combine multiple deep learning models, typically discriminative or generative, to form a comprehensive approach [84, 85]. Figure 2.5 illustrates the common ML algorithms employed in IDSs.

2.8.1 Shallow ML Models

Traditional models for machine learning, also known as shallow models. These models encompass various techniques such as support vector machines (SVM), Naïve Bayes (NB), K-nearest neighbour (KNN), random forest (RF), decision trees, artificial neural networks (ANN), ensembles learning (EL), and hybrid approaches. These methods have undergone extensive research over several decades, resulting in well-established methodologies. In addition to prioritising detection efficacy, they also address practical concerns, such as data management and detection efficiency within IDS.

- **Support Vector Machine (SVM):** SVMs strategically find an n-dimensional feature space hyperplane with greatest margin. SVMs are sensitive to hyperplane noise, even with tiny training sets. SVMs solve linear issues, whereas kernel functions handle nonlinear data. Nonlinear data can be separated using these functions. Kernel tricks are widely utilized in both SVMs and other ML algorithms.

- **K-Nearest Neighbor (KNN):** The fundamental concept behind KNN is rooted in the manifold hypothesis. According to this hypothesis, if most of a sample's neighbours belong to a class, the sample likely does too. As a result, the outcome of the categorization is extremely dependent on the top k closest neighbours. The parameter k plays a crucial role in the performance of KNN models. Smaller values of k result in more complicated models that are more prone to overfitting. In contrast, greater values of k result in simplified models with diminished fitting capabilities.

- **Naïve Bayes:** The Naive Bayes method relies on the assumptions of attribute

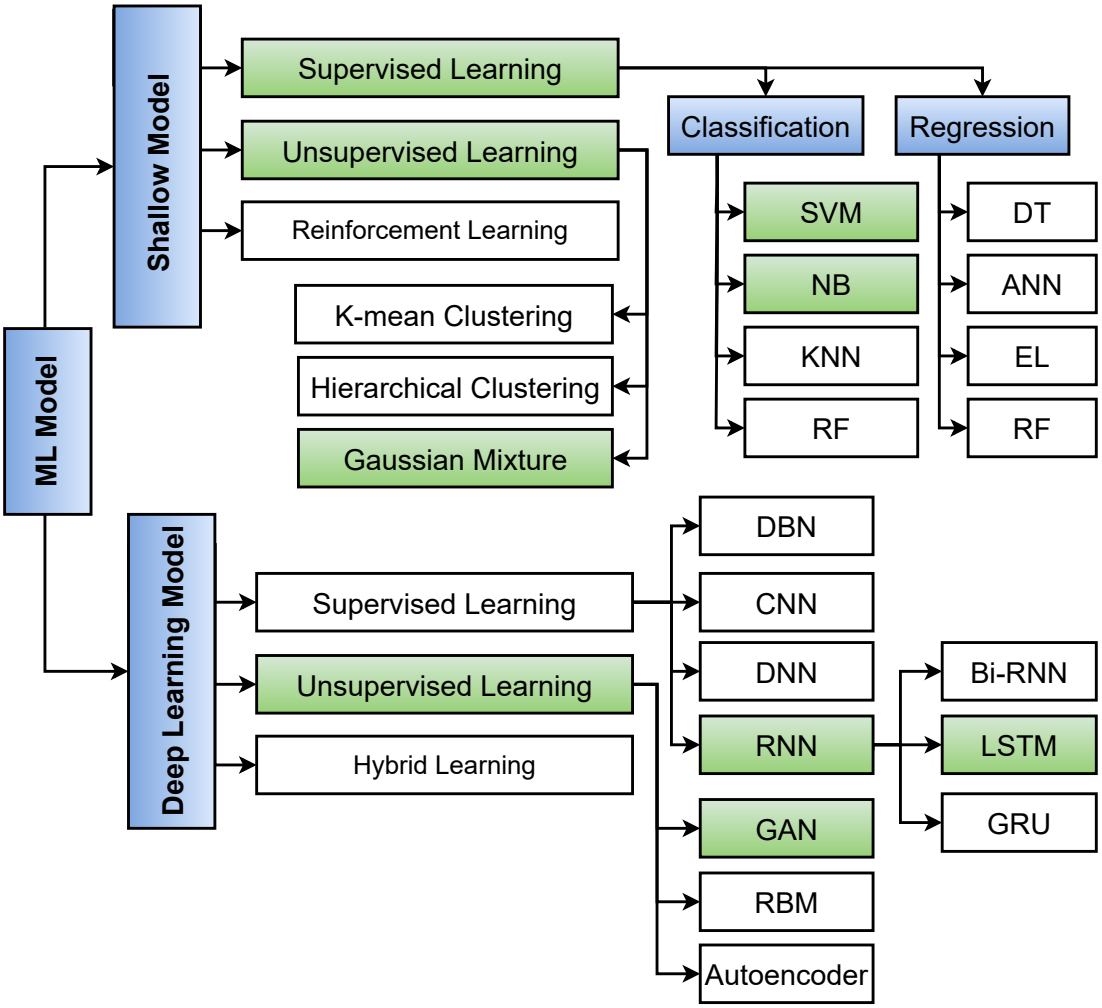


Figure 2.5: Classification of Machine Learning Techniques

independence and conditional probability to function. It determines class conditional probabilities based on sample properties. The sample is then assigned to the most likely category. However, achieving the optimal result relies on satisfying the attribute independence assumption, which is often challenging in real-world scenarios.

- **Logistic Regression (LR):** LR is a parametric logistic distribution-based logarithmic linear model that predicts class probabilities. This classifies sample x into the highest probability class of $k = 1, 2, ..., K - 1$. LR model construction and training are simple. However, LR is not well-suited for handling nonlinear data, which restricts its applicability.

- **Decision tree:** The decision tree algorithm is employed for data classification through

a series of rules. Its tree-like structure aids interpretation. This algorithm possesses the capability to automatically eliminate irrelevant and redundant features. The learning process involves selecting features, creating a tree, and then pruning it. During training, the algorithm identifies the most relevant attributes and creates child nodes from the parent node. Advanced classifiers like random forests and extreme gradient boosting (XGBoost) use numerous decision trees.

- **Artificial Neural Network (ANN):** The concept behind ANNs is to emulate the functioning of the human brain. An ANN is made up of an input layer, many hidden layers, and an output layer. ANNs are capable of performing a wide variety of functions, particularly nonlinear ones. However, ANNs are complex, thus training them takes some time. The backpropagation algorithm used to train ANNs is inefficient for deep networks. ANNs are shallow models, unlike deep learning models detailed in Section 2.8.2

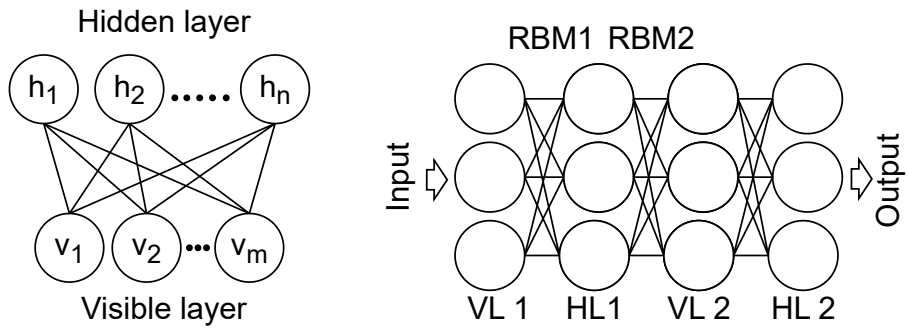
- **Ensembles Learning (EL):** Every classifier has its own advantages and disadvantages. Combining numerous under performing classifiers into a single robust one is a typical strategy for taking use of each one's strengths. Ensemble methods are employed for this purpose, involving the training of multiple classifiers. Through a voting process, these classifiers ultimately contribute to the final outcomes.

2.8.2 Deep Learning Models

Deep learning models include various supervised learning models, such as deep neural networks (DNNs), convolutional neural networks (CNNs), deep brief networks (DBNs), and recurrent neural networks (RNNs). On the other hand, unsupervised learning models include generative adversarial networks (GANs), restricted boltzmann machines (RBMs), and autoencoders. Since 2015, there has been a meteoric rise in the use of IDS based on DL. They can process big datasets and outperform shallow models. Deep learning research focuses on network construction, hyperparameter selection, and optimisation.

- **Restricted Boltzmann Machine (RBM):** An RBM is a stochastic neural network that follows the Boltzmann distribution. It consists of two layers, one visible and one hidden, as illustrated in Figure 2.6 (a), which are not connected within each other

but are fully connected between layers. The visible layer is v_i, and the hidden layer is h_i. RBMs share weights between layers bidirectionally. RBMs are trained using the contrastive divergence algorithm, a type of unsupervised learning, and they are commonly utilised for tasks such as feature extraction and denoising.



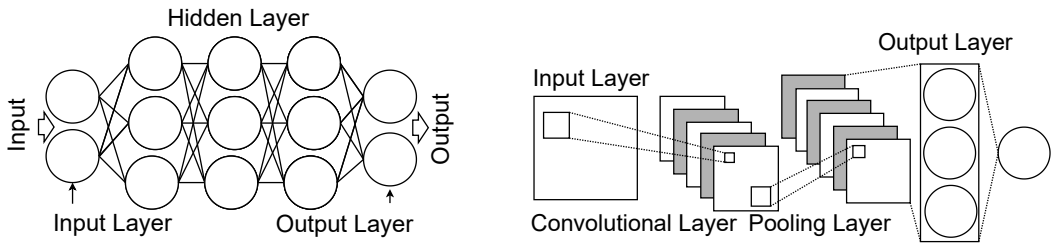
(a) RBM (b) DBN

Figure 2.6: The structure of RBM and DBN

- **Deep Brief Network (DBN):** As shown in Figure 2.6 (b), a DBN is made up of several RBM layers followed by a softmax classification layer. The DBN training is broken up into two parts: supervised fine-tuning and unsupervised pretraining [86]. Each RBM is greedy-layer-wise pretrained. Labelled data is used to learn softmax layer weights. In the context of attack detection, DBNs serve the dual purpose of feature extraction and classification, making them valuable for identifying and classifying attacks [87].

- **Deep Neural Network (DNN):** DNNs with multiple layers can be created using a layer-wise pretraining and fine-tuning method, shown in Figure 2.7 (a). This technique uses unlabeled data to train the network's parameters. Following that, the network is fine-tuned using labelled data, which is a stage of supervised learning. DNNs' success is mostly due to unsupervised feature learning.

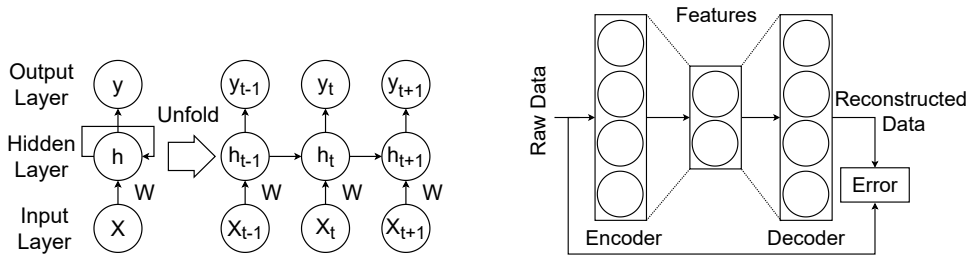
- **Convolutional Neural Network (CNN):** CNNs are meant to mimic the human visual system (HVS), advancing computer vision [88, 89, 90]. Figure 2.7 (b) shows how CNNs alternate convolutional and pooling layers. Convolutional layers extract features, while pooling layers make them more generalizable. For attack detection,



(a) DNN (b) CNN

Figure 2.7: The structure of DNN and CNN

CNNs must transform input data into matrices since they operate on 2-dimensional (2D) data.



(a) RNN (b) Autoencoder

Figure 2.8: The structure of RNN and Autoencoder

- **Recurrent Neural Network (RNN):** RNNs are specifically designed for processing sequential data, with widespread applications in natural language processing (NLP). Sequential data requires contextual analysis. To capture the contextual information, each unit in an RNN receives the current state and previous states, as depicted in Figure 2.8 (a). However, a common issue with RNNs is the vanishing or exploding gradients due to the repetitive nature of the network structure. Many RNN variants have been proposed to resolve this issue, including bidirectional RNNs [91], gated recurrent units (GRUs) [92], and long short-term memory (LSTM) [93]. LSTM introduced by Hochreiter and Schmidhuber in 1997, incorporates three gates to manage memory flow: forget gate, input gate, and output gate. GRU, developed by Chung et al. in 2014, simplifies the architecture by combining the forget and input gates into one update gate.

- **Autoencoder:** An autoencoder is a neural network architecture composed of two

symmetric components: an encoder and a decoder, as depicted in Figure 2.8 (b). The encoder extracts important characteristics from raw data, while the decoder reconstructs it. Training reduces the encoder input-decoder output divergence. The encoder has caught the important data properties if the decoder can rebuild the data from the extracted features. This approach uses unsupervised data. Denoising and sparse autoencoders are used for noise reduction and sparsity regularisation.

2.9 Intrusion Detection Systems for IoT

IDSs are important safety measures for network security, whether they are made of hardware or software. The major purpose of these defences is to protect individual devices, nodes, and entire networks against malicious attacks and violations of predetermined policies. IDSs are designed to proactively detect attempts at unauthorised access to a system and alert the system administrator immediately so that they can take appropriate action [94]. The attacks identified by IDSs can be broadly categorised into two classes: internal and external. Internal attacks originate from inside the network or system with the intention of elevating privileges to gain unauthorised access to data or services. In contrast, external attacks originate from external entities attempting unauthorised network access. Figure 2.9 provides a comprehensive overview of various types of IDS. In the following discussion, we examine IDSs' complex components, methodologies, deployment strategies, placement methods, and cutting-edge innovations in IoT environments.

2.9.1 Anatomy of IDS: Components and Deployment Tactics

IDS typically consists of three basic components, which are as follows:

1. **Sensing and Monitoring:** The first component is responsible for identifying any deviations from established network traffic norms that are not typical. Its purpose is to promptly identify network anomalies.

2. **Analysis and Detection:** The analysis and detection module is the core of the IDS. This core component detects and assesses harmful attacks quickly.

3. **Notification and Reporting:** The third component, which is critical for intrusion

detection, sends timely notifications to the system administrator. This fast notification enables administrators to take the necessary actions. In addition to this, this component is used as a source for in-depth reports that explain the nature of the attacks.

2.9.2 IDS Deployment Tactics

IDS Deployment Tactics: IDS deployment typically uses two different approaches.

1. **Host-Based (HIDS):** In this approach, the IDS is housed inside the target device.

2. **Network-based (NIDS):** In this approach, the IDS is dispersed throughout the network. Here, the IDS assumes responsibility for keeping track of and examining all network traffic, system calls, and running activities.

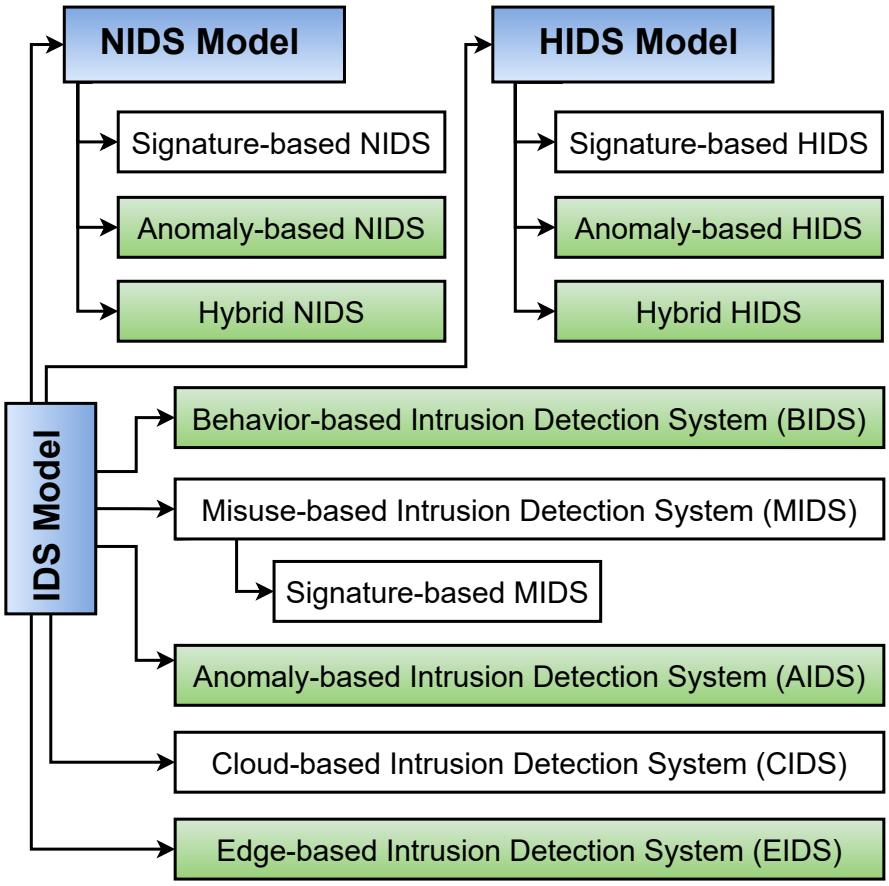


Figure 2.9: Various Intrusion Detection System Types

2.9.3 IDS Methodologies

- **Signature-based NIDS [95]:** This IDS uses predetermined patterns or signatures to identify network traffic threats. It compares incoming data to a library of signatures and sends out alerts when a match is found. This makes it easier to find known threats quickly. Signature-based IDSs are quick and effective, but only against known threats. This approach has a very low false positive rate (FPR) for known attacks, but it is ineffective against new attacks and even new variants of known attacks. This is because the database does not yet contain the signatures of newly developed attacks. As a result, in an era where attackers are growing smarter and constantly devising new attacks and versions of existing attacks, this technique becomes less effective. The approach also has the drawback of requiring a very large database to hold a very large number of signatures.

- **Anomaly-Based NIDS [96]:** This methodology identifies intrusions based on system behavior. By establishing a normal activity baseline, any deviations are flagged as potentially malicious and alerts are generated. This method is effective at detecting new attacks; however, it produces an excessive number of false positives, as deviations from normal behaviour do not always indicate an attack.

- **Hybrid NIDS [97]:** This strategy combines multiple approaches to enhance attack detection accuracy. It utilises the benefits of both signature-based and anomaly-based techniques by combining them. While anomaly-based evaluation finds anomalous behaviours, signature-based analysis detects known threats. As a result of this integration, the system can successfully defend against both known attacks and newly developed, innovative attacks. The hybrid approach provides a comprehensive defence strategy by overcoming the limitations of individual methods. It adapts to changing attack strategies, optimises detection, and identifies both known and unknown attacks.

2.9.4 IDS Placement Strategies

The placement of IDSs is typically guided by one of the following strategies [98]:

- **Centralised Strategy:** In this strategy, the IDS is located in a centralised component of the system, either at the node borders or within specific nodes. It has the ability to

examine all incoming and outgoing Internet traffic when installed at the border router. This strategy, however, may ignore attacks that occur outside the border router or nodes.

- **Distributed Strategy:** In this approach, each node in the system hosts an IDS, which enables cooperative intrusion detection via inter-node communication.

- **Hybrid Strategy:** The hybrid strategy combines the centralised and spread methods to find a good mix between detecting accuracy, algorithm complexity, energy use, memory use, and communication costs [99]. A hybrid technique is attractive but requires resolving data availability issues. Successful IDSs are dependent not only on anomaly detection models but also on access to meaningful data [100] for robust analysis and decision-making.

2.10 Related Work

In this subsection, we offer a high-level overview of why we selected the IoT security domain as the focus of our research. Each chapter delves into detailed discussions of related works pertinent to the specific contribution.

In the IoT ecosystem, there are presently several security solutions that rely on hybrid, anomaly, and signature implementations. However, the range of potential attacks has grown due to the constantly changing technological environment and the exponential increase in data created by these technologies. This development has made current IoT security methods less effective [101]. Makhdoom et al. [102] have made significant efforts to identify general and layerwise IoT threats. Within the context of the survey paper, the primary focus of the researcher is to concretely define the structure of malicious activity made against the IoT environment. They also talked about effective attack techniques and the use of IoT botnets for DDoS attacks. Despite substantial research on IoT security and solutions, there is still a need to enhance security solutions that use minimal energy and memory while providing excellent accuracy.

Mosenia et al. [76] utilized the Cisco seven-level reference model [103] in order to illustrate a variety of various attack scenarios. They evaluated IoT targeted attacks and mitigating techniques. The authors stressed the importance of taking a proactive approach

to IoT security.

The field of network data analysis has seen the emergence of a multitude of security frameworks and tools, each of which has its own unique set of constraints that leave critical parts unaddressed [80]. On the other hand, security solutions [104] favour an alternative methodology, frequently reliant on predetermined attributes and utilising proprietary extraction methods that are distinguished by their inflexibility. While this methodology could potentially be appropriate for specific applications, its absence of flexibility and customizability presents difficulties in situations that demand flexibility. Because network data is so varied, it is important to be able to adapt feature extraction methods to different data patterns in order to get accurate and useful results.

A large number of researchers have studied the IoT in order to provide insights into its complex ecosystems. Some have described IoT problems. Scholarly articles [105, 106, 107] extensively explore security concerns, delving into a multitude of attack types. Additionally, security holes in IoT systems are highlighted in a paper [108], with a focus on Zigbee technology flaws. Further challenges in the IoT have been highlighted by other researchers, as demonstrated by the research mentioned in [109, 110] along with others.

The methodology proposed by M. Bhuyan et al. in [111] for evaluating DDoS attack information metrics employs various information entropy measures such as Renyi's entropy, Shannon entropy, and Hartley entropy. Additionally, they utilize Kullback-Leibler divergence information distance measures to identify DDoS attacks effectively. This comprehensive approach enables a thorough examination of both legitimate and malicious attack traffic. However, its drawback lies in its demand for considerable computing power and resources, rendering it unsuitable for IoT networks.

This limitation underscores the need to develop lightweight security solutions tailored specifically for IoT networks. Given the resource constraints inherent in IoT devices, implementing heavy-duty security mechanisms like those proposed by Bhuyan et al. may prove impractical. Therefore, there's a compelling motivation to innovate security solutions that are efficient, resource-friendly, and optimized for the unique demands of IoT environments. By addressing these challenges, we can ensure robust protection against DDoS attacks while preserving the operational integrity of IoT networks.

Network traffic monitoring can be transparent or non-transparent. In non-transparent

mode [112], IDS nodes query or probe, alerting malicious nodes and increasing network congestion. In transparent mode [113], IDS nodes observe without adding overhead but face a trade-off between coverage and energy. Hence, we motivate the development of an IDS placement strategy for transparent monitoring to optimize coverage and energy use.

Numerous research studies [114, 115] have adopted a hybrid placement strategy that harnesses the advantages of both centralized and distributed approaches. However, these studies often focus solely on detecting single attacks [116, 117, 118]. Given the possibility of multiple concurrent attacks in a network, this scenario motivates us to develop solutions capable of detecting and mitigating multiple attacks simultaneously.

2.11 Summary

In this chapter, we first present a brief background on the IoT ecosystems and IoT architecture, followed by IoT applications. An exhaustive investigation into the various operating systems that are commonly used in the IoT domain is also carried out. We also analyse various simulators used in the literature. Subsequently, we have presented a brief discussion of the various IoT attacks in a layerwise manner. Following that, we explored a wide variety of datasets that are open to the public in order to evaluate the overall accuracy of offered security solutions in terms of attack detection, energy consumption, memory utilisation, and scalability. Additionally, we have investigated a variety of ML and DL models that are used in the IoT ecosystem. We have also reviewed the important literature on IDS for the IoT ecosystem. Finally, we discussed significant related works, such as attack detection and mitigation strategies used to enhance IoT network security. In the next chapter, we introduce an edge-based machine learning (ML) method that makes it easier for IDS to find DDoS attacks in IoT networks.

"As IoT devices proliferate, machine learning becomes the sentinel, tirelessly analyzing patterns to protect against emerging cyber threats."

- Andrew Ng

CHAPTER

3

Machine Learning for IEEE 802.15.4e/TSCH: An Energy-Efficient Approach to Detect DDoS Attacks

3.1 Introduction

Internet of Things (IoT) is a booming area of research in the current technological world with smart devices and applications like smart cities, smart health care, smart transportation, smart agriculture, etc. A complete IoT ecosystem has different layers as shown in Figure 3.1. The physical layer, MAC layer, adaption layer or 6LoWPAN layer, network layer, transport layer and application layer.

It coordinates millions of smart devices to support various intelligent applications to provide comfort to life. All the IoT devices communicate with each other and the environment intelligently without much human intervention. It is estimated to reach 50 billion devices by 2020 [119] and hence is one of the fastest-growing fields of computing. Even though it plays a vital role in raising the standards of life, it suffers from limitations like limited computation capabilities and security threats due to many components in setting up the IoT environment. It becomes challenging to implement encryption, authentication, etc., due to weaker computation capability and unattended automated environments. IoT environments are more vulnerable to security breaches at different layers. Therefore, to improve IoT environment security, various machine learning and deep learning algorithms are adopted.

Due to the resource constraint nature of the devices, new protocols are designed for

the IoT environment at different layers. Figure 3.1 illustrates the protocol stack comparison between IoT and traditional networks, while Figure 3.2 displays the IETF Suite of Protocols tailored for Industrial IoT environments.

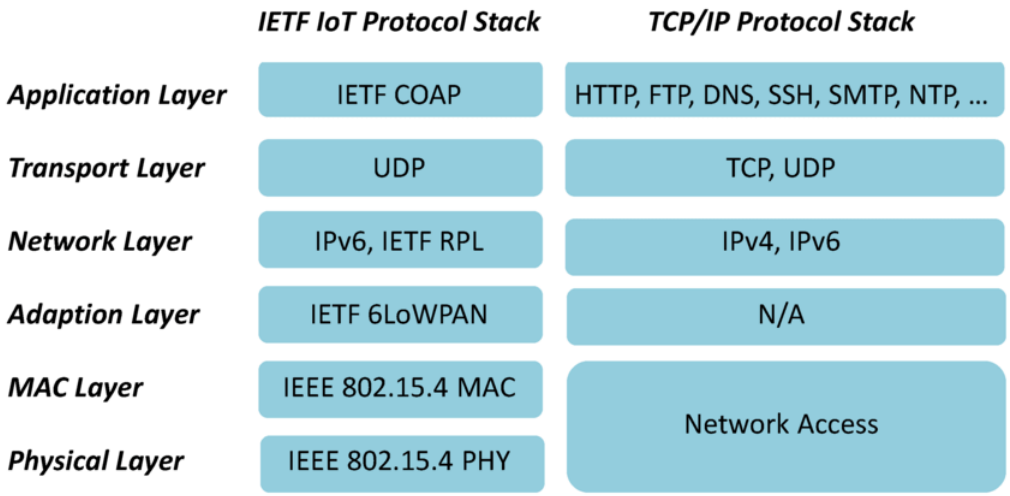


Figure 3.1: The comparison of protocol stacks of IETF IoT and TCP/IP [1]

These devices are connected as *Low Power Wireless Personal Area Network over IPv6 (6LoWPAN)*. 6LoWPAN networks are connected to internet using a *6LoWPAN Border Router (6BR)*. *6BR* acts as an gateway router for *6LoWPAN* to connect with internet. The link layer and physical layer uses IEEE 802.15.4 for the IoT network. Resource constraint devices are connected directly to internet which make them more vulnerable to the security threats from outside as well from inside the 6LoWPAN networks. Although message encryption and authentication is provided with IPsec [120] and a lightweight DTLS [121] and at link layer using IEEE 802.15.4 [122], IoT environment are prone to many attacks. It is challenging to deal with security threats in IoT due to several reasons such as direct access to internet, resource scarcity, newly designed IoT protocol stack, etc.

IoT applications are vulnerable to attacks at multiple surfaces [123] in a system, Physical attacks, Protocol-based attacks, Data and Storage based attacks, and Application and Software-based attacks. Our focus is on protocol-based attacks, and these are further classified as: Connectivity protocol-based attacks include connecting technologies such as RFID, Zigbee, NFC, WiFi. Network protocol-based attacks consider the network layer RPL and 6LoWPAN protocols. A communication-based protocol such as TCP/UDP at Transport layer and CoAP, MQTT at the application layer.

Over the last decade, ML-based security solutions have been implemented in a wireless network to reduce the attack surface [124] [85]. Network traffic investigates with the help of an *artificial neural network (ANN)*. They give a good result with complex non-linear learning. The prevalent ANN-based IDS install in all node and train to identify either an attack class traffic or benign traffic. They use a strong CPU or GPU in the training process.

In general, ANN-based IDS installed distributed manner is only feasible if the count of IDS is minimum. One strategy to achieve the purpose is to fix in devices and routers. As per our argument, it is useless to practice ANN-based IDS with this strategy. All devices were performing offline processing, learning, and high computational complexity. This activity is unmanageable to run in a resource-constrained network.

This chapter introduces an edge-based ML strategy that is efficient, trained to mimic IoT network traffic patterns, and whose performance incrementally improves overtime. The architecture of Edge-based IDS is illustrated in Figure 3.3. First, the features of the IDS is external and internal traffic analyzed dedicatedly. We note that while training the Edge-based ML approach, no more than one instance is stored in memory at a time. This parameter is used to enhance the response time with a fair trade-off in detection performance.

The goal of using an edge-based ML approach is because of all types of traffic analysis, and feature selection is achieved dedicatedly. From our analyses, we observed that edge-based ML strategy enhance the packet processing rate by a factor of five, and perform adequately than closely related works. The significant features of this security method as follows:

- The proposed approach generates both types of traffics (i.e., benign and spiteful traffic) utilizing the Contiki cooja simulator [125] [126]. It also examines a well-known dataset [79] [127] for the DDoS attack identification.
- A novel edge-based ML solution for the resource-constrained network, which is lightweight and energy-efficient.
- A feature selection is dynamically preparing and deriving certain contextual features from network traffic.
- The proposed approach incorporates distributed edge-based ML strategies. In a distributed environment, Edge devices communicate with each other.

- The proposed method can adapt hybrid traffic (i.e., internal traffic and external traffic) in the IoT ecosystem. Hence DDoS attacks can be identified and depreciate the future event of DDoS attacks.

- The intended approach achieves the IEEE 802.15. 4e/TSCH traffic characteristics of DDoS attack. It classifies the attack applying edge-based ML with High Accuracy and minimum response time.

3.2 Background and Related Work

This section provides an overview of the relevant background work to be specifically addressed in this chapter. Section 3.2.1 describes the 6LoWPAN and TSCH models in a comprehensive manner, while Section 3.2.2 provides an overview of the IoT attacks that will be discussed in this chapter. Further, Section 3.2.3 elaborates on the related works that are especially relevant to this chapter.

3.2.1 6LoWPAN and TSCH

6LoWPAN provides an efficient way to forward IP packets in low power lossy network (LLN). It makes IP header compression. 6LoWPAN incorporates routing specifications, causing it to enlarge the network range by connecting the devices in a mesh. The IEEE 802.15. 4e/TSCH protocol specially designed for Low power wireless networks. It is a MAC protocol and does *Time Division Multiple Access (TDMA)*. It is compatible with IEEE 802.15.4 physical layer [128] and 6LoWPAN network layers. This protocol very much recognized when it gives wire-like reliability in LLN. The IETF 6TiSCH Working Group involved in the 6LoWPAN standard and IEEE 802.15.4-2015 TSCH integration process. Based on the integration process, many RFC and Internet drafts are composed [129].

3.2.2 Attacks on IoT

IoT networks are prone to various attacks due to the end-node accessibility through the Internet, the lossy nature of networks, and resource constraints of the nodes. The nature of these attacks varies vastly with the varying objectives of the attacks, like interrupting network traffic, exhausting network resources, disrupting the topology, etc. This chapter is

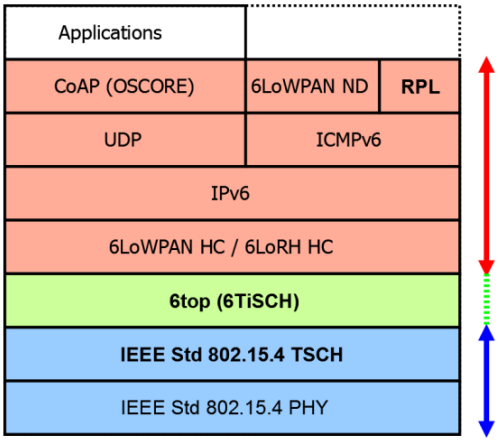


Figure 3.2: 6TiSCH-Stack: IETF Suite of Protocols for Industrial IoTs

focused on DDoS attacks, BashLite and Mirai botnet, as given below.

DDoS attack

It is a malicious attempt to interrupt the normal operation of a certain server, service, or network by flooding it with internet traffic. In Section 2.6 of Chapter 2, we explored a multitude of DDoS attacks and other forms of attacks within the IoT ecosystem.

Botnet attack

Originally, the word botnet referred to a robot network. It is composed of IoT nodes that host along with the affected nodes. By exploiting IoT vulnerabilities, many attacks are launched using IoT nodes to inject botnet malware into them. Botnets are able to control internet-connected devices remotely via remote access [130], [131]. This way, IoT nodes can be used for hosting robots that can execute attacks while the attacker remains anonymous. The botnet attacks are correlated with various attacks (i.e., ransomware, remote command execution, brute force attacks, backdoor installation, and DDoS attack). The most common botnet attacks among IoT botnets are:

- Mirai: It is a very popular IoT botnet for DDoS attacks that can infect 4,000 to 6000 IoT devices in just a few minutes [130].
- Bashlite: It is also known as Torlus and Gafgyt, which target IoT nodes that have firmware based on Linux [131].

3.2.3 Related Work

Related work is divided into two-part 1) ML-based security solution. 2) Generalised DDoS attack detection method. The machine learning-based security approach is implemented in wireless networks widely, as shown in Table 3.1. However, these approaches normally design for resource-rich devices, not for resource-constrained devices. ML training and executing phase is expensive in terms of computing and storage. Executing phase runs on router and gateways. The significant limitations of this approach are required a massive amount of resources and computational power.

Table 3.1: ML-based related works.

Solution\Year	**Detail**
E. Hodo et al. [124] 2016	It used an artificial neural network (ANN) as an intrusion detection system (IDS) to identify DoS/DDoS attacks.
M. K. Putchla et al. [132] 2017	It used Deep Learning and Gated Recurrent Neural Networks (GRNN). It is the first paper that implemented GRNN in the WSN.
C. Li et al. [133] 2018	It uses a deep learning model. It comprises three layers: 1) Input layer, 2) recursive forward and reverse layer 3) output layer. It also used are RNN, LSTM, CNN.
A. Guerra et al. [134] 2019	This paper used the sandwich feature selection method: this method combines Fisher's score, Pearson's correlation coefficient ρ, and wrapper step. Its boosts up the detection accuracy.
M. Shafiq et al. [135] 2020	This paper gives the objective is an efficient solution with a modified feature selection mechanism. A novel CorrAUC is applied for feature selection. It also integrates Shannon entropy and soft bijective function.

D. Yin et al. [136] this approach, DDoS attack originated device identified quickly. This approach also mitigates the DDoS attack in the IoT ecosystem. However, the drawback of this approach is the additional overhead expected for the cosine similarity module. Hence this approach not suitable for resource constraint networks.

As per H. Chen et al. [137] identify and alleviate DDoS attacks using Hilbert-Huang transformation and trust evaluation. However, the drawback of this solution is that it requires added resources (i.e., memory, battery, etc.). Therefore, this approach is not fit for the 6LoWPAN network.

Table 3.2: Non ML based related works.

Solution\Year	Detail
D. Yin et al. [136] 2018	SD-IoT incorporates a controller pool, hybrid switch, and IoT devices. The approach is based on cosine similarity of the IoT packet rate.
H. Chen et al. [137] 2019	identify and alleviate DDoS attacks using Hilbert-Huang transformation and trust evaluation. It requires added resources (i.e., memory, battery, etc.).
Y. Jia et al. [138] 2020	In this approach, two major components are incorporated (i.e., flow filter, flow handler). The flow filter is also subdivided into two parts 1) flow filtration and 2) DDoS discovery.

As per Y. Jia et al. [138], DDoS attacks identified using the FlowGuard model. this model contains a flow filter and flow handler. This solution executes on an edge-centric device and gives valid results. However, this solution also has disadvantages, like its outcomes show more FNR and FPR.

3.3 Proposed Works

This section performs the edge-based IDS: it contains three steps: 1) packet preprocessing, 2) feature extractor, and the last 3) intrusion detection. Additionally, we analyze the complexity of the IDS and give runtime performance.

3.3.1 Overview

Edge-based IDS, based on Naive Bayes, and designed for the DDoS attack identification. This IDS execution is based on the Bayes theorem [139]. Each edge-based IDS module is capable of detecting DDoS attacks. Edge-based IDS is designed to operate on a scalable IoT network and edge device (i.e., 6BR router). Edge-based ML enable IDS has been composed with low computing complexity and small memory storage.

The edge-based ML approach is comprised of the following components:

- Packet capture: During experiments 3.4.1 and 3.4.2, we gather packets in Contiki cooja using Wireshark.
- Feature Extractor (FE): In this component, Feature Extracted using Principle Com-

ponent Analysis (PCA) [140].

- Fitting Naive Bayes to the Training data Set: In this component, we use generated data and available data set [79]. We utilized the GaussianNB classifier to fit the training dataset.

- Classification and Result: This component is adopted to predict the test set result. And classify DDoS attack traffic and benign traffic.

3.3.2 Naive-bayes classifier

The proposed model is an intelligent IDS based UDP flood attack detection system which uses a naive-bayes classifier. It is a Edge-based ML enable IDS that analyses all type of traffic received from internal and external device. UDP flood affects the transport layer of the protocol stack, leads to denial of service to legitimate users, and consumes resources in a resource-constrained environment.

The algorithm is implemented at edge node of the IoT environment. Since the features are independent of each other, naive-bayes is considered suitable for detecting the anomaly. Figure 3.3 shows the experimental setup and placement of the IDS in the IEEE 802.15.4e (6TiSCH) network.

The naive bayes classifier is a supervised learning technique that uses bayes theorem for the classification and prediction of data. It is generally used in situations where the features are not dependent on each other. It gives equal weightage to all the features. It is simple, robust, easy to implement, and applied for two-class and multi-class problems. The basic flow chart of the proposed model is displayed in Figure 3.4. The model is trained with features of the captured UDP packets like the source IP address, port number, packet size, and correct labels, namely, normal and abnormal (anomalous) data. After the model is built, it can detect a new sample as normal or an anomaly. We can apply the naive bayes to our model by using the equation below.

$$P(b/A) = \frac{P(A/b)P(b)}{P(A)} \tag{3.1}$$

where b indicates the class labels, normal and abnormal and A is the feature vector. It can be written as where the a_i s are the features. We can ignore the denominator as it

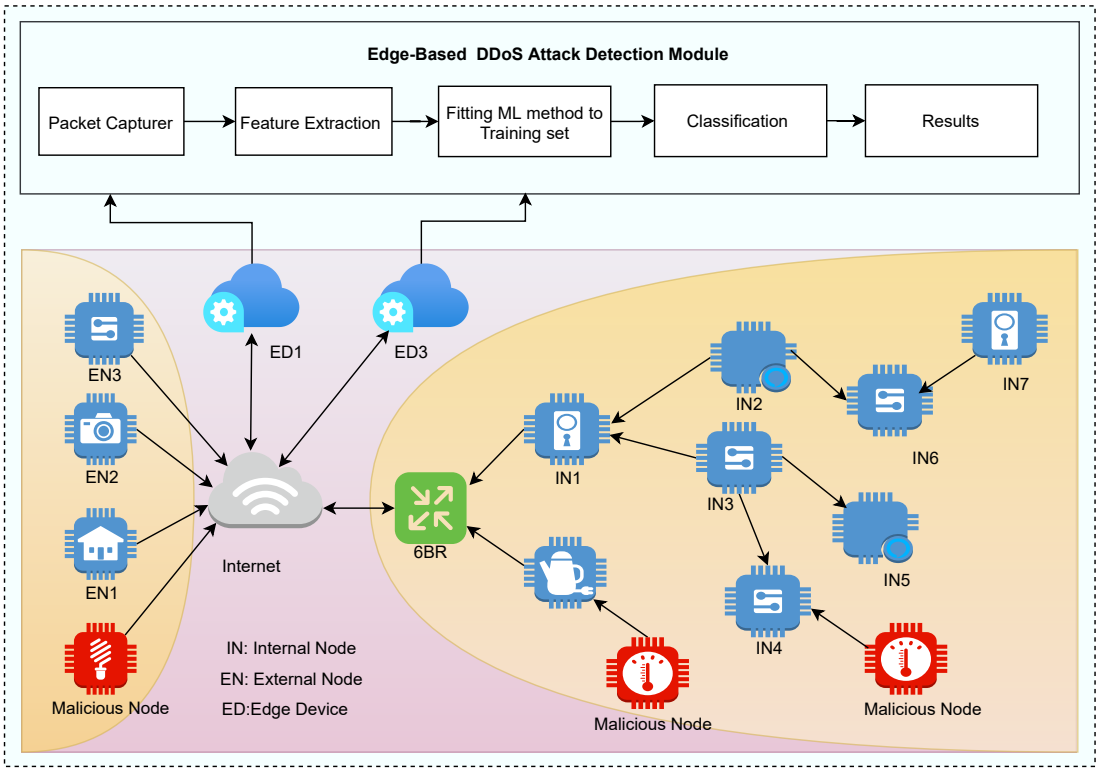


Figure 3.3: Experimental setup and IDS module architecture

is constant for a given input. Here, b is the proof, and A is the ground. The hypothesis presented is that the predictors/features (b/A) are independent. The behavior of one distinct feature does not influence the other. Therefore it is named naive.

The features like $a_1, a_2, a_3, ...a_n$ substitute for A and expanding the chain rule. The equation (3.2) can be written as follows:

$$P\left(b/a_1, \ldots, a_n\right) \alpha P(b) \prod_{i=1}^{n} P\left(a_i/b\right) \tag{3.2}$$

$P\left(a_i/b_j\right)$ is evaluated for each a_i in A and b_j in b, and also $P\left(b\right)$ where b is either normal or abnormal data is calculated. Once these are evaluated, the model becomes ready.

When a new UDP packet arrives, the target label is found out using maximum a posteriori (MAP), i.e., it is assigned the label for the class that has a maximum probability. The *class variable* (b) has two results, like (yes and no). Different cases have multivariate

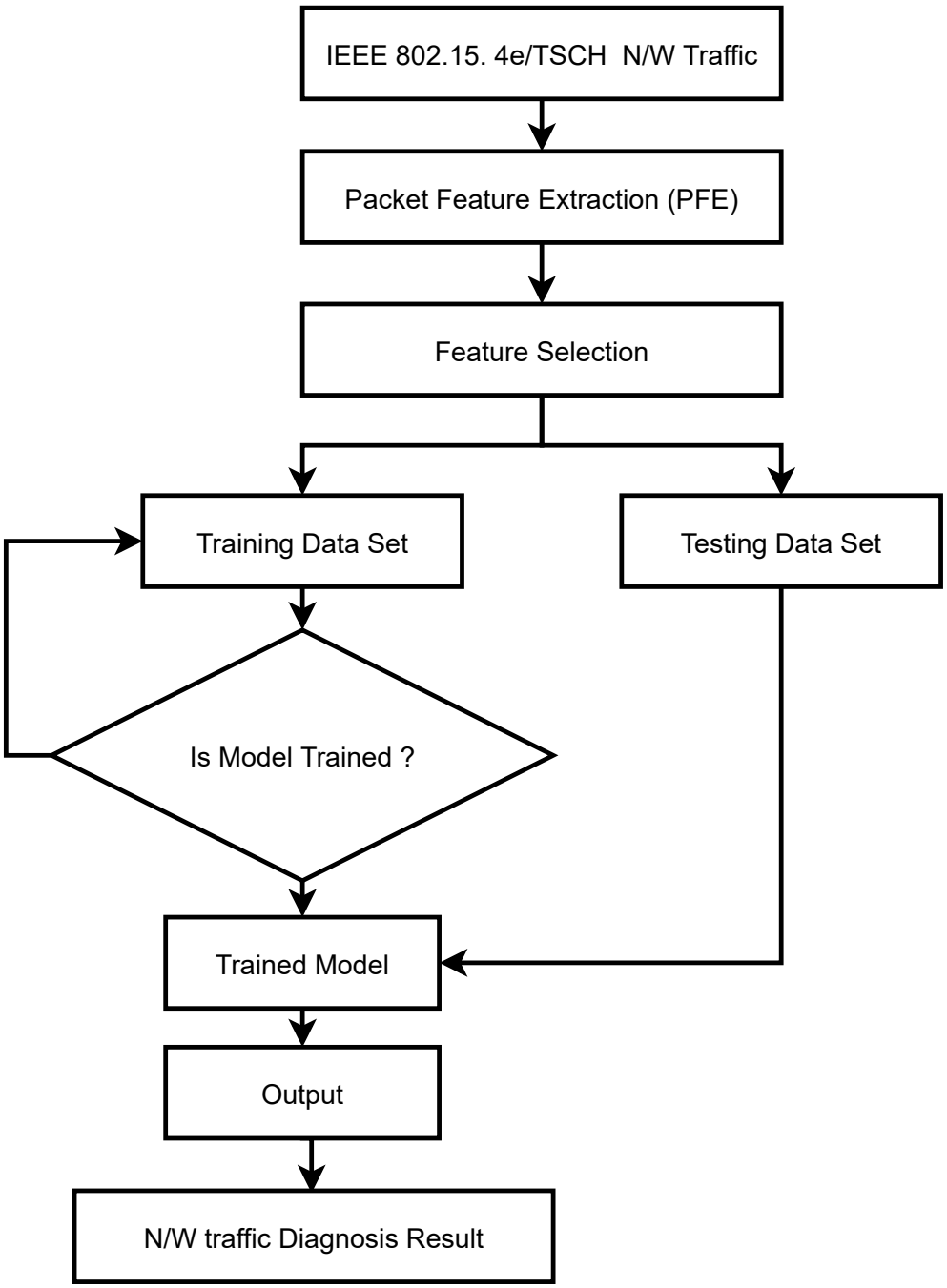


Figure 3.4: Flow chart of IEEE 802.15. 4e/TSCH DDoS attack detection system

classification. Therefore, the highest probability of class b is calculated as:

$$b = \arg\max_{b} P(b) \prod_{i=1}^{n} P\left(a_i/b\right) \tag{3.3}$$

Where, b represents the parameter or value that maximizes the expression, $P(b)$ denotes

the probability of observing parameter b, $\prod_{i=1}^{n}$ signifies the product operation over n terms, $P(a_i/b)$ represents the conditional probability of observing data a_i given the parameter b, and $\arg\max$ refers to the argument that maximizes the function or expression.

In essence, the equation seeks to find the parameter b that maximizes the joint probability of observing all data points $a_1, a_2, ..., a_n$ given b, thereby capturing the optimal parameter value that best explains the observed data.

3.4 Experiments and results analysis

In order to illustrate the proposed approach, some assumptions are built for the IoT network. Our approach assumes that the network consists of *Tmotes Sky* CC2420 IEEE 802.15.4-compatible radio chip [141], and it can be achieved in *Contiki cooja* [125] [126] emulator environments. The edge device has more computing power and an extra hardware component. To corroborate the intended approach of identifying DDoS attacks. Figure 3.3. exhibited IEEE 802.15.4e (6TiSCH) network.

In experiment part we conduct three type of experiments as follows:

3.4.1 Non-attack circumstance:

In non-attack circumstances, the *legitimate devices* (D_L) practice skywebsence web server for gaining their services. During the experiment, we consider 8, 16, 32, and 64 IoT devices. IoT network traffic is screened at 6BR, and traffic is forwarded toward the *edge device (ED)* to implement ML solution. In ED, we employ the *Naive Bayes* ML approach for accurately detecting DDoS attacks in IEEE 802.15.4e (6TiSCH) network.

3.4.2 DoS/ DDoS attack circumstance:

In attack circumstances, legitimate and malicious devices (D_M) practice skywebsence web-server for aveling IoT services. Different tools are used for injecting DDoS attacks (i.e., libcoap, hping, etc.). These tools generated traffic hinder the skywebsence web-server activity by composing recurring CoAP connections. It exploits the default timeout scheme of the skywebsence web-server. The assailant can be modified the default timeout scheme by impelling the attack. DDoS attack injected using 2, 4, 6, and 8 malicious nodes are

shown in Figure 3.5.

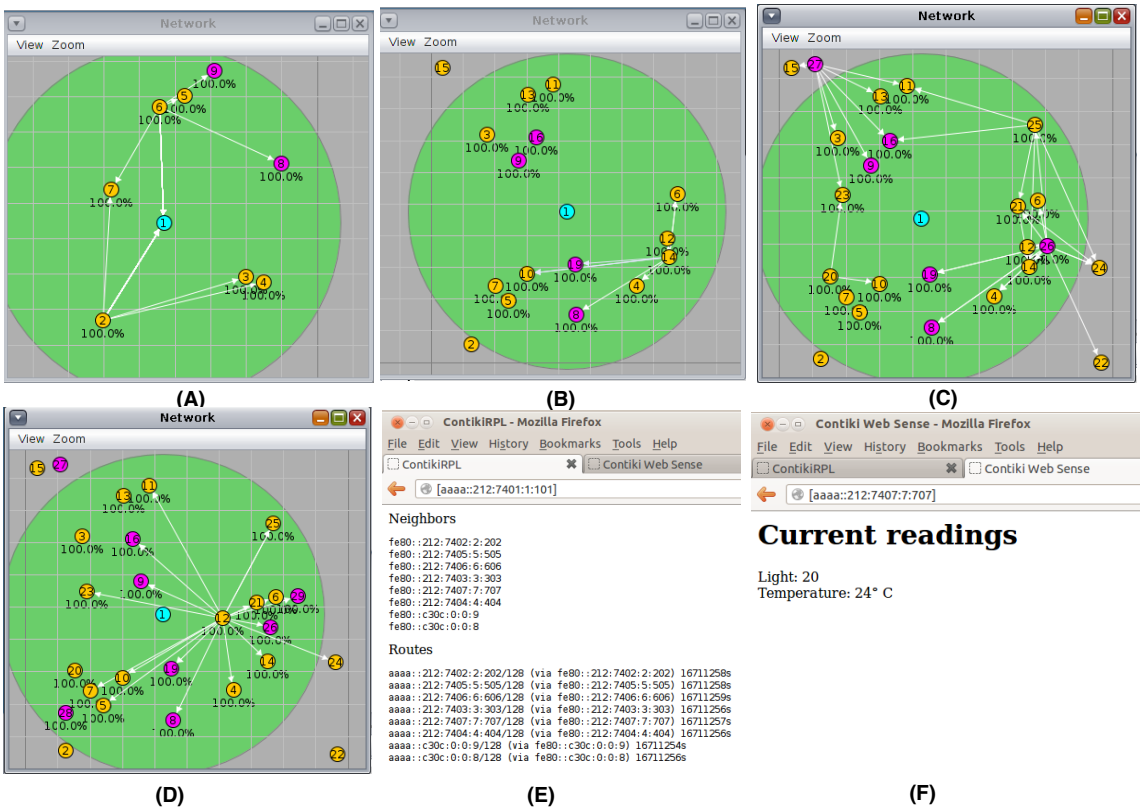


Figure 3.5: Snapshot of DDoS attack scenario in IEEE 802.15.4e (6TiSCH)

3.4.3 Execute experiment with proposed solution:

The intended approach autonomously operates on an edge device. It does not demand extra computing storage capacity from resource-constrained devices in IEEE 802.15.4e (6TiSCH) network. Hence, our approach is lightweight. The performance evaluation is based on *accuracy*, *precision*, *recall*, and *F-measure* are given below:

The performance analysis evaluation is realized using a stratified percentage split evaluation scheme, where 80% of total data is used for training and 20% for testing in randomly selected. The classification performances are expressed by the following performance parameters which describe the performance of binary classification over generated data set and Kitsuni data set [127].

Accuracy: It is the ratio of the number of correct predictions to the total number of

input samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

Precision (PRC): Precision is about when it predicts yes, how often is it correct.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.5}$$

Recall (RCL): It is referred to as the TPR or sensitivity

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.6}$$

F-measure (FM): It is the balance between the PRC and the RCL

$$F - \text{ measure } = 2 * \frac{RCL * PRC}{RCL + PRC} \tag{3.7}$$

Where TP, TN, FP, FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

From experiments 3.4.1 and 3.4.2, we generate a massive amount of normal and attacked IoT traffic. We conduct experiments 3.4.1 and 3.4.2 recursively. In experiment 3.4.3, we implement the *Naive Bayes* method in IEEE 802.15. 4e/TSCH network. This ML approach is efficiently identifying DDoS attacks using generated data and the kitsuni dataset [127].

Table 3.3: Evaluation Metrics for generated data and Kitsuni Mirai dataset

Dataset Used	**Generated Dataset**				**Kitsuni Mirai Dataset**			
No. of IoT Node	**16N**	**32N**	**64N**	**128N**	**16N**	**32N**	**64N**	**128N**
Energy Usage of IoT network (mJ)	48260	83040	87540	135000	46108	84520	85916	134080
Training Time (Sec.)	23	29	35	40	39	50	84	102
Accuracy (%)	96.4%	97.6%	98%	98.2%	98%	98%	98.5%	99.3%

Table 3.3 presents the comparative study of the proposed edge-based ML approach with generated data from Contiki cooja [125] and Kitsuni dataset [127]. This table clearly shows energy usage of network, training time, and accuracy for generated data and kitsuni dataset comparatively stable than other closely related work.

Table 3.4 shows a comparative analysis with the recent (2016–2020) approaches. The shortcomings of the current methods, which are perceived from Table 3.4, are as follows:

- Solutions are not based on resource constrain IEEE 802.15.4e (6TiSCH) network.
- Solutions planted extra overhead on IEEE 802.15.4e (6TiSCH) node.

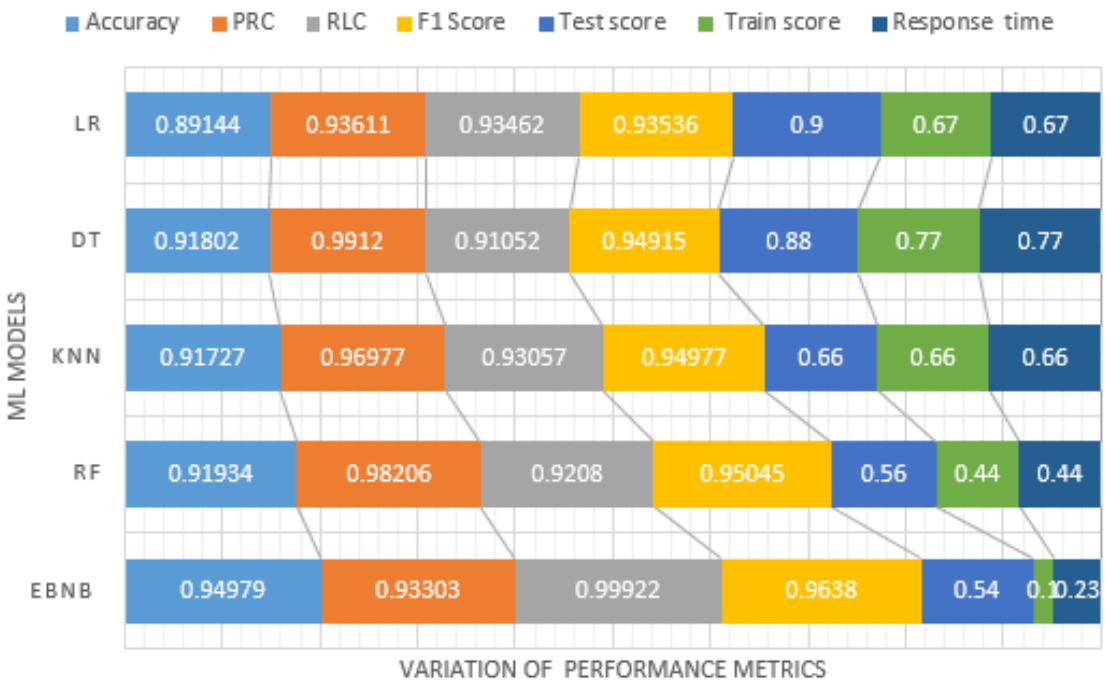


Figure 3.6: Evaluation Metrics with various ML models

Based on the experimental outcomes, it is clear that the proposed approach is scalable and gives comparable memory utilization ROM/RAM, Energy Usage, Response Time, and Accuracy. The average memory utilization and energy usage measured by the intended solution are $35834B/5378B$ and 85916 mJ. Table 3.4 also exhibits average accuracy and response time are 98.7% and $(24.2 - 68.9)$ $Sec.$ respectively.

We use four distinct ML approach in the IEEE 802.15.4e (6TiSCH) shown in Figure 3.6. In each ML approach in our experimental setup, we run experiments recursively. Figure 3.6 shows our EBNB approach outperformed compared to other ML models.

The different DDoS attack datasets are compared with the proposed approach, as shown in Figure 3.7. It shows that the proposed solution performs better with the combination of Generated data and Kitsuni Mirai dataset [127] than other DDoS data set.

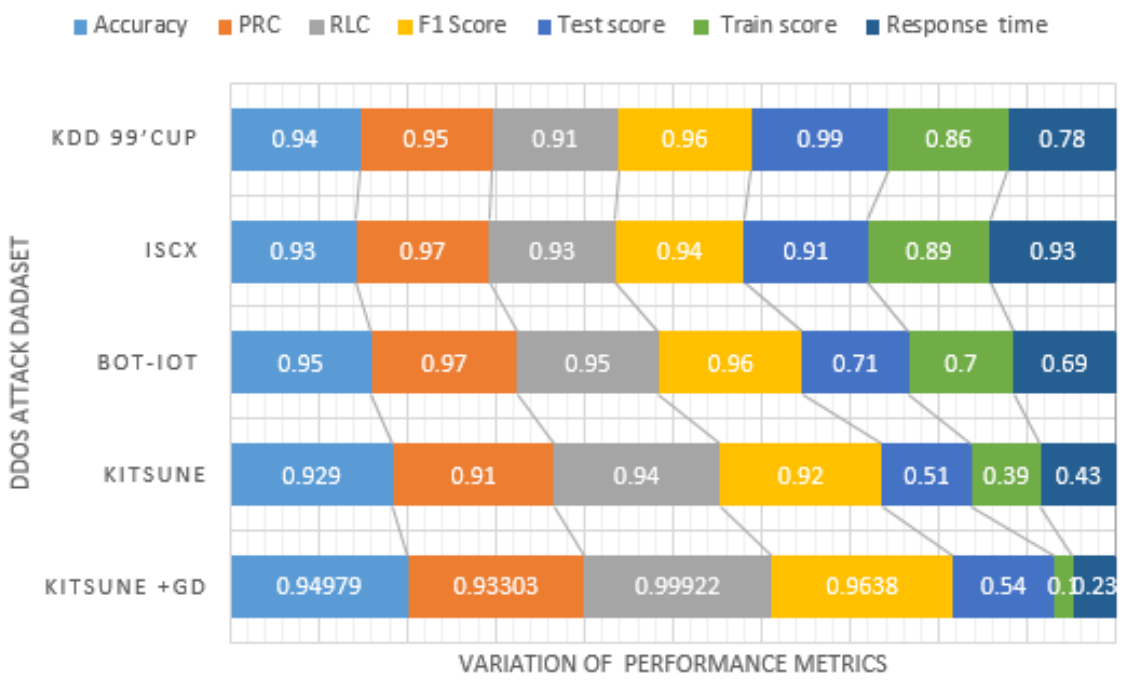


Figure 3.7: Evaluation Metrics with different dataset

Table 3.4: Comparison of the proposed strategy with the closely related works

Refrences	Simulated Environment	MEMU (in byte) ROM/RAM	Energy Usage (mJ)	Scalability	Accuracy (%)	Response Time (Sec.)
E. Hodo et al. (2016)[124]	N/A	42317/8738	76290	N	97.4 %	(33.8-98.5) Sec.
M. K. Putchala et al. (2017)[132]	NA	54914/12587	96413	N	97%	(54.8- 113.2) Sec.
C. Li et al. (2018)[133]	Open Flow SDN	N/A	116857	Y	98.60%	(94.7-198) Sec.
A. Guerra et al. (2019)[134]	N/A	N/A	98082	N	98.80%	(N/A)
M. Shafiq et al. (2020) [135]	N/A	45672/9832	120580	N	96%	(28.1-169.8) Sec.
Proposed Solution	**Contaki, EBNB**	**35834/5378**	**85916**	**Y**	**98.7%**	**(24.2-68.9) Sec.**
MEMU =MEMory Utilization, N/A = Not Applicable, EBNB= Edge Based Naive Bayes						

3.5 Summary

In this chapter, we developed an edge-based ML enable approach that observes the IEEE 802.15.4e (6TiSCH) network traffic and shield against the DDoS attack. The naive bayes method is analyzed generated DDoS attack traffic and well known DDoS attack dataset. The performance of the proposed approach is assessed on the Contiki cooja IEEE 802.15.4e environment and kitsuni data set. We also compare the proposed method with different DDoS datasets. The comparative analysis with current solutions with closely related work is also offered. Experimental outcomes show that the intended solution can identify the DDoS

attacks. Hence, DDoS attacks are depreciated as the proposed solution identifies the DDoS attack using the EBNB solution. Accuracy and response time outperformed compared to other ML models. The results exhibit that the average accuracy and response time are 98.7% and $(24.2 - 68.9)$ *Sec.* respectively, compared with the closely related work. In the upcoming chapter, we present a strategy for identifying and mitigating LrDDoS attacks within the IoT ecosystem. This methodology involves the utilization of a Packet Inspecting Agent (PIA).

"IoT security requires a proactive mindset; it is about anticipating threats before they materialize."

- Philip Lieberman

CHAPTER

4

LORD: LOw Rate DDoS Attack Detection and Mitigation Using Lightweight Distributed Packet Inspection Agent in IoT Ecosystem

4.1 Introduction

The IoT is a large ecosystem of devices and objects. Within this ecosystem, there are various types of devices, including resource-constrained ones like sensors and actuators, which utilise the IPv6 protocol [142]. These resource-constrained devices often function within networks known as 6LoWPAN. These networks were developed explicitly to allow low-power wireless communication and primarily make use of the IEEE 802.15.4 standard in the 2.4-GHz frequency band [143]. However, 6LoWPAN networks connect directly to the Internet, which means that assailants could get into these resource-constrained devices from anywhere on the Internet.

IoT applications span numerous domains, including smart communities, residential automation, and logistics, among others. These applications and services are extremely valuable. As a result, ensuring the integrity, confidentiality, and availability of the data linked with these services becomes the primary concern. The main threat to the 6LoWPAN network is the DDoS attack. It is a type of Denial-of-service (DoS) attack [144] in which several vulnerable devices with limited resources are simultaneously targeted. In the 6LoWPAN network, the DDoS attacks are basically classified into two categories: low-rate DDoS and high-rate DDoS (i.e.,LrDDoS and HrDDoS) [145]. In [146], a detailed analysis of the various security measures available to prevent DDoS assaults on wireless networks.

The purpose of an HrDDoS attack is to deny the services in the 6LoWPAN network to verified users. The assault is achieved by transferring plenty of traffic generated from a large number of endangered nodes. It also utilises the network bandwidth of the 6LoWPAN network. The main shortcoming of an HrDDoS attack is that it can be identified by security solutions due to its traffic characteristics (i.e., high volume). As an outcome, attackers are choosing the LrDDoS attack [147]. LrDDoS attacks are strenuous to recognise because the assault traffic characteristics are similar to genuine traffic. Such assaults exploit the 6LoWPAN protocol stack vulnerabilities rather than depleting resources and network bandwidth. As the intruder sends malicious packets with a low rate, it does not get caught by security solutions constructed based on network-level traffic characteristics. The goal of the assault is to degrade the quality of service (QoS) encountered by an authorised end user instead of desisting from the services given by the 6LoWPAN network to the verified users.

Several DDoS (mainly HrDDoS) attack detection approaches for the wireless network have been proposed. Many of them are standard *Anomaly Detection Systems (ADS)*. The LrDDoS attacks are intelligent and have intermittent behavior. Detecting this type of attack using ADS is strenuous. On the 6LoWAPAN network, a limited study is available on the energy-efficient, lightweight solution to detect LrDDoS attacks. The LrDDoS attack starts from endangered nodes in both internal and external networks of 6LoWPAN. Hence, the supervision of both the internal and external network-level traffic of 6LoWPAN is crucial. The endangered nodes are controlled by the intruder. The present security mechanisms have a few gaps. These security solutions do not examine the real-time 6LoWPAN network traffic for LrDDoS attack identification. The current security solutions are non-adaptive and inadequate to identify internal LrDDoS. It also consumes more energy. To address these shortcomings, a novel distributed intelligent agent-based defence method is designed that identifies internal as well as external LrDDoS attacks in the 6LoWPAN network. The significant features of this defence approach are:

1. *Generation and analysis of attack and non-attack traffic:* The proposed method generates attack and non-attack traffic using Contiki cooja simulator. It also examines network traffic from both authorized and compromised nodes for the LrDDoS attack identification.

2. *Distributed:* The suggested method incorporates distributed lightweight energy efficient

Packet Inspection Agent (PIA). In a distributed environment PIA interacts with each other.

3. *Adaptability:* The proposed method is able to adapt to both external as well as internal network traffic in the IoT ecosystem. Therefore LrDDoS attack can be detected, and minimize the future occurrence of LrDDoS attack.

4. *Low false alarm rate:* The proposed method supervises the network level traffic characteristics of LrDDoS attack. The solution identifies the attack using PIA. Hence, both *False Negative Rate (FNR)* and *False Positive Rate (FPR)* are minimized.

The rest of the chapter is organized as follows. Section 4.2 contains background and related work. Section 4.3 explains the proposed security approach. Section 4.4 presents the experimental setup and implementation of LrDDoS attack in Contiki OS. Section 4.5 describes the analysis of the experimental results. Finally, in Section 4.6 we summarize the chapter.

4.2 Background and Related Work

This section presents a systematic overview of attack in the IoT ecosystem. We also discuss previous studies used for a secure IoT ecosystem.

4.2.1 IoT attack

LrDDoS attack:

LrDDoS attack aims to consume resources and bandwidth steadily [148]. This type of attack creates adequately low rate traffic to the targeted device or application. The variations among LrDDoS and LRDoS attack are shown in Figure 4.1.

In these attacks, spiteful network traffic differs in terms of time period (ΔT), Burst-Rate (BR), and Burst-Width (BW). It also reduces latency and throughput. LrDDoS and LRDoS attacks can remarkably reduce the availability of devices or application due to packet loss and disparity in RTT. The LrDDoS attack works differently from conventional DDoS attacks. The LrDDoS primarily targets the vulnerabilities possible in the *TCP congestion control scheme*. Due to this LrDDoS attack detection is a challenging and difficult task.

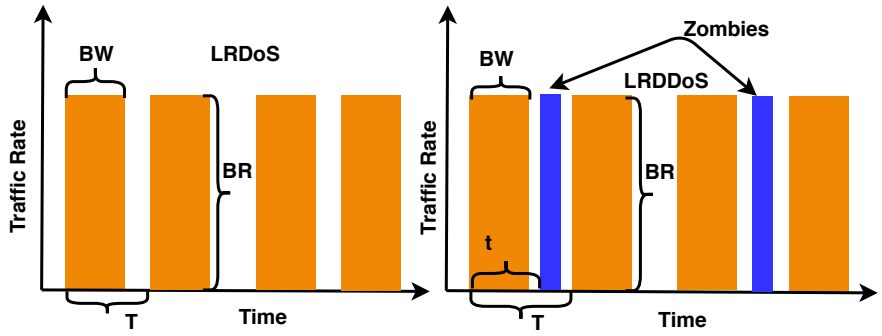


Figure 4.1: The variations among LrDDoS and LRDoS attack [2]

Automatic repeat request (ARQ) typically occurs because of network congestion. The TCP congestion control scheme comprises a couple of mechanisms as follows: 1) Duplicate ACK, which shows that the earlier sent packet is not received at the end device. The sender resends the missing packet using *Additive-increase and Multiplicative-decrease (AIMD) mechanism.* A legitimate sender depreciates the congestion window and decreases the packet sending rate that results in massive degradation of device and application availability. 2) Retransmission timeout (RTO), targeting global minimum RTO limit. If the router bottleneck link is equal to the minimum RTO limit then sender ceases sending packets due to the timeout limit. When the sender resends the missing packet to the end receiver, the new burst appears at sender side and the sender again enters into timeout state. Assailant recurs this activity to reduce TCP throughput by LrDDoS attack traffic that is very much similar to legitimate traffic.

4.2.2 Related Work

DDoS attacks are categorized into two classes on the basis of the protocol [149]. They are application and network layer DDoS attacks. Application-layer DDoS attacks disrupts authorized user's services (i.e., memory, CPU, I/O bandwidth, database bandwidth, and sockets) by consuming the enormous server resources. Network-level DDoS attacks occur by utilizing various protocol packets (i.e., DNS, ICMP, UDP, and TCP). It consumes I/O bandwidth of the network [149].

In [150], entropy metric-based approach was used to detect Low Rate DDoS attack. Depending on the various applications, request, response, and acknowledgment packet size are standardized. In the attack scenario, packet size varies for different attacks. This can

be utilized to detect low rate DDoS attacks to some extent. However, the drawbacks of this approach are totally depends on the packets analysis windows, restricted scalability, and large attack detection time.

The empirically DDoS attack information metrics evaluation approach proposed by M. Bhuyan *et al.* in [111] used different information entropy measures (i.e., Renyi's entropy, Shannon entropy, and Hartley entropy). They also apply *Kullback–Leibler* divergence information distance measures to detect DDoS attack. Using stipulated metrics helps a detailed analysis of legitimate and malicious attack traffic. However, the shortcoming of this approach is that it requires extra computing power and resources. Therefore, this method does not fit for IoT network.

H. Chen *et al.* [151] propose a hybrid solution that incorporates trust evaluation as well as *Hilbert-Huang Transformation* to identify *Low-Rate Denial of Service (LDoS)* attack. This hybrid approach executes an *intrinsic mode function (IMF)*. If the Kolmogorov-Smirnov and correlation coefficient value of the IMF component are higher than 0.4 and 0.3, respectively. the values indicate a higher trust rate of IMF function that is utilized to detect LDoS attack. However, the drawback of this hybrid solution is that it demands more resources (i.e., memory, battery, etc.). So this type of approach not suitable for resource constraint IoT ecosystem.

4.3 LrDDoS Detection and Mitigation Approach

In this section, we exhibit the proposed solution to detect and mitigate LrDDoS attack in the IoT ecosystem. This ecosystem contains heterogeneous devices in case of computing and storage capacity. Our *Packet Inspecting Agent (PIA)* module executes on the *6BR node* and intermediate nodes for packet analysis. It also performs a mitigation strategy to safeguard LrDDoS attack.

4.3.1 Security Architecture

The architecture consists of four types of nodes (i.e., 6LoWPAN border-router node (N_{6BR}), Packet inspecting agent node (N_{PIA}), Legitimate node (N_L), and Spiteful node (N_S)). Every IoT enabled devices are connected and controlled by the 6BR node. The 6BR and PIA nodes have extended computing as well as storage capacity that is used for the generating

Total Variation Metric (TVM) and *Packet Flow Count (PFC)* of IoT network traffic. The topology of the experimental setup under consideration in this research paper is presented in Figure 4.3. The LrDDoS attack occurs in the IoT network (i.e., skywebsenceserver) via 6BR. The distributed PIA node filters internal IoT network traffic. Hence, 6BR and PIA are the best spots in the IoT network for the intended LrDDoS attack identification approach. The illustrative layout of the proposed approach is exhibited in Figure 4.2.

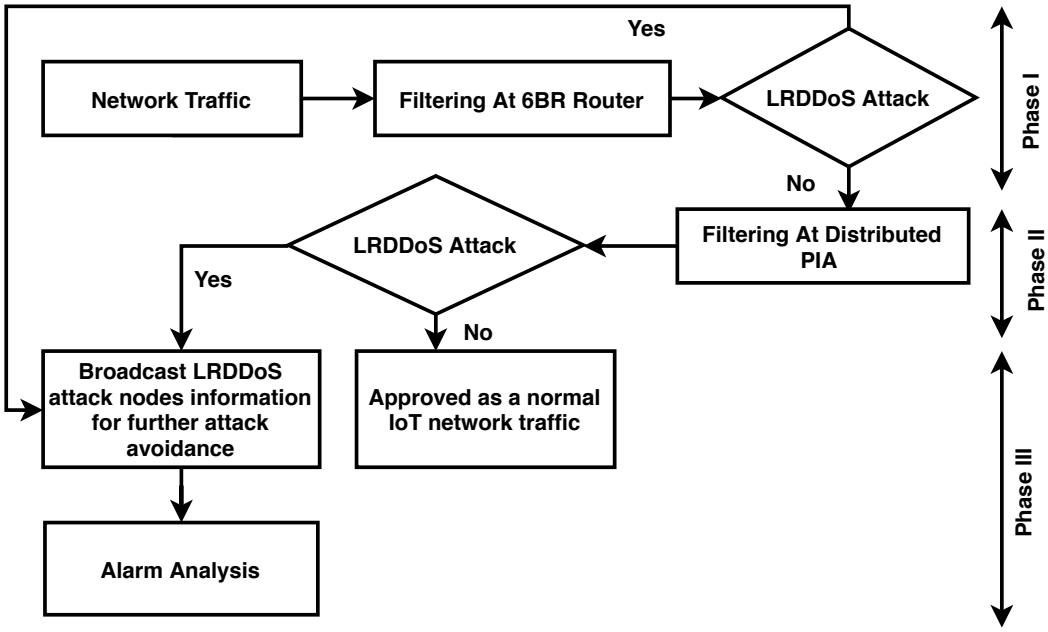


Figure 4.2: Illustrative layout of proposed approach.

4.3.2 Proposed Approach

The proposed approach incorporates three phases as follows: 1) LrDDoS attack detection at N_{6BR}, 2) LrDDoS attack detection at N_{PIA}, and 3) Mitigation of DDoS attack. In the first phase, input traffic is analyzed by packet arrival time. Our security approach identifies the legitimate IoT ecosystem traffic from DDoS attack traffic by utilizing *generalized total variation metric (GTV)*. In the GTV metric calculation, two consecutive traffic samples, and threshold $\Delta_{traff.}$ are used. The $\Delta_{traff.}$ is discovered by heuristics that depends on LrDDoS attack variations as shown in Figure 4.1 and deployed circumstances.

Generalized total variation metric (M_{GTV})

Let ϕ_1 and ϕ_2 imply a couple of discrete distributions across k distinct random values. Then the M_{GTV_μ} is represented with the following equation [152]:

$$M_{GTV_{\mu(\phi_1,\phi_2)}} = \left(\sum_{i=1}^{k} \left(|\phi_{1_i}(A_x) - \phi_{2_i}(A_y)|\right)^{\mu}\right)^{\min\left(1,\frac{1}{\mu}\right)} \tag{4.1}$$

where $\mu > 0$, A_x, A_y are the random unique samples and $\phi_1, \phi_2 \in \sum$ as explained previously. When $\mu \geq 1$, the difference between the two distributions (ϕ_1, ϕ_2) is also called the *Minkowski distance* [153].

$$M_{TV_{\mu(\phi_1,\phi_2)}} = \left(\sum_{i=1}^{k} \left(|(\phi_1)_i(A_x) - (\phi_2)_i(A_y)|\right)^{\mu}\right)^{\frac{1}{\mu}} \tag{4.2}$$

The *total variation (TV)* metric is a particular instance of GTV when $\mu = 1$ and scaling factor 1/2.

Total variation metric (M_{TV})

The total variation metric among P and Q is over k various random values. The following equation illustrates the total variation metric [154].

$$M_{TV(\phi_1,\phi_2)} = \frac{1}{2}\left(\sum_{i=1}^{k} |(\phi_1)_i(A_x) - (\phi_2)_i(A_y)|\right) \tag{4.3}$$

LrDDoS attack Detection algorithm at N_{6BR}

In this mechanism, before utilizing M_{TV} extracts packet properties (i.e., protocol, packet arrival time, source and destination IP address) to obtain the variation metric between distinct samples. We estimate the packet frequency and probability distribution by applying the following equations:

$$I_i = \frac{f_i}{\sum_{i=1}^{K} f_i}, \quad \text{and} \quad P(x_i) = \frac{x_i}{\sum_{i=1}^{k} x_i} \tag{4.4}$$

Where f_i and x_i indicate the packet frequency and feature instances of the ith sample within time period (k, K). The average packet inter-arrival time (A_{PIAT}) is measured from every incoming packet arrival time. With the help of those statistical information, we improve the M_{GTV} and achieve broad spacing within two distributions (i.e., A_x, and A_y),

which is shown in Equations (4.5) and (4.6).

$$\chi = \frac{\phi_a \times A_{PIAT}}{(\psi - PI_i)} \tag{4.5}$$

$$VM(\phi_1, \phi_2) = \chi \left(\sum_{i=1}^{k} (|(\phi_1)_i (A_x) - (\phi_2)_i (A_y)|)^{\mu} \right)^{\min\left(1, \frac{1}{\mu}\right)} \tag{4.6}$$

Here ϕ_a indicates the average probability distribution for packet characteristic and ψ is the number of characteristics considered. The variation metric (VM) distinguishes LrDDoS attack traffic and legitimate traffic. As per VM value of LrDDoS, attack traffic is larger than legitimate IoT network traffic. The important steps of our LrDDoS attacks detection approach is shown in Algorithm 4.1. We take $\psi = 4$, for four level packet characteristic analysis during probability distributions.

Algorithm 4.1 LrDDoS Attack Detection at N_{6BR}

Require: All external traffic (EX_i) **from** (N_{EL})**,** (N_{ES})
Ensure: Detection of LrDDoS attack in IoT network.
Initialisation : External traffic probability $P(EX_i)$,
Sample time window for external traffic (ET_S), where $i = 1, 2, 3, ...n$, $(ET_S) = \{ET_{s1}, ET_{s2}, ET_{s3}, ..., ET_{sN}\}$, and N is the total occurrences within sampling time windows (ET_S); average packet inter-arrival time (A_{PIAT}); scaling SC,

1: External traffic (EX) received from (N_{EL}) and (N_{ES}) within sampling time windows (ET_S).
2: Estimate packet intensity PI_i, probability $P(x)$ utilizing Equation (4.2) and further calculate A_{PIAT} within sampling time windows (ET_S).
3: Estimate the individual sample on distribution ϕ_1 and ϕ_2

$$\phi_1 (A_x) = \sum_{m=0}^{l} M_{GTV_\mu} \phi_m (A_x) \tag{4.7}$$

$$\phi_2 (A_y) = \sum_{m=l}^{l+1} M_{GTV_\mu} \phi_k (A_y) \tag{4.8}$$

4: **if** $(A == 0)$ for sampling time windows (ET_S) **then**
5: measure total variation (TV): Using Equations (4.5) and (4.6).
6: **else if** $(A_r == A_v)$ **then**
7: Estimate vertical variation applying $VM_{\phi_1\phi_2}$ (Equations (4.5) and (4.6))
8: **else if** $(A_r == A_h)$ **then**
9: Estimate horizontal variation applying $VM_{\phi_1\phi_2}$ (Equations (4.5) and (4.6))
10: **end if**
11: Compare against variation measure threshold, $VM_{\phi_1\phi_2} \geq (\Delta_{traff.})$ then LrDDoS attack alarm arise and initiate the algorithm 4.2

In the second phase of LrDDoS attack detection, internal network traffics (IX) are examined at the PIA. The DDoS attack traffic is obtained using network sniffer [155]. The network traffic is collected upto sample time windows (T_S) and produced the Packet Flow Behavior Graph (G_{PFB}). It also measures the packet count associated with the various IP address flows.

Algorithm 4.2 LrDDoS attack detection using N_{PIA}

Input: Internal Traffic (IX) from (N_{IL}), (N_{IS}); threshold ($\delta_{traff.}$) and ($\Delta_{ipcount}$)
Output: Detection of LrDDoS attack in IoT network.
Initialisation : Packet flow behavior graph (G_{PFB_i}),
Sample time window (T_S), where $i = 1, 2, 3, ...n$, (T_S) = $\{T_{s1}, T_{s2}, T_{s3}, ..., T_{sN}\}$, and N is the total occurrences within sampling time windows (T_S).
1: Internal traffic (IX) received from (N_{IL}) and (N_{IS}) within sampling time windows (T_S).
2: Generate 3-hop packet flow behavior graph (G_{PFB_i}).
3: Keep track on frequency flow (FF_i) of IP address (IP_i)
4: **if** ($IX \geq \delta_{traff.}$ && $FF_i \geq \Delta_{ipcount}$)
for 2 successive sampling time windows (T_S) **then**
5: LrDDoS attack detected and initiate Algorithm 4.3
6: **else**
7: Normal internal traffic (IX).
8: **end if**

The internal network traffic (IX) analysis is based on the IP address with the corresponding IP flow. The IP address and frequency flow are denoted as IP and FF_i, respectively. FF_i measures differently for LrDDoS attack traffic compared to legitimate traffic. The threshold ($\Delta_{traff.}$) and ($\Delta_{ipcount}$) values of the IP address with the corresponding IP flow are estimated for each attack instance. The threshold values are determined by utilizing heuristics because it is based on LrDDoS attack scaling and deployed network scenario. It is observed that the ($\Delta_{traff.}$) and ($\Delta_{ipcount}$) values obtained at the separate sample time (T_S) intervals show minor variations. From experiments and observations, we can determine the threshold values that covere all variations of LrDDoS attack traffic. The functioning of the LrDDoS attack identification process is represented in Algorithm 4.2.

The last phase of the proposed approach is the mitigation phase. This phase takes the output from phase one and two (i.e., IP address, protocol) and acquaints this information to other PIA and N_{6BR} nodes. Using this information other IoT node blocks the IP address with the corresponding protocol. The detailed steps of the mitigation phase are given in

Algorithm 4.3.

Algorithm 4.3 Mitigation Algorithm for LrDDoS

Input: IP_i Address of the attack node, and attacking node; Protocol.
Output: Block traffic flow of attacking node with corresponding protocol
Initialisation : $6BR$ is a 6LoWPAN Border Router,
$INFO_{GET}$ indicates LrDDoS attack detection alarm generate from PIA or $6BR$.
1: Identify the attacking node IP address and protocol from algorithm 4.1 and algorithm 4.2
2: **if** ($LrDDoS = TRUE$ && $INFO_{GET} \leftarrow 6BR$) **then**
3: Generate traffic flow blocking rule
X_{Block}(Attacking node IP, Corresponding protocol, Block).
4: **else**
5: IX_{Block}(Attacking node IP, Corresponding protocol,
Block).
6: **end if**
7: Set this blocking rule to $6BR$ and other PIA.

4.4 Experimental setup and implementation

Figure 4.3. presents the experimental setup adopted for implementation, in which *Contiki OS*, *Cooja simulator*, *Wireshark*, and N_{PIA} are used. The experimental setup and design incorporate two circumstances: A) non-attack circumstance and B) LrDDoS attack circumstance. The comprehensive explanation of individual experiments are explained as follows.

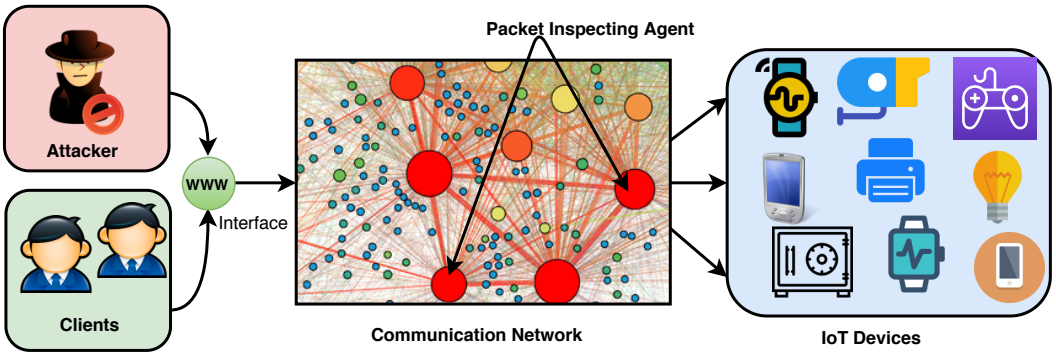


Figure 4.3: Experimental setup adopted for IoT implementation

4.4.1 Non-attack circumstance

In non-attack circumstances, the end user requests for IoT service (i.e., temperature, humidity) using the skywebsence web server. The experiment is carried out with 8, 16, 32, and 64 IoT nodes as shown in Figure 4.4. LrDDoS attack network traffic is filtered at N_{PIA} (i.e., N_{6BR} and IDS capability node). In N_{PIA}, we employ packet capturing tools like wireshark and keep the records of *Packet Flow Behavior Graph* (G_{PFB}). The average G_{PFB} at non-attack condition is shown in Table 4.1.

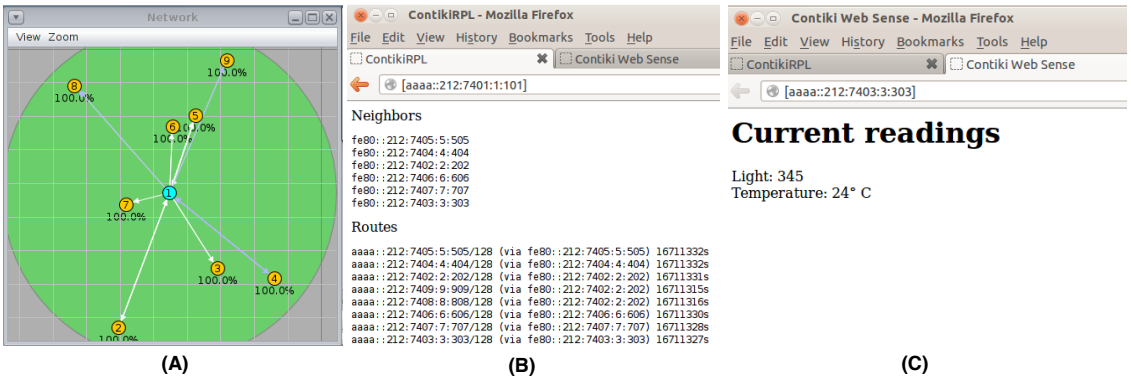


Figure 4.4: Snapshot of non-attack scenario using the skywebsence web server

Table 4.1: Packet Flow Behavior Graph (G_{PFB}).

Scenario	Tools/Simulator	G_{PFB} (30 sec) Distinct No. of Nodes			
		8 N	16 N	32 N	64 N
Non-Attack	Contiki Cooja	13320	15549	15553	15506
LrDDoS Attack	TorsHammer	262	276	290	294
	Longcat	292	308	320	326
	Contiki Cooja	277	292	305	310

4.4.2 LrDDoS attack circumstance

In the LrDDoS attack circumstance, legitimate and spiteful nodes are considered. This node demands IoT services through the skywebsence web-server. The spiteful nodes are originating LrDDoS attacks. In our experiments, we consider 2, 4, 6, and 8 nodes as spiteful nodes, as shown in Figure 4.5. This node descends the IoT services by forming recurring *Constrained Application Protocol (CoAP)* connections with the skywebsence web-server. The spiteful nodes also alter the timeout rate of the skywebsence web-server. As per our

experiments, we determined *3000 Milliseconds* as a time out value. The node requests for IoT service every *3000 Milliseconds*.

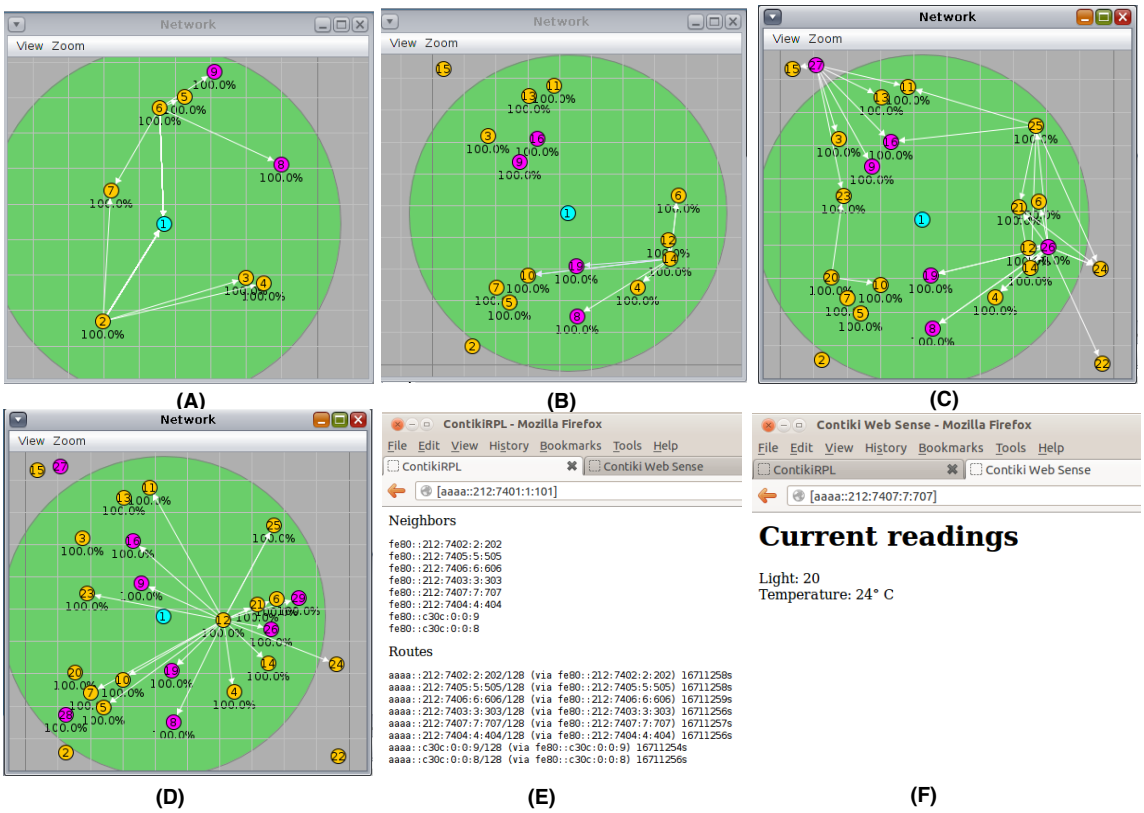


Figure 4.5: Snapshot of LrDDoS attack scenario using the skywebsence web server

4.5 Experimental Outcomes and Analysis

To examine the effectiveness of the intended security approach, we experiment and install our security approach in simulated as well as real-world circumstances. We consider CAIDA [156], MIT Laboratory [157] dataset for LrDDoS attack traffic and legitimate traffic is also used for the evaluation. In this section, we also exhibit and explain the results. We conclude by a comparative study of the proposed security method with the existing security solution presented in Table 4.2.

The security solution analysis incorporates three measures as follows: 1) probability distribution (PD), 2) variation metric (VM), and 3) frequency-rate variation of network traffic. The probability distribution measure of spiteful and legitimate traffics [158] [157] Illustrated in Figure 4.6. In Contiki cooja scenarios, traffic that arrives at N_{6BR} and N_{PIA}

Table 4.2: The comparative study of the intended security solution with the existing security methods.

Method Applied	**LW**	**AD**	**Response time (RT)**		**Average FNR (%)**		**Average FPR (%)**	
			CIADIA/ MIT	Generated Data	CIADIA/ MIT	Generated Data	CIADIA/ MIT	Generated Data
Du *et al.* (2015)	NA	NA	NA	Medium	NA	7.9	NA	9.25
H. Bhuyan *et al.* (2016) [111]	NA	NA	Medium	NA	NA	NA	4.89	NA
Wu *et al.* (2017)	NA	NA	NA	Fast	NA	18.64	NA	7.45
Chen *et al.* (2018)	Yes	NA	(26-118) Sec.	NA	4.84	NA	NA	NA
Proposed Method	Yes	Yes	(1-3.47) Sec.	(1-3) Sec.	5.15	3.82	5.41	5.12
LW=Light weight, AD = Adaptability, RT = Response time, NA= Not Available								

receive both spiteful and legitimate traffics. Hence, we have measured the PD of LrDDoS traffic and legitimate traffics.

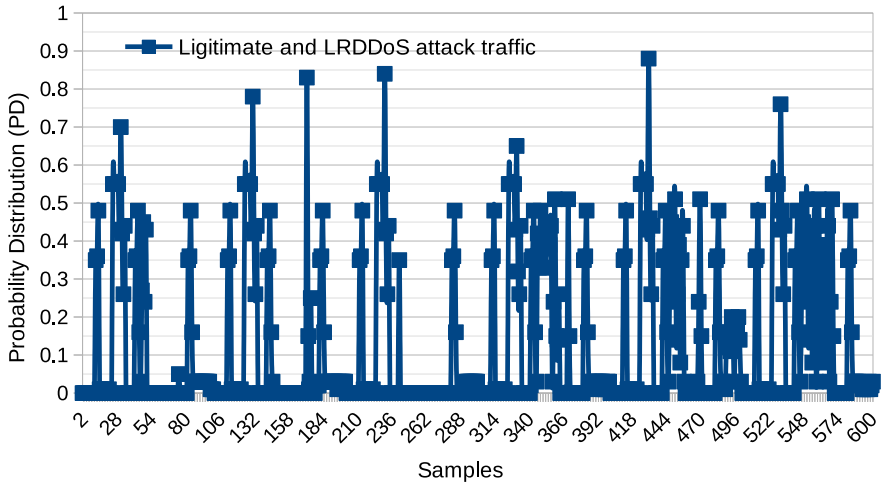


Figure 4.6: Probability distribution of LrDDoS attack and non-attack data

The variation metric is used to differentiate between LrDDoS attack traffic and legitimate traffic. In variation metric, we adopted μ values within 1 and 4 including for T_S. The VM estimate for CAIDA [158], MIT Lincoln Laboratory [157] dataset are exhibited in Figures 4.7, 4.8, and 4.9 respectively. They show the spacing and VM threshold ($\Delta_{traff.}$) among spiteful and legitimate traffic. The spacing value and $\Delta_{traff.}$ are recorded as 0.96 and 0.53, respectively. Figure 4.9 shows that VM can clearly differentiate the LrDDoS attack traffic from substantial traffic.

We also measure frequency-rate variation to distinguish LrDDoS traffic from legitimate traffic. In the attack scenario, assailant practice arbitrary delays the packet flow. The traffic flows are consistent during attack situation. However, packet flow consists of a constant delay

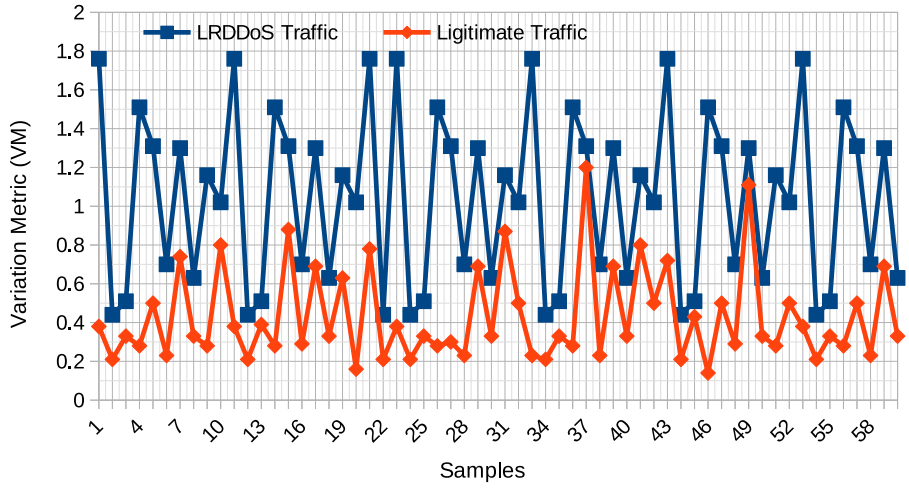


Figure 4.7: Variation Metric: a) Benchmark data-set (CAIDA and MIT data traffic)

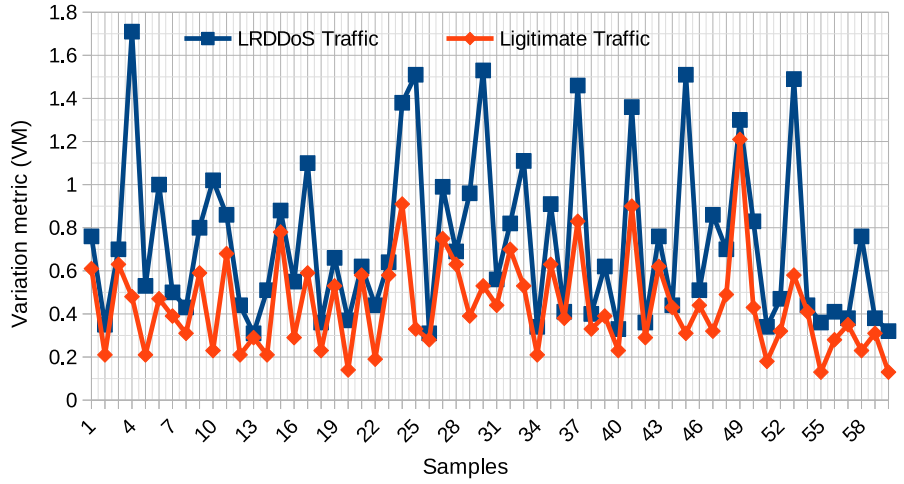


Figure 4.8: Generated data-set using Contiki cooja

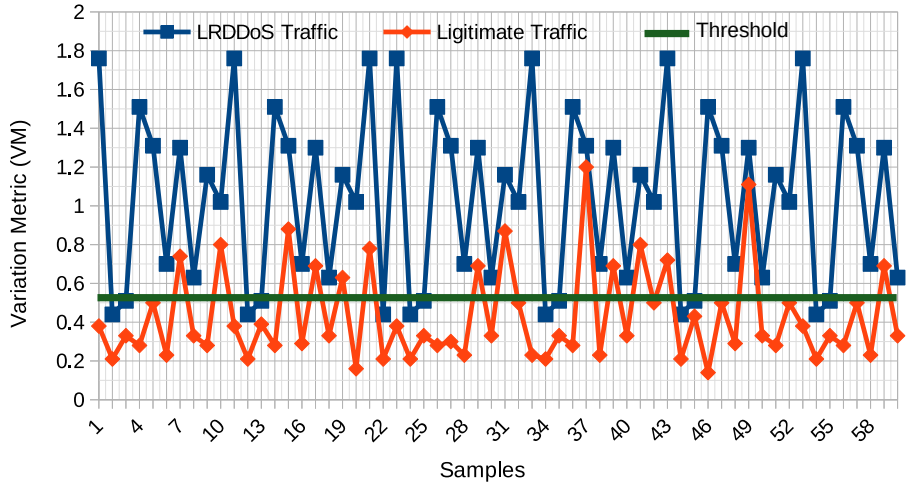


Figure 4.9: Threshold analysis

within two periods. To obtain LrDDoS attack packet flow with IP address and the frequency variation rate, we examined delays (0 *to* 0.36) *Seconds* towards a target. Like MIT, and CAIDA datasets, we estimate frequency-rate variation with the generality component $\mu = 1$ *to* 4 for the sampled traffic window T_S. The Frequency-rate variation threshold ($\delta_{traff.}$) and IP-Address count ($\Delta_{ipcount}$) are recorded as 0.68 and 310, respectively. Figures 4.10, 4.11, and 4.12 show the ($\delta_{traff.}$) calculated spacing and ($\delta_{traff.}$) in LrDDoS and authentic traffic. The threshold value 0.69 and spacing rate $S = 0.25$ are utilized to classify legitimate traffic and LrDDoS traffic.

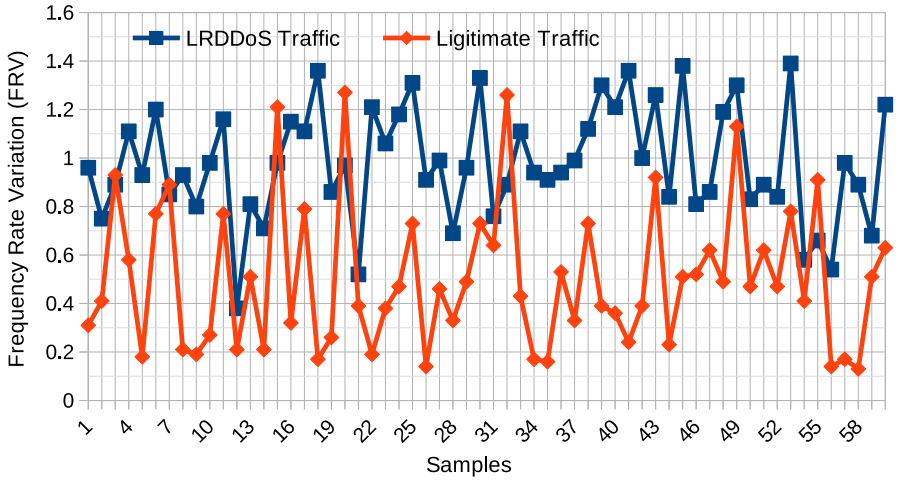


Figure 4.10: Frequency-rate variation: a) Benchmark data-set (CAIDA and MIT data traffic)

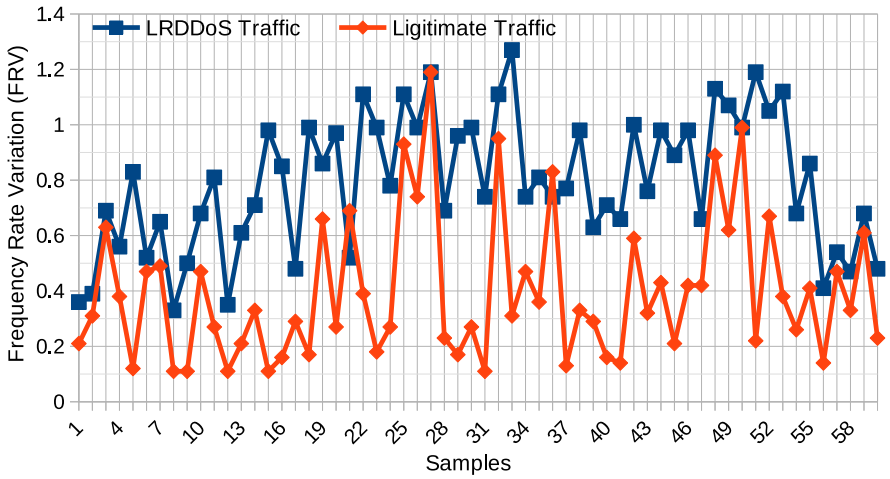


Figure 4.11: Frequency-rate variation: b) Generated data-set using Contiki cooja

The comparative study of the intended security solution with the existing security

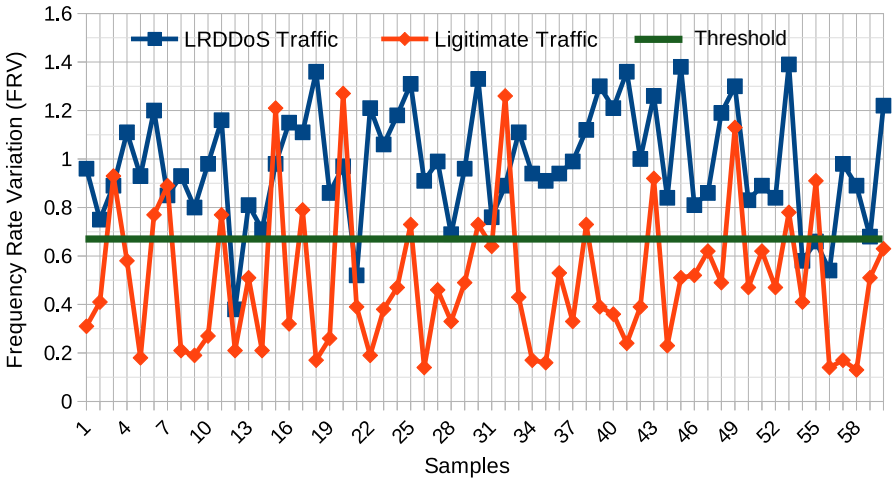


Figure 4.12: Frequency-rate variation: c)Threshold analysis

methods are presented in Table 4.2. An analysis is achieved concerning LrDDoS attack, *False Negative Rate (FNR)*, *False Positive Rate (FPR)*, and *Response Time (RT)*. The *FNR* and *FPR* are measured using the following equations (4.9) and (4.10).

$$FNR = \frac{FN}{FN + TP} \tag{4.9}$$

and

$$FPR = \frac{FP}{FP + TN} \tag{4.10}$$

where ,

FP = Legitimate activity wrongly classified.

TN = Legitimate activity recognized accurately.

FN = Spiteful activity wrongly classified.

TP = Spiteful activity recognized accurately

The performance study, the experiment analysis is repeated several times with various LrDDoS configuration. The comparative outline of the experimental outcomes are presented in Table 4.2. The small variation in the G_{PFB} is noted for a high number of nodes. Hence, the experimental results for up to 64 IoT nodes are presented in this paper. To verify the lightweight and adaptive feature (AF), we execute each experiment with expanding number (8 to 256) of nodes. During each trial, we calculate FNR and FPR values. The overall FNR and FPR obtained by the proposed defense method are 5.15%, and 5.41% respectively. It also identifies the LrDDoS attack traffic in (1-3.47) *seconds*. It is concluded that the

intended method identifies internal and external LrDDoS attack by improved FNR, FPR, AF, and enhanced RT.

4.6 Summary

In this chapter, we develop a defense method which observes diverse types of LrDDoS traffic of the IoT ecosystem and propose LrDDoS attack detection and mitigation method. The approach is based on the total variation metric, packet flow behavior graph, and IP address frequency. The performance of our approach has been assessed using various benchmark dataset (CAIDA and MIT DDoS dataset) as well as attack traffic by using verticle and horizontal scaling in Contiki cooja simulator environment. Our approach detects the LrDDoS attack effectively with minimum FNR and FPR. It additionally supports lightweight and adaptability property. The experimental outcomes exhibit that the average FNR and FPR on the benchmark and generated data set are 5.15%, 3.82%, 5.41%, and 5.12% respectively, which are comparable with the state of the art schemes. The next chapter of the thesis will provide a comprehensive analysis of MrDDoS attacks in the IoT environment, including both LrDDoS and HrDDoS attacks. In the course of our evaluation, we have conducted exhaustive evaluations of the accuracy of attack detection, network and system parameters.

"The marriage of machine learning and IoT security empowers devices to not only follow rules but to learn and adapt, staying ahead of cyber threats."

- Jeff Dean

CHAPTER

5

OPTIMIST: Lightweight and Transparent IDS with Optimum Placement Strategy to Mitigate Mixed-rate DDoS Attacks in IoT Networks

5.1 Introduction

In the last few years, various sectors like smart health monitoring systems, smart vehicles, smart home appliances, and smart cities have witnessed steady increases in the usages of IoT [159]. IoT devices are battery-operated, energy-constrained nodes with limited computation and storage capacities. IoT devices generate data by sensing the environment and send the generated data to a remote server through the Internet for further analysis/processing. IoT devices can be queried through the Internet for their generated data by external entities, and the obtained data is used for various critical/non-critical applications. Therefore, maintaining the authenticity, integrity, confidentiality, and availability of the generated data is very crucial, and the violation of any of them may incur serious consequences. As the IoT nodes are externally accessible through the Internet, any security vulnerability of the IoT devices can be exploited. IoT devices are manufactured by various vendors, which can purposefully insert some backdoor vulnerabilities to launch attacks. For example, an IoT device can be compromised and turned into a bot by an external malicious entity. That bot can be used to launch various kinds of attacks by generating malicious traffic flows. Among these malicious flows, DoS and DDoS attacks are very harmful to IoT systems as these attacks disrupt the availability of systems. Therefore, internal traffic flows also

need to be monitored/analyzed by IDS systems [160]. DDoS attacks can be classified into high-rate DDoS (HrDDoS), and low rate DDoS (LrDDoS) attacks. HrDDoS aims to disrupt the IoT system completely, whereas LrDDoS aims to partially degrade the IoT system performance, making LrDDoS detection more challenging compared to HrDDoS. Many existing solutions are there to detect HrDDoS, whereas very few solutions are proposed for LrDDoS. To the best of our efforts, we could not find out existing work which is designed to detect mixed-rate DDoS (MrDDoS) attacks (HrDDoS and LrDDoS). Motivated by this fact, this paper proposes the IDS solution OPTIMIST, which can detect and mitigate MrDDoS attacks. The OPTIMIST IDS module is based on a LSTM model, which is trained using publicly available as well as in-house generated datasets. However, the distribution of the flows of these datasets exhibits some network-specific bias which is used to generate the datasets. As a result, though the trained model shows high accuracy when tested with the flows of the same dataset, the model's performance fails to meet the expectation when run in network scenarios whose flow distributions are different from the training datasets. Motivated by the above facts, a novel training method is proposed for OPTIMIST where WGAN-generated artificial flows from the datasets are mixed with the original flows to reduce the biases of the datasets.

One crucial design challenge for any IoT solution is the IDS placement/deployment problem, i.e., where to run an IDS solution. Few IDS solutions are centralized in nature and run on the border router, through which all external Internet traffic flows are passed. Few existing works have proposed to run IDS on all IoT devices, which is redundant and reduces network lifetime. An alternative hybrid solution is to run IDS in border router along with few selected nodes, where each node is responsible for monitoring a small subset of the network. In this case, the design problem is to select an optimum number of nodes that can balance network coverage and energy. Few cluster-based solutions have been proposed where the network is divided into clusters, and cluster heads run IDS. The monitored nodes are queried by cluster heads for various network statistics, and the collected statistics are either analyzed locally in cluster heads or reported to sink. This query-response scheme imposes network overhead resulting in energy depletion of the nodes. Moreover, the presence of IDS nodes are exposed in the network and the malicious flows generated by compromised IoT nodes can easily avoid the IDS nodes. In transparent IDS solutions,

IDS nodes transparently sniff/eavesdrop on flows surrounding them without making their presence visible to other nodes. Though there are very few existing works on transparent IDS solutions, their placement strategies are non-optimal. Motivated by the above discussion, this paper proposes an optimal IDS placement strategy for a transparent IDS solution that can balance network coverage and energy overhead.

The above discussions are the motivations of the proposed work, namely, *A lightweight and transparent IDS with **o**ptimum **p**lacement **s**trategy to **mi**tigate mixed-rate DDo**S** attack in Io**T** system* (OPTIMIST). The contributions of our work are summarized as given below:

- Unlike existing works which focus either on high-rate or low-rate DDoS, this work provides a solution for mixed-rate DDoS attack detection, which can detect and mitigate both high and low-rate DDoS attacks.

- A novel training method is proposed to build the IDS solution. WGAN is used to generate artificial flows from public datasets as well as in-house generated dataset to reduce the distribution bias of the datasets. The WGAN-generated flows are mixed with the public and in-house generated training datasets and used for LSTM model training

- A novel hybrid IDS placement algorithm is proposed, which runs transparently without incurring any network overhead. The IDS node selection is optimized, which balances energy overhead and IDS coverage. The problem is formulated as the weighted minimum vertex cover problem of a K-uniform hypergraph, and an approximation solution is provided.

- Extensive experiments on Contiki and FIT IoT-LAB testbed are done for competitive performance analysis of the proposed scheme. The results show that our proposed scheme is most effective in detecting the attacks while consuming minimum energy compared with existing benchmark protocols.

The rest of the chapter is organized as follows. Section 5.2 provides some basic background knowledge about IoT, and attacks on IoT. Existing literature surveys on IDS placement and IDS solutions are presented in Section 5.3. In Section 5.4, IDS placement problem formulation and the proposed solution are described in details. Section 5.5 provides

a detailed description on the proposed IDS solution. Performance evaluation of the proposed work is given in Section 5.6 with detailed experimentation setup and competitive result analysis. Finally, in Section 5.7 we summarize the chapter.

5.2 Background

This section discusses a few important aspects of IoT and IoT security. IoT as low power lossy network (LLN) is introduced in Section 5.2.1. The Section 6.2.3 discusses about DDoS attacks on IoT networks.

5.2.1 IoT as low power lossy network

IoT are networks of resource-constrained nodes which sense various environment parameters and report generated data to sink(s). These nodes are connected to the Internet through a border router (BR/6BR). The underlying network is of low power lossy network (LLN) type comprising interconnected nodes of constrained energy (powered by battery), memory, and computing capacity. Nodes are interconnected by lossy wireless links of short communication ranges and low data rates. The *IPv6 Routing Protocol for LLNs* (RPL), is standardized for LLN as a proactive distance-vector routing protocol. The routes are formed from 6BR to each node as *destination oriented directed acyclic graph* (DODAG). Depending on the resource constrain, a node may store (storing mode) or may not store (non-storing mode) routing information locally. In non-storing mode, only 6BR has the entire DODAG topology information, and all messages are forwarded to 6BR to get routed to destinations. As a result, 6BR needs to do source routing to forward a message toward a destination.

5.2.2 Attacks on IoT

IoT networks are prone to various attacks due to the end-node accessibility through the Internet, the lossy nature of networks, and resource constraints of the nodes. A few well-known attacks are rank attacks, black-hole attacks, sink-hole attacks, version number attacks, buffer reservation attacks, bot attacks, DoS attacks, and DDoS attacks. The nature of these attacks varies vastly with the varying objectives of the attacks, like interrupting network traffic, exhausting network resources, disrupting the topology, etc. This paper is focused on

DDoS attacks, which can be broadly classified into two categories, as given below.

High Rate DDoS attack (HrDDoS)

In the HrDDoS, the assailants flood the network with malicious flows to interrupt the availability of IoT services. The interruption is created by exhausting resources like channel bandwidth, router buffer, CPU, etc. Attacks can be of transport or network layer flooding [161, 162] such as *user datagram protoco*l (UDP) flooding, *transmission control protocol* (TCP) SYN flooding, *Internet control message protocol* (ICMP) flooding, etc.

Low-rate DDoS Attacks (LrDDoS)

A LrDDoS attack is difficult to identify because of its low-rate and intermittent traffic behavior, which is quite similar to the legitimate traffic [3]. These attacks intend to increase latency and decrease the network's throughput to some extent for genuine users rather than disrupting the IoT services entirely. A LrDDoS attack model can be described by three parameters which are the off-time phase, on-time phase, and time interval. In an off-time phase, no attack packet is sent. During an on-time phase, the assailant sends malignant messages. The time interval Δ phase maintains time among two successive attack packet generations. Figure 5.1 shows the LrDDoS attack model with the three parameters.

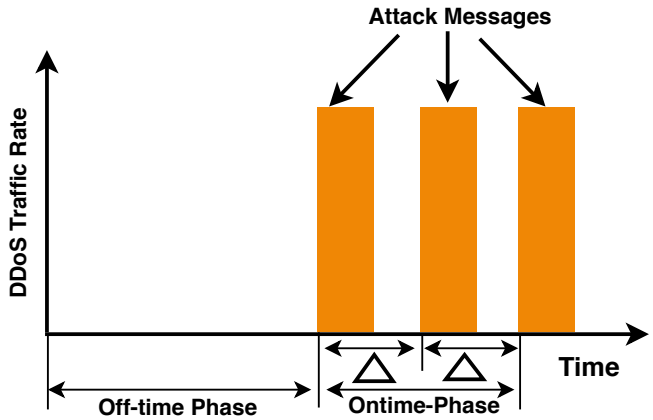


Figure 5.1: LrDDoS attack model adapted from [3]

Mixed-rate DDoS Attacks (MrDDoS)

MrDDoS is the type of attacks which includes both the HrDDoS and LrDDoS type of attacks.

5.3 Related Work

This section is organized into two parts. Section 5.3.1 reviews existing works on IDS placement while in Section 5.3.2, various existing IDS solutions are discussed.

5.3.1 IDS placement

Thakkar et al. [163], and Bruno et al. [160] have surveyed various IDS placement strategies and have categorized them into groups of centralized, distributed, and hybrid placement strategies. In a centralized placement strategy, an IDS instance is run in one dedicated high-resource node like a border router (6BR). The works [164, 165] are examples of centralized placement strategy. Though centralized placement can monitor all external traffic, some malicious internal flows generated by compromised LLN devices may remain undetected. In distributed placement strategy, lightweight (like rule or signature-based) IDS instances are run in all the LLN nodes. The work [113] is an example of distributed placement strategy. Though this strategy enables host-based IDS (HIDS) [166, 167], running IDS all the time on all nodes is redundant, draining energy from low-resourced nodes rapidly. The hybrid placement strategy combines both the benefits of centralized and distributed strategies. In this strategy, a centralized entity monitors external traffic, while a few of the LLN nodes are selected as IDS nodes to perform the role of watchdogs by monitoring the behavior of a subset of the nodes. The works [112, 168] are examples of hybrid placement strategies. As a subset of LLN nodes run IDS, network-based IDS (NIDS) [166] is applicable.

The task of monitoring can be performed in transparent or non-transparent mode. In non-transparent mode [164, 165, 112], the IDS nodes gather network status by querying or probing monitored nodes making malicious nodes aware of the presence of IDS nodes in a network. Additionally, these extra messages increase network congestion and energy overhead. In transparent mode [113, 167, 168], IDS nodes can sniff/eavesdrop packets to gather network information without adding any network overhead while keeping their

presence in the network transparent to malicious nodes. In a transparent IDS placement scheme, there is a trade-off between IDS coverage and energy overhead. On the one hand, if energy overhead is reduced by selecting a small number of IDS nodes, the IDS system is unable to eavesdrop on all traffic flows, resulting in poor IDS coverage. On the other hand, if IDS coverage is improved by increasing IDS running nodes, it decreases network lifetime. Consequently, optimization techniques are needed to balance IDS coverage and energy overhead.

No existing work was found which provides an optimum IDS placement solution for transparent monitoring mode. Accordingly, in this work, we propose a novel IDS placement algorithm for transparent monitoring, which is able to provide an optimum balance between IDS coverage and energy overhead based on system requirements.

5.3.2 IDS solutions

A number of IDS techniques are available in the literature which can be broadly classified into two categories as signature-based and anomaly-based [160] [33].

Signature-based IDS [160]

In this strategy, IDS is trained to learn behavior patterns or signatures of previously known attack flows. The trained model is then used to classify observed behaviors of network flows. However, signature-based IDS cannot detect unknown (zero-day) attacks or modified/evolved known attacks since their signatures are unknown to the IDS.

Few examples of signature-based IDS solutions for HrDDOS are given next. Li et al. [169] proposed a collaborative blockchain-enabled IDS framework for the IoT ecosystem. This approach incrementally builds and updates the signature database in the IoT network. It is also verifiable without the requirement of a trusted third party. Yadav et al. [170] proposed an automated machine learning (ML) model for IoT-enabled smart energy grids. Results are presented using an IoT dataset, showing the potential of the proposed approach in smart energy infrastructures.

Following are few examples of signature-based IDS solutions for LrDDoS. Perez-Diaz et al. [171] proposed a modular architecture that can detect and mitigate LrDDoS attacks in SDN-enabled networks. The IDS module is trained with six distinct types of ML models.

Even though it is hard to find LrDDoS attacks, the study shows that the suggested approach has a 95% detection rate. Liu et al. [172] proposed a LrDDoS attack detection method for wireless networks. In this method, the authors built a multidimensional sketch structure based on network traffic characteristics. This approach also preserves the baseline stability of network traffics and correctly differentiates LrDDoS attack traffics from normal network traffics.

Anomaly-based IDS [33]

In this strategy, an IDS solution first profiles the expected behavior of a given system and then tries to detect any deviant behaviors from the learned behavior profile. Though anomaly-based IDS can detect zero-day attacks, it usually suffers from a high false-positive rate as it is difficult to learn all possible normal behaviors of a given system in a finite time.

Few works have proposed anomaly-based IDS solutions for HrDDoS attacks. For example, Tabassum et al. [173] proposed a privacy-preserving IDS based on distributed incremental learning. To reduce the computation costs, the work has used a pre-processing method to eradicate redundant features. The work used non-negativity constraint-based autoencoders supporting distributed IDS. This approach minimizes and allocates the loads among IoT devices. Hussain et al. [174] proposed a two-fold approach to detect DDoS attacks. First, the premature attack activities are scanned, and then the ML model is trained for DDoS attack detection in the IoT ecosystem. The model is trained with distinct datasets to identify HrDDoS assaults exclusively. Abdelmoumin et al. [175] proposed a distributed IDS module that incorporates principle component analysis and 1-SVM AML-IDS. They enhanced 1-SVM AML and PCA models using ensemble learning and hyper-parameter tuning to identify HrDDoS attacks. The authors trained and tested these improved models on malicious and benign IoT network flows. Saharkhizan et al. [176] proposed an IDS comprising multiple LSTM models for attack detection on IoT systems. The model reportedly reached 99.91% accuracy. Li et al. [177] proposed a solution using LSTM and Bayes (LSTM-BA) models. DDoS attack detections by LSTM model can be of high or low confidence. Low confidence data is further analyzed using the Bayes model to further enhance the accuracy. However, the proposed LSTM-BA model is not suitable for resource-constrained devices.

Few works used anomaly-based IDS for LrDDoS attack mitigation. For example, Garcia. et al. [178] proposed an AI-based method for detecting LrDDoS attacks. This method continuously observes network traffic and organizes packets into conversation flows. The LrDDoS security model integrates deep learning and clustering analysis to enhance LrDDoS attack detection accuracy. Liu et al. [179] designed a LrDDoS attack detection technique. This technique is a combination of the self-adjusting SVM algorithm and APSO optimization. The self-adjusting SVM technique improves the generalization capabilities. Similarly, the APSO algorithm was employed to enhance the attack's adaptability. The outcomes demonstrated outstanding detection performance and accuracy, varying between 92.36% and 96.65%.

None of the above-mentioned works has trained their models with GAN/WGAN generated artificial traffic flows to reduce the bias of the trained models. In this work, we have generated artificial flows from the training datasets with WGAN and trained the LSTM model to make the OPTIMIST IDS more robust compared to existing solutions. None of the existing work has proposed an IDS solution that can detect both HrDDoS and LrDDoS types of attacks. Accordingly, the proposed OPTIMIST IDS is trained with both HrDDoS and LrDDoS to detect and mitigate both types of attacks.

5.4 Proposed OPTIMIST IDS placement

This section is divided into two parts. The problem formulation for IoT IDS placement is described in Section 5.4.1 and the solution for IoT IDS placement is proposed in Section 5.4.2.

5.4.1 IDS placement problem formulation

OPTIMIST proposes a transparent IDS placement strategy as it reduces network and energy overheads compared to non-transparent placement strategies. If the presence and locations of IDS nodes are unknown to the other IoT nodes, the IDS can be termed as transparent. In transparent IDS system, the IDS running nodes monitor and process surrounding traffic flows only by eavesdropping, and do not query the monitored nodes about their traffic/system status. To quantify transparent IDS coverage, we define the term K-hop IDS coverage scheme, which guarantees to monitor any flow of length K-hops or

more. However, in a K-hop IDS coverage scheme, few of the flows of length less than K may (not necessarily) remain unmonitored. There can be two solutions for an IDS system to eavesdrop or transparently monitor (without querying monitored nodes) packets in an IoT network.

1. IDS nodes can monitor any ongoing flow in promiscuous mode within its one-hop neighborhood.

2. IDS nodes can eavesdrop a K-hop flow if it is an intermediate node in the flow path.

To reduce the total network energy consumption for a distributed IDS solution, number of IDS running nodes need to be minimized without compromising the K-hop coverage property of the IDS system where K is predetermined by the network administrator. The value of K is a trade-off between energy and security. Larger values of K require less number of IDS running nodes which saves overall network energy but come with a cost of few unmonitored flows of lengths upto $K-1$. However, periodical selection of different sets of IDS nodes minimizes the unmonitored flow problem. To achieve K-hop coverage property, an IDS node needs to placed in every possible paths of degree K. An optimum number of IDS nodes can be selected by finding the K path vertex cover of the IoT network topology graph. In case of promiscuous mode eavesdropping, in addition with the packets for which an IDS node is an intermediate/destination node, the IDS node also needs to capture and process all the on-going transmissions within its communication range. The above reason cause fast energy depletion of overall energy of an IoT network [112, 180]. To reduce energy consumption, the second IDS solution is preferable where an IDS node processes only the packets for which it is an intermediate/destination node. The flows of an IoT network follow the routes established by RPL DODAG. If non-storing mode is used for the DODAG RPL routing, a node can eavesdrop all the packets which are generated from or destined for a node within its DODAG sub-tree. However, in non-storing mode, packets of most flows take non-optimal paths via root to reach destinations, which incurs delay and consumes network energy unnecessarily. If the storing mode of RPL routing is used, though the flow paths are optimal, an IDS node can eavesdrop a flow only if it is an intermediate node in the path between source and destination. An example with 3-hop IDS placement for DODAG-based IDS placement scheme is illustrated in Figure 5.2. For

non-storing mode, the flow from source S to destination D is passed through the IDS node, whereas in storing mode, the same flow gets unmonitored. To guarantee that all flows of length greater than or equal to K-hop are monitored by an optimum number of IDS nodes, K path vertex cover solution on the DODAG can be used. As a fresh IDS selection happens each time DODAG is re-created, the problem of undetected flows of K-hop IDS coverage is minimized. However, while selecting IDS nodes, the residual energy of the nodes also needs to be considered. Accordingly, the problem is formulated into a weighted minimum vertex cover problem of a K-uniform hypergraph as given next.

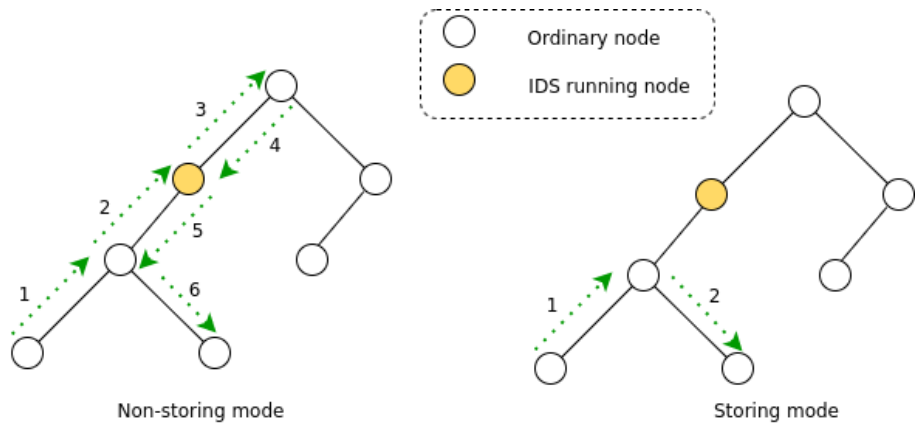


Figure 5.2: An example with 3-hop IDS placement for DODAG based scheme with non-storing and storing mode

An undirected hypergraph $\mathbb{H}$ consists of a vertex set $\mathbb{V}$, and a collection of non-empty subsets of $\mathbb{V}$, namely hyperedges, forming the set $\mathbb{E}$. In hypergraph, a hyperedge can connect an arbitrary number of vertices. A hypergraph is called K-uniform, if $|\mathbb{E}_m| = K, \ \forall \mathbb{E}_m \in \mathbb{E}$. Consider the DODAG as an undirected tree. For each node/vertex of the undirected tree, create a hyperedge for each of the K-hop length paths from that node. Consequently, each created hyperedge is a set of K vertices. The set of all such hyperedges $\mathbb{E}$, along with the set of vertices $\mathbb{V}$ form the K-uniform hypergraph $\mathbb{H}$. The algorithm for creating K-uniform hypergraph from an undirected tree graph is given in Algorithm 5.1. If an IoT node i has remaining energy $\mathcal{E}_i$, a weight $1/\mathcal{E}_i$ is assigned to that corresponding vertex $\mathbb{V}_i$ of the hypergraph $\mathbb{H}$. Let the function $\omega(\mathbb{V}_i)$ gives the assigned weight of $\mathbb{V}_i$. The minimum weight vertex cover solution VC of $\mathbb{H}$ contains the minimum number of vertices of $\mathbb{H}$ such that $\mathbb{E}_m \cap S \neq \varnothing, \ \forall \mathbb{E}_m \in \mathbb{E}$. In other words, VC contains at least one vertex from each of the K-hop paths from all vertices of the undirected tree, and the total remaining

energy of the selected nodes are maximized. Finding vertex cover of a graph is known to be NP-Hard. However, an approximation solution of a K-uniform hypergraph can be found with K approximation ratio by solving the problem as a binary (integer) program, which is described next.

Algorithm 5.1 K-uniform hypergraph creation from an undirected tree

Input: Undirected tree graph $G = \{V, E\}$, K
Output: K-uniform hypergraph $\mathbb{H} = \{\mathbb{V}, \mathbb{E}\}$

```
1: 𝕍 ← V, 𝔼 ← ∅
2: for all V_i ∈ V do
3:     declare QUEUE Q
       {//Create initial 1 hop paths from V_i}
4:     for all V_j adjacent to V_i sysmodel do
5:         declare STACK S
6:         S.PUSH(V_i)
7:         S.PUSH(V_j)
8:         Q.ENQUEUE(S)
9:     end for
       {//Extend the initial 1 hop paths from V_i}
10:    while !Q.EMPTY() do
11:        STACK TS = Q.DEQUEUE()
           {//If a path is of K-hop, insert into the hyperedge set}
12:        if TS.SIZE == K then
13:            declare SET hE
14:            while !TS.EMPTY() do
15:                V_t = TS.POP()
16:                hE.INSERT(V_t)
17:            end while
18:            𝔼.INSERT(hE)
19:            continue
20:        end if
           {//Keep growing the paths}
           {//Get the last added node of a path}
21:        V_l = TS.POP()
           {//Get the predecessor node of the last added node}
22:        V_l2 = TS.TOP()
23:        for all V_m adjacent to V_l do
24:            if V_m != V_l2 then
25:                declare STACK SC
26:                SC ← TS
27:                SC.PUSH(V_l)
28:                SC.PUSH(V_m)
29:                Q.ENQUEUE(SC)
30:            end if
31:        end for
32:    end while
33: end for
```

Let $x(\mathbb{V}_i) \in \{0, 1\}$ denote the decision variable for vertex whether to include it in the solution set VC of the vertex cover problem of the hypergraph $\mathbb{H}$ defined above. Then the objective of the minimum weight vertex cover solution is given below.

Formulation 5.1:

Minimize $\sum_{\mathbb{V}_i \in \mathbb{V}} (\omega(\mathbb{V}_i) \cdot x(\mathbb{V}_i))$
Subject to the constraints:

1. $\sum_{\mathbb{V}_j \in \mathbb{E}_m} x(\mathbb{V}_j) \geq 1, \quad \forall \mathbb{E}_m \in \mathbb{E}$
2. $x(\mathbb{V}_i) \in \{0, 1\} \quad \forall \mathbb{V}_i \in \mathbb{V}$

As integer program is a NP-Hard problem, we apply linear programming relaxation to remove the integer constraints of the decision variables to allow them to be real number in the range [0,1]. The resulting formulation of the linear programming is given below.

Formulation 5.2:

Minimize $\sum_{\mathbb{V}_i \in \mathbb{V}} (\omega(\mathbb{V}_i) \cdot x(\mathbb{V}_i))$

Subject to the constraints:

1. $\sum_{\mathbb{V}_j \in \mathbb{E}_m} x(\mathbb{V}_j) \geq 1, \quad \forall \mathbb{E}_m \in \mathbb{E}$
2. $x(\mathbb{V}_i) \leq 1 \quad \forall \mathbb{V}_i \in \mathbb{V}$
3. $x(\mathbb{V}_i) \geq 0 \quad \forall \mathbb{V}_i \in \mathbb{V}$

5.4.2 IDS placement solution

The problem of K-hop minimum IDS node selection with maximum residual energy is solved in two stages as described below.

Solution:

1. Create a K-uniform hypergraph from the undirected version of given DODAG using Algorithm 5.1.

2. Formulate the problem of K-hop minimum IDS node selection with maximum residual energy as shown in **Formulation 5.2**.

3. Solve the linear programming problem of **Formulation 5.2**.

4. For each $\mathbb{V}_i \in \mathbb{V}$, if $x(\mathbb{V}_i) \geq 1/K$, include $\mathbb{V}_i$ in the solution set VC.

It can be shown that the solution set VC is a K approximation solution for the minimum weighted vertex cover of K-uniform hypergraph $\mathbb{H}$. The proof for 2-uniform graph is given in [181] which can be easily extended for K-uniform hypergraph. First it needs to be shown that the given solution set VC is a vertex cover of K-uniform hypergraph $\mathbb{H}$. The constraint 1 of the **Formulation 5.2** ensures that the step 3 in **Solution** assigns values of $x(\mathbb{V}_i)$ such that for each hyperedge $\mathbb{E}_m \in \mathbb{E}$, $\sum_{\mathbb{V}_j \in \mathbb{E}_m} x(\mathbb{V}_j) \geq 1$. In a K-uniform hypergraph, all the hyperedges contains exactly K vertices. The above two facts imply that for each hyperedge $\mathbb{E}_m$, $\exists \mathbb{V}_j \in \mathbb{E}_m$, such that $x(\mathbb{V}_j) \geq 1/K$. As the step 2 of the given solution includes all the $\mathbb{V}_i$ with $x(\mathbb{V}_i) \geq 1/K$ in the solution set VC, definitely VC contains atleast one element from each of the hyperedges. Thus, VC is a vertex cover for hypergraph $\mathbb{H}$.

Next, it needs to be shown that the solution set VC is a K approximation of the optimal solution S^* for minimum weighted vertex cover of K-uniform hypergraph $\mathbb{H}$. Let Z be the optimal value of the linear program of **Formulation 5.2** obtained by step 3 in **Solution**. As S^* is a feasible solution of the linear program, clearly, $Z \leq \omega(S^*)$ which gives a lower bound of $\omega(S^*)$, where $\omega(S^*)$ is the total weight of the optimum solution S^*. We have,

$$
\begin{aligned}
& Z = \sum_{\mathbb{V}_i \in \mathbb{V}} (\omega(\mathbb{V}_i) \cdot x(\mathbb{V}_i)) \\
\implies & Z \geq \sum_{\mathbb{V}_i \in \mathbb{V}: x(\mathbb{V}_i) \geq \frac{1}{K}} (\omega(\mathbb{V}_i) \cdot x(\mathbb{V}_i)) \\
\implies & Z \geq \sum_{\mathbb{V}_i \in \mathbb{V}: x(\mathbb{V}_i) \geq \frac{1}{K}} (\omega(\mathbb{V}_i) \cdot \tfrac{1}{K}) \\
\implies & Z \geq \sum_{\mathbb{V}_i \in S} (\omega(\mathbb{V}_i) \cdot \tfrac{1}{K}) \\
\implies & Z \geq \tfrac{1}{K} \sum_{\mathbb{V}_i \in S} (\omega(\mathbb{V}_i)) \\
\implies & Z \geq \tfrac{1}{K} \omega(S) \\
\implies & \omega(S) \leq K \cdot Z \text{ [rearranging the terms]} \\
\implies & \omega(S) \leq K \cdot \omega(S^*) \text{ [because } Z \leq \omega(S^*)\text{]}
\end{aligned}
$$

As seen in the above inequality, the approximation solution is at most K times the

optimal solution. Hence, the provided solution steps provides a K approximation solution for the minimum weighted vertex cover of K-uniform hypergraph $\mathbb{H}$.

The time complexities for each solution steps of the proposed solution are given next. The first step of **Solution** uses Algorithm 5.1. The outer loop of line 2 runs in $O(|V|)$ time. The number of iterations for the loop from line 10 to 32 is same as the total number of ENQUEUE operations on the queue Q declared in line 3. From a starting vertex, one ENQUEUE operation is done on Q for each new vertex explored along a path from that starting vertex. As the input graph is an undirected tree represented in adjacency list format, from a starting vertex at most $|V|$ vertices can be explored. Therefore, for a starting vertex, the combined number of iterations for the loops of lines 4 and 23 can be at-most $|V|$. Therefore, total ENQUEUE operations on Q for a starting vertex is at most $O(|V|)$ and consequently, the loop of line 10 runs $O(|V|)$ iterations. The nested loop of line 14 runs in $O(1)$ as there are exactly K vertices in a fully explored path. The copy operation on line 26 takes $O(1)$ time as the maximum size of TS is K. All other operations used on all statements of the algorithm take $O(1)$ time. Therefore, the total running time of Algorithm 5.1 is $O(|V|^2)$. The second step of the **solution** formulates the linear programming as **Formulation 5.2**. To create the objective function, weight assignments for vertices are done in $O(|V|)$ time. From the analysis of Algorithm 5.1 it is clear that the total number of K-uniform hyperedges created by Algorithm 5.1 from an undirected graph is bounded by $O(|V|^2)$. Therefore, it takes $O(|V|^2)$ time to create the constrain 1 of **Formulation 5.2**. The creation of constraints 2 and 3 takes $O(|V|)$ time. As a result, the second step of the **solution** takes $O(|V|^2)$ time. In the third step of the **solution**, the simplex method is used to solve the linear programming problem of **Formulation 5.2**, which takes polynomial time. Finally, the step four of **Solution** takes $O(|V|)$ time. The above-mentioned solution steps are run centrally in 6BR, which is not resource-constrained. The details of the proposed IDS module are described in the next section.

5.5 Proposed OPTIMIST IDS solution

This section describes the proposed IDS solution OPTIMIST for detecting and mitigating mixed-rate DDoS attacks in IoT networks. Section 5.5.1 gives the description of the proposed model. Pre-processing steps for the proposed model are given in Section 5.5.2. The training

process of the IDS model is described in Section 5.5.3. The overall steps for the detection and mitigation of MrDDoS attacks of the OPTIMIST is given in Section 5.5.4. Section 5.5.5 discusses about the computational complexity of the proposed model.

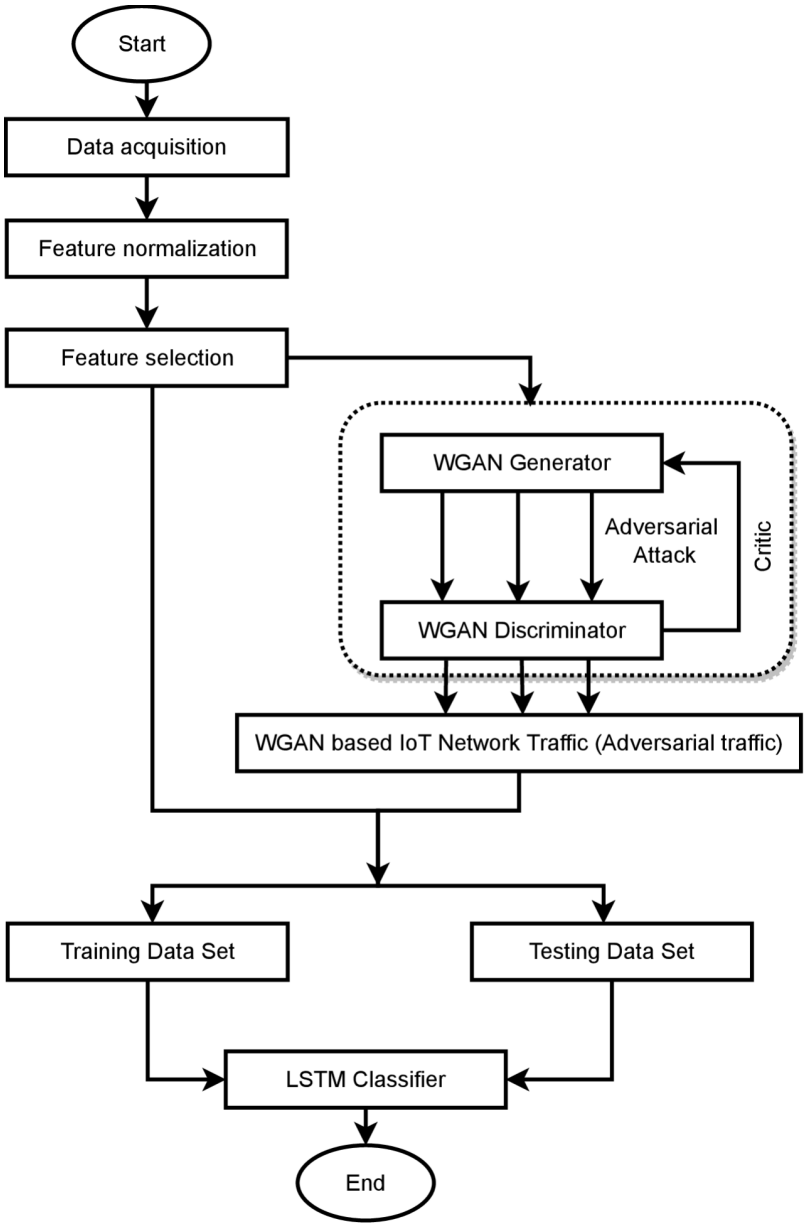


Figure 5.3: Flowchart of IDS creation process

5.5.1 Model description

A recurrent neural network (RNN) has feedback connections to learn the temporal dependencies of the features. Long short-term memory (LSTM) [182] is a special RNN to overcome the exploding/vanishing gradient issues of RNN. LSTM models are extensively used for speech/text recognition, cyber-security, etc. As IoT attack flows have temporal relations among themselves, LSTM models provide high accuracy for intrusion detection in IoT systems. Being lightweight, trained LSTM models are suitable for running in resource-constrained IoT devices. Accordingly, this work uses offline LSTM model training, and the trained model is deployed in a few of the selected IoT end devices for online attack detection. This work assumes that the IoT nodes of the system have the required amount of computational resources and storage to perform online detection with the trained LSTM model without hampering its primary task of environment sensing. However, the offline training using publicly available datasets induces some bias in the trained model towards the distribution of the data points of the used datasets. As a result, the trained model performs poorly when test inputs belong to some different distributions. To remove this training bias, we propose a novel training method. We first train a WGAN model with the datasets to generate artificial new data points and mix the new data points with the original data points before training the LSTM model. The overview of the proposed scheme is illustrated in Figure 5.3. Brief descriptions of the WGAN and LSTM models are given below.

Wasserstein GAN (WGAN)

The concept of the Generative Adversarial Networks (GAN) model is put forward by Goodfellow [183]. GAN consists of two neural networks known as the generator (ζ), and the discriminator/ critic (ϑ). The generator generates new training data samples from the distribution of the training data set, adding some random Gaussian noise. The discriminator module classifies the data as real (from actual domain) or fake (generated by generator), and the feedback is fed to the generator as in Figure 5.4

$$\min_{\zeta} \max_{\vartheta} E_{R\ \sim P(R)}\left[\vartheta\left(R\right)\right] - E_{F\ \sim P(F)}\left[\vartheta\left(\zeta\left(F\right)\right)\right] \tag{5.1}$$

However, as mentioned by A. Aggarwal at el. [184], a traditional GAN model may

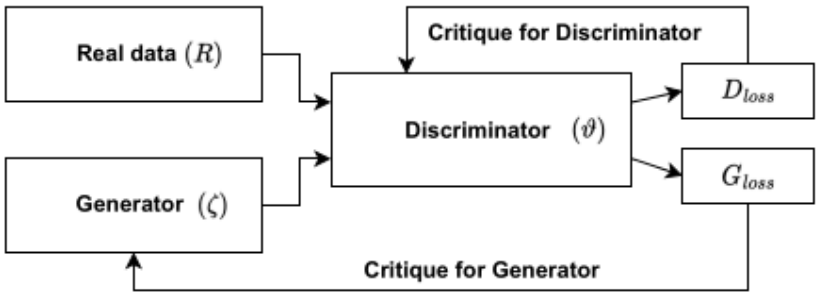


Figure 5.4: WGAN Network

be unable to produce output because of vanishing gradient, mode collapse problems, etc. Many authors [185] [186] have proposed enhancements over GAN. Wasserstein GAN [185] model is one of such improvements. Wasserstein distance is calculated between fake and real data distributions using Equation (5.1). It is also known as Critic loss. The Wasserstein distance loss function is of two types like Discriminator loss (D_{loss}), and Generator loss (G_{loss}) functions. These loss functions are mathematically represented below [185].

$$D_{loss} = \min_{\vartheta} E_{R\ \sim P(R)} \left[log\ \vartheta\left(R\right) \right] \tag{5.2}$$

$$G_{loss} = \min_{\zeta} E_{R\ \sim P(F)} \left[log\left(1 - \vartheta\left(\zeta\left(F\right)\right)\right) \right] \tag{5.3}$$

The loss function score depends on real and adversarial data. Based on score, WGAN generates high-quality adversarial data. WGAN discriminator (ϑ) provides a critic score. This score decides the difference between real and fake data. A critic score < 0 indicates real traffic data, and a score > 0 indicates that the given traffic is fake/adversarial.

LSTM model

Long short-term memory (LSTM) is a special kind of recurrent neural network (RNN) that can overcome the exploding/vanishing gradient issues. A LSTM cell includes three types of gates (i.e., forget gate, input gate, and output gate). The internal estimation procedure of LSTM cell is shown in the Equations (5.4) - (5.6).

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \qquad f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{5.4}$$

Table 5.1: Dataset Information

Dataset Name	ToN_ IoT(2020) [81]	IoT-23 (2020) [77]	Kitsune (2019) [80]	BoT-IoT (2018) [187]	Generated Data (GD)
Simulation/Testbed	Simulation	Simulation	Simulation	Simulation	Simulation and Testbed
Num of attack types	9	15	9	6	2
Data format	Raw, Log & sensor	Raw & Log	Raw	Sensor	Pcap file
Num of features	46	22	23	10	10
Dataset size	64GB	23 GB	20GB	69.3GB	17295 packet flow

Table 5.2: Feature selection using SHAP from five datasets

Datasets	Selected features
TON_IoT	src_pkts, src_ip_bytes, dst_pkts, dst_ip_bytes, ts, src_ip, src_port, dst_ip, dst_port, service, duration, src_bytes, dst_bytes, conn_state
IoT-23	ts, id_orig.h, id_orig.p, id_resp.h, id_resp.p, service, duration, orig_bytes, resp_bytes, conn_state, local_orig, local_resp, missed_bytes, history, orig_pkts, orig_ip_bytes.
Kitsune	Src_mac-ip_bw_obt, src_ip_bw_obt, channel_bw_obt, sock_bw_obt, cl_ibt_obt, socket_ibt_obt, src_mac_pr_obt, src_ip_pr_obt, cl_pr_obt, sock_pr_obt
BoT-IoT	Time, Bytes, src_mac, src_Ip, des_mac, des_Ip, src_Port, dst_Port, conn_state
GD	src_ip, dst_ip, src_pkts, service, duration, src_bytes, dst_bytes, id_resp.p, id_resp.h, conn_state

$$O_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \qquad \tilde{C} = tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{5.5}$$

$$C_t = f_t * \left(C_{t-1} + i_t * \tilde{C}_t\right) \qquad h_t = o_t * tanh\left(C_t\right) \tag{5.6}$$

Where f, i, O represent forget gate, input gate, and output gate, respectively. W and b represent weighted matrices and biases respectively. The new state and candidate state are C and $\tilde{C}$. Input, output, and input time are denoted as x, h, t. The sigmoid function is denoted as $\sigma(\cdot)$.

5.5.2 Model pre-processing

The pre-processing steps for the proposed OPTIMIST IDS model comprise three phases which are data acquisition, feature normalization, and feature selection. All of the phases are described subsequently.

Data acquisition

This work has used publicly available mixed-rate DDoS (LrDDoS and HrDDoS) attack datasets [81], [77], [80], [187]. The descriptions of the data sets are given in TABLE 5.1. It is observed that the procured datasets contain very few Low-rate DDoS data samples. Accordingly, a number of flows with low rate DDoS attack and non-attack are generated in-house using Contiki cooja and FIT IoT-LAB [62]. The flow packets are captured using the Wireshark tool [188]. The attack and non-attack experiments are described in Section 6.5.1. An additional dataset is created by the data points generated by extracting features (refer Section 5.5.2 and 5.5.2) from the captured flows. Further, additional data points for each dataset are generated by the WGAN model to make the LSTM model training more robust. WGAN model training description is given in Section 5.5.3.

Feature normalization

As the data sets are acquired from various sources, they have irregular central tendencies. Therefore, we normalize all data attributes using the min-max normalization method given below.

$$X_{norm} = \frac{X_{real} - X_{min}}{X_{max} - X_{min}} \tag{5.7}$$

Where X_{real} is the real value, X_{norm} is the normalized value, and the X_{min} and X_{max} are the smallest and highest values from real values, respectively.

Feature selection (FS)

The features having significant contributions to the mixed-rate DDoS attacks are chosen, while the redundant and insignificant features are discarded to reduce computation. This work used the SHAP (SHapley Additive exPlanations) method [189] for feature selection from mixed-rate DDoS attack datasets. The advantages of this method are as follows:

1. *Local Interpretability (LI)*: Each feature gets a SHAP score. This score indicates the impact of the feature across the complete dataset.

2. *Global Interpretability (GI)*: It exhibits how much a particular feature contributes towards the target (attacks).

TABLE 5.2 shows the extracted features by SHAP method form five datasets.

5.5.3 Model training

The model training of OPTIMIST has two phases. In the first phase, a WGAN model is trained on the datasets (refer to Section 5.5.2) to generate artificial data points. Table 5.4 shows the WGAN network setup, and Table 5.3 shows the training parameters. The Gradient Penalty helps with training stability. Leaky ReLU increases the training process's resilience and prevents a vanishing gradient.

Table 5.3: WGAN parameters

HP Name	Value
B_size	64
Critic_iters	3
Learning_rate	0.002
Optimizer	RMSprop
Lambda	10
HL_AF	LeakyReLU
HP: Hyper-parameter; B: Batch; HL : Hidden layer; AF: Activation Function	

Table 5.4: WGAN Configuration

Layer (Type)	CONFIG
IP_Noise ()	(N, 20)
IP_N_MrDDoS ()	(N, 41)
Concat_Input ()	(N, 61)
Dense	(N, 32)
Leaky_ReLu (0.2)	(N, 32)
Dense	(N, 8)
Leaky_ReLu (0.2)	(N, 8)
Dense	(N, 2)
Leaky_ReLu (0.2)	(N, 2)
IP_Noise ()	(N, 20)
IP_N_MrDDoS ()	(N, 41)
B: Batch; N: None; CONFIG: Configuration	

In the second phase of the training, the LSTM model is trained with the datasets (real and artificial) to classify mixed-rate DDoS attacks. To mitigate the over-fitting issue, dropout and batch normalization strategies are used. These strategies change the network design in each training epoch to reduce the chance of overfitting and increase the training speed. LSTM model comprises an input layer, three hidden layers, and an output layer. The detailed structure and hyper-parameters of the proposed LSTM model are shown in Table 5.6 and 5.5, respectively. The LSTM input layer contains 16 neurons. There are 3 LSTM layers composed of 32 memory blocks. The hidden layers of LSTM have the ReLu activation function. The output layer uses a sigmoid activation function.

Table 5.5: LSTM parameters

HP Name	Value
A_F_Input	ReLu
A_F_Output	Sigmoid
Epoch	100
Learning Rate	0.002
Window size	5
Optimizer	RMSprop
Dropout prob.	0.2
Train data	64% dataset
Validation data	16% dataset
Test data	20% dataset
HP: Hyper-parameter; A_F: Activation Function	

Table 5.6: LSTM Configuration

Layer (Type)	Configuration
LSTM_1	(N, N, 32)
B_norm_1	(B (N, N, 32))
Dropout_1	(N, N, 32)
LSTM_2	(N, N, 32)
B_norm_2	(B (N, N,32))
Dropout_2	(N, N, 32)
LSTM_3	(N, N, 32)
B_norm_3	(B (N, N, 32))
Dropout_3	(N, 32)
Dense_1	(N, 1)
Activation_1	(N,1)
B: Batch; N: None	

5.5.4 OPTIMIST IDS solution

The topological ordering of the OPTIMIST IDS solution for detection and mitigation is given in Figure 5.5. The heavy task of IDS model training for OPTIMIST is done offline with the novel method described in Section 5.5.3. The trained LSTM model is deployed in all IoT nodes. However, the IDS modules in all IoT nodes are in an idle state initially. Once a DODAG is created for the IoT network, the OPTIMIST IDS placement algorithm is executed in the 6BR to select the IoT nodes to act as IDS nodes. 6BR unicasts a message to each of the selected nodes to activate their respective OPTIMIST IDS modules. If a new instance of DODAG is created, the 6BR instructs the current IDS nodes to put their IDS module in an idle state and instructs the newly selected IDS nodes to activate their IDS modules. For the duration when the IDS module is active, an IDS node eavesdrops flows, extracts features, and classifies them with the trained LSTM model. If a DDoS (high-rate or low-rate) attack is detected, the IDS node reports the attack information to the 6BR node. The 6BR node broadcasts the malicious node information to all other nodes of the network and instructs them to block all the traffic flows originating from the malicious sources.

5.5.5 Time complexity of LSTM

The LSTM algorithm is local in space and time. Hence, activation values aren't stored. They only store and update derivatives based on Mozer's recurrent back-propagation method [93]. As a result, the LSTM method is extremely efficient. The time complexity of the LSTM is $O((N_{ou} \times N_{hu}) + (N_{ou} \times N_{mcb} \times S_{mcb}) + (N_{hu} \times N_{fc}) + (N_{mcb} \times S_{mcb} \times N_{fc}) = O(W)$, where N_{fc} is the number of units connected in a forward direction to hidden units, gate units, and memory cells. N_{hu} = number of hidden units, S_{mcb} = memory cell block size, N_{mcb} = number of memory cell blocks, and N_{ou} = number of output units. As a result, the storage complexity of the LSTM model is also $O(W)$ and independent of the length of the input sequence.

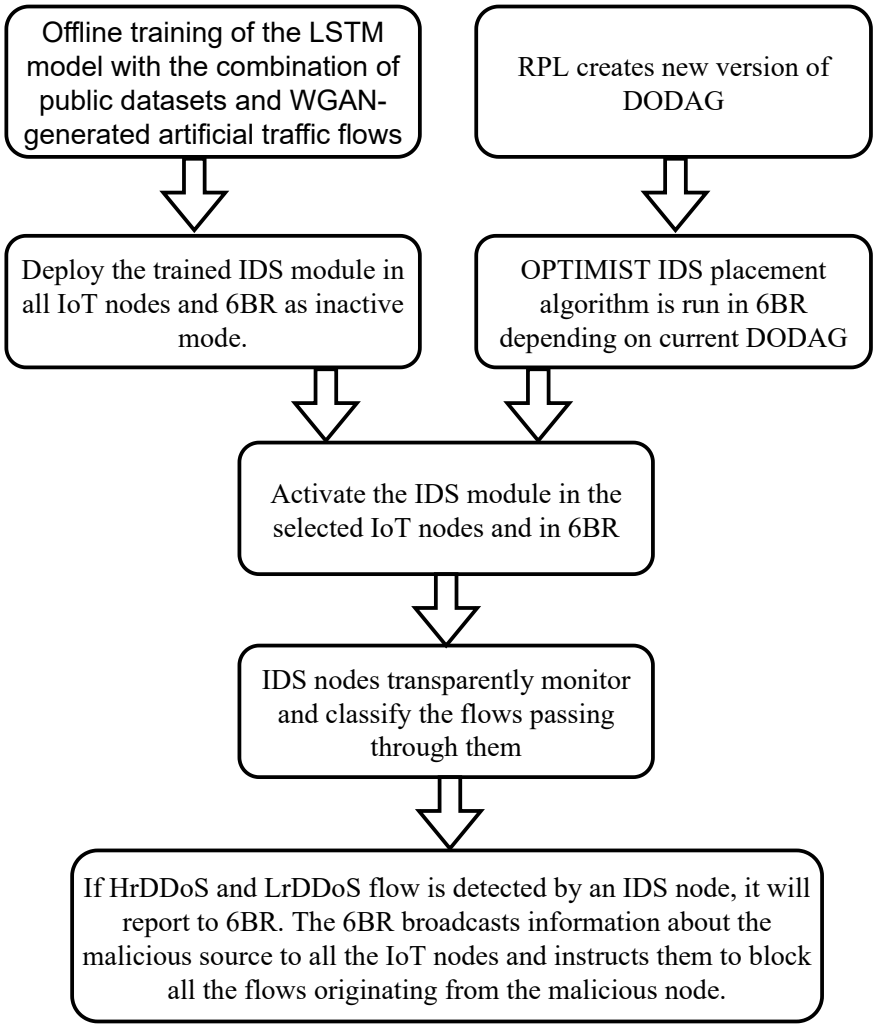


Figure 5.5: Topological order of OPTIMIST IDS

5.6 Performance evaluation

This section is divided into three subsections. Section 6.5.1 describes the experiment environments and setups. Section 6.5.2 defines the metrics to evaluate the performances of OPTIMIST. In Section 6.5.3, the performance of OPTIMIST is evaluated, and the competitive result analysis is done.

5.6.1 Experiment environments and setups

An IoT scenario is considered for performance evaluation of the OPTIMIST, as shown in Figure 6.10. LLN nodes are randomly deployed for sensing purposes that have multi-hop path connectivity among themselves. LLN nodes are connected to the Internet through a 6BR node of ample storage and computation power. The internal LLN nodes of the IoT system are accessible by external nodes through the 6BR node using the Internet. As shown in the figure, both internal, as well as external nodes can be malicious in nature. The scenario is created and run in Contiki cooja [46] simulation, and FIT IoT-LAB [62] test-bed environment. The experimental parameters of Contiki cooja and FIT IoT-LAB are presented in Table 6.4.

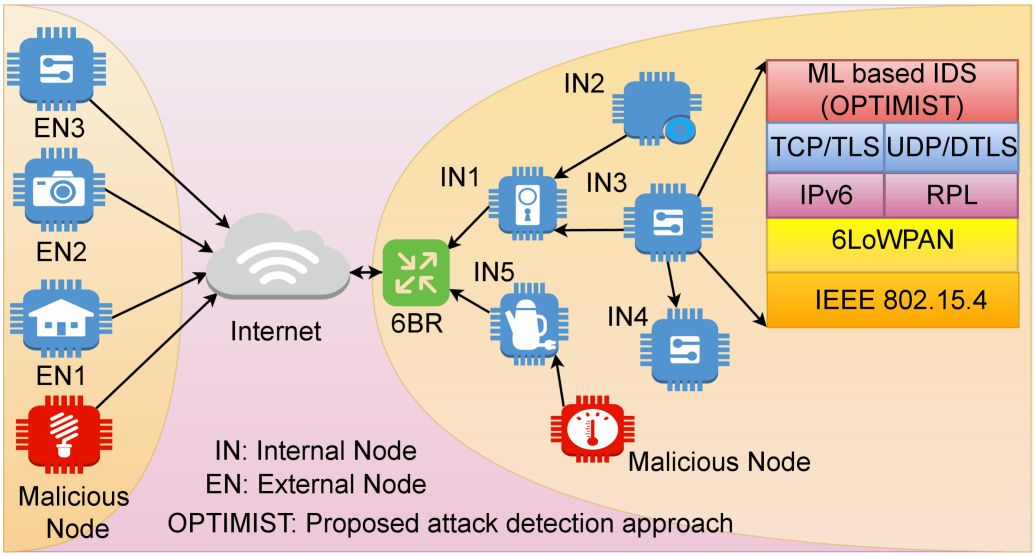


Figure 5.6: IoT network setup for experimentation

Simulation and test-bed are used to generate a in-house dataset which includes normal and LrDDoS attack flows using 8, 16, 32, and 64 IoT nodes. In attack scenarios, 25% of the

Table 5.7: Simulation and real-time test-bed parameters

Parameter name	**Simulation**	**Real time testbed**
Operating system	Contiki 3.0, Contiki 4.5	Contiki-NG
Simulator/Testbed	Cooja Cooja	FIT IoT-LAB
Network size	8,16, 32, 64 nodes	
Radio Environment	UDGM	
Node Type	Tmot Sky	IoT-Lab A8
Routing Protocol	RPL	RPL Lite
RPL Objective Function	MRHOF - ETX, OF0	MRHOF - ETX
MAC/adaptation layer	Contiki MAC/6LoWPAN	
Transmitter output power	(dBm) 0 to -25	
Receiver sensitivity	(dBm) -94	
Radio frequency	2.4GHz	
Attack Modeled	Mixed Rate DDoS attack	
Experiment Duration	60 minutes	

nodes are deployed as malicious nodes.

Figure 5.7: Snapshot of 4, 6 malicious nodes during mixed rate DDoS attack

During the performance evaluation of the OPTIMIST, the malicious nodes are used to launch MrDDoS attacks. Based on the proposed IDS placement algorithm, a few of the IoT nodes are selected to activate the OPTIMIST IDS solution to detect mixed-rate DDoS

attacks. The IDS running nodes are changed over time with each newly created DODAG versions by RPL.

5.6.2 Performance metrics

The following metrics are defined to evaluate the performance of OPTIMIST IDS solution.

- *Accuracy (ACC)*: It denotes the percentage of correctly classified flows as true attack or true legitimate flows with respect to total number of flows. Accuracy is given by:

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \tag{5.8}$$

  where, *FP=Legitimate flow wrongly classified; FN=Attack wrongly classified; TP=Attack recognized accurately; TN=Legitimate flow recognized accurately.*

- *Precision (PREC)*: It is the percentage of correctly predicted MrDDoS attack flows out of all predicted MrDDoS attack flows. It is calculated as below.

$$PREC = \frac{TP}{TP + FP} \times 100 \tag{5.9}$$

- *Recall (REC)*: It indicates the percentage of predicted MrDDoS attack flows by the classifier out of all real MrDDoS attack flows of the system. Recall, also known as sensitivity, and is estimated by

$$REC = \frac{TP}{TP + FN} \times 100 \tag{5.10}$$

- *F1-score*: It indicates the overall efficiency of the proposed OPTIMIST combining *PREC* and *REC*. It is the harmonic mean of the *PREC* and *REC* as given below.

$$F1 - score = 2 \times \frac{PREC \times REC}{PREC + REC} \times 100 \tag{5.11}$$

- *Memory Consumption (MEMC)*: It shows the percentage of memory utilization of the IoT devices to run OPTIMIST throughout the experimentation.

- *CPU energy (ENEC)*: The metric measures the energy consumed by CPU in IoT devices throughout the experimentation.

- *Throughput (THP)*: This metric measures the ratio of the network throughput in the presence of MrDDoS attacks with respect to the observed network throughput in the absence of MrDDoS attacks. The ratio is shown as percentage.

$$THP = \frac{throughput\ in\ MrDDoS\ sneraio}{throughput\ in\ normal\ sneraio} \times 100 \tag{5.12}$$

5.6.3 Result Analysis

For the evaluation of the OPTIMIST IDS solution, experiments are run on simulation as well as a testbed (refer to Section 6.5.1). The results are presented in two parts as follows. Section 6.5.3 shows the performances of the offline training process of the IDS model, and Section 5.6.3 presents the competitive analysis of online OPTIMIST performances on placement strategy and attack detection with existing protocols.

Offline training performance evaluation

First, we assessed the performances of different ML models like Support Vector Machine (SVM), Gated Recurrent Unit (GRU), Convolution Neural Network (CNN), and Transformer, on the IoT-23 [77] and in-house generated datasets. Figure 5.8 shows the comparative performances of different ML models with respect to the metrics ACC, PREC, REC, and F1-score. As LSTM model outperforms other ML models, it is chosen for the OPTIMIST IDS.

Consequently, the LSTM model is trained and tested with other publicly available datasets [81], [77], [80], [187] and the test results for each datasets are depicted in Figure 5.9. As shown in Figure 5.9, the ACC and PREC scores of the trained model are (94.12%-95.49%) and (92.56%-94.93%) respectively. For REC and F1 scores, the scores are (89.64%-93.49%) and (89.7%-93.34%), respectively. However, when the model is tested with artificial data generated from the WGAN model (described in Section 5.5.1), the LSTM performance degraded drastically, as shown in Figure 5.10. This is because the trained model is biased with the distributions of the training datasets. However, the distribution of WGAN-generated artificial data points (attack flows) is a little different from the distribution of the training

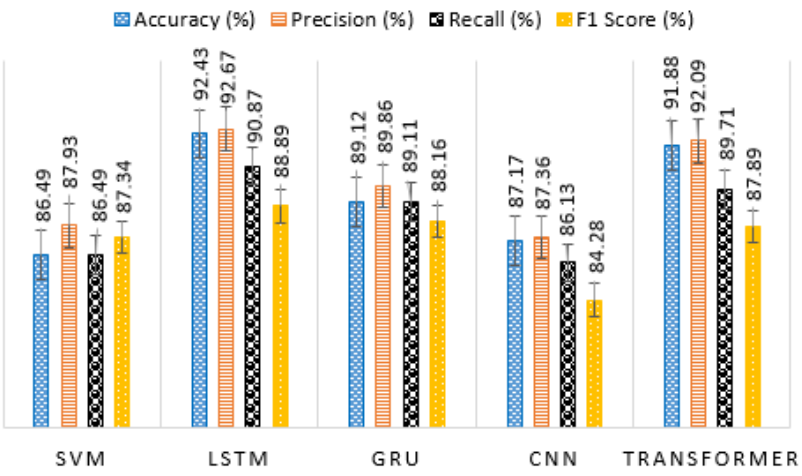


Figure 5.8: Training performances of ML models on IoT-23 and generated dataset

datasets. Figure 5.11 shows the comparative accuracy trends when tested with and without adversarial traffic samples.

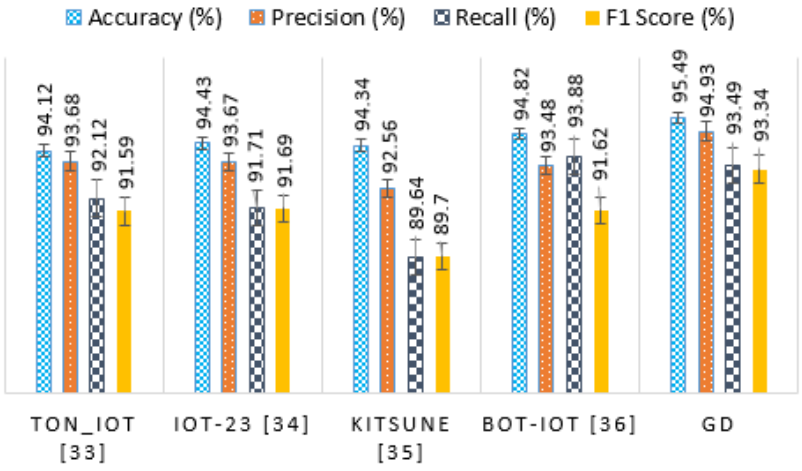


Figure 5.9: Performance of LSTM model over public datasets

To increase the robustness of the OPTIMIST IDS for any distribution of attack flows, WGAN-generated artificial traffic flows of MrDDoS attacks are mixed with the existing training samples, and then the model is again trained and tested. It took less than 100 epochs for the model to converge. Figure 5.12 shows the trend of loss and accuracy with increasing epochs. The model performance is quite satisfactory with ACC = 98.40%, PREC = 95.40%, REC = 96.49%, and F-score = 96.30% as depicted in Figure 5.13.

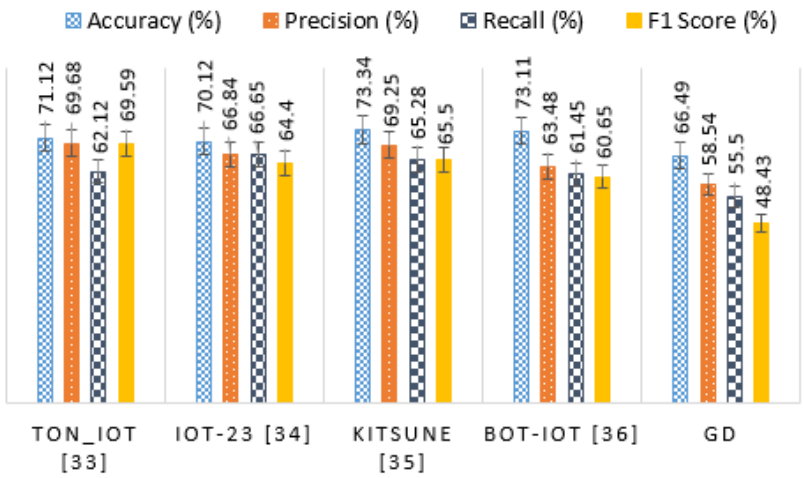


Figure 5.10: Effect of adversarial data on LSTM

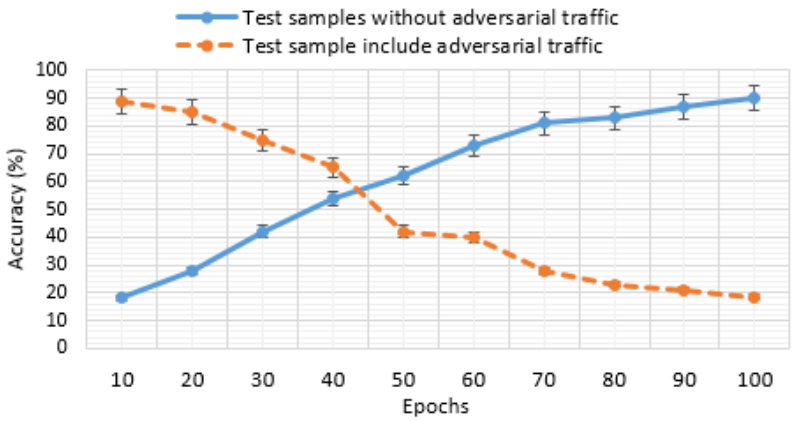


Figure 5.11: Model performance evolution with Epochs

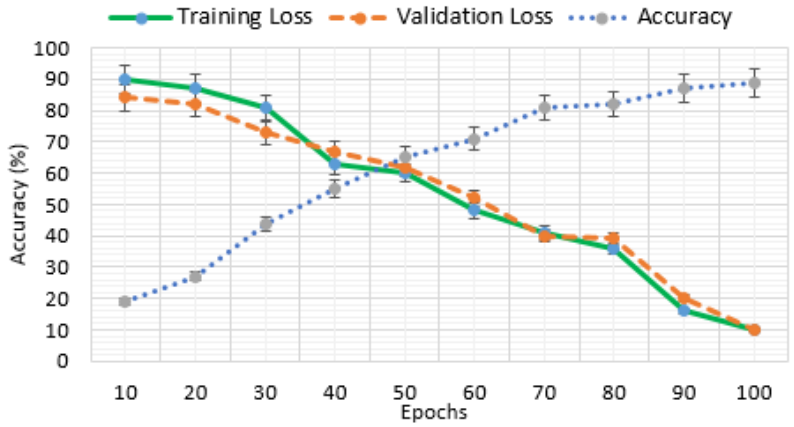


Figure 5.12: Training testing loss and Accuracy

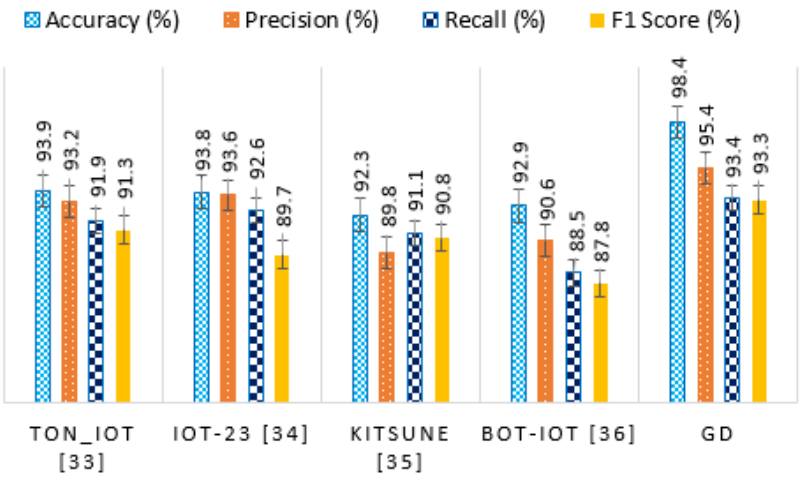


Figure 5.13: WGAN-LSTM based result

Table 5.8: Comparison of the proposed strategy with the closely related works

Ref	IDS placement	Target Attacks	Mitigation	ENEC (mJ)	MEMU (Byte)	THP(Kbps)	ACC	PREC	REC	F1 Score
[190]	Centralised	DoS	No	9482	N/A	N/A	98.20%	N/A	N/A	N/A
[177]	Centralised	HrDDoS	No	6745	55184	N/A	98.03%	98.21%	97.55%	97.87%
[171]	Distributed	LrDDoS	Yes	10869	63940	N/A	95.01%	95.46%	94.51%	94.98%
[191]	Distributed	HrDDoS	Yes	5128	N/A	52961	96.30%	93.24%	92.40%	96.20%
[176]	Distributed	HrDDoS	No	12814	68315	56531	98.99%	96.51%	N/A	95.67%
[179]	Centralised	LrDDoS	No	N/A	N/A	N/A	94.19%	95.85%	95.33%	95.56%
[174]	Centralised	HrDDoS	No	16439	N/A	N/A	98.89%	99.01%	98.74%	98.87%
[178]	Distributed	LrDDoS	No	N/A	N/A	N/A	97.00%	96.00%	96.65%	96.98%
[175]	Distributed	HrDDoS	No	N/A	N/A	N/A	96.70%	100%	77.80%	87.50%
OPTIMIST	**Distributed**	**MrDDoS**	**Yes**	**5407**	**45296**	**53824**	**98.40%**	**95.40%**	**96.49%**	**96.30%**

ENEC: ENErgy Consumption, MEMU: MEMory Usage, ACC: ACCuracy, PREC:Precision, REC: RECall, THP: Throughput, N/A: Not Applicable

Online performance evaluation

In this section, the online performances of OPTIMIST regarding the placement strategy and attack detection are evaluated. The experiments are conducted with 8, 16, 32, and 64 nodes with a malicious node ratio of 25%. System and network level metrics are measured in two scenarios when no IDS module is running on nodes and when optimally selected nodes are running the IDS module. For both cases, the duration of these experiments is 5400 seconds. The `size` command and `powertrac` tool are used to extract memory usage and energy consumption information respectively from the Contiki cooja simulation experiments. Similarly, for the FIT IoT-LAB testbed experiments, the `Sysstat` [192] tool is used to get the system level information like energy consumption and memory usage while `iperf` [193] tool is used to get throughput information of the experiments. Figure 5.14 shows the comparative performances of system and network level metrics for the two cases of experiments. The results show that THP metric performance is improved at the cost of

increased energy and memory overheads in the scenarios with IDS running nodes. This is because OPTIMIST IDS detects and mitigates malicious flows, which helps normal flows increase network bandwidth utilization.

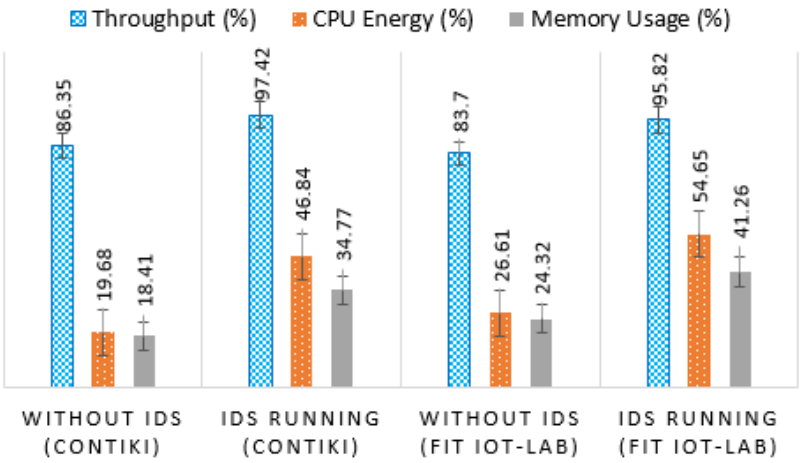


Figure 5.14: Contiki and FIT IoT-LAB result

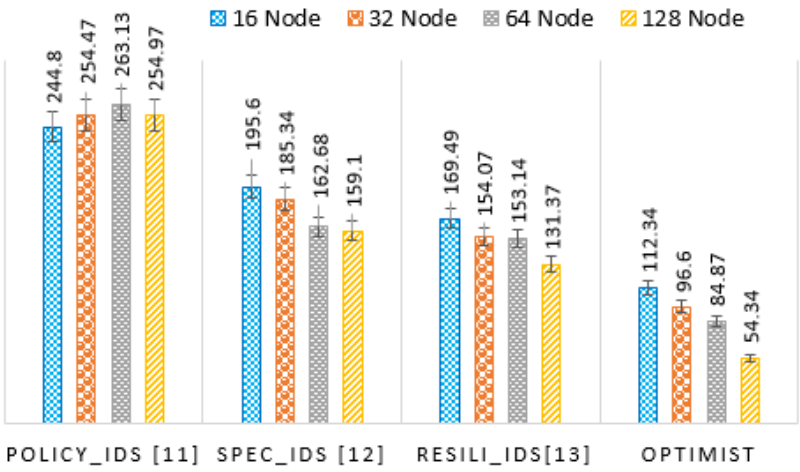


Figure 5.15: Average energy consumption comparison

Figure 5.15 and Figure 5.16 depict the comparative results of the IDS placement strategies with the works [167, 112, 168]. The IDS modules of the compared works are replaced by the OPTIMIST IDS model to assess the performances solely based on placement strategies. In [167], all nodes run IDS. Though the IDS monitors flows transparently and has a high detection rate, running IDS in all nodes results in high energy consumption. The placement strategy of the work [112] is cluster-based, which relies on query-response messages between IDS nodes and monitored nodes. Though only a few nodes (cluster heads)

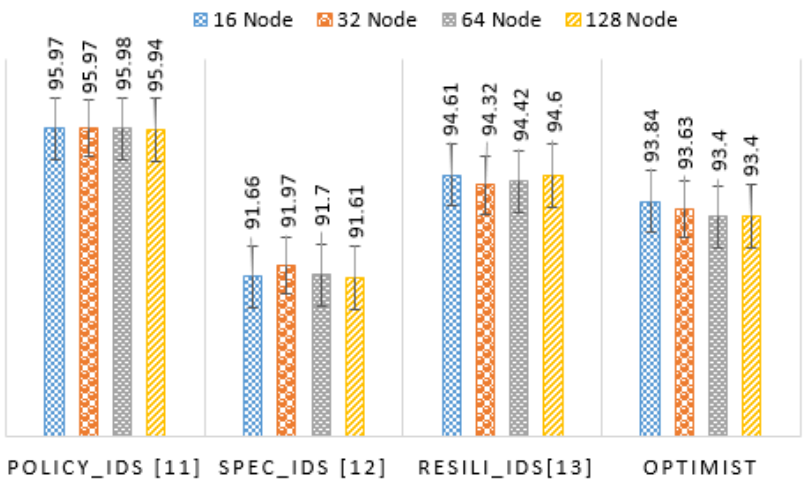


Figure 5.16: Attack detection rate in (%)

run IDS, energy consumption is still higher due to the query-response message overhead. The non-transparent nature of the proposed IDS also causes a low detection rate as malicious nodes create malicious flows avoiding the IDS nodes or can send false status reports. The work [168] has an optimum 1-hop vertex cover placement strategy where the IDS nodes monitor flows transparently. The optimum placement strategy has resulted in low energy consumption and a high detection rate compared to the previous two works. OPTIMIST IDS placement strategy has a K-hop vertex cover strategy, further reducing energy consumption. The detection rate of OPTIMIST is compatible with [168] as the placement algorithm is run using RPL DODAG structure rather than network topology, making most of the flows pass through the IDS nodes. However, few flows of length less than K can evade IDS nodes and remain undetected.

The online model performances of OPTIMIST on attack detection, along with system parameters, are compared with existing IDS models and the results are given in tabular format in the TABLE 5.8. The work of Mahdis et al. [176] has the best scores for attack detection but at the cost of huge system overheads. The proposed IDS of the work [191] consumes lesser energy as the flows of IoT devices are SDN managed where the IDS is distributed among switches. The IoT devices are directly connected to SDN switches with singe-hop links. As the flow packets are forwarded by switches, IoT devices are relieved from the task of packet forwarding, which saves energy. However, the model performs poorly in attack detection. It can be observed from the table that OPTIMIST can best balance

detection performances and resource overheads. OPTIMIST is the only solution that can detect MrDDoS attacks with competitive accuracy, whereas other protocols are focused only on HrDDoS attacks.

5.7 Summary

In this chapter, we proposed a lightweight distributed IDS solution, OPTIMIST, for IoT networks with an optimum placement strategy. OPTIMIST is trained to detect both high-rate and low-rate DDoS attacks on IoT systems. The placement problem of OPTIMIST is formulated as the weighted minimum vertex cover problem of a K-uniform hypergraph, and an approximation algorithm is used as the solution. The placement strategy is transparent in nature to reduce network overhead and to make other IoT nodes unaware of the presence of IDS nodes. The K-coverage strategy is proposed to reduce the redundancy of IDS nodes and energy consumption. To build the IDS model, WGAN-generated artificial training samples are used along with the real training data to train a LSTM model. This novel method of training is able to remove the model bias for dataset distributions. Extensive evaluations are done both on simulation and testbed platforms. The results show that the OPTIMIST IDS system can efficiently detect both high-rate and low-rate DDoS traffic flows while having a comparatively low system overhead. In the next chapter, we develop a multiple-mix attack detection and mitigation security solution using shadow honeypots with roaming strategies. This strategy aims to strengthen the IoT ecosystem's security against a variety of attacks.

"IoT security is not a one-time task; it is an ongoing commitment to protect the digital fabric of our lives."

- Raj Samani

CHAPTER

6

RENO: Roving Shadow Honeypot for Multiple-Mix-Attack Detection in IoT Networks

6.1 Introduction

Nowadays, the IoT is experiencing rapid growth in various sectors, including smart industries, transportation, building automation, environmental monitoring, and personal healthcare. This is due to their ubiquitous connectivity, which enables them to interact and share data, intelligence, and decision-making abilities with other technologies [194]. This interconnectedness creates seamless user experiences that significantly enhance people's daily lives, as evidenced by the widespread use of IoT devices. However, with external internet access, IoT devices become more susceptible to data privacy breaches and cybersecurity attacks [195]. Among these threats, rank attacks, buffer overflow (BOF) attacks, and DDoS attacks pose significant risks to IoT networks. These attacks can disrupt network topology, cause packet loss, and affect network availability, making the IoT infrastructure highly unstable when targeted simultaneously. In May 2021, a Synopsys survey revealed a concerning lack of confidence in IoT devices and network security. Although 72% of manufacturers expected attacks on IoT devices and networks within 12 months, only 18% were implementing preventative measures [196].

The inadequate security measures and absence of dedicated security solutions for IoT networks also expose them to multiple-mix attacks. As a result, a significant gap is seen between security needs and the security capabilities of the currently available IoT

devices. These devices are vulnerable for two main reasons: limited processing capacity and heterogeneity in terms of protocols, software, and hardware [197]. More specifically, it is usually not possible for IoT devices with limited radio bandwidth, computing power, battery power, and memory to run security solutions that require a lot of computing and communication power and are also sensitive to latency [198]. Due to this, it is impossible to implement extensive and comprehensive security solutions. Furthermore, due to the diversity of IoT devices, designing and executing a scalable security solution is extremely challenging [199].

Traditional network security solutions include vendor software patches, widely-used end-point defensive systems like antivirus, and fixed perimeter network defensive systems like firewalls and IDS. However, due to the diversity of IoT devices and use cases, conventional security systems are incapable of handling IoT network infrastructures [200, 201]. The well-known intrusion detection systems (IDSs), namely Bro and SNORT, are fixed and mainly use signature-based approaches on traditional IP-only networks [202]. Traditional anomaly detection systems are also inefficient in IoT ecosystems due to the large number of IoT devices and the dynamic nature of IoT networks. Therefore, there is a need for a security solution to detect multiple-mix attacks on IoT networks.

Many existing security solutions detect only one type of attack, such as a DDoS attack, a rank attack, or a BOF attack, whereas only a few solutions detect multiple-mix attacks. To the best of our knowledge, no work detects multiple-mix attacks. Motivated by this fact, this thesis integrates the IDS solution RENO, which can detect multiple-mix attacks. The RENO IDS module is based on a shadow honeypot model, which is a combination of honeypots and anomaly detection systems (ADSs). This creative security concept combines the best elements of ADS and honeypots [4, 203]. Figure 6.1 shows that ADS and honeypots have different tradeoffs in terms of scope and accuracy when finding multiple-mix attacks. Due to the fixed location of shadow honeypots in the IoT ecosystem, any attacker may identify a shadow honeypot. The attacker may also inform other attackers of the presence of the shadow honeypot. Thus, an attacker may learn about the shadow honeypot and find a way to circumvent it, rendering it useless. In this case, we used the idea of "roving shadow honeypots" to make it harder for the attacker to find the shadow honeypot, which boosts the shadow honeypot's ability to detect multiple-mix attacks. The shadow honeypot

is moved to a different IoT device that is most likely to be attacked in IoT networks. Based on the present state of the IoT network, the notion of Markov chain analysis is utilised to detect the most likely IoT device to be attacked. In addition, by utilising IP shuffles and the concept of turning services on and off, shadow honeypots will move throughout the IoT network to determine which device is the most likely to be attacked by using an attack score.

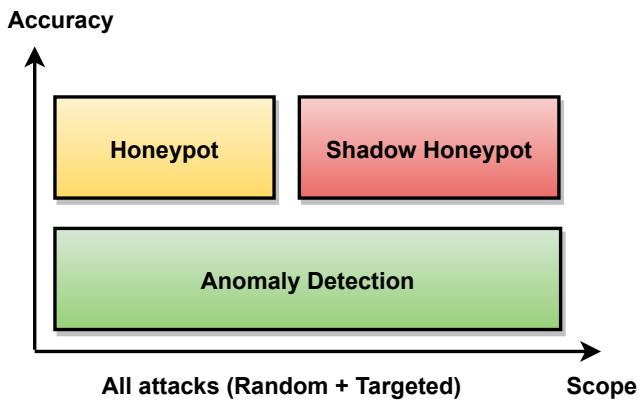


Figure 6.1: Honeypots and ADS with respective scope and accuracy [4].

The above discussions are the motivations of the proposed work, namely "RENO", which includes *lightweight shadow honeypot (LWSHP)* modules to detect multiple-mix-attacks in IoT networks. The contributions of our work are summarized as given below:

1. Unlike existing works that have focused on a single type of attack, this work offers a solution for multiple mixed attacks that can identify *rank attacks, BOF attacks, and DDoS attacks.*

2. A lightweight shadow honeypot installation based on an *attack score (AS)* and a Markov chain analysis of IoT networks is demonstrated.

3. Extensive experiments are conducted on the *Contiki-Cooja* and the *FIT IoT-LAB* testbeds to evaluate the competitive performance of the proposed RENO security solution. In comparison to current state-of-the-art techniques, the findings demonstrate that our suggested security solution is scalable and best at identifying attacks while consuming the least amount of power and memory.

The remainder of this chapter is organized as follows: Section 6.2 presents a concise

description of the shadow honeypots architecture, roving strategy, rank attack, BOF attack, and DDoS attack. Existing literature surveys on ADS/honeypot/LWSHP placement and security solutions are presented in Section 6.3. Section 6.4 presents a security solution (RENO). Section 6.5 presents the experimental setup, implementation of the proposed approach in Contiki OS, and real FIT IoT-LAB testbed. It also presents experimental results and a comparison of the proposed approach with existing security approaches. Finally, in Section 6.6, we summarize the chapter indicating future directions.

6.2 Background

This section discusses a few crucial concepts that are used in our proposed solutions. Section 6.2.1 introduces the *Lightweight Shadow Honeypot (LWSHP) Architecture.* Section 6.2.2 discusses about roving LWSHP strategy. The multiple-mix-attacks on IoT networks is introduced in Section 6.2.3.

6.2.1 Lightweight shadow honeypot (LWSHP) architecture:

As shown in Figure 6.2, the architecture as a whole is made up of three functional modules, such as filtering, anomaly detection, and shadow honeypot. The network traffic is routed through the filtering module. It uses an authorised list, and by examining the traffic that corresponds to this list, it discards a malicious packet. The second module, the anomaly detection system, gets information from the traffic filtering module. This module contains a number of ADS that begin shadow testing or attack detection. The ADS design module prefers to set the detector sensitivity to the highest possible level in order to reduce the number of *false positives (FP).* In order to protect against variant attacks, the normal service code module and the shadow testing module exchange their internal states. The level of instrumentation used in the LWSHP will be determined by how much delay the network administrator imposes on suspicious traffic. The LWSHP recognises a genuine attack and reports it to the other functional modules to prevent further attacks. Figure 6.3 is an illustration of the shadow honeypot process.

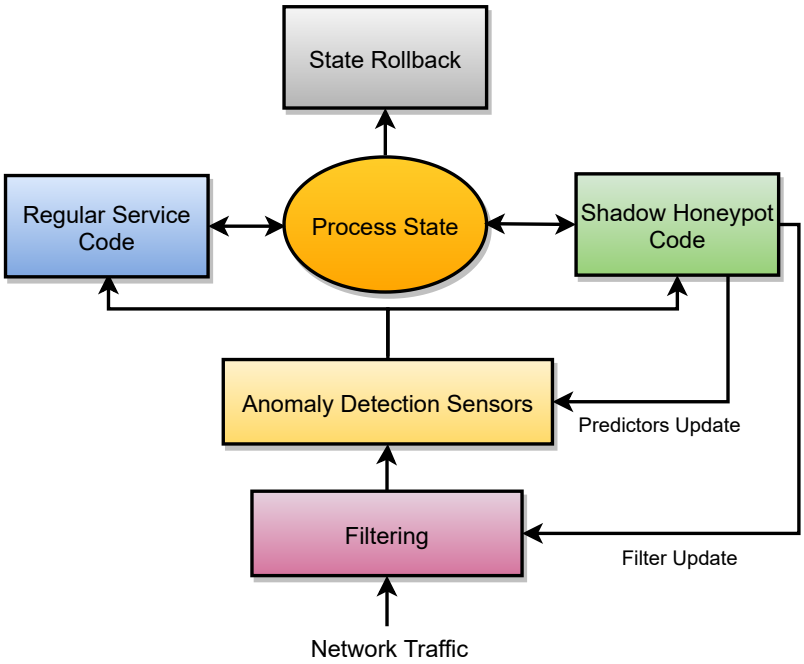


Figure 6.2: Shadow honeyPot Architecture [5]

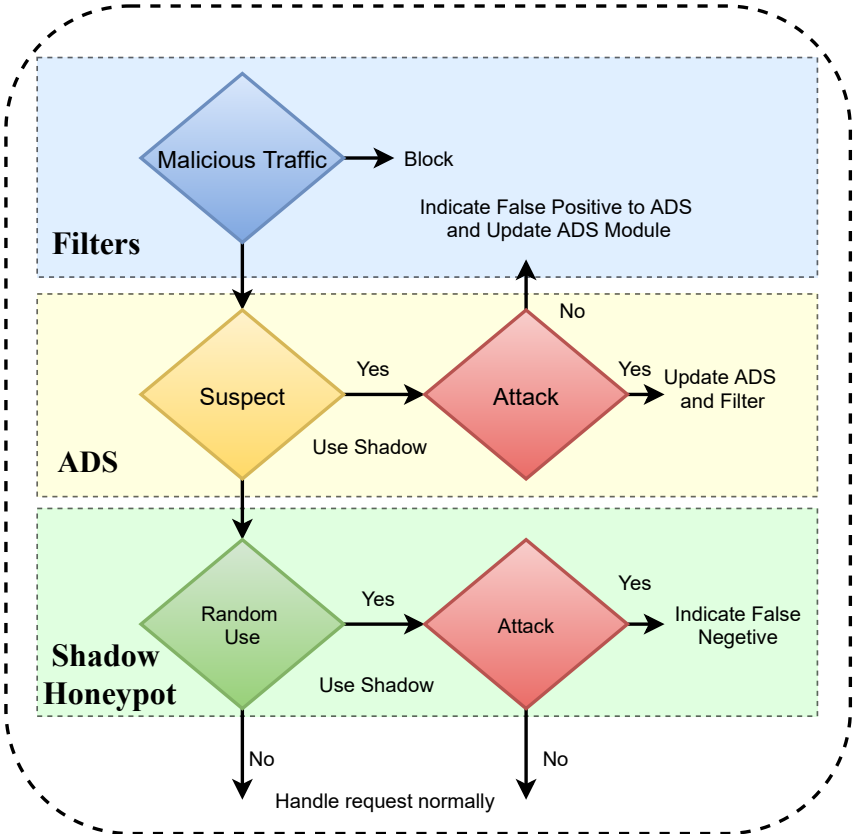


Figure 6.3: Shadow honeypot workflow.

6.2.2 Roving LWSHP:

With the help of a capture daemon, roving LWSHPs send information from one IoT node to another. It is also dependent on many characteristics, such as the intensity of an attack, the frequency of attacks, network traffic, etc. This strategy helps to record network activity and also makes it more likely to get network information from the IoT network, which is likely to be probed at a certain time. This method turns off the data capture daemons at all other IoT nodes while enabling the data capture daemon of LWSHP on IoT devices that are most likely to be attacked. In a low-interaction honeypot, the honeyd daemon is used, whereas in a high-interaction honeypot, the Xebek daemon [203] uses the aid module's ON/OFF activity. The process of daemon switching is spontaneous. The script is performed on the IoT node itself. In this scenario, IoT network data gathering is conducted in a distributed manner.

6.2.3 Attacks on IoT

IoT networks are vulnerable to a variety of threats since end nodes may be accessed over the Internet, networks are lossy by nature, and nodes have limited resources. There are several well-known attacks, including rank attacks, black-hole attacks, sink-hole attacks, version number attacks, buffer reservation attacks, bot attacks, DoS attacks, and DDoS attacks. The nature of these attacks varies greatly depending on their aims, such as disrupting network traffic, draining network resources, changing the topology, and so on. This chapter is mostly about multiple-mix attacks (*rank attacks, BOF attacks,* and *DDoS attacks,*) which are explained below.

RPL rank attack

RPL is a routing protocol in *Low-Power and Lossy Network (LLN).* It is a handy protocol that enables various communication types (i.e., multipoint-to-point traffic, point-to-multipoint traffic, and point-to-point traffic). The rank property is the crucial policy for preferring root nodes in DODAG. IoT nodes have a rank that defines their proper position with respect to the DODAG root and with respect to distinct nodes. Typically, rank increases from root to node and decreases in the direction approaching the root. Therefore, there is a massive burst in control packet traffic in the IoT ecosystem. The nodes being resource-constrained

illicitly face exhaustion of their batteries and power. Figure 6.4 provides a notion regarding the rank attack on the default RPL topology. The attacking node with ID 4 alters its rank value R and impacts adjacent nodes and connected child nodes to construct a fraudulent topology.

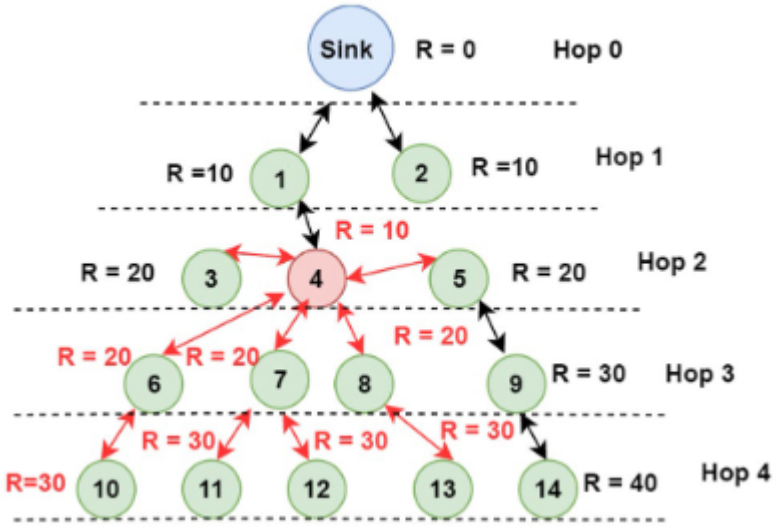


Figure 6.4: Rank attack on default RPL topology

BOF attack

IoT devices are manipulated by software-based attacks like buffer overflows. BOF attacks are common and include the well-known stack smashing [204, 205]. A buffer overflow allows the return address of a function on the stack to be modified while the attack initiation. In unprotected implementation and operation, when a function returns, the control may be moved to the place where malicious code is stored, as demonstrated in Figure 6.5. In order to undertake this type of attack, an unprotected buffer variable is identified in the program and fed with a particular input value, causing the stack frame to be overflowed. The return address is to be altered in order to jump to a new place.

DoS/DDoS attack

DoS and DDoS attacks are responsible for resource exhaustion. DDoS attacks add numerous packets with fabricated source addresses into the sky-websence web server. These source

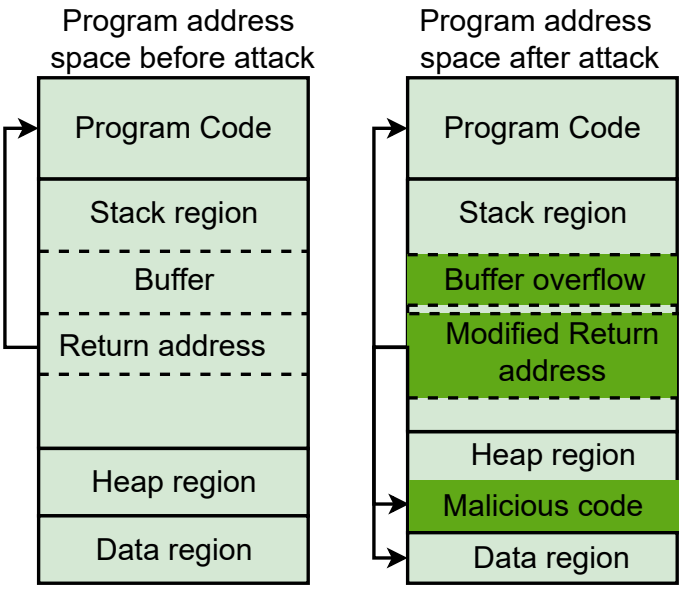


Figure 6.5: Buffer overflow attack

addresses not matched in bounded caches, which leads to congestion and interrupt services. If the DDoS attack succeeds, the attacked sky-websence web server will likely be utilized as a puppet to attack different IoT ecosystem, resulting in substantial losses. We discussed various varieties of DDoS attacks in the IoT ecosystem in Section 2.6.

6.3 Related work

This section is organized into two parts. Section 6.3.1 reviews existing works on ADS placement while in Section 6.3.2, various existing ADS/Honeypot solutions are discussed.

6.3.1 ADS/Honeypot/LWSHP placement

Chatterjee et al. [206] and Franco et al. [207] have surveyed various ADS placement strategies and honeypot systems. They categorized them into groups of centralized, distributed, and hybrid ADS placement strategies. In a centralized placement strategy, an ADS instance is run on one dedicated high-resource node like a border router (6BR). The works [208], [209] are examples of centralized ADS strategies. Though centralized ADS strategies can monitor all external traffic, some malicious internal flows generated by compromised IoT devices may remain undetected. The centralized honeypot framework (U-PoT) [210] automatically

creates a honeypot from UPnP device description documents and is extendable to any device type or vendor that uses UPnP for communication. S. Dowling et al. [211] also proposed a centralised honeypot for *Wireless Sensor Networks (WSN)*. The intended honeypot is installed on the ZigBee gateway in the wireless network. It captures all suspicious traffic to the gateway and can be later analyzed. *Secure Shell (SSH)* was chosen for the honeypot because it is the entry point that is most often attacked.

In distributed ADS strategy [212], Gaussian Mixture-based Correntropy, and an ensemble one-class statistical learning model, which is designed to monitor and recognize zero-day attacks in real-time IoT networks effectively. The work [213] is an example of a distributed ADS strategy based on Federated learning, Fourier mixing sublayer, Autoencoder, and Transformer. This strategy evaluation is based on accuracy, memory use, and energy consumption. Vishwakarma and Jain [214] suggested a self-adaptive honeypot to detect and identify DDoS attacks. The suggested approach that has been presented involves collecting logs of assaults made against ThingPot [215] honeypots and then using those logs to train ML classifiers. The authors thought about putting virtual box images of ThingPot on IoT devices in a network and putting the ML classifier on the router.

The hybrid placement strategy combines both the benefits of centralized and distributed strategies. In this strategy, a centralized entity monitors external traffic while a few of the IoT nodes are selected as ADS nodes to perform the role of watchdogs by monitoring the behavior of a subset of the nodes. The works [114], [115] are examples of hybrid placement strategies.

6.3.2 ADS/Honeypot/LWSHP solutions

A number of ADS/Honeypot/LWSHP techniques are available in the literature which can be broadly classified into three categories as RPL Rank attack, BOF attack and DDoS attack.

RPL rank attack

Kandhoul et al. [116] suggested a safe routing strategy for IoT networks that is based on Deep Q-learning and defends against sinkhole attacks and routing attacks. This strategy incorporates the benefits of both the value-based and policy-based approaches. A Markov decision process is used to simulate the learning process. Simulations with the actual

data trace demonstrate an accuracy of 89.6%. The RFTrust concept was suggested by K. Prathapchandran et al. [216] based on a trust-based, lightweight solution to securing IoT networks. It mainly detects sinkhole attacks in IoT networks based on RPL. It improves the IoT network's trusted routing by identifying and eliminating sinkhole nodes. To detect sinkhole attacks, this method additionally employs Subjective Logic (SL) and Random Forest (RF). The performance of this strategy is evaluated using simulation. R. Sahay et al. [217] came up with a good way to make the RPL protocol more secure against the Worst Parent Attack. This method improved RPL, was based on RPL, and is henceforth known as ERPL. In the process of topological building, the proposed ERPL achieves its purpose by decreasing the candidate set of parent nodes to the ideal set. As a result, ERPL assures that nodes select a parent from a set of optimum nodes, making IoT-LLNs resistant to WPA. The suggested work was compared in terms of energy usage and packet delivery ratio. Nandhini PS et al. [218] suggested a Rank Attack Detection (RAD) algorithm that keeps the integrity of control packets by using a non-cryptographic hash algorithm. The rank attack recognised warning is not a distinct control packet; rather, it is an attachment to the control message. Using the Cooja Simulator, the performance of a suggested method is assessed under multiple topologies. The studies of performance are based on packet overhead, energy usage, and precision. Additionally, it achieves 96% accuracy, consumes 40% less energy on average, and reduces the number of unnecessary packets from 44,523 to 1060.

BOF attack

Saeed A et al. [117] presented an IDS method by using random neural networks (RNNs) to build an intelligent security architecture. At the time of compilation, the application's source code is additionally instrumented in order to identify memory accesses that fall beyond the application's normal parameters. The idea relies on the generation of tags that are connected with each memory allocation and the subsequent insertion of tag-checking instructions for each memory access. To prove that the suggested security solution works, it is installed on an existing IoT system. Suspicious sensor nodes within the system's operational range and unusual base station activity are found with 97.23% accuracy, showing that the solution is useful. Xu B et al. [219] developed an architecturally better hardware security

architecture for detecting buffer overflow threats. Instruction monitoring and verification are one component of the design. This component is utilized to track the behavior of programs as they are being executed. Secure tag validation is another method, which is utilized to keep an eye on the characteristics of each and every bit of RAM. At the time of compilation, the automated extraction tools get the monitoring model and secure tag associated with each memory segment. The experimental study demonstrates that the offered strategies are capable of detecting a variety of buffer overflow threats. It also has low-performance penalties and low overheads. Fernando A et al. [220] proposed SIoT, a tool for analyzing and protecting distributed IoT systems from BOF attacks. This method includes a cutting-edge algorithm that effectively finds linkages between programs. Such linkages allow us to construct an inter-program perspective, which we can then share with a standard buffer overflow static analysis tool. Our approach has been constructed on top of the LLVM compiler and is utilized to secure BOF attacks. Our approach generates code that is just as secure as that achieved through more conventional analysis, yet programs instrumented with our approach incur an average of less than 6% in additional runtime and program size.

DoS/DDoS attack

Da. Yin et al. [118] proposed a Software Defined Network (SDN) based honeypot that mainly recognized and mitigated DDoS attacks in the IoT ecosystem. This research paper has a two-phase system, the first phase defends scan-based attacks and the second phase incorporates DDoS attack mitigation approach adopting SDN-based honeypots which enhance IoT ecosystem security. This approach effectively identifies and mitigates various attacks like SYN flood, Telnet-based, SSH-based, and scan-based. However, the downside of this method is the approach contains two phases, that consume added power and lower network lifetime. M. Aniruddha et al. [221] presented a honeypot methodology for detecting DoS attacks on IoT systems. Models like this are frequently deployed in online servers to divert DoS attacks away from the primary server. This approach prevents a DoS attack from shutting down the IoT. The model reportedly reached 94.04% efficiency. However, the proposed honeypot solution deal with a single type of DoS attack. Chronos, a revolutionary time-based ADS system, was proposed by Salahuddin et al. [222]. The ADS system utilizes

an autoencoder and time-based characteristics across different time frames to identify DDoS attacks anomalously and effectively. The author also created a threshold selection heuristic to improve the accuracy of DDoS assaults. The Chronos technique outperforms the time-based system by employing a less sophisticated anomaly detection pipeline, while outperforming flow-based systems with higher precision, according to a comparative analysis. Mihoub A et al. [223] suggested a security architecture with two components: DoS/DDoS detection and mitigation. The detection component uses a "Looking-Back" multi-class classifier, while the mitigation component applies countermeasures to particular packet types. The `BoT-IoT dataset` is being used to test this architecture, and the tests so far have shown promising results. The comparative analysis of the state-of-art work is summarised in Table 6.1.

Table 6.1: Comparative summary of related works

Thread Type	**Author & Year**	**Major Feature(s)**	**Inference(s)**
RPL Rank Attack	K. Prathapchandran et al. (2021) [216]	Trust based model	• Random forest and subjective logic implemented to detect rank attack. • Minimum false-positive, false negative rate, and high detection accuracy • Energy and memory usage issues are not addressed
	R. Sahay et al. (2022) [217]	Packet eavesdropping and negative behaviors of nodes	• Distributed lightweight weight intrusion detection approach for IoT network. • High accuracy and precision. • Scalability and energy efficciency are open challenges
	Nandhini et al. (2022) [218]	Control packet's integrity and non-cryptographic hash algorithm	• It reduces energy usage by random sampling. • No new control messages for assault detection. • They compare accuracy with three different topologies.
	Kandhoul et al. (2022) [116]	Deep Q-learning model	• The Q learning approach is used to cope with high-dimensional data. •This article suggests a security measure against several rank assaults. • Real data trace simulations exhibit great accuracy.
Buffer Overflow Attack	S. Ahmed et al. (2016) [117]	Random neural networks (RNNs) used	• RNN based intelligent security framework used for IoT system. • Minimum false positive result, but communication management and scalability are challenges.
	B. Xu et al. (2018) [219]	Secure architectural hardware used	• Hardware real-time behavior monitoring (HRTBM) and program off-line behavior analysis (POLBA) executed to detect BFO attack. • Energy efficciency and scalability issue are not examine.
	F. Teixeira et al. (2019) [220]	Array-Bound Checks inserted to guard buffer	• The proposed approach provides typical static analyses with distributed system. • It reduce overhead and runtime, but scalability, and energy efficiency are open challenges
DDoS Attack	M. Anirudh et al. (2017) [221]	Honeytrap model implimentation	• Statistical analysis based approach to detect and prevent DDoS attack. • Energy efficciency and memory usage are open challenges
	Da Yin et al. (2018) [118]	Software defined network (SDN) based honeypot	• Two phase attack detection and mitigation approach to identify DDoS attacks. • Achieve good accuracy. • The drawbacks of this approach take huge amount of power and lower network lifetime
	Salahuddin et al. (2021) [222]	ADS system based on autoencoder	• Autoencoder based ADS detect DDoS attacks effectively. • Threshold selection is based on a heuristic to improve attack detection accuracy. • It gives low false positive, This approach take massive computing power and response time.
	A. Mihoub et al. (2022) [223]	Looking-Back concept used	• ML based approach to detect and mitigate DDoS attack. • Energy efficciency, adaptability, and scalability are open challenges.

6.4 Proposed solution

This section demonstrates the proposed security solution (RENO) to detect and mitigate multiple-mix attacks (i.e., Rank attacks, BOF attacks, and DDoS attacks) in the IoT ecosystem. Section 6.4.1 gives the network assumptions for IoT networks. Section 6.4.2 gives the description of the proposed model. Data collection using Contiki Cooja and FIT IoT-LAB for the proposed model is given in Section 6.4.3. The proposed RENO solution, analysed based on Markov chain analysis modeling, is described in Section 6.4.4.

6.4.1 Network model assumptions

RENO model has been designed with the following assumptions:

- The network model is completely based on the IoT ecosystem.
- IoT devices contain different capabilities in terms of their storage, processing, and energy consumption.
- Every IoT network incorporates one highly resource-rich device which known as 6BR. Assuming the 6BR device is genuine and cannot be compromised by an assailant.
- IoT devices are tiny in size and resource-constrained in nature. Their work is based on sensing, monitoring, updating, and processing. These procedures take huge power consumption. Hence device may get drained.

6.4.2 Security model description

Honeypots and honeynets can be necessary for understanding and defending against multiple-mix-attacks on IoT networks by luring attackers and making them think they have gotten into the real systems. Using honeypots and honeynets, in conjunction with other security measures (such as *firewalls* and *ADS*), can provide a formidable barrier against multiple-mix-attacks. Shadow Honeypot [4] is a special honeypot to detect multiple-mix attacks with high accuracy. Shadow Honeypot models are extensively used for enterprise networks and IoT environments. As IoT attack flows have different characteristics among themselves, RENO security models provide high accuracy for multiple-mix attack detection in IoT networks. Being lightweight, shadow honeypot models are suitable for running in resource-constrained

IoT devices. Accordingly, this work uses a roving shadow honeypot model run on selected IoT devices for multiple mix attack detection. This work assumes that the IoT nodes of the system have the required amount of computational resources and storage to perform attack detection with the RENO security model without hampering its primary task of environment sensing. The conceptual architectural view of the proposed RENO scheme is illustrated in Figure 6.6. Brief descriptions of the RENOs sub-components (1: LWSHP roving based on attack score (AS) and Markov chain analysis; 2: LWSHP that merges the best characteristics of the honeypot and ADS) are given next.

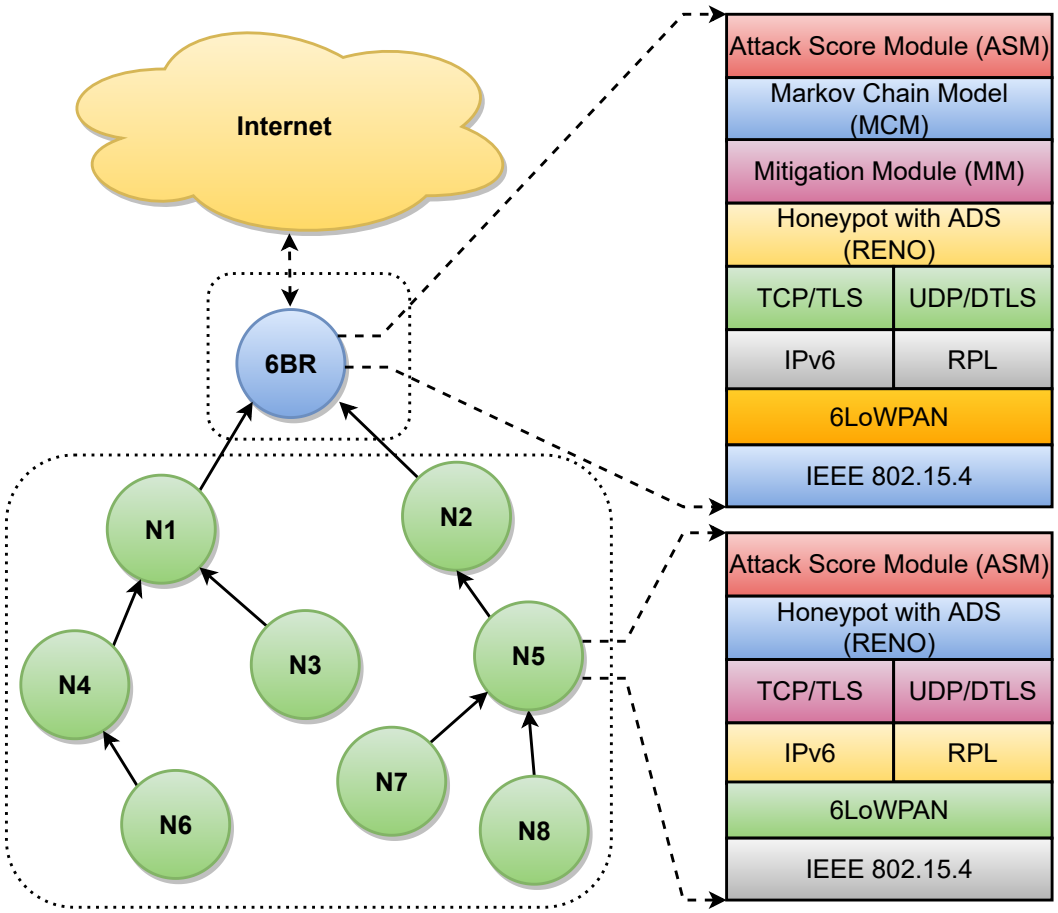


Figure 6.6: Conceptual architectural view of the proposed RENO in IoT ecosystem

RENO-first component

Considering Figure 6.6, ASM is executed on each IoT node based on attack score, and Markov chain model (MCM) analysis is used for a roving process by activating the Xebek

daemon [203] for the LWSHP. At this stage, the attack score variable AS varies from 0 *to* 1 on the IoT network. The AS value of a node is higher than that of other nodes in the IoT ecosystem. Appropriate AS is switched back to 0 on every additional node after the LWSHP is relocated. The functional pattern of the intended approach is demonstrated in Figure 6.6.

The LWSHP module is executed continuously on a 6BR router to examine external and internal network traffic. The MCM always runs on a 6BR router. Based on the LWSHP result, updated vulnerability information at the mitigation module is used to minimize further attacks. The IoT node executes ASM and detects the vulnerabilities and priorities on a scale of $(1-3)$ [224]. Scale 1 indicates high preference and frequency of attacks. We estimate the AS, and then MCM is utilized to achieve roving on the basis of AS. The AS estimation process is elaborated subsequently.

The attack score $0 \leq t_i\left(e_j^{\beta}\right) \leq 1$ to the ith object upon the experience of the jth event of an attack β, where AS provides the subsequent identifiers, which qualitatively represent the attack score of an object:

$$Compromised : t_i\left(e_j^{\beta}\right) = 1$$

$$Threatend : 0 \leq t_i\left(e_j^{\beta}\right) < 1$$

$$Unthreatened : t_i\left(e_j^{\beta}\right) = 0$$

The above attack assessment scheme gives an attack score that precisely depicts the series of attack incidents. In other words, the imperiled node should have the maximum attack rating for the other endangered nodes. Based on this presentiment, we determine the normalized attack score, for user i one step before being compromised by attack β. The normalized attack score is estimated as follow.

$$t^{\star}(\beta) = \left\{ \frac{t_i\left(e_j^{\beta}\right)}{max_{k\epsilon I}\left(I^{\star}\left(e_j^{\beta}\right)\right) t_k\left(e_j^{\beta}\right)} \mid t_i\left(e_{j+1}^{\beta}\right) = 1, t_i\left(e_j^{\beta} < 1\right) \right\} \tag{6.1}$$

In this first component, IoT network traffic is analyzed with the help of ASM and generate alert file. The alert file contains attribute like $Node\_ID$, $Node\_rank$, $Timestamp$, $Protocol$, etc. Choosing proper attributes is a vital exercise in this research work. During

every time interval t, ASM generates n alerts. The alert set is expressed as follow:

$$Attack\_score = \sum_{n=1}^{3} (alert\_count_n) \times \left(2^{3-n} - 1\right) \tag{6.2}$$

The Markov chain has to be applied to $n+1$ states for n IP addresses under consideration where n states correspond to each of the n IP addresses being under attack and $(n+1)^{th}$ state corresponds to those IP address that are not under attack. The most attacked IP address is found by observing the threat score on a particular IP address for 60 time intervals. Then the transition probabilities are calculated using the following two formulae:

$$P_{ij}^{t} = P\left[Q_{t+1} = j \mid Q_t = i\right] \tag{6.3}$$

$$P\left[Q_{t+1} = j \mid Q_t = i\right] = \frac{No.of event\left((Q_{t+1} = i) \bigcap (Q_t = j)\right)}{Totalevent\left(Q_t = i\right)} \tag{6.4}$$

where P_{ij}^{t} is termed as transition probability at timestamp t, $Q_t = i$ is a system in class i, $Q_{t+1} = j$ is a system in class j during the period $t+1$. The event $(Q_{t+1} = i) \bigcap (Q_t = j)$ represents that the system is in class j after been in class i.

The transition probability is utilized to obtain $Node\_ID$ with high probability of attack among all $Node\_ID$. The LWSHP is moved on the high attack probability node ID. Hence, the maximum attackers' activities and attack patterns could be gathered and analyzed with the help of LWSHP.

RENO-second component

For making the illusion of a real IoT ecosystem to the assailants, the approach adopted many LWSHP as exhibited in Figure 6.7.

The LWSHP is an autonomous secure module that identifies multiple mix attacks in the IoT ecosystem. The operational flow of LWSHP is shown in Figure 6.7. The associated steps are as follows 1) Rank attack detection, 2) BOF attack detection, and 3) DDoS attack detection. If the LWSHP module encounters rank inconsistencies, then the module classifies it as a rank attack. The Algorithm 6.1 described the rank attack detection pseudocode.

To detect BOF attack more accurately with the minimum *False Positive Rate (FPR)* the network traffic is rerouted towards the LWSHP module. It contains shadow enable ()

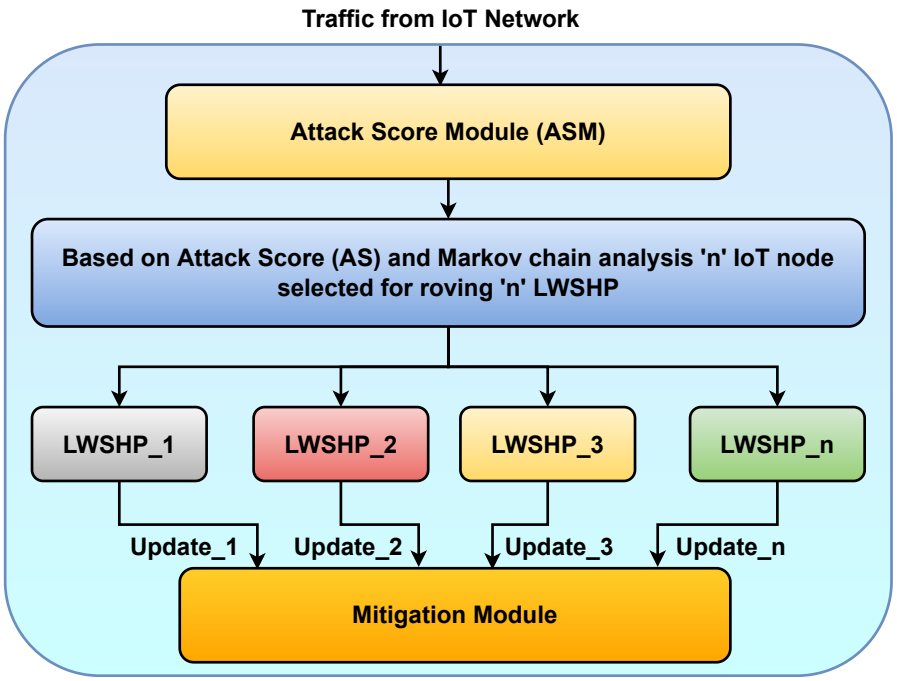


Figure 6.7: LWSHP and its operational flow (Second Component)

macro, and there task is to decide which code (i.e., shadow and regular) should run and examine the status of the shared-memory variable. The shadow pseudocode for BOF attack is describe in Algorithm 6.2.

In DDoS attacks, many zombie devices (devices controlled by the attacker) flood the targeted devices. In this process, inter-packet spacing is shorter. Hence attack is easily identifiable by ADS. There may be instances like the inter-packet spacing is shorter for non-malicious network traffic and modified DDoS attack. Unlike the conventional flooding attack, it is an intelligent attack and tough to recognize due to its enigmatic traffic behavior [118]. To detect DDoS attacks accurately and reduce *False-Positive (FP)* alarms. As LWSHP analyzes network traffic. The Algorithm 6.3 described the DDoS attack detection pseudocode.

6.4.3 Data collection

In our experiments, we used Contiki Cooja and the FIT IoT-LAB setup. Multiple mix attacks have been discovered for eight-node IDs, namely 1, 2, 3, 4, 5, 6, 7, and 8. The findings are shown over 60 different time intervals (T). The length of a certain T is equal to

Algorithm 6.1 Rank attack detection using LWSHP

Require: X: List of IoT nodes
Ensure: Rank attack detection

```
N_R: Node rank, NP_R: Node parent rank, N_F: Node fault
for <IoT node in X> do
    if N_R + RankIncreased_MinHop < NP_R then
        N_F = N_F + 1
    end if
end for
for <IoT node in X> do
    if N_F < T_N_F then
        Raise alarm
    end if
end for
```

60 *Sec.* Table 6.2 displays the AS associated with a particular IoT node ID. According to the analysis shown in Table 6.3, we find various IoT node IDs that have a higher attack score for a given T. The first thing we assume is that no IP address in the starting period should have the same attack score as T1, T2, T3, T4, T5, T6, T7, and T8. When each IoT node ID records an AS of zero for any given time period, it is assumed that no IP addresses are under attack; the attacked IoT node count is assumed to be 0.

When more than two IoT node IDs have an identical AS in a particular time period (T), we presume that the most-attacked IoT node ID has been attacked the least over the previous attack periods. Table 6.2 exhibits the AS for an individual IoT node ID using the ASM. Based on Table 6.2, Table 6.3 provides the IoT node ID that was most exposed throughout each time interval.

6.4.4 RENO security solution

The topological ordering of the RENO security solution for multiple-mix attack detection and mitigation is given in Figure 6.8. Tasks involving ASM and MCM execution are completed in an online fashion, as detailed in Section 6.4.2. The LWSHP description is given in Section 6.4.2. The ASM gets the traffic from the IoT network. This module creates a list of $n_{nodeIDs}$, each with a different attack score. On a 6BR node, a Markov chain analysis is performed using $n_{nodeIDs}$ and an attack score. The RENO security model is deployed in all IoT nodes. However, the RENO modules in all IoT nodes are initially inactive. After the IoT network traffic has been analysed by ASM, the MCM module is performed in the

Algorithm 6.2 BOF attack detection using LWSHP

Require: IoT network traffic
Ensure: BOF attack detection

```
int BufferOverflowFunc()

if shadow enable() then
    char *Lbuffer = pmalloc(100)
    char *Rbuffer = pmalloc(30)
else
    char *Lbuffer = Lbuffer[100]
    char *Rbuffer = Rbuffer[30]
    ...
    func1(Lbuffer, sizeof(Lbuffer))
    func2(Rbuffer, sizeof(Rbuffer))
    ...
    if shadow enable() then
        pfree(Lbuffer)
        pfree(Rbuffer)
        return 0
    end if
end if
```

Algorithm 6.3 DDoS attack detection using LWSHP

Require: IoT network traffic
Ensure: DDoS attack detection

```
PFBG : Packet flow behaviour graph, ADS : Anomaly Detection System
LWSHM : Lightweight shadow honeypot module

if PFBG > Threshold && FF > IP_count Threshold then
    inform an attack by ADS and reroute to LWSHM
    if attack exposed then
        signal responded to ADS and filtering element
    else
        signify false_positive (FP)
    end if
else
    utilize LWSHM
    if attack exposed then
        signify false_negative (FN)
    else
        signal responded to ADS and filtering element
    end if
end if
```

6BR to determine which IoT nodes will serve as RENO nodes. 6BR unicasts a message to each of the selected nodes to activate their respective RENO security modules. MCM forecasts the most targeted node after specified time intervals. The 6BR gives instructions to the already active RENO nodes, telling them to put their RENO security modules into an inactive state, and it gives instructions to the newly chosen RENO nodes, telling them to turn on their RENO security modules. A RENO node is capable of detecting multiple-mix attacks for as long as the RENO security module is active. The RENO node notifies the 6BR node of an assault if a multiple-mix attack is discovered. The 6BR node broadcasts information about the malicious node to all other network nodes and advises them to stop any traffic flows coming from malicious sources.

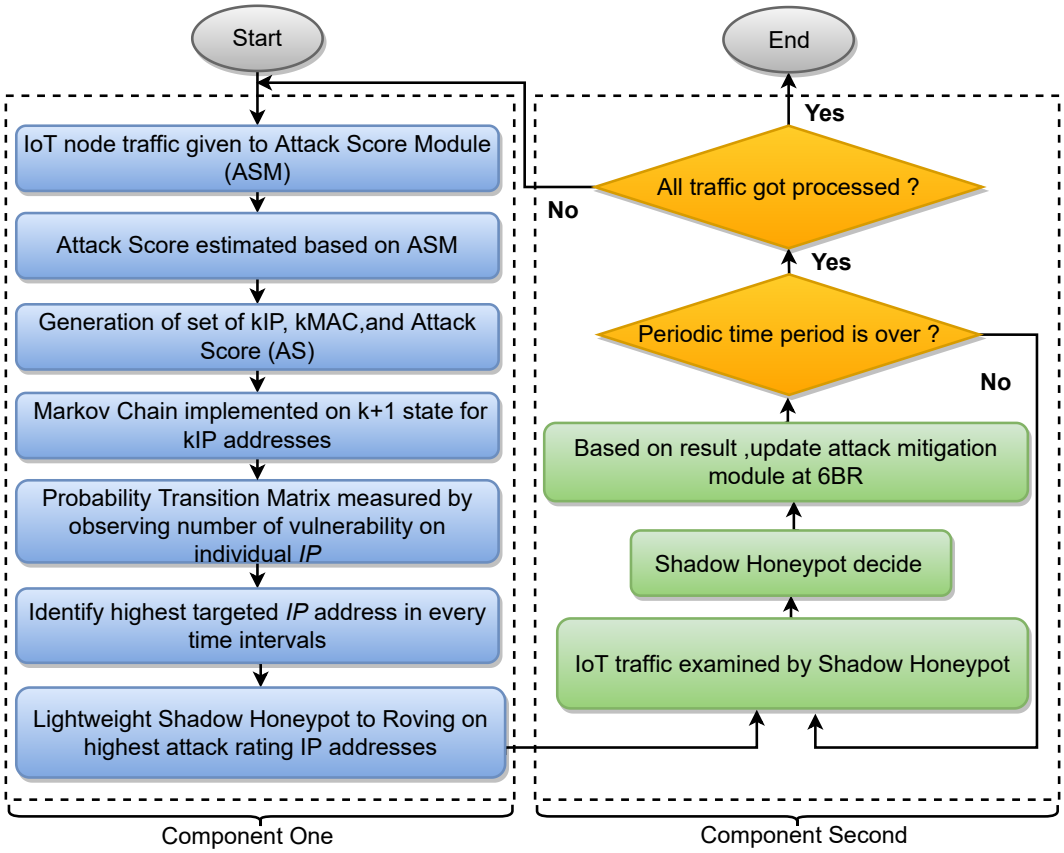


Figure 6.8: Topological order of RENO security solution

Table 6.2: The Number of Vulnerability per $Node\_ID$ collected from Attack Score Module (ASM)

Node ID	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20
①	0	0	0	0	2	4	6	8	0	9	16	17	14	0	26	14	26	18	0	27
②	0	0	2	4	3	7	8	5	0	3	10	18	19	0	24	17	28	15	0	29
③	0	2	3	8	9	7	9	14	0	2	13	10	12	0	20	17	29	14	0	22
④	0	5	6	6	4	10	14	16	0	0	22	23	14	0	20	12	14	16	0	23
⑤	0	0	3	2	5	8	7	11	0	4	15	10	17	0	25	18	27	16	0	23
⑥	0	9	2	10	6	10	15	17	0	12	14	15	9	0	23	13	25	16	0	19
⑦	0	7	5	4	3	5	8	19	0	5	17	19	14	0	10	15	18	10	0	24
⑧	0	0	2	7	7	5	6	9	0	4	10	0	5	0	10	15	16	10	0	15
Node ID	**T21**	**T22**	**T23**	**T24**	**T25**	**T26**	**T27**	**T28**	**T29**	**T30**	**T31**	**T32**	**T33**	**T34**	**T35**	**T36**	**T37**	**T38**	**T39**	**T40**
①	10	23	10	0	6	40	36	28	10	0	28	19	20	18	6	0	0	0	11	14
②	13	24	12	9	9	37	38	15	12	0	22	23	20	15	4	0	0	0	10	11
③	10	20	13	10	12	37	39	19	13	0	24	27	23	14	0	12	14	0	10	17
④	10	15	16	16	14	19	24	18	16	0	18	12	18	16	0	14	12	0	21	17
⑤	9	10	10	22	14	18	27	20	10	0	22	19	17	16	5	10	11	0	25	20
⑥	12	13	10	25	26	19	25	17	10	0	25	23	28	16	3	30	14	0	24	20
⑦	13	17	12	14	13	25	28	19	12	0	19	19	18	10	0	10	9	0	19	17
⑧	4	5	12	12	12	25	16	16	12	0	0	10	11	10	9	10	12	0	10	0
Node ID	**T41**	**T42**	**T43**	**T44**	**T45**	**T46**	**T47**	**T48**	**T49**	**T50**	**T51**	**T52**	**T53**	**T54**	**T55**	**T56**	**T57**	**T58**	**T59**	**T60**
①	0	0	36	43	11	0	14	10	46	38	27	21	40	0	16	48	36	21	0	30
②	0	9	34	38	11	0	14	9	58	35	29	20	37	0	14	57	38	26	0	27
③	0	10	42	40	12	0	10	11	49	34	22	23	37	0	20	52	39	29	0	29
④	0	11	45	33	13	0	9	11	34	46	23	18	19	0	10	49	24	24	0	11
⑤	0	11	26	36	23	0	10	9	37	36	23	21	18	0	10	38	27	23	0	12
⑥	0	12	23	25	21	0	11	14	45	26	19	24	19	0	13	34	25	18	0	13
⑦	0	10	11	12	21	0	9	0	48	30	24	19	25	0	10	30	28	12	0	0
⑧	0	10	10	0	5	0	0	10	46	30	15	0	25	0	0	11	16	16	0	0

Table 6.3: List of the most Attacked Node ID

Time Periods	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20
Most Attacked Node ID	Null	6	4	6	3	6	6	7	Null	6	4	4	2	Null	5	5	5	1	Null	2
Time Periods	**T21**	**T22**	**T23**	**T24**	**T25**	**T26**	**T27**	**T28**	**T29**	**T30**	**T31**	**T32**	**T33**	**T34**	**T35**	**T36**	**T37**	**T38**	**T39**	**T40**
Most Attacked Node ID	6	2	4	6	6	1	3	1	4	Null	1	3	6	1	8	4	3	Null	5	5
Time Periods	**T41**	**T42**	**T43**	**T44**	**T45**	**T46**	**T47**	**T48**	**T49**	**T50**	**T51**	**T52**	**T53**	**T54**	**T55**	**T56**	**T57**	**T58**	**T59**	**T60**
Most Attacked Node ID	6	6	4	1	5	Null	2	6	2	4	2	3	1	Null	3	2	3	3	Null	1

6.4.5 Markov chain analysis modelling

The Markov chain model that we construct has a total of nine states. In our experiments, we look at eight states that correspond to the eight IoT node IDs. The ninth state is when no IoT node is under attack. Using Equation (6.2), Equation (6.3), and satistical information from Table 6.2 and Table 6.3, we determine the matrix of transition probabilities P at a specific time, as shown below:

Initially, there is no IoT node under attack in the IoT network. Therefore, we obtain an initial attack probability (A_p) as

$$A_p(0) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$$

$$P = \begin{pmatrix} 0.054 & 0.122 & 0.180 & 0.125 & 0.134 & 0.079 & 0.158 & 0.000 & 0.148 \\ 0.214 & 0.075 & 0.143 & 0.000 & 0.175 & 0.107 & 0.160 & 0.111 & 0.015 \\ 0.275 & 0.075 & 0.075 & 0.019 & 0.123 & 0.123 & 0.075 & 0.068 & 0.167 \\ 0.121 & 0.041 & 0.098 & 0.189 & 0.169 & 0.000 & 0.162 & 0.179 & 0.041 \\ 0.054 & 0.122 & 0.180 & 0.125 & 0.134 & 0.079 & 0.158 & 0.000 & 0.148 \\ 0.169 & 0.000 & 0.128 & 0.056 & 0.152 & 0.179 & 0.138 & 0.125 & 0.053 \\ 0.214 & 0.075 & 0.143 & 0.000 & 0.175 & 0.107 & 0.167 & 0.111 & 0.008 \\ 0.121 & 0.000 & 0.041 & 0.130 & 0.163 & 0.000 & 0.230 & 0.123 & 0.192 \\ 0.000 & 0.111 & 0.056 & 0.212 & 0.158 & 0.268 & 0.126 & 0.000 & 0.069 \end{pmatrix}$$

As per the experimental study, after time duration 't' state probability of the nine sate system can be predicted as below:

$$A_p(t) = A_p\,(0)\,P$$

As a result, we achieve

$$A_p(t) = [0.000\ 0.111\ 0.111\ 0.212\ 0.222\ 0.333\ 0.111\ 0.111\ 0.054]$$

Hence, we can predict the node ID ⑥ (aaaa:c30c:0:0:6) is the one that is most targeted.

In Figure. 6.9 exhibits the probability of the attack on an eight-node IoT system at a time instant 't' in a percentage format. It assists in roving the LWSHP to the particular node address, which is very likely to be attacked.

6.5 Performance evaluation

This section is divided into three subsections. Section 6.5.1 describes the experiment environments and setups. Section 6.5.2 defines the metrics to evaluate the performances of RENO. In Section 6.5.3, the performance of RENO is evaluated, and the competitive result analysis is done.

6.5.1 Experiment environments and setups

An IoT scenario is considered for the performance evaluation of the RENO, as shown in Figure 6.10 (a). IoT nodes are randomly deployed for sensing purposes and have multi-hop path connectivity among themselves. IoT nodes are connected to the Internet through a

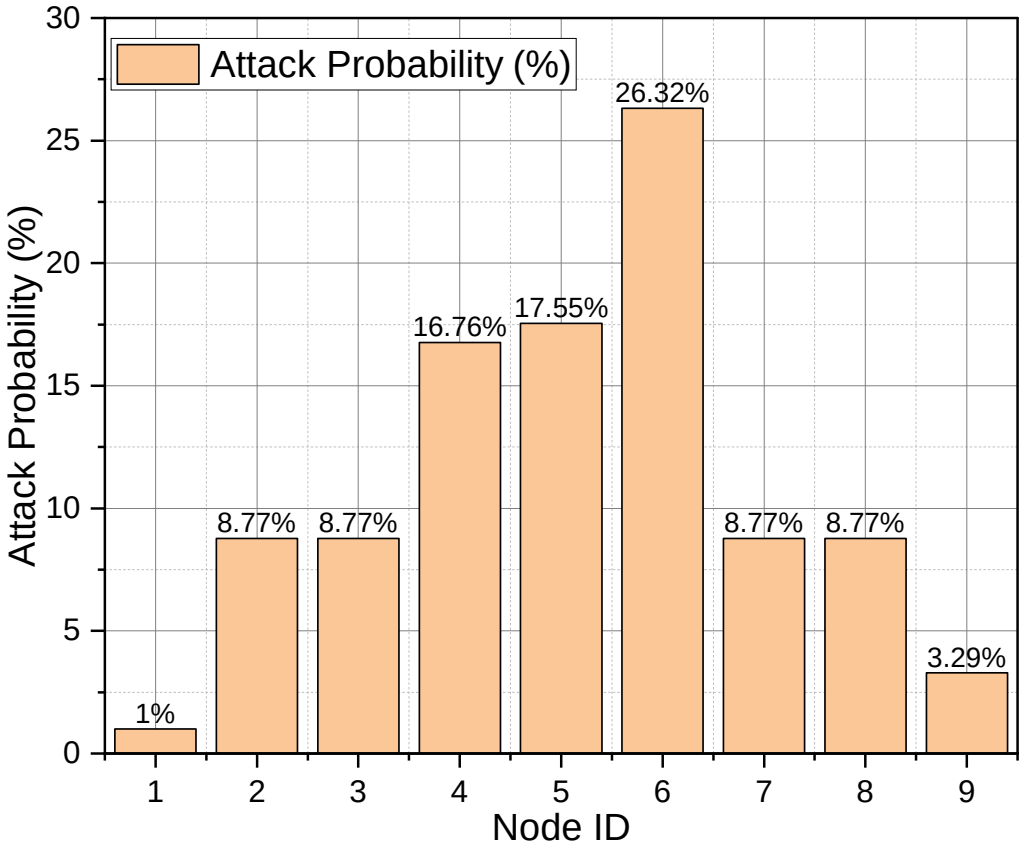
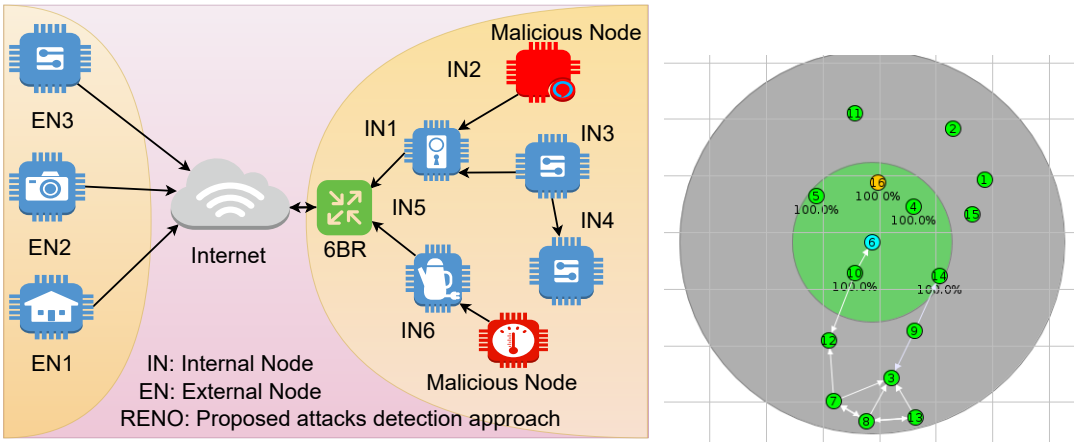


Figure 6.9: Details of attack on IoT node IDs

6BR node with ample storage and computation power. The internal IoT nodes of the IoT system are accessible by external nodes through the 6BR node using the Internet. As shown in Figure 6.10 (a), both internal and external nodes can be malicious in nature. The scenario is developed and tested in the Contiki Cooja [46] simulation and the FIT IoT-LAB [62] testbed environments. The experimental parameters of Contiki Cooja and FIT IoT-LAB are presented in Table 6.4. The comprehensive explanation of all simulation and testbed-based experiments is as follows:

Case 1: Non-attack Scenario (Experiment 5.1): In non-attack cases, all internal and external ($IN + EN$) legitimate nodes request for IoT service (i.e., temperature and humidity) using the sky-websence web server. The experiment is conducted using 8, 16, 32, 64, and 128 IoT nodes, as shown in Figure 6.10 (b). Network traffic is filtered at distributed ASMs.

Case 2: Multiple-Mix-Attack Scenario (Experiment 5.2): In attack scenarios, various



(a) (b)

Figure 6.10: IoT network setup for experimentation

Table 6.4: Simulation and real-time test-bed parameters

Parameter name	Simulation	Real time testbed
Operating system	Contiki 3.0, Contiki 4.5	Contiki-NG
Simulator/Testbed	Cooja Cooja	FIT IoT-LAB
Network size	8,16, 32, 64 nodes	
Radio Environment	UDGM	
Node Type	Tmot Sky	IoT-Lab A8
Routing Protocol	RPL	RPL Lite
RPL Objective Function	MRHOF - ETX, OF0	MRHOF - ETX
MAC/adaptation layer	Contiki MAC/6LoWPAN	
Transmitter output power	(dBm) 0 to -25	
Receiver sensitivity	(dBm) -94	
Radio frequency	2.4GHz	
Attack Modeled	BOF attack, DDoS attack, Rank attack	
Experiment Duration	60 minutes	

attacks are generated from malicious nodes. This experiment contains two types of nodes *(i.e., Legitimate Node (LN) and Malicious Node (MN))*. These nodes request services to the sky-websence web server. We exert 2, 4, 6, and 8 spiteful nodes during these experiments, as shown in Figure 6.11. These nodes initiate various attacks using IoT protocol stacks.

Case 3: Multiple-Mix-Attack with RENO Solution (Experiment 5.3): In this scenario, Case 2 is executed with a RENO security solution. At the initial stage, simulation

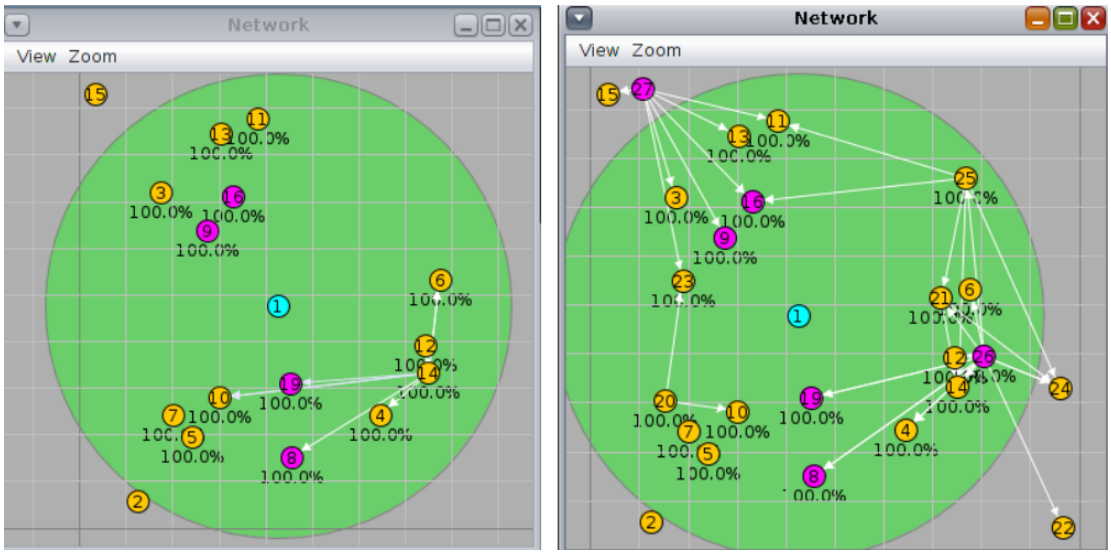


Figure 6.11: Snapshot of 4, 6 malicious nodes during these experiments

and a testbed are used to generate an in-house dataset that includes normal and multiple-mix attack flows using 8, 16, 32, and 64 IoT nodes. In this scenario, we choose 15%, 30%, and 45% of the IoT nodes that are malicious. These malicious nodes are used to start multiple-mix attacks. The 6BR node has RENO, ASM, MCM, and MM modules that are always active. Based on the MCM analysis, a subset of IoT nodes is chosen to activate the RENO security solution in order to identify multiple-mix assaults. Over time, the active nodes of the RENO security system are reshuffled based on the attack score given by ASM.

6.5.2 Performance metrics

The following metrics are defined to evaluate the performance of the RENO security solution.

- *Throughput (THP)*: This metric measures the ratio of the network throughput in the presence of multiple mix attacks with respect to the observed network throughput in the absence of multiple mix attacks.
- *CPU energy (ENEC)*: The metric measures the energy consumed by CPU in IoT devices throughout the experimentation.
- *Memory consumption (MEMC)*: It shows the percentage of memory utilization of the IoT devices to run RENO throughout the experimentation.
- *False alarm rate*: The false alarm rates are estimated by applying *Specificity* and

Sensitivity. *Specificity* estimates the proportion when the genuine traffic is identified accurately, and *Sensitivity* estimates the proportion when the attack traffic is identified accurately. Both *Specificity* and *Sensitivity* are represented as follows:

$$Specificity\ (\alpha) = \frac{TP}{TP + FN} \qquad Sensitivity\ (\beta) = \frac{TN}{TN + FP} \tag{6.5}$$

where, *TP=True Positive (get nuine nodes identified accurately) FN=False Negative (genuine nodes identified wrongly) TN=True Negative (malicious nodes identified accurately) FP=False Positive (malicious nodes identified wrongly)*

The FPR is the proportion when the genuine node is incorrectly classified as a malicious node. FNR is the proportion when the malicious node is incorrectly classified as a genuine node. The FPR and FNR can be defined as $FPR = 1 - \alpha$, and $FNR = 1 - \beta$.

- *Accuracy (ACC)*: The accuracy of the proposed RENO is defined as its ability to effectively identify multiple-mix attacks in an IoT network, in addition to the typical behavior in the IoT network. This has been calculated with Equation (6.6).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.6}$$

- *Attack detection time (ADT)*: It indicates the total amount of time needed to accurately identify multiple mixed attacks.

6.5.3 Result analysis

For the evaluation of the RENO security solution, experiments are run on simulation as well as a testbed (refer to Section 6.5.1). The results are presented in two parts as follows. Section 6.5.3 shows the performances of the RENO on the Contiki Cooja simulation, and FIT IoT-LAB test-bed environment, and Section 6.5.3 presents the competitive analysis of the RENO performances on placement strategy and attack detection with existing protocols.

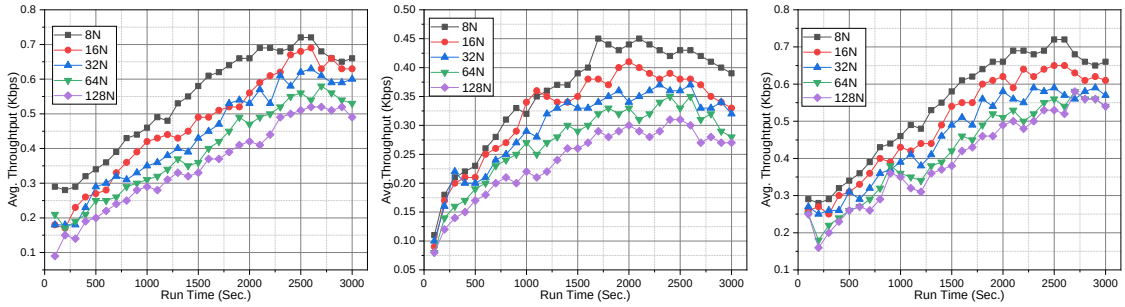
Simulation and testbed-based performance evaluation

We conduct three experiments: one in which there is no attack, one in which there is an attack, and one in which there is an attack employing RENO security solutions using Contiki Cooja [46] and FIT IoT-LAB [62]. All of these experiments run on different numbers of nodes.

The IoT network traffic is analysed using `collect view methods` and the `iperf` [193] tool in Contiki Cooja and FIT IoT-LAB, respectively. First, we evaluated the throughput performances of these experiments, which are depicted in Figure 6.12 (a), Figure 6.12 (b), and Figure 6.12 (c). The comparative threshold analysis with various IoT node counts and associated experiment run times is shown in Figure 6.12 (a). The figure illustrates that an increase in the number of nodes leads to a gradual drop in the average throughput. This occurs as a result of factors such as the number of devices connected to the IoT network, network congestion, packet loss, errors, etc.

In Experiment 6.2, the same number of nodes and a malicious node are used to perform a multiple-mix attack. As shown in Figure 6.12 (b), the average throughput suffers a significant drop of 34%-45% depending on the number of IoT nodes used and the corresponding experiment run duration. This occurs as a result of malicious nodes causing problems in the functioning of RPL, resulting in end-to-end delay and packet drop.

Experiment 6.3 is the one in which Experiment 6.2 with the RENO security solution is carried out. As can be seen in Figure 6.12 (c), the implementation of the RENO security solution results in an improvement of 32%-44% in the average throughput. In RENO, AS and MCM analysis are used to make LWSHP work. As a result, it is much easier to spot attacks that use a combination of methods.



(a) (b) (c)

Figure 6.12: Avg. throughput for 50 *min* network execution in (a) Non-attack Scenario, (b) Multiple-Mix-Attack Scenario, and (c) Multiple-Mix-Attack with RENO solution

Average energy overhead: In experiments 6.1, 6.2, and 6.3, we utilised the `powertrac` and the `Sysstat` [192] tool to determine average power utilisation per node and average energy consumption with the help of two types of RPL modules: RPL only and RPL collects. Figures 6.13 (a) and Figure 6.14 (a) show how power utilisation and energy usage vary with

the number of IoT nodes in a non-attack scenario. In assault situations, power utilisation per node and energy consumption increase by 46.8% and 37.8%, respectively, as illustrated in Figures 6.13 (b) and Figure 6.14 (b). This is due to while under attack conditions, we are unable to achieve adaptive routing paths, local optimization, in-time packet processing, and other similar goals. During experiment 6.3, we put the RENO security solution into action and found that the average power usage per node and energy consumption are 1.57 mW per node and 98749 mJ for the whole IoT ecosystem. These results are depicted in Figure 6.13 (c) and Figure 6.14 (c), respectively.

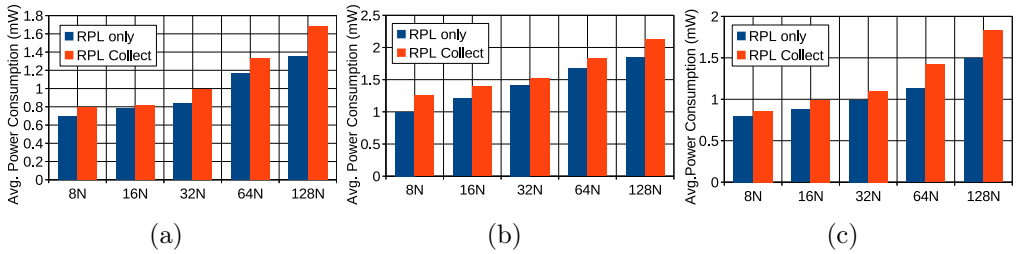


Figure 6.13: Avg. power per node for 50 *min* network execution in (a) Non-attack Scenario, (b) Multiple-Mix-Attack Scenario, and (c) Multiple-Mix-Attack with RENO solution

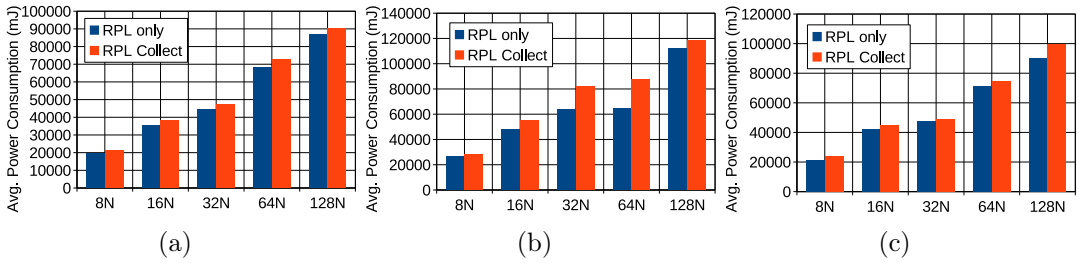


Figure 6.14: Avg. energy consumption for 50 *min* network execution in (a) Non-attack Scenario, (b) Multiple-Mix-Attack Scenario, and (c) Multiple-Mix-Attack with RENO solution

Average multiple-mix attack detection time: Experiment 6.3 is run numerous times with a changing number of IoT nodes to calculate the average attack detection time, and attack detection time is recorded for each simulation. As the number of attack nodes rises, the attack detection time decreases from 263.72 $ms.$ to 246.21 $ms.$, as shown in Figure 6.15 (a). Therefore, the detection time will be reduced according to the number of attack nodes that are present. Since the time it takes the RENO to find a multiple-mix attack in an IoT network is a concern for how effectively the attack detection module performs.

FPR and FNR: To analyse FPR and FNR performance metrics, Experiment 6.3 is

carried out using a range of 8 to 128 IoT nodes. Experiment 6.3 is performed 100 times, and the average FPR and FNR are calculated using Equation (6.5), as shown in Figure 6.15 (b). The average FPR and FNR go down as the number of IoT nodes, including malicious nodes, rises, as seen in the figure. The reason for this is due to an increasing number of IoT nodes makes it possible to execute RENO on the appropriate nodes and investigate multiple-mix attacks.

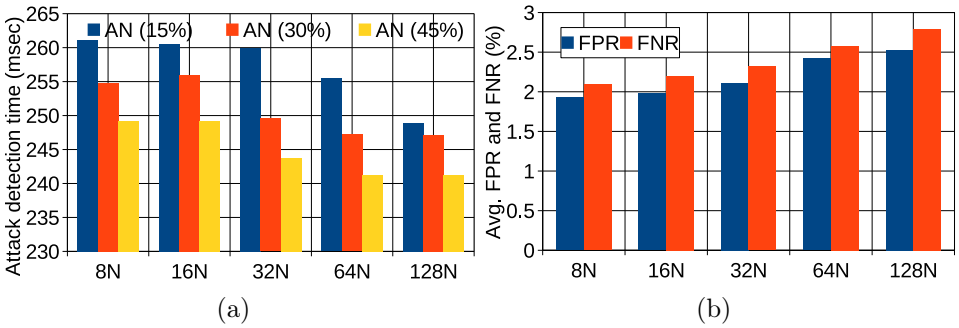


Figure 6.15: Avg. throughput, power and energy for 50 *min* network execution in attack scenario

Memory consumption: In experiment 6.3, we used the `size command` and the `Sysstat` [192] tool to assess memory utilisation in Contiki Cooja and FIT IoT-LAB, respectively. The RENO security module has several submodules built within it, including ASM, MCM, LWSHP, and MM. The 6BR node incorporates all of these sub modules, and other IoT nodes that contain ASM are in an active state. Based on the ASM score, the LWSHP node is also active. The 6BR node is a powerful node (like a PC, laptop, etc.) that claims more ROM than other IoT nodes. The amount of ROM required to host the ASM and LWSHP modules in resource-constrained nodes is smaller than the amount of ROM available in constraint nodes (i.e., 48k in sky mote). Figure 6.16 illustrates the memory overhead-related analyses. The overhead bar depicts the ASM and LWSHP's true overhead in the IoT ecosystem. In addition, we estimate the amount of RAM that is used by each of RENO's individual submodules, as seen in Figure 6.16. During the experiments, we use `Sky motes`, and the total amount of RAM that they have is 10 *kb*. As a result, RENO modules with 0.981 *k* additional RAM requirements can work efficiently on IoT nodes.

In the testbed experiments, we run non-attack scenarios, multiple mix-attack scenarios, and attacks with RENO solutions. As mentioned in Section 6.5.2, the performance metrics are used to assess the proposed solution (RENO). The experimental analysis shows Avg.

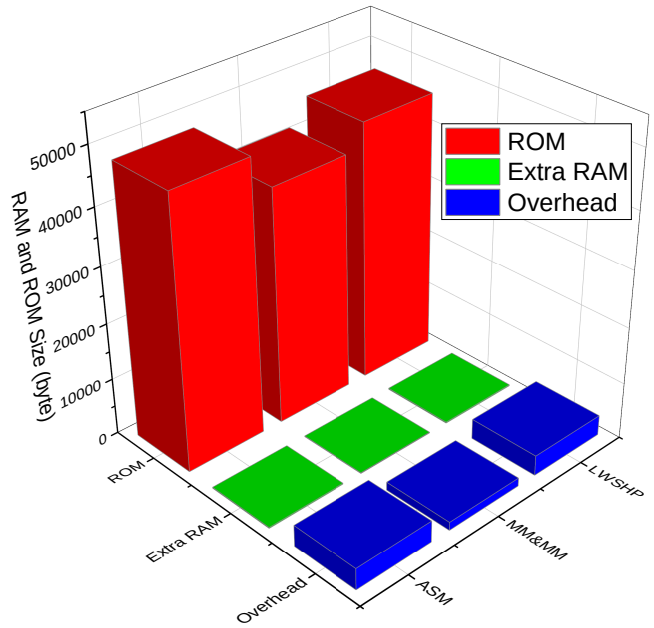


Figure 6.16: RAM and ROM Usage

total energy consumption, Avg. power utilisation per node, and Avg. throughput for the entire IoT network. Table 6.5 presents the experimental results of the Contiki Cooja simulation and the FIT IoT-LAB real testbed over a 50-minute time interval (i.e., each experiment runs for 50 minutes).

Table 6.5: Network energy, power consumption per node, and throughput for IoT ecosystem (During RENO solution running on Contiki Cooja and FIT IoT-LAB).

Part A: During Contiki cooja simulation															
	N/W Energy (mJ)					**Node Power (mW)**					**Throughput (Kbps)**				
Attack Scenario	**8N**	**16N**	**32N**	**64N**	**128N**	**8N**	**16N**	**32N**	**64N**	**128N**	**8N**	**16N**	**32N**	**64N**	**128N**
Case 1	19901	32041	49012	57389	78839	0.75	0.81	0.98	1.15	1.29	0.732	0.711	0.694	0.651	0.621
Case 2	24834	43913	58736	70023	87189	1.11	1.29	1.42	1.53	1.83	0.441	0.422	0.412	0.398	0.381
Case 3	20110	33102	51341	58489	80010	0.83	0.91	1.01	1.23	1.34	0.729	0.698	0.689	0.649	0.598
Part B: During FIT IoT-LAB real-time test-bed execution															
	N/W Energy (mJ)					**Node Power (mW)**					**Throughput (Kbps)**				
Attack Scenario	**8N**	**16N**	**32N**	**64N**	**128N**	**8N**	**16N**	**32N**	**64N**	**128N**	**8N**	**16N**	**32N**	**64N**	**128N**
Case 1	21368	34833	50068	59189	80572	0.81	0.88	1.05	1.18	1.34	0.673	0.648	0.637	0.621	0.609
Case 2	28936	45892	59351	71105	87968	1.21	1.34	1.46	1.53	1.98	0.475	0.442	0.411	0.378	0.369
Case 3	22979	34135	52623	59134	81462	0.82	0.97	1.15	1.25	1.41	0.661	0.639	0.592	0.598	0.499

Comparison of the proposed security solution with the closely-related works

The RENO security solution's performance on multiple-mix attack detection, as well as system parameters, are compared to existing security solutions, and the results are

summarized in tabular fashion in Table 6.6.

In [225], all nodes run security solutions. It is a security solution based on deep learning. According to the findings, it is scalable with performance criteria such as FPR, FNR, accuracy, and ADT, giving values of 3.47%, 5.73%, 93.58%, and (592.37-987.45) ms, respectively. The author of [226] suggested a security solution that makes use of a 1-hop optimum vertex cover placement approach in which IDS nodes monitor flows in a transparent manner. According to the paper results, the energy consumption, memory utilisation, FPR, FNR, and accuracy are 178529 mJ, 7500/47507, 3.97%, 4.11%, and 95.36%. In [216], the results of the proposed model show the performance parameters, such as energy use, FPR, FNR, and accuracy, which are given as 137034 mJ, 4.31%, 5.24%, and 93.21%, respectively. However, our RENO is scalable and detects multiple mix attacks with high accuracy, minimum energy, minimum power per node, lowest memory consumption, and minimum attack detection time, which is comparable to state-of-the-art works.

In [117], the BOF attack detection and prevention approach is carried out on every IoT node. With this method, each IoT node does security analysis in addition to its other duties, such as sensing and making decisions, without the need for any extra hardware. In addition to this, it has a throughput on average of 0.681 kbps, an energy level of 168352 mJ, a memory use (RAM/ROM) of 9893/52680, an average attack detection time of (723.35–965.32) ms., FPR and FNR values of 3.15% and 3.68%, respectively, and an attack detection accuracy of 97.23%. The author suggested hardware security modules in [219]. The hardware component keeps an eye on execution traces in real-time to provide information about how the system is behaving. It has additional hardware that occupies more space and consumes more energy (143681 mJ). This approach gives an attack detection time range of (260.32–548.13) ms. on average. In [220], the author uses standard static analysis to look at a distributed system from a global point of view. This point of view makes it easier to find BOF attacks. This system is also scalable and can make use of supplementary hardware. It requires additional memory in the form of RAM/ROM (11585/54983) and additional power (160251 mJ). On the other hand, as can be seen in Table 6.6, the RENO security solution that has been proposed has a higher accuracy rate, a lower FPR/FNR value, a better throughput, a lower energy consumption, a lower FPR/FNR value, a shorter amount of time needed to detect an attack, and it provides support for scalability.

In [221], the author employed a honeypot to enable a security system to recognise DDoS assaults while retaining efficiency using a verification technique. In this method, all traffic analysis is handled by one centralised machine. This approach requires the use of supplementary hardware and a greater amount of energy. In addition, the proposed security solution has an attack detection accuracy of 94.04%, an average attack detection time of (763.36–895.46 ms), and FPR and FNR values of 4.24% and 4.61%, respectively. In [227], the author improves single-learner AML-IDS models like PCA and 1-SVM to construct efficient, scalable, and distributed intelligent IDS for IoT network intrusion detection. This distributed AI/ML model can run indefinitely on IoT nodes, which use more power than conventional RENO security models. Based on results, it is scalable with performance criteria such as energy consumption, memory utilisation, FPR, FNR, accuracy, and ADT, giving values of 12984 mJ, 8460/51783, 3.62%, 3.95%, 96.72%, and (550.73-635.28) ms, respectively. The author offers in [223] a unique security architecture with two modules: attack detection and prevention. The detection module provides fine-grained detection using the "Looking Back" approach. Based on the detection module, particular packet types receive the relevant mitigating countermeasures. The installation of this design necessitated the usage of 143846 mJ of energy. FPR and FNR values (2.34% and 2.86%, respectively) as a function of attack detection time (703.83-869.44) ms. However, the proposed RENO security solution reduces FPR and FNR values with minimum overhead. The following are the summarised flaws of existing security methods identified in Table 6.6:

- Existing security systems evaluate external traffic for just one form of attack identification (i.e., Rank attack, BOF attack, and DDoS attack). As a result, internal traffic attacks remain undetected.

- The memory footprint is not considered by the security solutions for attack detection.

- While detecting attacks on an IoT network, current security solutions do not take into account the network's overall energy consumption or its per-node power consumption.

- There are problems with the scalability of the security solutions.

- The FPR and FNR of the security solutions are greater.

Table 6.6: Comparison of the proposed strategy with the closely related works.

Thread Type	Ref.	Simulation/ Testbed	System type	THP (%)	ENEC (mJ)	RAM/ROM (byte)	FPR (%)	FNR (%)	ACC (%)	SCA	ADT (ms)
Rank attack	[225]	Ⓢ	Ⓓ	N/A	N/A	N/A	3.47	5.73	93.58	✓	(592.37-978.45) ms.
	[216]	Ⓢ	Ⓒ	0.732	137034	N/A	4.31	5.24	93.21	✗	N/A
	[226]	Ⓢ	Ⓓ	N/A	178529	**6580/49507**	3.97	4.11	95.36	✓	(547.21-684.53) ms.
BOF attack	[117]	Ⓣ	Ⓒ	0.681	168352	9893/52680	3.15	3.68	97.23	✗	(723.35-965.32) ms.
	[219]	Ⓢ	Ⓓ	N/A	143681	N/A	N/A	N/A	N/A	✓	**(260.32-548.13) ms.**
	[220]	Ⓢ Ⓣ	Ⓒ	N/A	160251	11585/54983	N/A	N/A	N/A	✓	N/A
DDoS attack	[221]	Ⓢ	Ⓒ	N/A	N/A	N/A	4.24	4.61	94.04	✗	(763.36-895.46) ms.
	[227]	Ⓣ	Ⓓ	N/A	129894	8460/51783	3.62	3.95	96.72	✓	(550.73-635.28) ms.
	[223]	Ⓢ	Ⓒ	N/A	143846	N/A	**2.34**	2.86	**99.81**	✗	(703.83-869.44) ms.
Proposed RENO		**Ⓢ Ⓣ**	**Ⓓ**	**0.892**	**98749**	**6983/50451**	**2.52**	**2.78**	**99.78**	✓	**(263.72-559.28) ms.**

Simulation: Ⓢ, Testbed Ⓣ, Distributed:Ⓓ, Central:Ⓒ, Throughput: THP, Attack detection time: ADT,
N/A: Not Available, SCA: SCAlability, ADT:Attack detection time, Ref.=References

6.6 Summary

In this chapter, a novel security solution named RENO that merges the roving ability, honeypots features, and ADS has been proposed. We implement and evaluate the RENO security solution, demonstrating that the suggested lightweight RENO works well in the IoT environment. The security solution improves performance by decreasing *false positive (FP)* and *false negative (FN)* rates to 2.52% and 2.78%, respectively, which outperforms existing security solutions. The suggested security method additionally makes use of a minimal average energy consumption, as well as per-node power use and memory footprints that are (ROM/RAM).

"Cybersecurity is much like a game of chess; attackers think multiple moves ahead, while defenders must cover all possible angles."

- James Scott

CHAPTER

7

Conclusions and Future Directions

In this thesis, several contributions have been made to detect and mitigate DDoS, BashLite, Mirai botnet, LrDDoS, MrDDoS, Rank attacks, and Sinkhole attacks within IoT ecosystems. This section emphasises the most important research contributions and their impacts. Additionally, we identify future research directions that include both immediate extensions to the work carried out in this thesis and emerging trends in the IoT field that will remain relevant in the coming years.

7.1 Summary of Contributions

In the first contribution, the proposed security solution against DDoS attacks encompasses notorious threats such as BashLite and Mirai botnets, which pose significant risks to IoT ecosystems. Our first contribution focused on investigating a novel security solution for identifying and choosing the best characteristics from network traffic data. Proposed security solutions improve the accuracy of attack detection methods by utilizing the power of ML strategies to identify attacks with unprecedented efficacy. Selecting relevant hyperparameters for training models like the Naive Bayes Classifier and the NOC-SVM has helped identify botnet attacks. Our proposed method has been rigorously evaluated on numerous datasets, including CICIDS2017, Kitsuni, BoT-IoT, IScX, KDD99 cup, N-BaIoT, and an in-house dataset. Throughout the evaluation, we conducted exhaustive comparisons against existing state-of-the-art approaches and showed comparable results.

In the second contribution, we proposed security solutions introducing a distributed, lightweight, and energy-efficient PIA paradigm. The various PIAs use PFC and TVM to analyse network packets within the IoT network. The proposed security approach effectively detects LrDDoS attacks, showing proficiency in analysing network traffic originating from authentic and compromised nodes. The internal and external network traffic of the IoT ecosystem is significantly covered by the proposed security solution. Through careful analysis, it can accurately find and stop LrDDoS attacks while reducing both FNR and FPR. However, while our suggested security solution has many advantages, one potential disadvantage is that it is based on simulated attack scenarios, which may not fully replicate the complexity and diversity of real-world LrDDoS attacks. Despite these issues, the proposed LrDDoS detection and mitigation security solution is noteworthy, providing a foundation for future advancements in enhancing IoT security.

In our third contribution, we provide a complete approach to detecting and mitigating MRDDoS attacks, which offers a number of significant advantages. First, this work provides a comprehensive approach that can identify and mitigate both high-rate and low-rate DDoS attacks, whereas earlier studies only handled these two types of attacks independently. In addition, the robustness of the Intrusion Detection System (IDS) is improved through the incorporation of Wasserstein Generative Adversarial Network (WGAN) approaches. The novel training method, which combines WGAN-generated artificial flows with public and in-house-generated data, significantly reduces data set distribution bias, thereby improving the IDS's accuracy. The proposal also includes a transparent and efficient IDS placement mechanism that optimizes IDS node selection to strike a balance between energy overhead and coverage. This optimization is formulated as the weighted minimal vertex cover problem of a K-uniform hypergraph, and approximate solutions are provided. Extensive experiments on the Contiki and FIT IoT-LAB testbeds show that the proposed strategy outperforms existing solutions. Additionally, it outperforms current benchmark protocols in terms of attack detection effectiveness while minimizing energy consumption. This multifaceted strategy provides a reliable and resource-conserving response to MRDDoS attacks on IoT networks.

In our fourth contribution, we present an exhaustive method for identifying and mitigating multiple-mix attacks, which offers a number of substantial advantages. The RENO

model (Roving lightwEight shadow hoNeypOt) that we propose introduces a lightweight and adaptable method for detecting multiple-mix attacks in IoT networks. RENO achieves this by extensively collecting data on attackers' activities, thereby enhancing the detection rate of multiple-mix attacks. Furthermore, the lightweight shadow honeypot can be easily relocated to another node or device within the IoT network, increasing its likelihood of being targeted. We use Markov chain analysis (MCA) to identify the most likely node or device for an attack based on the existing IoT network profile. The RENO model's applicability and efficacy are confirmed by the MCA modeling procedure. We conducted a performance comparison of the RENO solution versus other relevant approaches in order to showcase its benefits. This evaluation shows the possible advantages of the RENO model by considering things like throughput, energy consumption, memory usage, false-positive and false-negative rates, accuracy, and attack detection time.

7.2 Future Research Directions

The contributions in this thesis can potentially be used as a foundation for future work, particularly in studies aimed at establishing more effective and lightweight security methods for IoT ecosystems. Despite the fact that the security mechanisms implemented in this document have demonstrated their ability to effectively overcome the limitations of common research conventions, as discussed previously, some limitations remain. These are some of the new paths that this work will take in the future.

7.2.1 Exploring the Potential of Inter-Fog Resource Sharing

In our thesis, we present four key contributions aimed at enhancing security within the Internet of Things (IoT) ecosystem. First, we employ machine learning (ML) techniques to detect Distributed Denial of Service (DDoS) attacks. Our approach involves distributing the detection module across resource-rich IoT nodes at the edge, capitalizing on the advantages offered by existing IoT infrastructure. However, we acknowledge that our solution does not explore Fog Computing or Inter-Fog Resource Sharing methodologies, leaving room for future investigation in this area.

The concept of inter-fog resource sharing plays a crucial role in optimizing the efficiency and performance of Internet of Things (IoT) networks, particularly in fog computing

environments. In the context of IoT, fog computing represents an edge computing paradigm where data processing occurs closer to IoT devices, reducing latency and enhancing real-time decision-making. This paragraph highlights the importance of further exploring inter-fog resource sharing within IoT networks and its potential impact.

- **Current Challenges in Fog Computing:** - Fog computing in IoT networks aims to process data locally or at the network edge to minimize latency and reduce the load on centralized cloud resources. However, heavy workloads or resource constraints within a fog layer can lead to a bottleneck, forcing requests to be forwarded to the cloud. This results in increased latency and potential resource underutilization.

- **Resource Sharing Benefits:** - Inter-fog resource sharing addresses these challenges by enabling neighboring fog layers to collaborate and share resources. Resources can encompass computing power, storage, and network bandwidth. Sharing resources helps distribute the processing load more efficiently, reducing the need to transfer requests to the distant cloud.

- **Prevention of Cloud Overload:** - Through inter-fog sharing, fog nodes can collaborate to process requests from nearby IoT devices, preventing unnecessary data transfers to the cloud. This not only reduces cloud overload but also minimizes data transmission costs and latency, making IoT applications more responsive.

- **Enhanced Scalability and Reliability:** - Resource sharing fosters scalability within the fog layer, allowing it to accommodate an increasing number of IoT devices and applications. It also improves the fault tolerance and reliability of the network, as fog nodes can compensate for failures in neighboring nodes.

- **Resource Allocation Algorithms:** - Developing efficient resource allocation algorithms is a key research direction in this area. These algorithms should consider factors such as workload, proximity of fog nodes to IoT devices, and available resources to make informed decisions about resource sharing.

- **Security and Privacy Considerations:** - While resource sharing enhances network efficiency, it also introduces security and privacy concerns. Future research must

address these issues, ensuring that shared resources are used securely and do not compromise sensitive IoT data.

- **Network Orchestration and Management:** - Implementing a robust network orchestration and management framework is essential to facilitate seamless resource sharing. This involves dynamically allocating and deallocating resources as per the workload and network conditions.

In conclusion, inter-fog resource sharing presents a promising avenue for optimizing IoT networks by distributing processing tasks and preventing unnecessary data transfers to the cloud. Future research should focus on resource allocation algorithms, security measures, and effective network management to harness the full potential of this approach, ultimately enhancing the efficiency and reliability of IoT applications.

7.2.2 IoT and Blockchain Integration for IoT Networks

In our thesis, contributions 2, 3, and 4 involve security measures and their corresponding mitigation strategies, all of which are executed in a distributed manner. However, throughout the distributed execution process, we have not implemented checks for device integrity. This omission raises concerns about the functionality of devices executing our solution. If a malicious node participates in the execution process, it could compromise the integrity of our solution and adversely affect its outcomes. To address this challenge effectively, integrating blockchain technology with our solution emerges as a promising approach. By enabling blockchain within our solution framework, we can enhance its resilience against malicious nodes and ensure the integrity of the execution process. Notably, we have not explored this method in our current solution, presenting a significant avenue for future research in our thesis work.

Research and development efforts in the intersection of IoT and blockchain are focusing on achieving the following goals:

- **Scalable Permissioned Networks:** Innovations are aimed at enabling permissioned IoT networks to scale efficiently by designing consensus mechanisms that can handle a large number of nodes. Sharding, sidechains, and off-chain solutions are explored to improve scalability.

- **Enhanced Throughput for Permissionless Networks:** In the context of permissionless IoT networks, researchers are working on consensus algorithms, like proof-of-stake (PoS) and delegated proof-of-stake (DPoS), to increase throughput while maintaining security and decentralization.

- **Hybrid Architectures:** Combining the strengths of permissioned and permissionless blockchains, hybrid architectures are emerging as a potential solution. These architectures aim to balance scalability and throughput requirements in diverse IoT scenarios.

- **IoT-Specific Use Cases:** Tailoring blockchain solutions for specific IoT use cases, such as supply chain management, smart cities, and healthcare, is an ongoing focus. These customizations aim to optimize blockchain architecture to meet the unique demands of different IoT applications.

In summary, the integration of blockchain technology into IoT networks is a dynamic field that continues to evolve. Addressing scalability and throughput challenges is essential to unlock the full potential of blockchain in the IoT domain. Ongoing research and innovations in consensus algorithms, network architectures, and use case-specific deployments are crucial steps toward achieving these objectives and harnessing the benefits of blockchain technology in the IoT ecosystem.

7.2.3 Threat Intelligence Sharing in IoT ecosystem

In our thesis contributions, we have leveraged intelligent and real-time testing methodologies to assess the presence of attacks. Additionally, we have integrated machine learning (ML) techniques by training ML models offline using diverse IoT datasets. However, it is important to note that our current approach lacks the utilization of intelligent sharing mechanisms in real-time within the IoT ecosystem. Thus, one potential avenue for future research within our thesis direction involves exploring the integration of Threat Intelligence Sharing in IoT ecosystems.

Threat intelligence sharing in the IoT (Internet of Things) ecosystem is a critical aspect of enhancing security in the rapidly evolving landscape of connected devices. As a future direction, it involves the collection, analysis, and dissemination of information related

to IoT-specific threats and vulnerabilities among various stakeholders, including device manufacturers, service providers, government agencies, and cybersecurity organizations. Here are some key aspects of how threat intelligence sharing in the IoT ecosystem may develop in the future:

- IoT-Specific Threat Feeds: Specialized threat intelligence feeds dedicated to IoT threats will become more prevalent. These feeds will include information on emerging vulnerabilities, malware targeting IoT devices, and attack patterns specific to IoT ecosystems.
- Collaborative Platforms: The development of collaborative platforms and consortiums where industry players, researchers, and government agencies can share threat intelligence in a trusted and secure manner. These platforms will facilitate information sharing among stakeholders to address common threats.
- Standardization: The establishment of standardized formats and protocols for sharing threat intelligence in the IoT space. Standardization ensures that threat data can be easily exchanged and understood across different organizations and systems.
- Privacy Considerations: Future threat intelligence sharing efforts will need to strike a balance between sharing critical information for security purposes while respecting user privacy and data protection regulations. Anonymization and data minimization techniques will be employed to mitigate privacy concerns.
- Real-Time Sharing: The move toward real-time threat intelligence sharing, allowing organizations to receive timely updates about emerging threats and vulnerabilities. This enables faster response and mitigation actions.
- Automated Threat Detection and Response: Integration of threat intelligence feeds with IoT security systems for automated threat detection and response. When a new threat is identified, devices can automatically adapt their security measures or notify administrators.
- Machine Learning and AI: Implementation of machine learning and artificial intelligence for analyzing threat intelligence data. These technologies can help identify patterns and anomalies in the massive amounts of data generated by IoT devices.

- Regulatory Incentives: Government agencies may introduce incentives or regulations that encourage organizations to share threat intelligence. This can help create a culture of cooperation in addressing IoT security threats.
- Vulnerability Prioritization: Development of tools and methodologies for prioritizing vulnerabilities based on their potential impact on IoT ecosystems. This helps organizations focus their resources on addressing the most critical threats.
- Global Collaboration: International collaboration on threat intelligence sharing to address IoT security threats that transcend borders. Cybersecurity threats often have a global reach, and cross-border cooperation is essential.
- Incident Response Coordination: Enhanced coordination among organizations during IoT security incidents. This includes predefined incident response plans and communication protocols for sharing threat intelligence during and after an incident.
- Sector-Specific Sharing: Tailoring threat intelligence sharing efforts to specific IoT sectors, such as healthcare, energy, or transportation, to address industry-specific threats and vulnerabilities.

In summary, the future direction of threat intelligence sharing in the IoT ecosystem involves creating a collaborative and standardized framework for sharing information about emerging threats and vulnerabilities. This proactive approach is essential to stay ahead of evolving IoT security challenges and to protect the integrity and functionality of IoT devices and systems.

7.2.4 Time Domain to Frequency Domain Analysis for IoT Security:

In our thesis, each contribution relies on time-domain traffic analysis. While some contributions employ ML-based security solutions, they predominantly utilize time-domain traffic datasets for both training and testing purposes. However, despite these efforts, our solutions encounter challenges in detecting concealed attacks such as LrDDoS attacks. Recognizing the significance of frequency-domain analysis in identifying such attacks, we highlight the importance of exploring this domain further. Detecting and mitigating threats within the IoT ecosystem through frequency-domain analysis presents a promising avenue for enhancing security measures.

The exploration of transitioning from the time domain to the frequency domain in the context of IoT security represents a promising future research direction. This approach involves analyzing IoT data and signals in the frequency domain, which can provide unique insights and solutions to enhance IoT security. Here's an explanation of why this is an important and promising avenue for research:

- Signal Analysis for Anomaly Detection: In the frequency domain, IoT signals and data can be decomposed into their constituent frequency components. Analyzing these components can help in detecting unusual patterns or anomalies that might signify security breaches or unauthorized activities. Researchers can develop advanced algorithms and techniques for more effective anomaly detection.

- Frequency-Based Cryptography: Traditional cryptographic techniques are often designed for the time domain. Exploring cryptographic methods that leverage the frequency domain can lead to innovative approaches for securing IoT communications. Frequency-based cryptography may offer increased resistance to certain types of attacks.

- Resilience Against Jamming: Frequency domain analysis can help in identifying and mitigating the effects of signal jamming and interference, which are common attack vectors in IoT networks. Research in this area can lead to improved resilience against jamming attacks.

- Energy-Efficient Security: Exploring the frequency domain can lead to energy-efficient security solutions for resource-constrained IoT devices. Frequency-based security mechanisms may require less computational power and energy consumption compared to traditional methods.

- Cross-Layer Security: Integrating frequency domain analysis into a cross-layer security approach can enhance the overall security posture of IoT systems. Researchers can investigate how frequency-related insights can be applied across different protocol layers.

- Machine Learning and Frequency Domain: Combining machine learning with frequency domain analysis can lead to intelligent security systems that adapt to evolving threats.

Machine learning algorithms can be trained to recognize abnormal frequency patterns associated with attacks.

In summary, transitioning from the time domain to the frequency domain represents an exciting future research direction for IoT security. It offers opportunities to develop novel security techniques, cryptography methods, and anomaly detection systems that can better protect IoT devices and networks from a wide range of threats.

Bibliography

[1] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[2] M. H. Bhuyan and E. Elmroth, "Multi-scale Low-Rate DDoS Attack Detection Using the Generalized Total Variation Metric," in *ICMLA*. IEEE, 2018, pp. 1040–1047.

[3] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew DDoS," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1069–1083, 2014.

[4] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis, "Detecting targeted attacks using shadow honeypots," 2005.

[5] P. Bhale, S. Biswas, and S. Nandi, "An Adaptive and Lightweight Solution to Detect Mixed Rate IP Spoofed DDoS Attack in IoT Ecosystem," in *India Council International Conference (INDICON)*. IEEE, 2018, pp. 1–6.

[6] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[8] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[9] M. S. Jalali, J. P. Kaiser, M. Siegel, and S. Madnick, "The internet of things promises new benefits and risks: a systematic analysis of adoption dynamics of IoT products," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 39–48, 2019.

[10] T. Simon, "Chapter seven: Critical infrastructure and the internet of things," *Cyber security in a volatile world*, vol. 93, 2017.

[11] J. M. Yusta, G. J. Correa, and R. Lacal-Arántegui, "Methodologies and applications for critical infrastructure protection: State-of-the-art," *Energy Policy*, vol. 39, no. 10, pp. 6100–6119, 2011, sustainability of biofuels. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421511005337

[12] F. Cirillo, D. Gómez, L. Diez, I. E. Maestro, T. B. J. Gilbert, and R. Akhavan, "Smart city IoT services creation through large-scale collaboration," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5267–5275, 2020.

[13] A. C. Şerban and M. D. Lytras, "Artificial intelligence for smart renewable energy sector in europe—smart energy infrastructures for next generation smart cities," *IEEE access*, vol. 8, pp. 77 364–77 377, 2020.

[14] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, "On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[15] N.-N. Dao, "Internet of wearable things: Advancements and benefits from 6G technologies," *Future Generation Computer Systems*, 2022.

[16] Z. Qadir, K. N. Le, N. Saeed, and H. S. Munawar, "Towards 6G Internet of Things: Recent advances, use cases, and open challenges," *ICT Express*, vol. 9, no. 3, pp. 296–312, 2023.

[17] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[18] W. Liao, S. Salinas, M. Li, P. Li, and K. A. Loparo, "Cascading Failure Attacks in the Power System: A Stochastic Game Perspective," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2247–2259, 2017.

[19] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "False data injection threats in active distribution systems: A comprehensive survey," *Future Generation Computer Systems*, 2022.

[20] S. Khanam, I. B. Ahmedy, M. Y. Idna Idris, M. H. Jaward, and A. Q. Bin Md Sabri, "A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things," *IEEE Access*, vol. 8, pp. 219 709–219 743, 2020.

[21] Statista, "Number of Cyber Attacks 2016-2022," https://www.statista.com/statistics/1201177/india-number-of-cyber-attacks/, 2022, accessed: March 24, 2024.

[22] R. H. Murofushi and J. J. Tavares, "Towards fourth industrial revolution impact: smart product based on RFID technology," *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 2, pp. 51–56, 2017.

[23] S. Lim, O. Kwon, and D. H. Lee, "Technology convergence in the Internet of Things (IoT) startup ecosystem: A network analysis," *Telematics and Informatics*, vol. 35, no. 7, pp. 1887–1899, 2018.

[24] J. Granjal, E. Monteiro, and J. Sá 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.

[25] R. Nath and H. V. Nath, "Critical analysis of the layered and systematic approaches for understanding IoT security threats and challenges," *Computers and Electrical Engineering*, vol. 100, p. 107997, 2022.

[26] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and Opportunities in Securing the Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2021.

[27] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on internet of things security: Requirements, challenges, and solutions," *Internet of Things*, vol. 14, p. 100129, 2021.

[28] A. Mukherjee, "Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015.

[29] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and Applications of Physical Layer Security Techniques for Confidentiality: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1773–1828, 2019.

[30] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[31] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[32] J. Manan, A. Ahmed, I. Ullah, L. Merghem-Boulahia, and D. Gaïti, "Distributed intrusion detection scheme for next generation networks," *Journal of Network and Computer Applications*, vol. 147, p. 102422, 2019.

[33] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," *Future Generation Computer Systems*, vol. 133, pp. 95–113, 2022.

[34] B. Min and V. Varadharajan, "Design and Evaluation of Feature Distributed Malware Attacks against the Internet of Things (IoT)," in *2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2015, pp. 80–89.

[35] A. O. Bang, U. P. Rao, A. Visconti, A. Brighente, and M. Conti, "An iot inventory before deployment: a survey on iot protocols, communication technologies, vulnerabilities, attacks, and future research directions," *Computers & Security*, p. 102914, 2022.

[36] R. Lohiya and A. Thakkar, "Application domains, evaluation data sets, and research challenges of IoT: A Systematic Review," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8774–8798, 2020.

[37] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[38] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE internet of things journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[39] K. Pretz, "Exploring the Impact of the Internet of Things: A new IEEE group is taking on the quest to connect everything," *The Institute*, 2013.

[40] K. Karunanithy and B. Velusamy, "Cluster-tree based energy efficient data gathering protocol for industrial automation using WSNs and IoT," *Journal of Industrial Information Integration*, vol. 19, p. 100156, 2020.

[41] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[42] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities," *Ieee Access*, vol. 5, pp. 26 521–26 544, 2017.

[43] Y. B. Zikria, H. Yu, M. K. Afzal, M. H. Rehmani, and O. Hahm, "Internet of things (IoT): Operating system, applications and protocols design, and validation techniques," pp. 699–706, 2018.

[44] B. Patel and P. Shah, "Operating system support, protocol stack with key concerns and testbed facilities for IoT: A case study perspective," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5420–5434, 2022.

[45] F. Javed, M. K. Afzal, M. Sharif, and B.-S. Kim, "Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2062–2100, 2018.

[46] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *International conference on local computer networks*, 2004, pp. 455–462.

[47] D. Willmann, "Contiki-A Memory-Efficient Operating System for Embedded Smart Objects," *2009*, 2009.

[48] A. Kurniawan, "Practical Contiki-NG," *Pract. Contiki-NG*, 2018.

[49] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, p. 101089, 2022.

[50] J. G. Ko, N. Tsiftes, A. Dunkels, and A. Terzis, "Pragmatic low-power interoperability: ContikiMAC vs TinyOS LPL," in *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2012, pp. 94–96.

[51] "TinyOS: An OS for Embedded, Wireless Devices," https://github.com/tinyos/tinyos-main, March 2021, accessed: 2021-03-13.

[52] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in *2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 79–80.

[53] R. Barry *et al.*, "FreeRTOS," *Internet, Oct*, vol. 4, 2008.

[54] T. B. Chandra, P. Verma, and A. Dwivedi, "Operating systems for internet of things: A comparative study," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–6.

[55] "An introduction to Arm Mbed OS 6," https://os.mbed.com/docs/mbed-os/v6.16/introduction/index.html, May 2023, accessed: 2023-05-13.

[56] A. Eswaran, A. Rowe, and R. Rajkumar, "Nano-rk: an energy-aware resource-centric rtos for sensor networks," in *26th IEEE International Real-Time Systems Symposium (RTSS'05)*. IEEE, 2005, pp. 10–pp.

[57] K. Baynes, C. Collins, E. Fiterman, B. Ganesh, P. Kohout, C. Smit, T. Zhang, and B. Jacob, "The performance and energy consumption of embedded real-time operating systems," *IEEE transactions on computers*, vol. 52, no. 11, pp. 1454–1469, 2003.

[58] A. Diaz and P. Sanchez, "Simulation of attacks for security in wireless sensor network," *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[59] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

[60] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, 2010.

[61] J. Fernandes, M. Nati, N. Loumis, S. Nikoletseas, T. P. Raptis, S. Krco, A. Rankov, S. Jokic, C. M. Angelopoulos, and S. Ziegler, "IoT Lab: Towards co-design and IoT solution testing using the crowd," in *2015 International Conference on Recent Advances in Internet of Things (RIoT)*. IEEE, 2015, pp. 1–6.

[62] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, and T. Noel, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *World Forum on Internet of Things (WF-IoT)*, 2015, pp. 459–464.

[63] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer, "WISEBED: an open large-scale wireless sensor network testbed," in *Sensor Applications, Experimentation, and Logistics: First International Conference, SENSAPPEAL 2009, Athens, Greece, September 25, 2009, Revised Selected Papers 1*. Springer, 2010, pp. 68–87.

[64] P. Demeester, P. Van Daele, T. Wauters, and H. Hrasnica, "Fed4FIRE–The Largest Federation of Testbeds in Europe," in *Building the future internet through FIRE*. River Publishers, 2022, pp. 87–109.

[65] S. Ziegler, L. Baron, B. Vermeulen, and S. Fdida, "F-interop–online platform of interoperability and performance tests for the internet of things," in *Building the Future Internet through FIRE*. River Publishers, 2022, pp. 603–611.

[66] K. Lorincz, D. J. Malan, T. R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: challenges and opportunities," *IEEE pervasive Computing*, vol. 3, no. 4, pp. 16–23, 2004.

[67] R. K. Shrivastava, S. P. Singh, M. K. Hasan, S. Islam, S. Abdullah, A. H. M. Aman *et al.*, "Securing Internet of Things devices against code tampering attacks using Return Oriented Programming," *Computer Communications*, vol. 193, pp. 38–46, 2022.

[68] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE network*, vol. 20, no. 3, pp. 41–47, 2006.

[69] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "MIS: Malicious nodes identification scheme in network-coding-based peer-to-peer streaming," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5.

[70] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN fragmentation attacks and mitigation mechanisms," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, 2013, pp. 55–66.

[71] M. Hossain, Y. Karim, and R. Hasan, "Secupan: A security scheme to mitigate fragmentation-based network attacks in 6lowpan," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 2018, pp. 307–318.

[72] Y. Benslimane, K. Benahmed, and H. Benslimane, "Security mechanisms for 6LoWPAN network in context of internet of things: A Survey," in *Renewable Energy for Smart and Sustainable Cities: Artificial Intelligence in Renewable Energetic Systems 2*. Springer, 2019, pp. 49–69.

[73] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, 2008.

[74] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59 353–59 377, 2021.

[75] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "Correlation-based traffic analysis attacks on anonymity networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 954–967, 2009.

[76] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Transactions on emerging topics in computing*, vol. 5, no. 4, pp. 586–602, 2016.

[77] M. J. Sebastian Garcia, Agustin Parmisano, "IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]," 2020. [Online]. Available: http://doi.org/10.5281/zenodo.4743746

[78] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, R. Budiarto *et al.*, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132 911–132 921, 2020.

[79] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[80] T. D. Yisroel Mirsky, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," 2018. [Online]. Available: http://arxiv.org/abs/1802.09089

[81] N. Moustafa, "ToN_IoT datasets," 2019. [Online]. Available: https://dx.doi.org/10.21227/fesz-dm97

[82] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.

[83] UNB. (Year of Access) ISCX Datasets. [Online]. Available: http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html

[84] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, 2020.

[85] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[86] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[87] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Systems with Applications*, vol. 167, p. 114170, 2021.

[88] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[89] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[90] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial intelligence review*, vol. 53, pp. 5455–5516, 2020.

[91] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[92] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[93] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[94] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[95] F. Erlacher and F. Dressler, "FIXIDS: A high-speed signature-based flow intrusion detection system," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–8.

[96] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *computers & security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[97] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[98] T. Escamilla, *Intrusion detection: network security beyond the firewall.* John Wiley & Sons, Inc., 1998.

[99] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," *Electronics*, vol. 8, no. 11, p. 1210, 2019.

[100] D.-W. Huang, F. Luo, J. Bi, and M. Sun, "An efficient hybrid IDS deployment architecture for multi-hop clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2688–2702, 2022.

[101] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, "A review of security standards and frameworks for IoT-based smart environments," *IEEE Access*, vol. 9, pp. 121 975–121 995, 2021.

[102] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the internet of things," *IEEE communications surveys & tutorials*, vol. 21, no. 2, pp. 1636–1675, 2018.

[103] The Internet of Things Reference Model. (2014) The internet of things reference model. [Online] Available. [Online]. Available: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf

[104] J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "New directions in automated traffic analysis," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3366–3383.

[105] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *Journal of network and computer applications*, vol. 149, p. 102481, 2020.

[106] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[107] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[108] A. Barua, M. A. Al Alamin, M. S. Hossain, and E. Hossain, "Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 251–281, 2022.

[109] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of Things (IoT): A review," *Journal of Computer Networks and Communications*, vol. 2019, pp. 1–14, 2019.

[110] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[111] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," *Pattern Recognition Letters*, vol. 51, pp. 1–7, 2015.

[112] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," *Information*, vol. 7, no. 2, p. 25, 2016.

[113] C. Cervantes, D. Poplade, and Nogueira, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *IEEE/IFIP Int. Symp. Intg Netw. Manag.*, 2015, pp. 606–611.

[114] M. Krzysztoń and M. Marks, "Simulation of watchdog placement for cooperative anomaly detection in bluetooth mesh intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102041, 2020.

[115] T. Sui, X. Tao, S. Xia, H. Chen, H. Wu, X. Zhang, and K. Chen, "A real-time hidden anomaly detection of correlated data in wireless networks," *IEEE Access*, vol. 8, pp. 60 990–60 999, 2020.

[116] N. Kandhoul and S. K. Dhurandher, "Deep Q learning based secure routing approach for OppIoT networks," *Internet of Things*, vol. 20, p. 100597, 2022.

[117] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani, "Intelligent intrusion detection in low-power IoTs," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 4, pp. 1–25, 2016.

[118] 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. 24 694–24 705, 2018.

[119] "IoT market growth," [accessed: 11-January-2021.]. [Online]. Available: https://www.forbes.com/sites/louiscolumbus

[120] N. Ferguson and B. Schneier, "A cryptographic evaluation of IPsec," 1999.

[121] P. Li, J. Su, and X. Wang, "ITLS/IDTLS: Lightweight end-to-end security protocol for IoT through minimal latency," in *Proceedings of the ACM SIGCOMM*, 2019, pp. 166–168.

[122] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15. 4," *Sensor network operations*, vol. 4, pp. 218–237, 2006.

[123] G. Nebbione and Calzarossa, "Security of IoT application layer protocols: Challenges & findings," *Future Internet*, vol. 12, no. 3, p. 5, 2020.

[124] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network IDS," in *ISNCC*. IEEE, 2016, pp. 1–6.

[125] "Instant Contiki," [accessed: 11-January-2021.]. [Online]. Available: http://www.contiki-os.org/start.html

[126] "Cooja Simulator," [accessed: 11-January-2021.]. [Online]. Available: http://anrg.usc.edu/contiki/index.php/CoojaSimulator

[127] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[128] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15. 4e in a nutshell: Survey and performance evaluation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1989–2010, 2018.

[129] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15. 4e (6TiSCH) configuration," *Internet Engineering Task Force RFC series*, no. RFC8180, 2017.

[130] M. Antonakakis, T. April, and Bailey, "Understanding the Mirai Botnet," in *26th USENIX security symposium*, 2017, pp. 1093–1110.

[131] M. M. Alani, "BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning," *Comput. Commun.*, vol. 193, pp. 53–62, 2022.

[132] M. K. Putchala, "Deep learning approach for IDS in the IoT network using gated recurrent neural networks (GRU)," 2017.

[133] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," *IJCS*, vol. 31, no. 5, p. e3497, 2018.

[134] A. Guerra-Manzanares, H. Bahsi, and S. Nõmm, "Hybrid feature selection models for machine learning based botnet detection in IoT networks," in *Conference on Cyberworlds*. IEEE, 2019, pp. 324–327.

[135] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: a malicious bot-iot traffic detection method in iot network using machine learning techniques," *IEEE Internet of Things Journal*, 2020.

[136] 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. 24 694–24 705, 2018.

[137] H. Chen, C. Meng, Z. Shan, Z. Fu, and B. K. Bhargava, "A novel Low-rate Denial of Service attack detection approach in ZigBee wireless sensor network by combining Hilbert-Huang Transformation and Trust Evaluation," *IEEE Access*, vol. 7, pp. 32 853–32 866, 2019.

[138] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, 2020.

[139] I. Rish *et al.*, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[140] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[141] "Tmote-Sky," [accessed: 11-January-2021.]. [Online]. Available: https://wirelesssensornetworks.weebly.com/blog/tmote-sky

[142] C. Gomez, J. Paradells, C. Bormann, and J. Crowcroft, "From 6LoWPAN to 6Lo: Expanding the universe of IPv6-supported technologies for the internet of things," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 148–155, 2017.

[143] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[144] N. Agrawal and S. Tapaswi, "A Lightweight Approach to Detect the Low/High Rate IP Spoofed Cloud DDoS Attacks," in *SC2*. IEEE, 2017, pp. 118–123.

[145] V. Adat, A. Dahiya, and B. Gupta, "Economic incentive based solution against DDoS attacks for IoT customers," in *ICCE*. IEEE, 2018, pp. 1–5.

[146] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, "A Review of Security in Internet of Things," *Wireless Personal Communications*, pp. 1–20, 2019.

[147] A. M. da Silva Cardoso, R. F. Lopes, A. S. Teles, and F. B. V. Magalhães, "Real-time DDoS detection based on complex event processing for IoT," in *IoTDI*. IEEE, 2018, pp. 273–274.

[148] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks and counter strategies," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 4, pp. 683–696, 2006.

[149] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against DDoS flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[150] P. Du and S. Abe, "IP packet size entropy-based scheme for detection of DoS/DDoS attacks," *IEICE TRANSACTIONS on Information and Systems*, vol. 91, no. 5, pp. 1274–1281, 2008.

[151] H. Chen, C. Meng, Z. Shan, Z. Fu, and B. K. Bhargava, "A Novel Low-Rate Denial of Service Attack Detection Approach in ZigBee Wireless Sensor Network by Combining Hilbert-Huang Transformation and Trust Evaluation," *IEEE Access*, vol. 7, pp. 32 853–32 866, 2019.

[152] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[153] H. Xu, W. Zeng, X. Zeng, and G. G. Yen, "An evolutionary algorithm based on Minkowski distance for many-objective optimization," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–12, 2018.

[154] H. Rahimian, G. Bayraksan, and T. Homem-de Mello, "Identifying effective scenarios in distributionally robust stochastic programs with total variation distance," *Mathematical Programming*, pp. 1–38, 2018.

[155] R. Hunt and S. Zeadally, "Network forensics: an analysis of techniques, tools, and trends," *Computer*, vol. 45, no. 12, pp. 36–43, 2012.

[156] CAIDA, "The distributed denial of service attack (2007) dataset," http://www.caida.org/data/passive/ddos-20070804-dataset.xml, accessed 2019-07-02.

[157] LLDOS, "MIT Lincoln Laboratory Datasets, MIT LLS-DDOS-0.2.2," https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets, accessed 2019-07-02.

[158] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th annual IEEE international conference on local computer networks*. IEEE, 2004, pp. 455–462.

[159] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2019.

[160] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, 2017.

[161] H. Li, J. Zhu, Q. Wang, T. Zhou, H. Qiu, and H. Li, "LAAEM: A method to enhance LDoS attack," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 708–711, 2016.

[162] H. Griffioen, K. Oosthoek, P. van der Knaap, and C. Doerr, "Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2021, pp. 940–954.

[163] A. Thakkar and R. Lohiya, "A review on ML and DL perspectives of IDS for IoT: recent updates, security issues, and challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3211–3243, 2021.

[164] P. Kasinathan, G. Costamagna, C. Pastrone, and M. A. Spirito, "DEMO: An IDS Framework for Internet of Things Empowered by 6LoWPAN," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2013, pp. 1337–1340.

[165] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things," *Int. J. Distrib. Sens. Netw.*, vol. 9, no. 8, p. 794326, 2013.

[166] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013.

[167] J. P. Amaral, L. M. Oliveira, and Rodrigues, "Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks," in *IEEE Int. Conf. Commun. (ICC)*, 2014, pp. 1796–1801.

[168] M. Al Qurashi, C. M. Angelopoulos, and V. Katos, "An Architecture for Resilient Intrusion Detection in IoT Networks," in *IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[169] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Future Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019.

[170] N. Yadav, L. Truong, and E. Troja, "Machine Learning Architecture for Signature-based IoT Intrusion Detection in Smart Energy Grids," in *IEEE Mediterr. Electrotech. Conf. (MELECON)*, 2022, pp. 671–676.

[171] J. A. Perez-Diaz, I. Valdovinos, and Amezcua, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155 859–155 872, 2020.

[172] X. Liu, J. Ren, H. He, Q. Wang, and C. Song, "Low-rate DDoS attacks detection method using data compression and behavior divergence measurement," *Computers & Security*, vol. 100, p. 102107, 2021.

[173] A. Tabassum, A. Erbad, A. Mohamed, and M. Guizani, "Privacy-preserving distributed IDS using incremental learning for IoT health systems," *IEEE Access*, vol. 9, pp. 14 271–14 283, 2021.

[174] F. Hussain, S. G. Abbas, I. M. Pires, S. Tanveer, U. U. Fayyaz, and N. M. Garcia, "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," *IEEE Access*, vol. 9, pp. 163 412–163 430, 2021.

[175] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280–4290, 2022.

[176] M. Saharkhizan, A. Azmoodeh, and Dehghantanha, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8852–8859, 2020.

[177] Y. Li and Y. Lu, "LSTM-BA: DDoS detection approach combining LSTM and Bayes," in *IEEE Int. Conf. Adv. Cloud and Big Data (CBD)*, 2019, pp. 180–185.

[178] N. Garcia, T. Alcaniz, A. González-Vidal, and J. Bernabe, "Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial Intelligence," *J. Netw. Comput. Appl.*, vol. 173, p. 102871, 2021.

[179] B. Liu, D. Tang, Y. Yan, and Z. Zheng, "TS-SVM: Detect LDoS Attack in SDN Based on Two-step Self-adjusting SVM," in *IEEE 20th Int. Conf. Trust, Sec. Pri. Comp. Commun. (TrustCom)*, 2021, pp. 678–685.

[180] R. H. Jhaveri, "MR-AODV: A Solution to Mitigate Blackhole and Grayhole Attacks in AODV Based MANETs," in *3rd Int. Conf. Advan. Comput. and Commun. Tech. (ACCT)*, 2013, pp. 254–260.

[181] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Chapter 35.4, Introduction to algorithms (Third Edition)," pp. 1123–1127, 2009.

[182] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3469–3477, 2020.

[183] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Adv Neural Inf Process Syst*, vol. 27, 2014.

[184] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, p. 100004, 2021.

[185] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning.* PMLR, 2017, pp. 214–223.

[186] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers & Security*, vol. 82, pp. 156–172, 2019.

[187] N. Koroniotis and N. Moustafa, "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019.

[188] C. Sanders, *Practical Packet Analysis, 3E: Using Wireshark to Solve Real-World Network Problems.* No Starch Press, 2017.

[189] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using Shapley Additive Explanations," *Expert Systems with Applications*, vol. 186, p. 115736, 2021.

[190] N. Moustafa, B. Turnbull, and K. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[191] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3559–3570, 2020.

[192] S. Godard, "SYSSTAT utilities home page," *Information and code available at http://sebastien. godard. pagesperso-orange. fr/index. html*, 2015.

[193] A. Tirumala, "Iperf: The TCP/UDP bandwidth measurement tool," *http://dast. nlanr. net/Projects/Iperf/*, 1999.

[194] S. Balaji, K. Nathani, and R. Santhakumar, "IoT technology, applications and challenges: a contemporary survey," *Wireless personal communications*, vol. 108, pp. 363–388, 2019.

[195] S. Tweneboah-Koduah, K. E. Skouby, and R. Tadayoni, "Cyber security threats to IoT applications and service domains," *Wireless Personal Communications*, vol. 95, pp. 169–185, 2017.

[196] Synopsys. (2020) Medical Device Security Report. [Online]. Available: https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/medical-device-security-ponemon-synopsys.pdf

[197] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, p. e4946, 2020.

[198] C. Avasalcai, C. Tsigkanos, and S. Dustdar, "Resource management for latency-sensitive IoT applications with satisfiability," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2982–2993, 2021.

[199] S. Khanam, I. B. Ahmedy, M. Y. I. Idris, M. H. Jaward, and A. Q. B. M. Sabri, "A survey of security challenges, attacks taxonomy and advanced countermeasures in the internet of things," *IEEE access*, vol. 8, pp. 219 709–219 743, 2020.

[200] Z.-K. Zhang, M. C. Y. Cho, and S. Shieh, "Emerging security threats and countermeasures in IoT," in *Proceedings of the 10th ACM symposium on information, computer and communications security*, 2015, pp. 1–6.

[201] 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.

[202] Y. Otoum and A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things," *Journal of Network and Systems Management*, vol. 29, no. 3, pp. 1–26, 2021.

[203] N. A. Quynh and Y. Takefuji, "Towards an invisible honeypot monitoring system," in *Australasian Conference on Information Security and Privacy*. Springer, 2006, pp. 111–122.

[204] G. Mullen and L. Meany, "Assessment of buffer overflow based attacks on an IoT operating system," in *2019 Global IoT Summit (GIoTS)*. IEEE, 2019, pp. 1–6.

[205] B. Xu, W. Wang, Q. Hao, Z. Zhang, P. Du, T. Xia, H. Li, and X. Wang, "A security design for the detecting of buffer overflow attacks in iot device," *IEEE Access*, vol. 6, pp. 72 862–72 869, 2018.

[206] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, p. 100568, 2022.

[207] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[208] A. Yahyaoui, T. Abdellatif, S. Yangui, and R. Attia, "READ-IoT: Reliable event and anomaly detection framework for the Internet of Things," *IEEE Access*, vol. 9, pp. 24 168–24 186, 2021.

[209] S. Mouti, S. K. Shukla, S. Althubiti, M. A. Ahmed, F. Alenezi, and M. Arumugam, "Cyber Security Risk management with attack detection frameworks using multi connect variational auto-encoder with probabilistic Bayesian networks," *Computers and Electrical Engineering*, vol. 103, p. 108308, 2022.

[210] M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, "U-pot: A honeypot framework for upnp-based iot devices," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, pp. 1–8.

[211] S. Dowling, M. Schukat, and H. Melvin, "A ZigBee honeypot to assess IoT cyberattack behaviour," in *2017 28th Irish Signals and Systems Conference (ISSC)*. IEEE, 2017, pp. 1–6.

[212] N. Moustafa, M. Keshk, K.-K. R. Choo, T. Lynar, S. Camtepe, and M. Whitty, "DAD: a distributed anomaly detection system using ensemble one-class statistical learning in edge networks," *Future Generation Computer Systems*, vol. 118, pp. 240–251, 2021.

[213] H. T. Truong, B. P. Ta, Q. A. Le, D. M. Nguyen, C. T. Le, H. X. Nguyen, H. T. Do, H. T. Nguyen, and K. P. Tran, "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Computers in Industry*, vol. 140, p. 103692, 2022.

[214] R. Vishwakarma and A. K. Jain, "A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019, pp. 1019–1024.

[215] M. Wang, J. Santillan, and F. Kuipers, "Thingpot: an interactive internet-of-things honeypot," *arXiv preprint arXiv:1807.04114*, 2018.

[216] K. Prathapchandran and T. Janani, "A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST," *Computer Networks*, vol. 198, p. 108413, 2021.

[217] R. Sahay, G. Geethakumari, and B. Mitra, "Mitigating the worst parent attack in RPL based internet of things," *Cluster Computing*, pp. 1–18, 2022.

[218] P. Nandhini, S. Kuppuswami, S. Malliga, and R. DeviPriya, "A Lightweight Energy-Efficient Algorithm for mitigation and isolation of Internal Rank Attackers in RPL based Internet of Things," *Computer Networks*, p. 109391, 2022.

[219] B. Xu, W. Wang, Q. Hao, Z. Zhang, P. Du, T. Xia, H. Li, and X. Wang, "A security design for the detecting of buffer overflow attacks in IoT device," *IEEE Access*, vol. 6, pp. 72 862–72 869, 2018.

[220] F. A. Teixeira, F. M. Pereira, H.-C. Wong, J. M. Nogueira, and L. B. Oliveira, "SIoT: Securing Internet of Things through distributed systems analysis," *Future Generation Computer Systems*, vol. 92, pp. 1172–1186, 2019.

[221] M. Anirudh, S. A. Thileeban, and D. J. Nallathambi, "Use of honeypots for mitigating DoS attacks targeted on IoT networks," in *2017 International conference on computer, communication and signal processing (ICCCSP)*. IEEE, 2017, pp. 1–4.

[222] M. A. Salahuddin, V. Pourahmadi, H. A. Alameddine, M. F. Bari, and R. Boutaba, "Chronos: Ddos attack detection using time-based autoencoder," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 627–641, 2021.

[223] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," *Computers & Electrical Engineering*, vol. 98, p. 107716, 2022.

[224] J. Holsopple, S. J. Yang, and M. Sudit, "TANDI: Threat assessment of network data and information," in *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, vol. 6242. International Society for Optics and Photonics, 2006, p. 62420O.

[225] S. Nayak, N. Ahmed, and S. Misra, "Deep learning-based reliable routing attack detection mechanism for industrial Internet of Things," *Ad Hoc Networks*, vol. 123, p. 102661, 2021.

[226] M. Al Qurashi, C. M. Angelopoulos, and V. Katos, "An architecture for resilient intrusion detection in ad-hoc networks," *Journal of Information Security and Applications*, vol. 53, p. 102530, 2020.

[227] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280–4290, 2021.

List of Publications

Publications Related to Thesis Work:

Refereed Journals:

1. **Pradeepkumar Bhale**, Debanjan Roy Chowdhury, Santosh Biswas, and Sukumar Nandi, "OPTIMIST: Lightweight and Transparent IDS with Optimum Placement Strategy to Mitigate Mixed-rate DDoS Attacks in IoT Networks," in IEEE Internet of Things Journal, vol. 10, no. 10, pp. 8357-8370, 15 May15, 2023. [Contribution 3]

2. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, "Effective injection of adversarial botnet attacks in IoT ecosystem using evolutionary computing," in Wiley Internet Technology Letters, e433, (2023) [Contribution 1]

3. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, "A Hybrid IDS for Detection and Mitigation of Sinkhole Attack in 6LoWPAN Networks," in Springer International Journal of Information Security, (Accepted) [Contribution 4]

4. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, " RENO: Roving Shadow Honeypot for Multiple-Mix-Attack Detection in 6LoWPAN," in Elsevier Internet of Things; Engineering Cyber Physical Human Systems, (Under Review) [Contribution 4]

5. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, " EDAS-6N: An Energy Aware Edge Assisted Mixed Rate DDoS Attack Detection and Mitigation in 6TiSCH Network, " in ACM Transactions on Internet of Things (TIOT), (Under Review) [Contribution 3]

6. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, "BLooM: BashLite and Mirai Botnet Detection using Machine Learning Algorithms in 6LoWPAN Network," in Elsevier Internet of Things; Engineering Cyber Physical Human Systems, (Accepted) [Contribution 1]

Refereed Conferences:

7. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi. "ML for IEEE 802.15. 4e/TSCH: Energy Efficient Approach to Detect DDoS Attack Using Machine Learning," in International Wireless Communications and Mobile Computing (IWCMC), Harbin City, China, 2021. [Contribution 1]

8. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi."LORD: LOw Rate DDoS Attack Detection and Mitigation Using Lightweight Distributed Packet Inspection Agent in IoT Ecosystem," in IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 2019. [Contribution 2]

9. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi."An Adaptive and Lightweight Solution to Detect Mixed Rate IP Spoofed DDoS Attack in IoT Ecosystem," in IEEE India Council International Conference (INDICON), Coimbatore, India, 2018. [Contribution 3]

10. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi." Energy Efficient Approach to Detect Sinkhole Attack Using Roving IDS in 6LoWPAN Network," Innovations for Community Services (I4CS). Communications in Computer and Information Science, vol 1139. Springer, Cham. (2020) https://doi.org/10.1007/978-3-030-37484-6_11 [Contribution 4] **(Youth Scientist Award 2020)**

11. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi." BRAIN: Buffer Reservation Attack PreventIoN Using Legitimacy Score in 6LoWPAN Network," Innovations for Community Services (I4CS). Communications in Computer and Information Science, vol 1139. Springer, Cham. (2020) https://doi.org/10.1007/978-3-030-37484-6_12 [Contribution 4]

Publications other than Thesis Work:

Refereed Journals:

12. Dipojjwal Ray, **Pradeepkumar Bhale**, Santosh Biswas, Sukumar Nandi, and Pinaki Mitra. "A Novel Energy-efficient Scheme For RPL Attacker Identification In IoT Networks Using Discrete Event Modeling," IEEE Access, 2023 (Accepted).

13. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, "ROC-IDS: Raptor-Optimized Convolutional Neural Network-based Intrusion Detection Systems for IoT Network," in Wiley Internet Technology Letters, (Under Review).

14. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi, "Adaptive and Scalable Security Solution for Low-Rate DDoS Attack Detection in IoT Ecosystems," in Wiley Internet Technology Letters, (Under Review).

Refereed Conferences:

15. **Pradeepkumar Bhale**, Dipojjwal Ray, Santosh Biswas, and Sukumar Nandi."WOMN: WOrMhole Attack DetectioN and Mitigation Using Lightweight Distributed IDS in IoT Network," IEEE GCON 2023, (Accepted).

16. **Pradeepkumar Bhale**, Santosh Biswas, and Sukumar Nandi. "LIENE: Lifetime Enhancement for 6LoWPAN Network Using Clustering Approach Use Case: Smart Agriculture," 21st International Conference, I4CS 2021, Bamberg, Germany, May 26–28, 2021, Proceedings (pp. 59-75). Springer International Publishing, 2021.

17. Dipojjwal Ray, **Pradeepkumar Bhale**, Santosh Biswas, Sukumar Nandi, and Pinaki Mitra. "DAISS: Design of an Attacker Identification Scheme in CoAP Request/ Response Spoofing," IEEE TENCON, Auckland, New Zealand, 2021, pp. 941-946.

18. Dipojjwal Ray, **Pradeepkumar Bhale**, Santosh Biswas, Sukumar Nandi, and Pinaki Mitra. "ArsPAN: Attacker Revelation Scheme using Discrete Event System in 6LoW-PAN based Buffer Reservation Attack," IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1-6. IEEE, 2020.

19. Sukanta Dey, **Pradeepkumar Bhale**, and Sukumar Nandi. "ReFIT: Reliability Challenges and Failure Rate Mitigation Techniques for IoT Systems," International Conference on Innovations for Community Services. Cham: Springer International Publishing, 2019.

20. Pranav Kumar Singh, Subhredu Chattopadhyay, **Pradeepkumar Bhale**, and Sukumar Nandi. "Fast and Secure Handoffs for V2I Communication in Smart City Wi-Fi Deployment," 14th International Conference, ICDCIT, Bhubaneswar, India, Proceedings 14 (pp. 189-204). Springer International Publishing, 2018.

Doctoral Committee

Chairperson:

Prof. Jatindra Kumar Deka
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, Assam, India.
Email: jatin@iitg.ac.in

Research Advisor:

Prof. Sukumar Nandi
Senior Professor
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, Assam, India.
Email: sukumar@iitg.ac.in

Prof. Santosh Biswas
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Bhilai, Chhattisgarh, India.
Email: santosh@iitbhilai.ac.in

Members:

Prof. Diganta Goswami
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, Assam, India.
Email: dgoswami@iitg.ac.in

Prof. Partha Sarathi Mandal
Professor
Department of Mathematics
Indian Institute of Technology Guwahati, Assam, India.
Email: psm@iitg.ac.in

Vitae

Pradeepkumar Gajendra Bhale Student Member, IEEE) received the Bachelor of Engineering degree in com- puter science and engineering from the Government Collage of Engineering Aurangabad, Aurangabad, India, in 2010, and the Master of Technology degree in information security from ABV-Indian Institute of Information Technology Gwalior, Gwalior, India, in 2014. He is currently pursuing the Doctorate of Philosophy degree with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India, under the supervision of Prof. Sukumar Nandi and Prof. Santosh Biswas. He worked as a Technical Associate with Tech Mahindra Pvt. Ltd., Pune, India, for a period of two years. After that, he joined as an Assistant Professor with Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar, India, and worked for two years. His current research interests are Internet of Things, wireless security, network security, and discrete-event system modeling. Pradeepkumar Bhale was awarded the State of Maharashtra Post Graduate Fellowship for pursuing his master's degree. Profile link: https://scholar.google.com/citations?user=hHpL0uEAAAAJ&hl=en.

Contact Information

E-mail: pradeepkumar@iitg.ac.in
bhale@ieee.org
bhalepradeepkumar.iitg@gmail.com

Address: S/o: Gajendra Makaji Bhale, At post: Kesar Jawalga,
Tq: Omerga, Dist: Osmanabad, Maharashtra- 413605, India