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 a□rm 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 e□orts 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 e□orts might be fruitless. During these moments, I shared my anxieties with him, and he consistently o□ered 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 o□ered 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 e□orts 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 o□ciating Director), as well as all Deans and administrative sta□. I also extend my heartfelt thanks to the faculty and sta□ 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 sta at IITG for their healthcare services. A special thanks goes to the library and computer centre sta 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 a ection!.

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 di cult 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, o□ering their unwavering love and encouragement.

The utmost level of gratitude is extended to my parents. Their e□orts 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	Arti cial 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 Identi cation
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-De ned Networking
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network
6BR	6LoWPAN Border Router
BOF	Bu er Over ow
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
$P(b \mid A)$	Conditional probability of class b given the feature vector A
$P(A \mid b)$	Conditional probability of feature vector A given class b
$P(b)$	Prior probability of class b
$P(a_i \mid b)$	Conditional probability of feature a_i given class b
$P(b \mid a_1, \ldots, a_n)$	Conditional probability of class b given individual features $a_1, \ldots, a_n$
$\propto$	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
$_1$ and $_2$	Discrete distributions across distinct random values
k	Number of distinct random values
A_x and A_y	Random unique samples
	A parameter in the M_{GTV} equation
M_{TV}	Total Variation Metric
$P(x)$	Probability distribution
I_i	Packet frequency
A_{PIAT}	Packet frequency

	Intermediate variable
VM	Variation metric
P^EX$_i$•	External tra c probability
ET$_S$	Sample time window for external tra c
A$_r$ A$_v$ A$_h$	Di erent scenarios for LRDDoS attack detection
T$_S$	Sample time window
IX	Internal tra c
traff:	Threshold for internal tra c
ipcount	Threshold for IP address ow
IP$_i$	IP Address of an attacking node
INFO$_{GET}$	Information received for LRDDoS attack detection
X$_{Block}$	Tra c ow blocking rule
K	Parameter representing the number of hops
H	Hypergraph
V	Vertex set
E	Edge set (hyperedges)
VC	Vertex Cover
Z	Optimal value of linear program
! ^V$_i$•	Weight assigned to vertexV$_i$
x^V$_i$•	Decision variable for vertexV$_i$
S‡	Optimal solution for the minimum weighted vertex cover
Generator ^ •	Neural network component in GAN that generates fake data
Discriminator ^#•	Neural network component in GAN that discriminate fake data
D$_{loss}$	Discriminator loss in WGAN
G$_{loss}$	Generator loss in WGAN
^•	Sigmoid function
i$_t$, f$_t$, O$_t$, C̃	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.
$t_i(e_j)$	Attack score for the i^{th} object after the j^{th} event of attack .
I	Set of objects (nodes) under consideration.
$I^{\ddagger}(e_j)$	Maximum attack rating of an object in set I for event e_j.
$Node_{ID}$	Identi cation 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.
$Total event(Q_t = i)$	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

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 re 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-e ective implementation of IoT solutions, improved e ciency and productivity in industries, new business models, increased connectivity, and greater awareness and understanding of IoT bene ts [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 nancial 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 signi cant market penetration. This gure 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/Paci c 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 di erent 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, nancial 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/Paci c 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 signi cant 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. Speci cally, 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 signi cant 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 signi cant 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 signi cant technological revolution, reshaping our interaction with the physical world [22, 23]. However, these technologies are also associated with signi cant security aws [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, Bu er Reservation attacks, Fragment Duplication, node capture, and IP Address Spoo ng. 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 speci c security goals like con dentiality, integrity, availability, authentication, authorization, non-repudiation, privacy, resilience, auditing, and accountability [28, 29, 30]. However, applying these security measures uniformly across di erent 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 identi es 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, Bu er Reservation attacks, sinkhole attacks, bu er over ow 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 tra c 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. Bu er reservation attacks can be prevented by improving bu er management algorithms and monitoring bu er use in real time. Anomaly detection and tra c analysis can discover sinkhole attack redirection patterns. Secure coding and runtime monitoring of programme execution can prevent bu er over ow

attacks. Machine learning techniques, network tra c 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 uni ed 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 identi cation 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. E ective 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 identi cation 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 xes 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 e ectiveness balances device capabilities with the cost of e cient 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 e cient 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 e cient detection and response to security threats, e ectively 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 di erent networks or locations. In the IoT ecosystem, devices frequently change their network connections, making it di cult for traditional IDS systems to e ectively 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 e ective 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 di erent types of attacks. By training on labeled datasets that include both normal and malicious network tra c, the algorithms can learn the characteristics of di erent attacks and make accurate predictions in real time.

ML-based IDS approaches o er several advantages in the context of IoT environments. They can adapt to evolving attack techniques and detect previously unknown attacks. This exibility allows them to detect sophisticated and emerging threats that are di cult 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 e ectiveness 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 re ning 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 e ective 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-e cient 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 tra c 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 e ective models like the Naive Bayes Classi er 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-e cient 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 tra c using the Contiki Cooja simulator. It detects LRDDoS attacks by analysing network tra c from real and compromised nodes. The technique can also be adapted to internal and external network tra c 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 tra c 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 rst 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.

ˆ Chapter 7 - Conclusions and Future Work: In the concluding chapter, signi cant conclusions are drawn based on the ndings 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 a ordable. The availability of cloud computing infrastructure supports large-scale IoT implementations since it o ers processing, analytics, and storage capabilities. Data analytics and arti cial 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 e ciently. 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 di erent 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 di erent 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 fth 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 simpli es 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 signi cant 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 e ciency, minimise downtime, and improve product quality.

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

- Energy Management: IoT helps save energy in homes, o ces, 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 e ciency. 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, e ective 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 tness 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 o er personalised information. In addition to facilitating communication, wearables enable users to receive noti cations, 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, o ering low power consumption, a small memory footprint, and real-time capabilities. Principal characteristics include preemptive multitasking, portability across microcontroller platforms, e cient 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 exibility 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 o ers a platform for developing connected, energy-e cient 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 Tra c: It is crucial to create and make use of authentic network tra c 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 tra c. As a result, it would construct tra c patterns based on distributions, which may not re ect 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 e cacy of IoT security solutions under di erent 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 tra c, device diversity, and energy usage. These tools include NS-3 and OMNeT++, which o er 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 ne-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 o er 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 inter-operability 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 tra c by running actual applications; instead, they create dynamic tra c or use patterns like exponential distributions. In contrast, genuine network tra c 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	Tra c 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 o ers its own unique characteristics and capabilities, catering to di erent 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, o ers 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

a ect millions of interconnected devices. This section o ers 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 di erent 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 signi cant portion of the WSNs functionality [66].

2. Device tampering attacks:
   It is a cyberattack that incorporates intentionally manipulating the rmware /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 classi ed 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 speci c 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 signi cant 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. Bu er Reservation Attack:
   In a bu er reservation attack, the attacker reserves the bu er space of the victim node by sending it incomplete packets. Nodes are unable to allocate additional bu er space for incomplete packets from other nodes due to their restricted resources. As a result, while the attacker occupies the bu er 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 les [72].

2.6.3 Application Layer Vulnerabilities

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

2. Bu er Over ow Attack: An attacker commits a bu er over ow attack when they place an excessive amount of data into a bu er, which causes the bu er to over ow 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 ood 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 nal ACK packet to nish the handshake. When an attacker uses a spoofed IP address or fails to reply, the server is left waiting inde nitely for the nal ACK while its resources are depleted. This kind of attack reduces the reliability of the server, which might a ect 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 tra c 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 e ciency. 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. Tra c Analysis Attack:

   Tra c analysis resembles eavesdropping. However, in this kind of attack, the attacker just examines the tra c rather than gaining access to the real contents. The objective is to infer the tra c 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 eld in IPv6 Routing Protocol for Lossy Networks (RPL) routing control messages. Using the version number eld, an attacker can cause routers to rebuild their routing tables by sending falsi ed 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 tra c, overwhelming the network. Consequently, the links between nodes become unavailable, disrupting communication. An example of such an attack is the HELLO ood 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

tra c from neighboring nodes. The compromised node broadcasts falsi ed information to lure a signi cant amount of tra c. 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 di erent geographical networks build a communication tunnel. This tunnel can be made with a wired or wireless connection with ample range and bandwidth, working at di erent 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 signi cant threat to data con dentiality, 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 tra c to exhausting network resources and disrupting the topology. This thesis is focused on DDoS attacks, which can be broadly

classi ed into three categories: High-Rate DDoS Attacks, Low-Rate DDoS Attacks, and Mixed-Rate DDoS Attacks. Low-Rate DDoS attacks are di cult to identify due to their low-rate and intermittent tra c behavior, which is similar to legitimate tra c [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 e cacy, 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 di erent datasets, which provide general information about them. In this thesis, we used one, two, and three datasets to develop e ective IDS solutions.

Table 2.3: Technical Details and Properties of Di erent 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		

ˆ IoT-23 [77]: IoT-23 is a dataset that includes network tra c 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 tra c and a variety of assaults, including Brickerbot and Mirai. The dataset is divided into training and test sets and includes both encrypted and unencrypted tra c in pcap format. The dataset also contains metadata les that list the devices and di erent 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 tra c and various types of attacks, such as DoS, port scans, and brute-force attacks. It consists of several scenarios and network topologies with various tra c characteristics, including IoT tra c. 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 di erent 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 tra c from several IoT device-targeting botnets, such as Mirai and Gafgyt. The dataset contains both normal tra c 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 tra c. The dataset consists of over 12,000 pcap les, each of which contains one minute of tra c. 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 e cacy 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 tra c 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 les, each of which contains 30 minutes of tra c, and is accessible in both pcap and CSV formats. In addition, it contains metadata les containing information about the devices and types of attacks. The TON_IOT dataset may be used to assess the e ectiveness 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 tra c data captured during the DARPA Intrusion Detection Evaluation Programme. Although it is not an explicitly IoT-related dataset, it may include some tra c 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 di erent 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 e ectiveness 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 e cient IDS systems to combat network attacks.

- ISCX [83]: The ISCX dataset comprises regular tra c 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 signi cant information for tasks like as classi cation, 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 speci c 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, arti cial 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 e cacy, they also address practical concerns, such as data management and detection e ciency within IDS.

- Support Vector Machine (SVM): SVMs strategically nd 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 over tting. In contrast, greater values of k result in simpli ed models with diminished tting capabilities.

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

Figure 2.5: Classi cation 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 classi es sample x into the highest probability class of $k \quad 1; 2; :::; K \quad 1$. 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 classi cation 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 identi es the most relevant attributes and creates child nodes from the parent node. Advanced classi ers like random forests and extreme gradient boosting (XGBoost) use numerous decision trees.

- Arti cial 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 ine cient for deep networks. ANNs are shallow models, unlike deep learning models detailed in Section 2.8.2

- Ensembles Learning (EL): Every classi er has its own advantages and disadvantages. Combining numerous under performing classi ers 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 classi ers. Through a voting process, these classi ers ultimately contribute to the nal 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 classi cation layer. The DBN training is broken up into two parts: supervised ne-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 classi cation, 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 ne-tuning method, shown in Figure 2.7 (a). This technique uses unlabeled data to train the network's parameters. Following that, the network is ne-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 identi ed 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 rst component is responsible for identifying any deviations from established network tra c 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. Noti cation and Reporting: The third component, which is critical for intrusion

detection, sends timely noti cations to the system administrator. This fast noti cation 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 di erent 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 tra c, 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 tra c threats. It compares incoming data to a library of signatures and sends out alerts when a match is found. This makes it easier to nd known threats quickly. Signature-based IDSs are quick and e ective, but only against known threats. This approach has a very low false positive rate (FPR) for known attacks, but it is ine ective 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 e ective. 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 identi es intrusions based on system behavior. By establishing a normal activity baseline, any deviations are agged as potentially malicious and alerts are generated. This method is e ective 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 bene ts of both signature-based and anomaly-based techniques by combining them. While anomaly-based evaluation nds 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 identi es 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 speci c nodes. It has the ability to

examine all incoming and outgoing Internet tra c 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 nd 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 o er 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 speci c 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 e ective [101]. Makhdoom et al. [102] have made signi cant e orts to identify general and layerwise IoT threats. Within the context of the survey paper, the primary focus of the researcher is to concretely de ne the structure of malicious activity made against the IoT environment. They also talked about e ective 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-o 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 rst 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 o ered 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 signi cant 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 nd 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-E cient 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 di erent 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 elds of computing. Even though it plays a vital role in raising the standards of life, it su ers 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 di erent 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 tra c investigates with the help of an arti cial 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 tra c or benign tra c. 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 x in devices and routers. As per our argument, it is useless to practice ANN-based IDS with this strategy. All devices were performing o ine 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 e cient, trained to mimic IoT network tra c 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 tra c 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-o in detection performance.

The goal of using an edge-based ML approach is because of all types of tra c 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 ve, and perform adequately than closely related works. The signi cant features of this security method as follows:

- The proposed approach generates both types of tra cs (i.e., benign and spiteful tra c) utilizing the Contiki cooja simulator [125] [126]. It also examines a well-known dataset [79] [127] for the DDoS attack identi cation.
- A novel edge-based ML solution for the resource-constrained network, which is lightweight and energy-e cient.
- A feature selection is dynamically preparing and deriving certain contextual features from network tra c.
- 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 tra c (i.e., internal tra c and external tra c) in the IoT ecosystem. Hence DDoS attacks can be identi ed and depreciate the future event of DDoS attacks.

- The intended approach achieves the IEEE 802.15. 4e/TSCH tra c characteristics of DDoS attack. It classi es 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 speci cally 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 e cient way to forward IP packets in low power lossy network (LLN). It makes IP header compression. 6LoWPAN incorporates routing speci cations, 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 doesTime 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 tra c, 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 signi cant 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 arti cial 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 rst 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 coe cient , and wrapper step. Its boosts up the detection accuracy.
M. Sha q et al. [135] 2020	This paper gives the objective is an e cient solution with a modi ed 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 identi ed 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 t 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., ow lter, ow handler). The ow lter is also subdivided into two parts 1) ow ltration and 2) DDoS discovery.

As per Y. Jia et al. [138], DDoS attacks identi ed using the FlowGuard model. this model contains a ow lter and ow 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 identi cation. 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 classi er to t the training dataset.

- Classi cation and Result: This component is adopted to predict the test set result. And classify DDoS attack tra c and benign tra c.

3.3.2 Naive-bayes classi er

The proposed model is an intelligent IDS based UDP ood attack detection system which uses a naive-bayes classi er. It is a Edge-based ML enable IDS that analyses all type of tra c received from internal and external device. UDP ood a ects 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 classi er is a supervised learning technique that uses bayes theorem for the classi cation 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 ow 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 $\hat{}b\neg A\bullet$ are independent. The behavior of one distinct feature does not in uence 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:

$$P\hat{}b\neg a_1; : : : ; a_n\bullet \; P\hat{}b\bullet \prod_{i\ 1}^{n} P\hat{}a_i\neg b\bullet \tag{3.2}$$

$P\hat{}a_i\neg b_j\bullet$ is evaluated for each a_i in A and b_j in b, and also $P\hat{}b\bullet$ 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 $\hat{}b\bullet$ has two results, like (yes and no). Di erent cases have multivariate

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

classi cation. Therefore, the highest probability of classb is calculated as:

$$b \quad \arg\max_{b} P\hat{}b\bullet \prod_{i\ 1}^{n} P\hat{}a_i \sim b\bullet \tag{3.3}$$

Where, b represents the parameter or value that maximizes the expression,P,ˆb• denotes

the probability of observing parameter b, $\prod_{i=1}^{n}$ signi es 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 nd the parameter b that maximizes the joint probability of observing all data points $a_1, a_2, \ldots, a_n$ 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 motes 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 tra c is screened at 6BR, and tra c 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. Di erent tools are used for injecting DDoS attacks (i.e., libcoap, hping, etc.). These tools generated tra c 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 modi ed 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 strati ed percentage split evaluation scheme, where 80% of total data is used for training and 20% for testing in randomly selected. The classi cation performances are expressed by the following performance parameters which describe the performance of binary classi cation 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.

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

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

$$\text{Precision} \quad \frac{TP}{TP \quad FP} \tag{3.5}$$

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

$$\text{Recall} \quad \frac{TP}{TP \quad FN} \tag{3.6}$$

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

$$F \quad \text{measure} \quad 2 \ddagger \frac{RCL \ddagger PRC}{RCL \quad PRC} \tag{3.7}$$

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 tra c. 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 e ciently 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 85916mJ . Table 3.4 also exhibits average accuracy and response time are98:7% 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 di erent 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 di erent 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. Sha q 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 tra c and shield against the DDoS attack. The naive bayes method is analyzed generated DDoS attack tra c 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 di erent DDoS datasets. The comparative analysis with current solutions with closely related work is also o ered. Experimental outcomes show that the intended solution can identify the DDoS

attacks. Hence, DDoS attacks are depreciated as the proposed solution identi es 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 ˆ24:2 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, con dentiality, 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 classi ed 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 veri ed users. The assault is achieved by transferring plenty of tra c 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 identi ed by security solutions due to its tra c characteristics (i.e., high volume). As an outcome, attackers are choosing the LrDDoS attack [147]. LrDDoS attacks are strenuous to recognise because the assault tra c characteristics are similar to genuine tra c. 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 tra c 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 veri ed 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-e cient, 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 tra c 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 tra c for LrDDoS attack identi cation. 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 identi es internal as well as external LrDDoS attacks in the 6LoWPAN network. The signi cant features of this defence approach are:

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

2. Distributed: The suggested method incorporates distributed lightweight energy e cient

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 tra c 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 tra c characteristics of LrDDoS attack. The solution identi es 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 [48]. This type of attack creates adequately low rate tra c to the targeted device or application. The variations among LrDDoS and LRDoS attack are shown in Figure 4.1.

In these attacks, spiteful network tra c di ers in terms of time period ˆ T•, Burst-Rate ˆBR•, and Burst-Width ˆBW •. 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 di erently 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 di cult 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 tra c that is very much similar to legitimate tra c.

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 di erent 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 di erent 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 tra c. However, the shortcoming of this approach is that it requires extra computing power and resources. Therefore, this method does not t 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 coe cient 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 tra c. 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 lters internal IoT network tra c. Hence, 6BR and PIA are the best spots in the IoT network for the intended LrDDoS attack identi cation 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 rst phase, input tra c is analyzed by packet arrival time. Our security approach identi es the legitimate IoT ecosystem tra c from DDoS attack tra c by utilizing generalized total variation metric (GTV) . In the GTV metric calculation, two consecutive tra c samples, and threshold $_{traff:}$ are used. The $_{traff:}$ is discovered by heuristics that depends on LrDDoS attack variations as shown in Figure 4.1 and deployed circumstances.

Generalized total variation metric $\hat{M}_{GTV}\bullet$

Let $_1$ and $_2$ imply a couple of discrete distributions across k distinct random values. Then the M_{GTV} is represented with the following equation [152]:

$$M_{GTV\hat{}\ _1;\ _2\bullet} \quad \left(\sum_{i\ 1}^{k} \hat{S}\ _{1i}\hat{A}x\bullet \quad _{2i}\hat{A}_y\bullet S\bullet\right)^{\min\check{S}1;\frac{1}{-}\bullet} \tag{4.1}$$

where $A0$, $A_x;A_y$ are the random unique samples and $_1;\ _2>P$ as explained previously. When $C1$, the di erence between the two distributions $\hat{}\ _1;\ _2\bullet$ is also called the Minkowski distance [153].

$$M_{TV\hat{}\ _1;\ _2\bullet} \quad \left(\sum_{i\ 1}^{k} \hat{S}\hat{}\ _1\bullet_i\hat{A}x\bullet \quad \hat{}\ _2\bullet_i\hat{A}_y\bullet S\bullet\right)^{\frac{1}{}} \tag{4.2}$$

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

Total variation metric $\hat{M}_{TV}\bullet$

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

$$M_{TV\hat{}\ _1;\ _2\bullet} \quad \frac{1}{2}\left(\sum_{i\ 1}^{k} S\hat{}\ _1\bullet_i\hat{A}x\bullet \quad \hat{}\ _2\bullet_i\hat{A}_y\bullet S\right) \tag{4.3}$$

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:

$$I_i \quad \frac{f_i}{\sum_{i\ 1}^{K} f_i}; \quad \text{and} \quad P\hat{x}_i\bullet \quad \frac{x_i}{\sum_{i\ 1}^{k} x_i} \tag{4.4}$$

Where f_i and x_i indicate the packet frequency and feature instances of the i^{th} sample within time period $\hat{k;K}\bullet$. The average packet inter-arrival time $\hat{A}_{PIAT}\bullet$ 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).

$$\frac{a \quad A_{PIAT}}{\hat{} \quad PI_i \bullet} \tag{4.5}$$

$$VM\hat{}_{1};\ _2\bullet \quad \min\check{S}1;\tfrac{1}{k}\bullet \sum_{i\ 1}^{k} \hat{S}\hat{}_{1}\bullet_i \hat{A}_x\bullet \quad \hat{}\ _2\bullet_i \hat{A}_y\bullet S\bullet \text{‘} \tag{4.6}$$

Here $_a$ indicates the average probability distribution for packet characteristic and is the number of characteristics considered. The variation metric $\hat{V}M\bullet$ distinguishes LrDDoS attack tra c and legitimate tra c. As per VM value of LrDDoS, attack tra c is larger than legitimate IoT network tra c. The important steps of our LrDDoS attacks detection approach is shown in Algorithm 4.1. We take 4, for four level packet characteristic analysis during probability distributions.

Algorithm 4.1 LrDDoS Attack Detection at N_{6BR}

Require: All external tra c $\hat{E}X_i\bullet$ from $\hat{N}_{EL}\bullet$, $\hat{N}_{ES}\bullet$
Ensure: Detection of LrDDoS attack in IoT network.
Initialisation : External tra c probability $P\hat{E}X_i\bullet$,
Sample time window for external tra c $\hat{E}T_S\bullet$, where i 1;2;3;:::n, $\hat{E}T_S\bullet$ $\tilde{E}T_{s1};ET_{s2};ET_{s3};:::;ET_{sN}\bullet$, and N is the total occurrences within sampling time windows $\hat{E}T_S\bullet$; average packet inter-arrival time $\hat{A}_{PIAT}\bullet$; scaling SC,

1: External tra c $\hat{E}X\bullet$ received from $\hat{N}_{EL}\bullet$ and $\hat{N}_{ES}\bullet$ within sampling time windows $\hat{E}T_S\bullet$.
2: Estimate packet intensity PI_i, probability $P\hat{x}\bullet$ utilizing Equation (4.2) and further calculate A_{PIAT} within sampling time windows $\hat{E}T_S\bullet$.
3: Estimate the individual sample on distribution $_1$ and $_2$

$$_1\hat{A}_x\bullet \quad \sum_{m\ 0}^{l} M_{GTV} \quad _m\hat{A}_x\bullet \tag{4.7}$$

$$_2\hat{A}_y\bullet \quad \sum_{m\ l}^{l\ 1} M_{GTV} \quad _k\hat{A}_y\bullet \tag{4.8}$$

4: if (A 0• for sampling time windows $\hat{E}T_S\bullet$ then
5: measure total variation $\hat{T}V\bullet$: Using Equations (4.5) and (4.6).
6: else if (A_r A_v) then
7: Estimate vertical variation applying $VM_{\ 1\ 2}$ (Equations (4.5) and (4.6))
8: else if (A_r A_h) then
9: Estimate horizontal variation applying $VM_{\ 1\ 2}$ (Equations (4.5) and (4.6))
10: end if
11: Compare against variation measure threshold, $VM_{\ 1\ 2}$ $C\hat{}_{\ traff:}\bullet$ then LrDDoS attack alarm arise and initiate the algorithm 4.2

In the second phase of LrDDoS attack detection, internal network tra cs $\hat{I}X \bullet$ are examined at the PIA. The DDoS attack tra c is obtained using network sni er [155]. The network tra c is collected upto sample time windows $\hat{T}_S \bullet$ and produced the Packet Flow Behavior Graph $\hat{G}_{PFB} \bullet$. It also measures the packet count associated with the various IP address ows.

Algorithm 4.2 LrDDoS attack detection using N_{PIA}

Input: Internal Tra c $\hat{I}X \bullet$ from $\hat{N}_{IL} \bullet$, $\hat{N}_{IS} \bullet$; threshold $\hat{}_{traff:} \bullet$ and $\hat{}_{ipcount} \bullet$
Output: Detection of LrDDoS attack in IoT network.
Initialisation : Packet ow behavior graph $\hat{G}_{PFB_i} \bullet$,
Sample time window $\hat{T}_S \bullet$, where $i \quad 1; 2; 3; :::n$, $\hat{T}_S \bullet \quad \tilde{} T_{s1}; T_{s2}; T_{s3}; :::; T_{sN} \bullet$, and N is the total occurrences within sampling time windows $\hat{T}_S \bullet$.
1: Internal tra c $\hat{I}X \bullet$ received from $\hat{N}_{IL} \bullet$ and $\hat{N}_{IS} \bullet$ within sampling time windows $\hat{T}_S \bullet$.
2: Generate 3-hop packet ow behavior graph $\hat{G}_{PFB_i} \bullet$.
3: Keep track on frequency ow $\hat{F}F_i \bullet$ of IP address $\hat{I}P_i \bullet$
4: if $(IX \ C \ _{traff:} \ \&\& \ FF_i \ C \ _{ipcount})$
for 2 successive sampling time windows $\hat{T}_S \bullet$ then
5: LrDDoS attack detected and initiate Algorithm 4.3
6: else
7: Normal internal tra c $\hat{I}X \bullet$.
8: end if

The internal network tra c $\hat{I}X \bullet$ analysis is based on the IP address with the corresponding IP ow. The IP address and frequency ow are denoted as IP and FF_i, respectively. FF_i measures di erently for LrDDoS attack tra c compared to legitimate tra c. The threshold $\hat{}_{traff:} \bullet$ and $\hat{}_{ipcount} \bullet$ values of the IP address with the corresponding IP ow 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 $\hat{}_{traff:} \bullet$ and $\hat{}_{ipcount} \bullet$ values obtained at the separate sample time $\hat{T}_S \bullet$ intervals show minor variations. From experiments and observations, we can determine the threshold values that covere all variations of LrDDoS attack tra c. The functioning of the LrDDoS attack identi cation 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  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 tra c is ltered 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 $\hat{G}_{PFB}\bullet$. 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 $\hat{G}_{PFB}\bullet$.

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 tra cs. Hence, we have measured the PD of LrDDoS tra c and legitimate tra cs.

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

The variation metric is used to di erentiate between LrDDoS attack tra c and legitimate tra c. In variation metric, we adopted 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 $_{traff:}$ • among spiteful and legitimate tra c. The spacing value and $_{traff:}$ are recorded as 0:96 and 0:53, respectively. Figure 4.9 shows that VM can clearly di erentiate the LrDDoS attack tra c from substantial tra c.

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

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

Figure 4.8: Generated data-set using Contiki cooja

Figure 4.9: Threshold analysis

within two periods. To obtain LrDDoS attack packet ow with IP address and the frequency variation rate, we examined delays ˆ 0 to 0:36• Secondstowards a target. Like MIT, and CAIDA datasets, we estimate frequency-rate variation with the generality component 1 to 4 for the sampled tra c window T_S. The Frequency-rate variation threshold ˆ $_{traff:}$ • and IP-Address count ˆ $_{ipcount}$ • are recorded as 0:68 and 310, respectively. Figures 4.10, 4.11, and 4.12 show the ˆ $_{traff:}$ • calculated spacing and ˆ $_{traff:}$ • in LrDDoS and authentic tra c. The threshold value 0:69 and spacing rate S 0:25 are utilized to classify legitimate tra c and LrDDoS tra c.

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

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).

$$FNR \quad \frac{FN}{FN \quad TP} \tag{4.9}$$

and

$$FPR \quad \frac{FP}{FP \quad TN} \tag{4.10}$$

where ,

FP = Legitimate activity wrongly classi ed.

TN = Legitimate activity recognized accurately.

FN = Spiteful activity wrongly classi ed.

TP = Spiteful activity recognized accurately

The performance study, the experiment analysis is repeated several times with various LrDDoS con guration. 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 identi es the LrDDoS attack tra c in (1-3.47) seconds It is concluded that the

intended method identi es 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 tra c of the IoT ecosystem and propose LrDDoS attack detection and mitigation method. The approach is based on the total variation metric, packet ow 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 tra c by using verticle and horizontal scaling in Contiki cooja simulator environment. Our approach detects the LrDDoS attack e ectively 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.

- Je 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, con dentiality, 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 tra c ows. Among these malicious ows, DoS and DDoS attacks are very harmful to IoT systems as these attacks disrupt the availability of systems. Therefore, internal tra c ows 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 sni /eavesdrop on ows 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 optimum placement strategy to mi tigate mixed-rate DDoS attack in IoT 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 arti cial ows from public datasets as well as in-house generated dataset to reduce the distribution bias of the datasets. The WGAN-generated ows 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 e ective 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, bu er 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 tra c, 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 protocol (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 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 tra c, some malicious internal ows 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 bene ts of centralized and distributed strategies. In this strategy, a centralized entity monitors external tra c, 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 sni /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-o 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 tra c ows, 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 classi ed 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 ows. The trained model is then used to classify observed behaviors of network ows. However, signature-based IDS cannot detect unknown (zero-day) attacks or modi ed/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 veri able 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 tra c and organizes packets into conversation ows. 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 arti cial tra c ows to reduce the bias of the trained models. In this work, we have generated arti cial ows 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 tra c ows only by eavesdropping, and do not query the monitored nodes about their tra c/system status. To quantify transparent IDS coverage, we de ne the term K-hop IDS coverage scheme, which guarantees to monitor any ow 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 H consists of a vertex set V, and a collection of non-empty subsets of V, namely hyperedges, forming the set E. In hypergraph, a hyperedge can connect an arbitrary number of vertices. A hypergraph is called K-uniform, if $\forall E_m \in E, |E_m| = K$. 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 E, along with the set of vertices V form the K-uniform hypergraph 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 E_i, a weight $1/E_i$ is assigned to that corresponding vertex V_i of the hypergraph H. Let the function $w(V_i)$ gives the assigned weight of V_i. The minimum weight vertex cover solution VC of H contains the minimum number of vertices of H such that $\forall E_m, \exists x \in V_C, x \in E_m$. 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 $H = (V, E)$

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

Formulation 5.1:

Minimize $\sum_{V_i \in V} \hat{!}\,\hat{V}_i \bullet x\hat{V}_i \bullet\bullet$

Subject to the constraints:

1. $\sum_{V_j \in E_m} x\hat{V}_j \bullet C1; \quad \brvbar\, E_m \in E$
2. $x\hat{V}_i \bullet \in \tilde{} 0;1 \bullet \quad \brvbar\, V_i \in V$

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 $\sum_{V_i \in V} \hat{!}\,\hat{V}_i \bullet x\hat{V}_i \bullet\bullet$

Subject to the constraints:

1. $\sum_{V_j \in E_m} x\hat{V}_j \bullet C1; \quad \brvbar\, E_m \in E$
2. $x\hat{V}_i \bullet B1 \quad \brvbar\, V_i \in V$
3. $x\hat{V}_i \bullet C0 \quad \brvbar\, V_i \in V$

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 $V_i \in V$, if $x(V_i) \geq 1/K$, include 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 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 H. The constraint 1 of the Formulation 5.2 ensures that the step 3 in Solution assigns values of $x(V_i)$ such that for each hyperedge $E_m \in E$, $\sum_{V_j \in E_m} x(V_j) \geq 1$. In a K-uniform hypergraph, all the hyperedges contains exactly K vertices. The above two facts imply that for each hyperedge E_m, $\exists V_j \in E_m$, such that $x(V_j) \geq 1/K$. As the step 2 of the given solution includes all the V_i with $x(V_i) \geq 1/K$ in the solution set VC, definitely VC contains atleast one element from each of the hyperedges. Thus VC is a vertex cover for hypergraph 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 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 $Z \leq w(S^*)$ which gives a lower bound of $w(S^*)$, where $w(S^*)$ is the total weight of the optimum solution S^*. We have,

$$
\begin{aligned}
& Z = \sum_{V_i \in V} w(V_i) \cdot x(V_i) \\
\Rightarrow\ & Z \geq \sum_{V_i \in V,\ x(V_i) \geq \frac{1}{K}} w(V_i) \cdot x(V_i) \\
\Rightarrow\ & Z \geq \sum_{V_i \in V,\ x(V_i) \geq \frac{1}{K}} w(V_i) \cdot \frac{1}{K} \\
\Rightarrow\ & Z \geq \sum_{V_i \in S} w(V_i) \cdot \frac{1}{K} \\
\Rightarrow\ & Z \geq \frac{1}{K} \sum_{V_i \in S} w(V_i) \\
\Rightarrow\ & Z \geq \frac{1}{K} w(S) \\
\Rightarrow\ & w(S) \leq K \cdot Z \text{ [rearranging the terms]} \\
\Rightarrow\ & w(S) \leq K \cdot w(S^*) \text{ [because } Z \leq w(S^*)]
\end{aligned}
$$

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 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 $T[]$ 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 $\hat{}\bullet$, and the discriminator/ critic $\hat{}\#\bullet$. 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

$$\min_{\#} \max E_{R \quad P\hat{}R\bullet} \#\hat{}R\bullet \quad E_{F \quad P\hat{}F\bullet} \#\hat{} \hat{}F\bullet\bullet \tag{5.1}$$

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 $\hat{D}_{loss}\bullet$, and Generator loss $\hat{G}_{loss}\bullet$ functions. These loss functions are mathematically represented below [185].

$$D_{loss} \quad \min_{\#} \; E_{R \quad P\hat{R}\bullet} \; \log \#\hat{R}\bullet \tag{5.2}$$

$$G_{loss} \quad \min \; E_{R \quad P\hat{F}\bullet} \; \log\hat{1} \quad \#\hat{} \quad \hat{F}\bullet\bullet\bullet \tag{5.3}$$

The loss function score depends on real and adversarial data. Based on score, WGAN generates high-quality adversarial data. WGAN discriminator $\hat{}\#\bullet$ provides a critic score. This score decides the di erence between real and fake data. A critic score @0 indicates real tra c data, and a score A0 indicates that the given tra c 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).

$$i_t \quad \hat{}W_i \quad h_{t \ 1}; x_t \quad b_i\bullet \qquad f_t \quad \%\!\!\%W_f \quad h_{t \ 1}; x_t \quad b_f\check{Z} \tag{5.4}$$

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 le
Num of features	46	22	23	10	10
Dataset size	64GB	23 GB	20GB	69.3GB	17295 packet ow

Table 5.2: Feature selection using SHAP from ve 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

$$O_t \quad \hat{W}_o \quad h_{t\ 1}; x_t \quad b_o \bullet \quad C \quad \tanh \hat{W}_c \quad h_{t\ 1}; x_t \quad b_c \bullet \tag{5.5}$$

$$C_t \quad f_t \ddagger \%C_{t\ 1} \quad i_t \ddagger C_t \check{Z} \quad h_t \quad o_t \ddagger \tanh \hat{C}_t \bullet \tag{5.6}$$

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 C̃. Input, output, and input time are denoted as x, h, t. The sigmoid function is denoted as ˆ •.

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 [31], [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 ows with low rate DDoS attack and non-attack are generated in-house using Contiki cooja and FIT IoT-LAB [62]. The ow 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 ows. 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.

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

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

Feature selection (FS)

The features having signi cant contributions to the mixed-rate DDoS attacks are chosen, while the redundant and insigni cant 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 ve datasets.

5.5.3 Model training

The model training of OPTIMIST has two phases. In the rst phase, a WGAN model is trained on the datasets (refer to Section 5.5.2) to generate arti cial 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 Con guration

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: Con guration	

In the second phase of the training, the LSTM model is trained with the datasets (real and arti cial) to classify mixed-rate DDoS attacks. To mitigate the over- tting issue, dropout and batch normalization strategies are used. These strategies change the network design in each training epoch to reduce the chance of over tting 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 Con guration

Layer (Type)	Con guration
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 o ine 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 ows, extracts features, and classi es 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 tra c ows 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 e cient. The time complexity of the LSTM is $O(N_{ou} \cdot N_{hu} \cdot (N_{ou} \cdot N_{mcb} \cdot S_{mcb}) \cdot (N_{hu} \cdot N_{fc}) \cdot (N_{mcb} \cdot S_{mcb} \cdot N_{fc}) \cdot O(W))$, 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 de nes 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 gure, 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 ows 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