TECHNIQUES TO DETECT RESOURCE ATTACKS IN RPL-BASED INTERNET OF THINGS

Thesis submitted to the Bharathidasan University
in partial fulfillment of the requirements for
the award of the degree of
DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

By

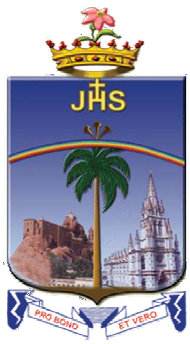
A. ARUL ANITHA, MCA., NET.,

(BDU Ref. No. 06470/Ph.D.K10/Comp.Sci./F.T./Nov. 2018)

Under the Guidance and Supervision of

Dr. L. AROCKIAM, M.C.A., M.Tech., M.B.A., CSM., BLIS., M.Phil., Ph.D.,

Associate Professor



PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

St. JOSEPH'S COLLEGE (Autonomous)

Special Heritage Status Awarded by UGC, Nationally Accredited at 'A++' Grade (4th Cycle) by NAAC
College with Potential for Excellence by UGC, DBT-STAR & DST-FIST Sponsored College

TIRUCHIRAPPALLI - 620 002, INDIA.

MAY 2022

St. JOSEPH'S COLLEGE (Autonomous)

Special Heritage Status Awarded by UGC, Nationally Accredited at 'A++' Grade (4th Cycle) by NAAC
College with Potential for Excellence by UGC, DBT-STAR & DST-FIST Sponsored College

TIRUCHIRAPPALLI-620 002.

Phone: 0431-2700320, Cell: 94439 05333 Fax: 0431-2701501
E-mail: larockiam@yahoo.co.in Website: www.sjctni.edu

Dr. L. Arockiam, ***MCA., M.Tech., MBA., CSM., BLIS., M.Phil., Ph.D.,***
Associate Professor in Computer Science
Director of MCA

CERTIFICATE

This is to certify that the thesis entitled **"Techniques to Detect Resource Attacks in RPL-based Internet of Things"** submitted by **Ms. A. Arul Anitha,** a research scholar in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli – 620002, for the award of the degree of **Doctor of Philosophy in Computer Science**, is a record of original work carried out by her under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the University and in my opinion the thesis has reached the standards needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Date: 13.05.2022 **(Dr. L. Arockiam)**

Place: Tiruchirappalli *Research Supervisor*

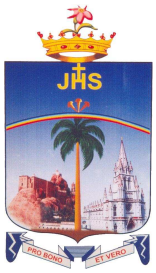
A. ARUL ANITHA
Research Scholar
Department of Computer Science
St.Joseph's College (Autonomous)
Tiruchirappalli–620002, India.

DECLARATION

I hereby declare that the work embodied in this thesis entitled **"Techniques to Detect Resource Attacks in RPL-based Internet of Things"**, is a researchwork done by me under the supervision and guidance of **Dr. L. Arockiam,** Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli – 620002, India. The thesis or any part thereof has not formed the basis for the award of any Degree, Diploma, Fellowship or any other similartitles.

Date: 13.05.2022 **(A. Arul Anitha)**

Place: Tiruchirappalli *Research Scholar*



PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

St. JOSEPH'S COLLEGE (Autonomous)

TIRUCHIRAPPALLI- 620 002

TAMILNADU, INDIA

CERTIFICATE OF PLAGIARISM CHECK

1	Name of the Research Scholar	**A. Arul Anitha**
2	Course of Study	Ph.D., Computer Science
3	Title of the Thesis/Dissertation	**Techniques to Detect Resource Attacks in RPL-based Internet of Things**
4	Name of the Research Supervisor	**Dr. L. AROCKIAM**
5	Department/Institution/Research Center	PG & Research Department of Computer Science St. Joseph's College (Autonomous) Tiruchirappalli-620 002
6	Acceptable Maximum Limit	10%
7	Percentage of Similarity of Content Identified	1%
8	Software Used	OURIGINAL
9	Date of Verification	04.05.2022

Report on Plagiarism check, item with % of similarity is attached

Signature of the Research Supervisor

Signature of the Candidate

Curiginal

Document Information

Analyzed document	Full Thesis (5).docx (D135401019)
Submitted	2022-05-04T07:04:00.0000000
Submitted by	Dorairajan
Submitter email	manavaidorai@gmail.com
Similarity	1%
Analysis address	manavaidorai.stjct@analysis.curiginal.com

Sources included in the report

W	URL: https://hal.archives-ouvertes.fr/hal-01206380/document Fetched: 2020-06-08T07:24:05.4030000	1
W	URL: https://arxiv.org/pdf/2003.11061 Fetched: 2021-01-13T09:47:33.8530000	1
W	URL: https://arxiv.org/pdf/2201.06937 Fetched: 2022-05-04T07:04:20.3730000	1
W	URL: https://congpu.github.io/lab/document/paper/ieee_ccwc_2018.pdf Fetched: 2022-05-04T07:04:50.7930000	3
W	URL: https://www.semanticscholar.org/paper/Mitigation-Mechanisms-Against-the-DAO-Attack-on-the-Wadhaj-Ghaleb/c0b30d6442016e8059a696a1c9bda762322dde65 Fetched: 2021-11-13T00:27:01.9000000	1
W	URL: https://downloads.hindawi.com/journals/wcmc/2021/8414503.pdf Fetched: 2021-11-02T14:02:20.4600000	1
W	URL: https://www.napier.ac.uk/~/media/worktribe/output-1319002/addressing-the-dao-insider-attack-in-rpls-internet-of-things-networks.pdf Fetched: 2020-01-02T14:27:30.6800000	2

ACKNOWLEDGEMENT

"How can I repay the Lord, for all the good,

He has done for me?" **Psalm 116:12**

With profound joy and gratitude to **God, the Almighty**, I wish to express my sincere thanks to those who have extended their constant support in one way or the other during this amazing research journey.

First of all, I am extremely grateful to **Rev. Mother Reginal, SAT,** the Superior General of the Congregation of Sisters of St. Anne, Tiruchirappalli, for giving me permissions to pursue my research as a Full-Time scholar in this esteemed institution. With grateful heart, I thank her for her prayers, blessings and support.

I would like to thank **Rev. Dr. Leonard SJ,** the Rector, **Rev. Dr. Peter SJ,** the Secretary, and **Rev. Dr. Arockiasamy Xavier SJ,** the Principal for their encouragement and support to accomplish my research.

My sincere gratitude is reserved for my Research Supervisor **Dr. L. Arockiam,** Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, for his invaluable insights and suggestions. I thank him for all the useful discussions and brainstorming sessions, especially during the difficult conceptual development stage of my research. Despite of his busy schedule, he is able to spend time for his scholars whenever they are in need.

Very special thanks to my Doctoral committee members **Dr. P. Calduwel Newton,** Assistant Professor, Department of Computer Science, Govt. Arts College, Thiruverumbur and **Dr. S. S. Manikadarasan,** Assistant Professor and Associate Director, Department of Computer Science, Adaikalamatha Arts and Science College,

Vallam, Thanjavur for their comments and suggestions in all the stages of my research journey to enhance the quality of my research work.

I would like to take this opportunity to thank **Prof. A. Charles**, the Head of the Department, **Dr. P. Jayapaul,** the former HoD, the **Faculty Members** and **Research Scholars** of Computer Science department for their support.

My special thanks to my fellow researchers-the 4th Research Group of **Dr. L. Arockiam** (**4G Team**). Their wonderful support and insightful research discussions were very helpful to accomplish my research in a systematic manner.

I extend my gratitude to my superiors **Rev. Sr. Angel, SAT** and **Rev. Sr. Carpus, SAT** and my community sisters for their care, prayers and support during the course of my research.

Finally, I am grateful beyond words to **my dearest family members**, St. Anne's Sisters, and friends who encouraged and inspired me throughout this intellectual journey.

A. Arul Anitha

ABSTRACT

The Internet of Things (IoT) is one of the hottest technologies which fuels innovation in all fields and every aspect of life. This technology transforms real-world objects into networks of smart devices. The 'things' in the IoT network are uniquely addressable. They can be monitored and controlled remotely over the Internet. Though it is a boon for enhancing modern life, the IoT has its own challenges while implementing it in large-scale environments. Among all the issues, security-related issues and challenges are the most prevalent and need to be considered. The smart devices in the IoT have limited resources in terms of memory, processing capacity, and energy. The widespread inclusion of such devices in the global network, combined with their severe constraints, exposes the IoT to new security challenges. Though there are many endeavours in security research, there are still a number of unresolved security issues.

IoT relies on the routing protocol for low power and lossy networks (RPL). The RPL protocol is vulnerable and is prone to several security threats and attacks. The Internet Control Message Protocol Version 6 (ICMPv6) consists of many RPL control messages for constructing the Destination Oriented Directed Acyclic Graph (DODAG) topology. Attackers modify these RPL control messages to initiate security breaches in the IoT networks.

Version Number Attack, DIS Attack, and DAO Attack are some of the attacks that are created using RPL control messages. These attacks consume more resources like energy, memory, and CPU time and reduce the lifetime of the constrained nodes. If these RPL resource attacks are undetected, the consequences can be severe. These RPL resource attacks target the precious resources of the IoT network and make them

exhausted soon. Moreover, these attacks increase the control traffic and lead to Denial of Service (DoS) attacks. Health care devices that are connected to the RPL networks put human lives at risk. For this reason, it is necessary to detect these attacks.

This research proposes techniques, namely VeNADet, DISDet, and DADTec, for detecting Version Number Attacks, DIS Attacks, and DAO Attacks, respectively, and an AdaBoost ensemble model (Ada-IDS) to safeguard the IoT networks from these three attacks. An Intelligent AdaBoost Architecture called ANIT-Ada, which integrates the three techniques and the Ada-IDS model.

The impacts of the three attacks are analysed separately in terms of power consumption, control overhead, and packet delivery ratio (PDR) and compared with the performance of the normal scenarios. Then the proposed techniques such as VeNADet, DISDet, and DADTec are implemented in the three attacker scenarios to detect Version Number Attack, DIS Attack, and DAO Attack respectively. The Ada-IDS utilizes the network traces of the normal and three types of attacker scenarios. The proposed three techniques are implemented in the root node, and the Ada-IDS ensemble model is installed in the border router of the ANIT-Ada architecture.

Whenever an attack is initiated from any node in the IoT network, the corresponding technique detects such attacks. The Ada-IDS monitors the incoming and outgoing traffic of the IoT network, and when it encounters any malicious activity from external sources or from the internal nodes, it analyses the packets and detects the attacks. Hence, Ada-IDS adds an additional layer of security to the RPL-based IoT networks. The proposed techniques are implemented in the Cooja Simulator. The ANIT-Ada architecture safeguards the IoT networks from RPL-based resource attacks.

LIST OF PUBLICATIONS

1. **A. Arul Anitha**, A. Stephen and Dr. L. Arockiam, "A Hybrid Method on Smart Irrigation System", International Journal of Recent Technology and Engineering (IJRTE), Volume 8, Issue 3, pp. 2995-2998, 2019. **(Scopus Indexed**)
2. **A. Arul Anitha** and Dr. L. Arockiam, "ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume 8, Issue 11, pp.2583-2588, 2019. **(Scopus Indexed)**
3. **A. Arul Anitha** and Dr. L. Arockiam, "Promoting a Clean and Hygienic Environment using IoT", International Journal of Recent Technology and Engineering (IJRTE), Volume 8, Issue 5, pp. 4722-4726, 2020. **(Scopus Indexed)**
4. **A. Arul Anitha** and Dr. L. Arockiam, "VeNADet: Version Number Attack Detection for RPL based Internet of Things", Solid State Technology, Volume 64, Issue 2, pp.2225-2237,2021. **(Scopus Indexed)**
5. **A. Arul Anitha** and Dr. L. Arockiam, "Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things", International Journal of Advanced Computer Science and Applications, Volume 12, Issue 11, pp.499-506, 2021. **(Scopus** and **WoS Indexed)**
6. **A. Arul Anitha** and Dr. L. Arockiam, "A Review on Intrusion Detection Systems to Secure IoT Networks", International Journal of Computer Networks and Applications", Volume 9, Issue 1, pp.38-50, 2022. **(Scopus Indexed)**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

6BR	6LowPAN Border Router
6LowPAN	IPv6 over Low -Power Wireless Personal Area Networks
AdaBoost	Adaptive Boosting
Ada-IDS	AdaBoost Intrusion Detection System
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANNIDS	Artificial Neural Network based Intrusion Detection Systems
APS	Artificial Pancreas System
AS	Autonomous Systems
BiLSTM	Bidirectional Long Short-Term Memory
CAGR	Compound Annual Growth Rate
CERP-IoT	Cluster of European Research Projects on the Internet of Things
CFS	Correlation-based Feature Selection
CNN	Convolutional Neural Network
COOJA	Contiki OS JAva simulator
CPU	Central Processing Unit
CSRC	Computer Security Resource Centre
CSV	Comma-Separated Values
DADTec	DAO Detection Technique
DAO	Destination Advertisement Object
DAO-Ack	DAO-Acknowledgement
DCNN	Distributed Convolutional Neural Network
DDoS	Distributed Denial of Service
DIO	DODAG Information Object
DIS	DODAG Information Solicitation

DISDet	DIS Attack Detection
DNS	Domain Name Systems
DODAG	Destination Oriented Directed Acyclic Graph
DoS	Denial of Service
DT	Decision Tree
DTM	Dynamic Threshold Mechanism
DTSN	Destination Trigger Sequence Number
ENIDS	Enhanced Network IDS
ESET	Essential Security against Evolving Threats
ETX	Expected Transmission Count
HTTP	Hyper Text Transport Protocol
IBOOS	Identity Based Offline/Online Signature
ICMPv6	Internet Control Message Protocol version 6
IDC	International Data Corporation
IDM	Intrusion Detection Mechanism
IDS	Intrusion Detection Systems
IETF	Internet Engineering Task Force
IGR	Information Gain Ratio
IIoT	Industrial Internet of Things
INTI	Intrusion detection of Sinkhole attacks on 6LoWPAN for InterneT of ThIngs
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
IPv6	Internet Protocol Version 6

KALIS	Knowledge-driven Adaptable Lightweight Intrusion Detection System
KNN	K-Nearest Neighbour
LDoS	Low-rate Denial of Service
LLN	Low Power Lossy Network
LPM	Low Power Mode
LSTM	Long-Short Term Memory
LTE	Long-Term Evolution
ML	Machine Learning
MLP	Multilayer Perceptron
MOP	Mode of Operation
MQTT	Message Queue Telemetry Transport
MRC	Maximum Response Code
MROFH	Minimum Rank Objective Function with Hysteresis
NB	Naïve Bayes
NFC	Near Field Communication
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology
OF	Objective Function
OPF	Optimum-Path Forest
P2P	Point-to-Point (P2P)
PBOM	Prospective Backward Oracle Matching Algorithms
PC	Personal Computer
PCA	Principal Component Analysis
PCAP	Packet Capture
PDR	Packet Delivery Ratio
PUF	Physically Unclonable Function

RF	Random Forest
RFC	Request for Comment
RFID	Radio Frequency Identification
RMS	Reconfigurable Manufacturing Systems
RNN	Random Neural Network
ROLL	Routing for Low-powered and Lossy networks
RPL	Routing Protocol for Low-power and Lossy Networks
SDN	Software-Defined Networking
SMA	Sub Miniature Version A
SQL	Structured Query Language
SVM	Support Vector Machine
SYN	SYNchronise
TCP	Transmission Control Protocol
UDGM	Unit Disk Graph Medium
UDP	User Datagram Protocol
UI	User Interface
UIDS	Unified Intrusion Detection System
UWB	Ultra-Wide Bandwidth
VeNADet	Version Number Attack Detection
VN	Version Number
VNA	Version Number Attack
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

Chapter – I

Introduction

CHAPTER – I
INTRODUCTION

1.1. Background

Internet of Things (IoT) is a collection of small uniquely identifiable devices with limited processing capacity and storage that are typically powered by batteries. Kevin Ashton, the forefather of the MIT Auto Identification Centre, first proposed the Internet of Things concept in 1999. IoT is a rapidly expanding network of networked "things" with sensors that collect and share data over the Internet without the need for human interaction [Ash, 21]. Virtually everything in IoT is "smart" because of the network connectivity, automatic data collection and taking actions analysis using sensors and actuators. Thus, various traditional devices are turned into smart products that can be monitored or controlled remotely.

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a standardised routing protocol for IoT. There are nodes and border routers in RPL networks that are connected to the Internet through a Destination-Oriented Directed Acyclic Graph (DODAG). IoT is vulnerable to routing attacks because most IoT devices have limited memory, storage, and energy. The rapid development of this resource-constrained technology and the inclusion of heterogeneous devices in the IoT lead to more challenges in this context [Jer, 21]. These attacks require mechanisms to detect them in order to safeguard the nodes in IoT network. A number of security challenges remain unresolved despite various research efforts. Determining the effectiveness of such attacks and proposing new techniques to detect these security threats and attacks in RPL networks are worthwhile endeavours.

1.2. IoT Terminology, Architecture and Applications

1.2.1. IoT Definition

There is no specific definition for IoT. Gartner defines IoT as the network of objects with embedded technology for sensing and interacting about their internal or external states [Gar, 21]. According to IBM, IoT is a huge network in which the connected things generate and share data [Jen, 16]. The Oxford Dictionary included the term IoT in the year 2013 and states IoT as: "A proposed development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data" [Dan, 18].

In General, IoT is a network of connected devices with distinct IP addresses that can detect, gather data, and communicate about their surroundings or about themselves, and embedded with those technologies.

1.2.2. A Typical IoT System

An IoT network consists of a collection of IoT devices, Sensors/Actuators, Connectivity, Gateway devices, Internet, User Interface, and Cloud storage. A typical IoT system with these components is illustrated in Fig.1.1.

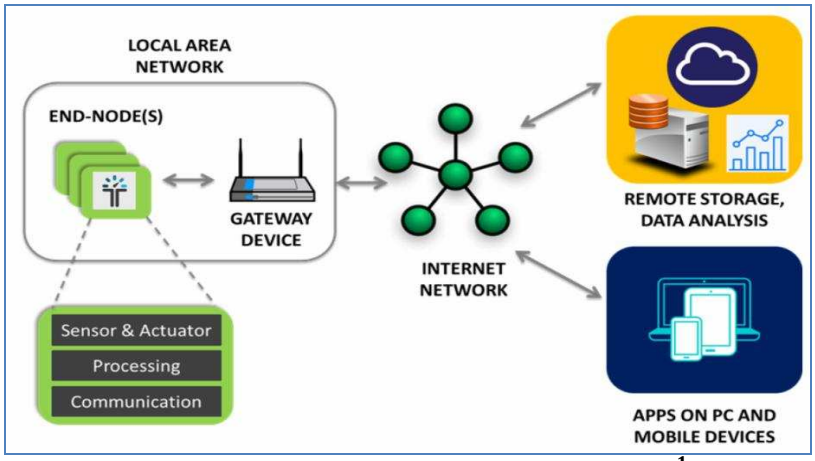


Fig. 1.1. The Components of IoT System[1]

[1]https://rb.gy/oq7ohu

- **End Nodes:** The 'things' in the IoT network are called end nodes. Depending on the application, the end nodes may perform sensing or actuation. Sensors and actuators are included in the end nodes' layer because they capture and represent the physical world in the digital domain. The properties of the devices and their environmental conditions are exchanged as it passes through physical devices to identify the physical world [Jes, 19].
- **Communication:** The sharing of information in IoT-based applications necessitates the use of appropriate communication technologies. In order to improve both efficiency and effectiveness, a better communication medium is required. The use of communication technology in IoT enables the delivery of specialized smart services by linking disparate things together. Devices used in the Internet of Things (IoT) must often run on very little power. Wireless Fidelity (Wi-Fi), Bluetooth, Zigbee, and Advanced Long-Term Evolution (LTE) are examples of IoT communication protocols. Radio Frequency Identification(RFID), Near Field Communication (NFC), and Ultra-Wide Bandwidth (UWB) for proximity services are examples of communication technologies that are relevant for the Internet of Things (IoT) [Fat, 17].
- **Gateway:** Gateway is also called as a fog node. Through the gateway, physical things are linked to the cloud. Gateway provides connectivity, security, and manageability between sensors/devices and the cloud. Gateways enable the devices in the IoT networks to periodically gather and transfer data to the cloud network [Jis, 21].
- **Internet:** Internet connectivity is the backbone of the Internet of Things. All the smart devices are connected to the Internet. Through the Internet, smart

devices are controlled remotely. Internet connectivity is the basic requirement for storing a large volume of sensor data and retrieving the data for analysis and decision making.

- **User Interface:** IoT data collection requires a medium for users to see and comprehend it. The user interface plays an important role here. A User Interface (UI) can be defined as the means by which a user and a computer system communicate with one another (or both). The voice-controlled devices and the dashboard buttons are also examples of user interfaces [Lev, 18].
- **Cloud Storage:** Large volumes of data are generated by IoT devices. This requires the cloud for the storage and retrieval process. The cloud infrastructure is made up of interconnected servers and storage. Big data collected from IoT processing units are stored in Cloud Units. It permits the transmission of data from smart objects and devices to the cloud. Machine learning and Artificial Intelligence (AI) are employed in these infrastructures to support IoT applications, which evaluate data from the device or thing in order to produce relevant information that may be used for service or decision making [Muh, 18].

1.2.3. Characteristics of IoT

IoT has some special characteristics that are listed below:

- **Intelligence:** The sensor in an IoT network senses its surroundings and based on the collected details, the actuator takes the decision automatically.
- **Interconnection:** The things on the Internet such as objects, sensors, actuators, people, etc. are connected to the Internet and other infrastructure using different types of communication technologies.

- **Addressing:** All ‘things’ in IoT are connected over the global network. Hence each thing should be uniquely addressed. The Internet Protocol version 6 (IPv6) is used for addressing. Using the 128 bits of the IPv6 protocol, the objects in the IoT networks are uniquely identified.
- **Scalability:** The Internet of Things' exponential expansion will necessitate the scalability of IoT. The IoT applications must be able to accommodate an expanding number of connected devices, users, application features, and analytical capabilities while maintaining a high level of service quality.
- **Heterogeneity:** The technologies, platforms, devices, and networks used in IoT are diverse in nature. Hence, it supports heterogeneity.
- **Resource-Constraint:** IoT is a resource-limited technology that has less energy, processing capacity, and memory. The devices are small in size and portable. Due to this resource constraint nature, IoT can be called a lightweight technology.
- **Dynamic and Self-Adapting:** IoT is dynamically adapting and taking actions based on the operational conditions, the user's context, or identified environments. These features are possible for IoT-connected devices and systems.
- **Self-Configuring:** A vast number of devices can work together to deliver specific functionality because of the IoT System's self-configuring potential. Smart gadgets can do everything on their own, from setting up networking to configuring themselves to downloading software and firmware updates.

1.2.4. Architecture of IoT

There is no rigid architecture for IoT. Depending upon the nature and need, the layers of the IoT architecture range from three-layer, four-layer, and five-layer.

The three-layer architecture is the most commonly used layered architecture with perception layer, network layer, and application layer. Each layer has its own security requirements too [Sye, 18]. The three-layered architecture of IoT is illustrated using Fig.1.2.

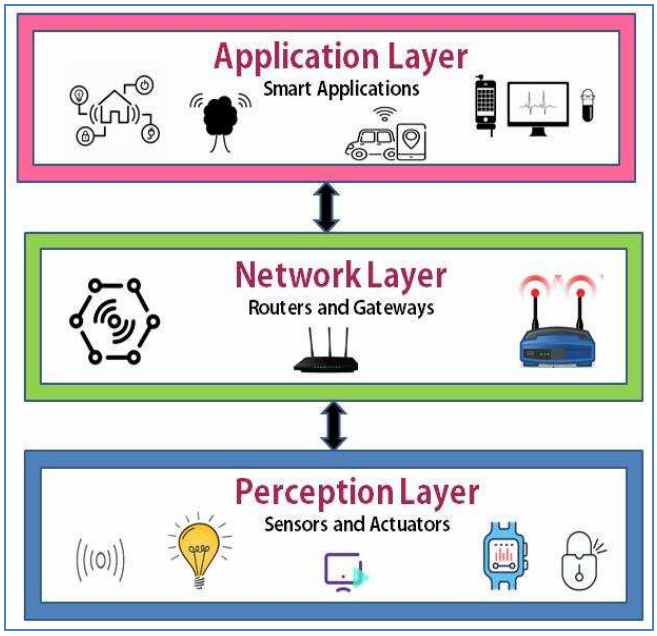


Fig. 1.2. Three-Layer Architecture of IoT

- **Perception Layer** is the physical layer that uses sensors to detect and collect data from the surrounding. This layer detects the smart item and the physical parameters for further processing. This primary goal of this layer is to retrieve the qualities of detecting items such as temperature, noise, leakage, and position through the use of sensors. The acquired data is then transformed into digital signals and sent up the stack to the network layer for processing.
- **Network Layer** is the central hub for IoT. It acts as a smart processing and management center. This layer transmits and processes the detected information from the objects in the perception layer. It links a wide range of network devices, including smart objects, servers, and more.

- **Application Layer** is often known as the business layer. It is the topmost layer of the Internet of Things architecture. To offer people application-specific services is the job of the application layer. Authentication, privacy data integrity are all guaranteed at this level of protection [Lat, 21]. In this research work, the widely used three-layer architecture is considered.

1.2.5. Applications of IoT

To make IoT more accessible, affordable, energy-efficient, and most importantly secure, new technologies and protocols are being added to the ecosystem. Due to high demand across several industries, the Internet of Things will continue to evolve. Due to the connectivity, simple implementation, advancements in sensors and wireless technology and the anytime, anywhere and anything features of IoT increase the applications of its domain in day-to-day life around the globe [Saf, 18]. Most common deployment of IoT applications are given below.

- **Home Automation:** The household devices such as home appliances, fans, lights, sensors or actuators with communication technologies and interface are considered as smart devices. The network of these devices forms a home Wireless Sensor Network (WSN). These smart objects are capable of communication, processing, sensing and actuating. They can sense, actuate, process data and communicate. At regular intervals these devices sense the surroundings and send the data to the home hub. The home hub is a device with storage capacity like a Personal Computer (PC), laptop, tablet or a smartphone. This home hub is capable of storing and managing the data collected by the sensors and acts as a mediator between the internal devices

and devices outside the home WSN. The Action to be performed based on the sensor data also can be commanded or controlled remotely by a smartphone or other devices [Keh, 21] [Bil, 17].

- **Smart Healthcare:** Smart Healthcare is one of the dominant domains of IoT. It necessitates storing aggregation and analysing the raw data collected from the healthcare related sensor data. Modern healthcare systems are under significant strain due to an ageing population and a corresponding rise in chronic illness, and demand for everything from hospital beds to doctors and nurses is at an all-time high. There must be a way to reduce the strain on healthcare systems while still providing high-quality care to patients who are most in need [Ste, 17]. The challenge is to combine data from various sensors and other sources that are heterogeneous in nature.
- **Industrial Automation:** As smart IoT devices improve in capability, they are becoming more widely used in industrial automation systems, accelerating reconfigurable manufacturing. The advent of Industry 4.0, also known as the fourth industrial revolution, ushers in a new era of highly customised manufacturing, as opposed to highly serialised manufacturing. The production resources in this vision are highly modularized, which gives them the adaptability they need to meet changing market demands. Reconfigurable Manufacturing Systems (RMS) enables rapid structural and functional modifications with little or no downtime [Vuk, 21].
- **Environmental Monitoring:** Due to industrialization, environment monitoring plays an important role to safeguard our surroundings. It consists of monitoring and analysing the quality of soil, water, air, weather and

monitoring the wildlife. Smart sensors are used to collect different environmental data and necessary actions are taken based on the collected data [Mee, 21].

- **Smart Transportation:** Vehicle-to-vehicle communication, smart parking, and other aspects of transportation are included in smart transportation. It provides better route ideas, parking reservations, energy-efficient lighting, and telematics for public transit, all while decreasing the risk of an accident or self-driving car [Fot, 19].
- **Security and Surveillance Systems**: With smart cameras, video feeds can be collected from as far as the end of the street. Smart security systems can recognise and prevent dangerous situations with real-time visual object identification [Fot, 19].
- **Smart Agriculture:** The soil quality checking, water level analysis, temperature level, humidity and precipitation are crucial variables to consider in smart agriculture. Irrigating the plants according to the weather condition and plant disease monitoring are also noteworthy research in smart agriculture [Tra, 19].

1.3. Issues and Challenges of IoT

The counterfeit hardware, heterogeneous platforms, data privacy, software faults, system management difficulties, security issues during communication, and remote device service management are vital issues for current IoT infrastructure. The issues and challenges of Internet of Things are depicted using Fig.1.3.

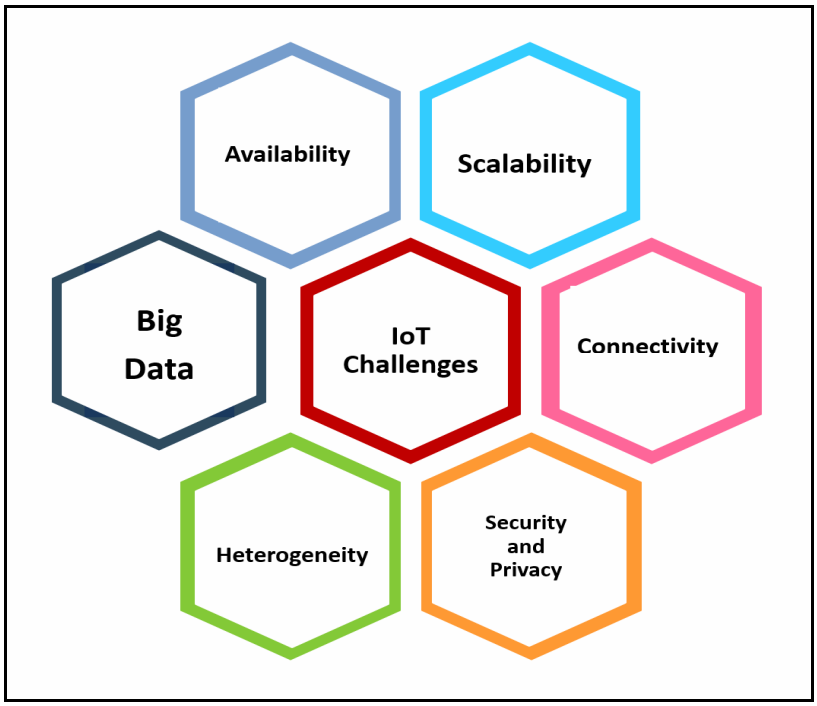


Fig. 1.3. Issues and Challenges of IoT

As it is given in Fig.1.3, there are various issues and challenges for IoT such as Connectivity, Security and Privacy, Scalability, Availability, Big Data and Heterogeneity. Among all these issues, Security is the most predominant challenge in IoT.

1.3.1. Security Challenges of IoT

When a new technology is launched, security is always the primary issue. Security is becoming a priority in any IoT network infrastructure. Many Internet-connected products are available without basic security features. Some manufacturers have proprietary security standards that are incompatible with the products of other manufacturers. Because these devices are connected to the Internet, hackers can take advantage of them. As the number of devices connected to the IoT applications grows, the risk of vulnerabilities increases as well [Rac, 21]. According to the Statistica research, there were 35.8 billion linked devices around the world in 2021,

which will be rising to 75.44 billion in 2025. The overall report of the Statistica Research is given in Fig.1.4.

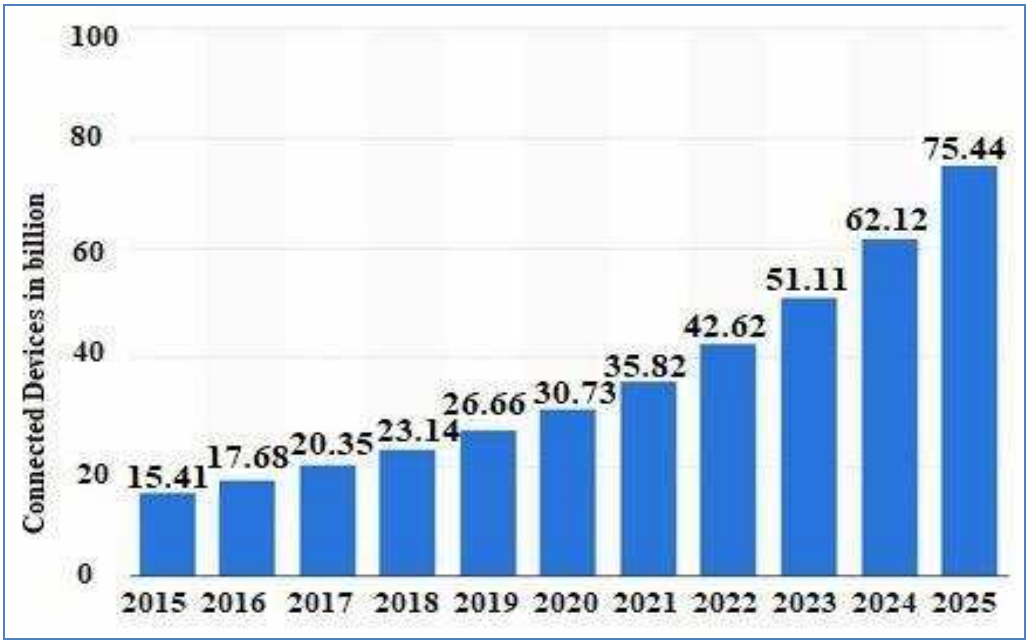


Fig. 1.4. No. of Connected Devices from 2015-2025 (Statistica)[2]

IoT application vulnerabilities increase as the number of connected devices increases. IoT technology has to ensure the security of devices, networks and data from IoT security attacks. While designing security solutions, these aspects must be considered.

- **Device Security:** The security mechanism has to protect the IoT devices from attackers. For smart operation, these devices require persistent Internet connectivity and energy. Tampering of these devices has to be prevented and the security of the devices must be guaranteed. The increasing demand for connected devices also increases the security spending for IoT devices [Ivi, 21].
- **Network Security:** Security in the network has also to be considered the secure communication between the IoT devices. It includes the construction of

[2]statista.com/statistics/471264/iot -number-of-connected-devices-worldwide

DODAG, maintaining the stability of the links, and securing end-to-end communication between the devices. The IoT devices necessitate Internet connectivity for communication with the outside world which hikes the associated threats and attacks. Though there are many tools and techniques to prevent the IoT from these threats, still network security research is an ongoing demand for preventing new types of attacks and security challenges [Vik, 19]. In this research, network security issues are considered.

- **Data Security:** IoT data are transmitted over the Internet and a large volume of data is warehoused in the cloud so that the data security has to be assured. As per the prediction of the International Data Corporation (IDC), by 2025, there will be 55.9 billion connected devices in the world which will generate more than 79.4 ZB by 2025, growing from 13.6 ZB in 2019 [Car, 21]. The voluminous escalation of the IoT devices boosts up the security threats and vulnerabilities too. The cloud provides capabilities for collecting, storing, processing, and managing massive amounts of IoT data continuously. As a result of the cloud's ability to aggregate and store data from all of these disparate sources, it can do predictive analytics. So security should be ensured for the IoT data residing in the cloud [Tia, 20]

1.4. Security Tools and Techniques for IoT

There are many tools and techniques in the market to provide security to the IoT nodes, networks, and IoT data. Some of the software and hardware solutions are enlisted here.

- **Antivirus/ Antimalware:** The vulnerabilities associated with software and applications used in the Internet of Things are prevented by the specially

designed Antivirus/Antimalware software. These IoT antivirus software products provide great protection against the threats and risks associated with connected devices. Essential Security against Evolving Threats (ESET) smart security, BullGuard, Avira Prime, Bitdefender Box 2, VipreAntivirus, Norton 360 with LifeLock, F-Secure Total with SENSE router, Avast Premium Security, and Comodo Cloud Antivirus are the important Antivirus software available in the market for providing better security to the IoT networks [Rad, 21].

- **Firewalls:** A firewall is hardware or software or a combination of both which acts as an investigator to analyze the entering and departing network packets and permit them only if they satisfy certain security rules. It is a medium to prevent the nodes in a network from external unauthorized sources. Due to the complexities and lightweight nature of IoT systems, the firewall is not deployed in embedded systems. Like the traditional network, IoT doesn't need a full-fledged firewall. In IoT networks, checking of the incoming and outgoing packets is enough. The edge devices of IoT require a dedicated firewall for resource-constrained devices. The firewall for IoT use-cases should be scalable and portable [Nav, 18]. Check Firewalls, Bitdefender Box, Cujo, Barracuda Cloud Generation Firewall, and Norton Core are some of the important embedded Firewalls available in the market for IoT.
- **Sandbox:** The glossary of National Institute of Standards and Technology (NIST) -Computer Security Resource Center (CSRC) defined Sandbox as "A system that allows an untrusted application to run in a highly controlled environment where the application's permissions are restricted to an essential

set of computer permissions. In particular, an application in a sandbox is usually restricted from accessing the file system or the network" [Nat, 21]. Sandbox is a technique to analyze cyber threats or attacks. It is an isolated virtual environment in which the unsafe software code can be executed without affecting the local resources or network. V-Sanbox, Cuckoo Sandbox, GoGuardian, Check Point's Threat Emulation, and Cybus are some of the available sandbox technologies for IoT.

- **Cryptography:** The large volume of data generated by the IoT devices in the real world are the target of potential cyber-attacks. Cryptography is one of the countermeasures to guard the IoT data from such attacks. It is a technique of protecting the data by using encryption with secret keys. Symmetric key and asymmetric key algorithms are the major two divisions of algorithms in cryptography. Due to restricted resources and complexities, cryptographic algorithms for IoT data should be lightweight. The data is divided into number of blocks, and the block cipher-based operations are involved in developing the lightweight cryptographic algorithms. TWINE, PRESENT, and OTR are the block-cipher based lightweight cryptographic solutions for IoT [Oka, 17]. E4 and Azure Sphere are also some lightweight cryptographic tools for protecting the IoT devices [Lil, 20]. There are hardware-based cryptographic solutions also existing. Physically Unclonable Function (PUF) offers hardware-related cryptography functionalities and provides better security to IoT devices [Zia, 21]. Lack of encryption, authentication, and compromised keys weaken the security of the nodes and IoT data.

- **Blockchain**: Blockchain is a linked collection of blocks with a decentralized database that utilizes a 'ledger' to record transactions in a chain of blocks. Each block in a blockchain comprises a hash of the previous block as well as transaction data, timestamps, and a digital signature, all of which are immutable [Dal, 21]. With the help of cryptography, communication technology, and the consensus mechanism, the blockchain has proven to be an extremely effective security system for IoT. The higher security standards of IoT are ensured by blockchain features, which include a Point-to-Point (P2P) decentralized network, an open and transparent multiparty consensus, and untampered data [Li, 21]. Blockchain acts as an invigilator to manage and secure IoT devices [Min, 18]. Helium, Chronicled, Arctouch, Filament, NetObjex, Hypr, Xage Security, and Grid+ are the topmost industries that provide blockchain-based security solutions to the Internet of Things [Sam, 21].
- **Intrusion Detection System (IDS):** An Intrusion Detection System is a software or hardware or a combination of both software and hardware tool. IDS monitors the network traffic, identifies the security threats in the network based on the algorithms and patterns used, and detects the attacks whenever it encounters any threats in the network [Jos, 20]. It safeguards the information system from intrusions that threaten confidentiality, integrity, and availability [Moh, 18a]. According to the placement of the IDS in the network, Intrusions detection systems are classified in the following manner [Elh, 18]:
    - **Centralized:** The IDS is implemented in a central node, either it can be a dedicated server or a router.

- **Distributed:** IDS is installed in all nodes or multiple nodes share the responsibility of monitoring the intrusions.
- **Hierarchical:** IDS is deployed in selected nodes and in a hierarchical manner the responsibilities are shared. Some IDS nodes have greater responsibilities and power than other nodes.
- **Hybrid:** Any permutation of the aforementioned strategies, the hybrid IDS is deployed in the IoT network.

Intrusion Detection System acts as an investigator that provides an additional layer of security to the IoT network and guards the nodes and network against external and internal attacks. The combination of IDS and Intrusion Prevention System (IPS) offer better security to the Internet of Things. Suricata, Snort,and Zeek are the important open-source IDS available for protecting the IoT networks [Bri, 21].

1.5. Motivation

Internet of Things (IoT) relies on the routing protocol known as RPL which is designed for small devices with limited resources. In RPL, ICMPv6 is the Internet Control Message Protocol which consists of many control messages for constructing the DODAG topology. Attackers modify these RPL control messages to initiate security breaches in the IoT networks. Version Number Attack, DIS flooding Attack, and DAO attack are some attacks that are created using ICMPv6 protocol. These attacks consume more resources like energy, memory, and CPU time and reduce the lifetime of the constrained nodes. If these RPL resource attacks are undetected, the consequences can be severe. These RPL resource attacks target the precious resources of the IoT network and make them be exhausted soon. Moreover, these attacks increase the control traffic and lead to Denial of Service (DoS) attacks and also

Distributed Denial of Service (DDoS) attacks. Health Care devices that are connected to the RPL networks put human lives at risk. For this reason, it is necessary to detect these attacks.

1.6. Problem Definition

IoT devices are resource-constrained. The voluminous inclusion of such devices in the global network tends to have a huge amount of vulnerabilities that can be easily exploited by an attacker. The unsecured IoT device that connected the Internet affects the security and resilience of the entire network. The traditional security tools and techniques are not suitable for IoT due to their heavyweight nature. The lack of encryption and authentication mechanism weakens the security of the IoT networks. Hence, an additional layer of security mechanism is required to safeguard the nodes and network. An intelligent Intrusion Detection System is necessary to monitor the IoT network traffic and analyze the packets to detect the attacks.

1.7. Research Objectives

The aim of the research work is to propose an intelligent system to detect the RPL-based resource attacks in IoT environment. The objectives of this work are enlisted below:

- To analyze the impacts of the Version Number attacks in IoT network using Contiki Cooja simulator and to propose a technique called **VeNADet** to detect the Version Number attacks
- To investigate the influences of the DIS flooding attacks using Cooja simulator and to propose a technique termed **DISDet** for the detection of such attacks

- To create a DODAG with DAO attacker and examine the negative effects caused by the attack and to propose a technique called **DADTec** to detect the DAO attacks
- To implement the normal and attack scenarios and to collect the packets from the simulation environments for developing an AdaBoost ensemble Model named **Ada-IDS** to enhance the detection process
- To develop an architecture called **ANIT-Ada** by integrating the three techniques and the Ada-IDS model

1.8. Scope and Limitations

This research work is to propose an intelligent system for securing the IoT network from RPL resource attacks. The proposed solution is suitable for the network layer of the IoT architecture. Three attacks such as Version Number attack, DIS flooding, and DAO attack are considered in this research. The vulnerabilities of these attacks and their consequences are analyzed by implementing the RPL network in Contiki Cooja Simulator. Three techniques named **VeNADet, DISDet,** and **DADTec** are proposed to detect these three attacks respectively. The network traces with normal and attacks packets are collected from the 6LoWPAN analyzer tool. The collected network traces are converted into comma-separated values **(.csv)** format and called as icmpv6.csv dataset. This dataset is used to develop an ensemble machine-learning based Intrusion Detection System termed **Ada-IDS** to detect the RPL resource attacks. The scope of the research work is illustrated using Fig.1.5.

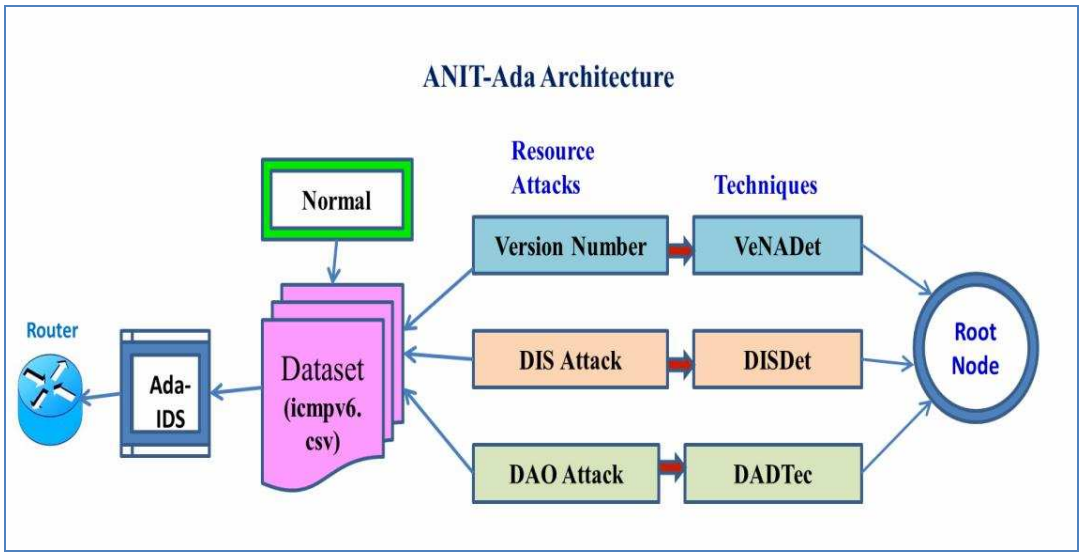


Fig. 1.5. Research Diagram

As it is given in Fig.1.5 the proposed Ada-IDS intelligent system is installed in a gateway device (border router) and the three techniques are implemented in the root node to protect the network from malicious events. The Ada-IDS model and the three techniques are combined to develop the ANIT-Ada architecture.

This proposed system safeguards the network from RPL resource attacks. The research work is limited to the detection of these three attacks only. The impact of the attacks such as power consumption, packet delivery ratio (PDR), and control overhead are analyzed in this work. Other metrics such as jitter, throughput, delay, latency, and bandwidth are beyond the scope of the research work.

1.9. Thesis Layout

Chapter 2 provides an in-depth look at the Intrusion Detection System, its working principles, and common routing attacks in RPL based Internet of Things and existing techniques for detecting such attacks

For a better understanding of the effects of RPL based resource attacks on IoT networks and their security mechanisms, **Chapter 3** experimentally investigates

Version Number Attack and provides insights and suggestions to reduce the effects of the investigated attack by proposing a technique called VeNADet.

Chapter 4 examines the DIS flooding attacks and their detrimental effects on the IoT network and proposes a solution termed DISDet to reduce their effects.

Chapter 5 investigates another ICMPv6 based attack, the DAO attack. DADTec is the security mechanism proposed to detect and overcome the negative impacts of the DAO attack.

In **Chapter 6,** the Ada-IDS ensemble model is elaborately discussed. It is an intelligent AdaBoost Ensemble Attack model to detect the RPL-based resource attacks such as Version Number Attack, DIS flooding Attack, and DAO attacks. This ensemble model is developed by capturing the network traffic from normal and attack scenarios. This Ada-IDS ensemble model is implemented in the Gateway (Border Router) of the IoT network for providing an additional layer of security. The individual techniques like VeNADet, DISDet, and DADTec are installed in the root node so it provides an inner layer of security to the IoT nodes. These three techniques and the Ada-IDS model are integrated to develop the **ANIT-Ada** architecture.

In **Chapter 7,** the overall summary of the Chapters and the research contributions are briefly narrated and the Thesis is concluded.

1.10. Chapter Summary

The enormous growth of IoT in all fields makes it an essential part of our day-to-day life. Though IoT gives a lot of benefits to the people, it has a number of challenges too. Among all the challenging issues, security is the predominant one. The voluminous inclusion of the connected devices and resource-constrained characteristics

of IoT lead to enormous security vulnerabilities. Hence, there is a need for safeguarding the IoT devices, data collected from the sensor nodes, and the IoT network from security threats. This Chapter explains the rudimentary concepts of IoT and the proposed research work briefly. The Problem definition, Research objectives, Motivation, and the Scope of the research work are discussed in detail. The various security tools and techniques available in the market are also highlighted.

The second chapter elaborately reviews the literature, explains different types of Intrusion Detection Systems in detail; the research issues are identified; the research flow diagram is explained briefly and the strengths and weaknesses of the existing attack detection techniques are tabulated.

Chapter – II

Literature Review

Chapter – II

Literature Review

2.1. Introduction

The Internet of Things (IoT) is a robustly evolving trend that incorporates technical, scientific, social, and economic implications. The IoT refers to an emerging paradigm consisting of a continuum of uniquely addressable things communicating with one another to form a worldwide dynamic network [Bor, 14]. The Cluster of European Research Projects on the Internet of Things (CERP-IoT) defined the Internet of Things as "a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual things have identities, physical attributes, virtual personalities, and use intelligent interfaces, and are seamlessly integrated into the information network" [Jai, 09].

The main intention of this chapter is to provide a complete overview of security challenges and attacks in the IoT environment and to analyse the security tool called the Intrusion Detection System (IDS) for securing smart devices and IoT networks. This chapter addresses security and privacy concerns related to the Internet of Things (IoT) and the countermeasures to overcome the security challenges.

This chapter continues to elucidate the role of machine learning algorithms for enhancing the performance of intrusion detection techniques, the analytical survey of the literature, and the research flow diagram. The various types of security attacks with a special highlight on RPL resource attacks and intrusion detection techniques for Internet of Things are reviewed to accomplish the research work.

2.2. Routing Protocol for Low-Power and Lossy Networks (RPL)

In order to regulate the routing protocol for IPv6 based objects, the Internet Engineering Task Force (IETF) created a working group called Routing for Low-powered and Lossy Networks (ROLL) in 2008, and this ROLL team standardized the RPL in 2012. RPL is adaptable to routes in a different path when it encounters problems in the existing one in heterogeneous networks [Jay, 21].

2.2.1. RPL Control Messages

RPL is a distance-vector protocol that finds a path dynamically between nodes. In RPL, the majority of network communication is directed upwards towards a single node called "root" (border router). RPL creates and upholds a logical topology called Destination Oriented Directed Acyclic Graph (DODAG). Each DODAG is identified by an RPL Instance ID, a DODAG ID, and a DODAG Version Number. The DODAG consists of a root node and a number of child nodes in a tree like structure [Con, 19]. For constructing the DODAG, RPL uses a number of Internet Control Message Protocol Version 6 (ICMPv6) messages. There are six fields in this ICMPv6 message, such as Type, Code, Checksum, Security, and a message body comprising of a Message Base and a number of options. The structure of the ICMPv6 is given in Fig.2.1.

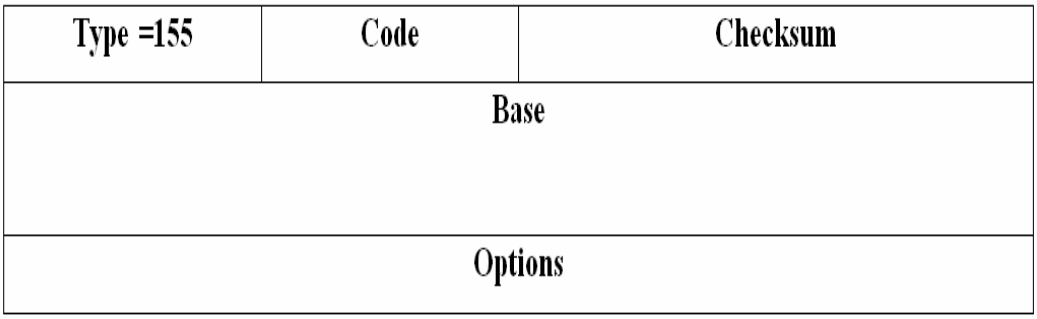


Fig. 2.1. ICMPv6 Message Format [Ana, 18]

Fig.2.1 explains the different types of fields in the ICMPv6 protocol. There are two types of messages in ICMPv6, such as error messages and informational messages. In the Type field, if there is a zero in the high-order bit, then it is treated as an error message. If there is one in the high-order bit, then it is an informational message. Hence, if the Type field contains up to 127 values, it is an error message; from 128 to 255, it is an informational message. If the Type field contains the value '155', then the informational messages are related to RPL control messages. In which the code field is used to identify the four types of control messages as given in Table 2.1

Table 2.1. Control Messages for the Code field [Ana, 18]

Code	Message
0x00	DODAG Information Solicitation (DIS)
0x01	DODAG Information Object (DIO)
0x02	Destination Advertisement Object (DAO)
0x03	DODAG Destination Advertisement Object Acknowledgement (DAO-ACK)

2.2.2. DODAG Construction

The DODAG root issues a DIO message to its neighbors and initiates the DODAG construction process. The DIO message consists of the routing metrics and constraints like the DODAG ID, the rank, and an Objective Function (OF). DODAG ID is the 128-bit address of the root node. Rank is the position of node in the DODAG with respect to the root node. The rank value is increased in downward and increased in upward direction. Objective Function defines how a RPL node selects and optimizes within a RPL instance based on the objects available. Hop Count, Expectation Transmission Count (ETX) and Minimum Rank Objective Function with Hysteresis (MROFH) are examples for Objective Functions.

When a node receives the DIO message, it adds the sender of the DIO message to its parent list, computes its own rank according to the Objective Function, and passes on the DIO message to its descendants with the updated rank information. When the DIO message reaches the leaf node, the route towards the root node is built through its parent list. To form end-to-end communication from root to other nodes, the leaf node issues a Destination Advertisement Object (DAO) control message to broadcast reverse route information and record the visited nodes along the upward routes. After receiving a DAO message, the DODAG root replies with a Destination Advertisement Object ACK (DAO-ACK) message to the source of the DAO message [Con, 19]. Hence, by sending the DIO, DAO, DAO-ACK, and DIS messages, the DODAG topology is constructed and it is directed towards the root node from other nodes.

Using the aforementioned control messages, the DODAG is built for communication among the nodes in the network. The DAO message is a unicast message that is sent by the child node to its parents after receiving the DIO message from its parent node. The DAO is used to disseminate the backward route details to record the visited nodes along the reverse path [Hon, 17]. The DODAG construction process with DAO and other control messages is explained using Fig. 2.2.

In Fig.2.2, there is a root node 'R' and four nodes ‘A’, ‘B’, ‘C’, and ‘D’. First, the root node ‘R’ broadcasts the DIO message to its adjacent nodes. Since node ‘A’and ‘D’ are its neighbor nodes, they receive the DIO message and send back the DAO message to ‘R’. Meanwhile, the nodes 'B' and ‘C’ send the DIS message to their neighbors in order to join the network.

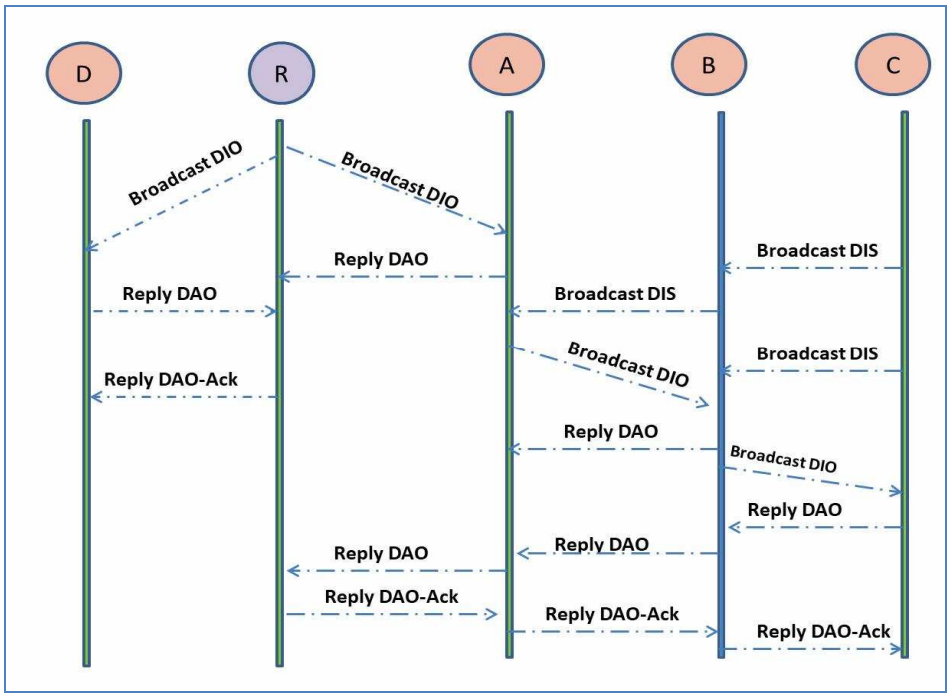


Fig. 2.2 DODAG Construction Process

The receivers of the DIS messages ‘A’ and ‘B’ are not yet connected to the DODAG. So, at present they never respond to the node ‘C’. After node ‘A’ joins the DODAG, it sends the updated DIO message to its neighbor 'B' and this process goes on. After getting the DIO message from the node ‘B’, the node 'C' replies with a DAO message to ‘B’ and the same is forwarded to node ‘A’ and from node ‘A’ it reaches the root node 'R'. In the same manner, the DAO-ACK message is sent from the root node to the leaf node. By undergoing these processes, the DODAG is constructed. The constructed DODAG topology with nodes ‘R’, ‘A’, ‘B’, ‘C’, and ‘D’ is shown in Fig. 2.3.

All nodes in Fig. 2.3 are directed towards the root node ‘R’. In general, the communication of the nodes is towards the upward direction. Downward data transmission is also permitted when it is required. The nodes send messages to the root node and other nodes using hop-by-hop method of data transmission.

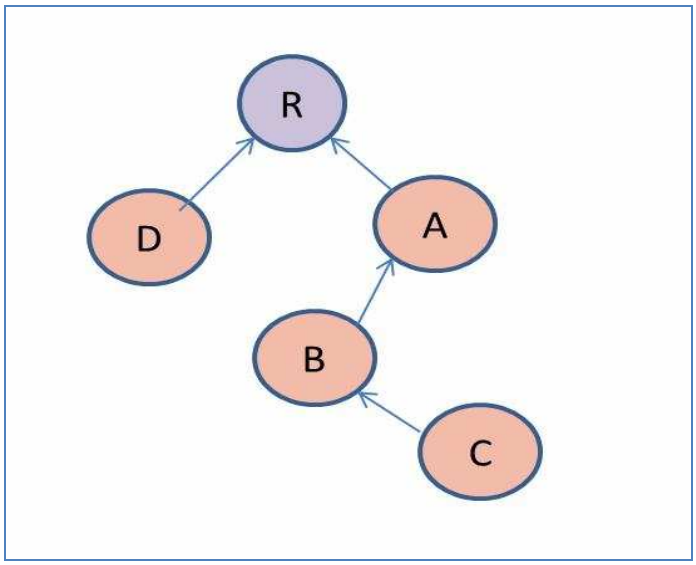


Fig. 2.3. DODAG topology

2.3. IoT and its Security Challenges

In a nutshell, [Key, 16] discussed the definition, characteristics, technologies, architecture, and applications of IoT and also highlighted the research issues and challenges regarding security, interoperability, data management, and energy issues related to IoT. In their detailed survey, [Vip, 17] discussed the history, background, and statistics of IoT and provided a security-based analysis of its architecture. The authors provided a set of security challenges and security requirements in the perception layer, network layer, support layer, and application layer of the IoT architecture. They also presented taxonomy of security issues and challenges as well as existing defense mechanisms for the IoT environment.

Threats and vulnerabilities rise robustly as the connected devices in the IoT increase. The security issues of the IoT are becoming troublesome with the limited resources and weak capabilities. Moreover, the enormous growth and adoption of IoT devices in day-to-day life indicates the urgency of addressing these security threats before deployment. The security market from 2019 to 2025 is given in Fig.2.4.

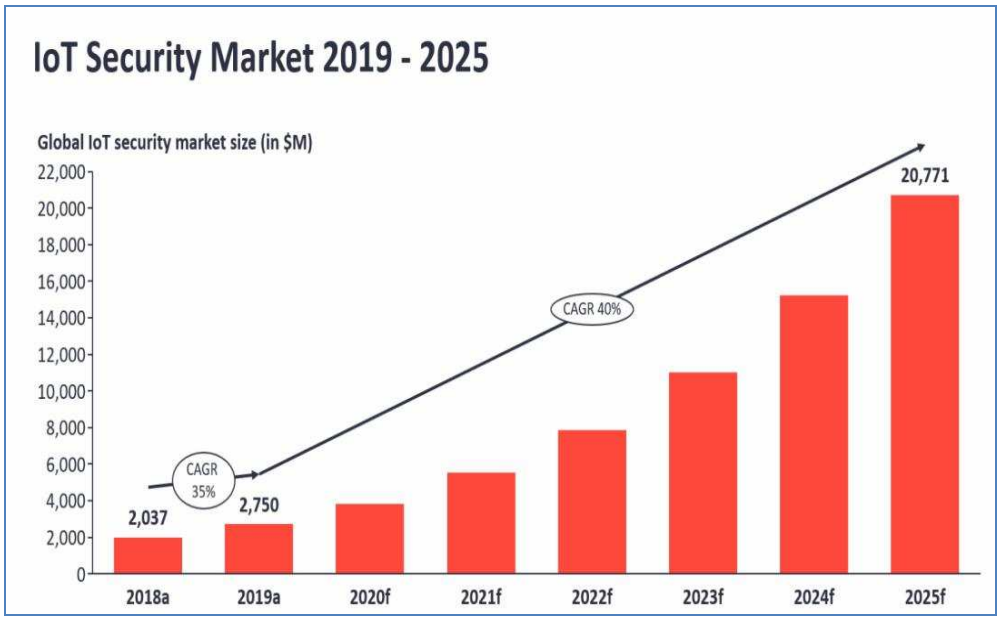


Fig. 2.4. IoT Security Market (2019-2025)[3]

According to IoT Analytics Research 2020, the IoT security market size was $2,750 million in 2019, and it is estimated to be the same as $20,771 million in 2025. The increase in the Compound Annual Growth Rate (CAGR) is 40% from the year 2019 to 2025. This underlines the rapid growth of security issues in IoT environments and the importance of securing the devices against various attacks.

2.3.1. Classifications of IoT Attacks

The security attack is an attempt to obtain, alter, destroy, remove, implant or reveal information without authorized access or permission. The security threats in the IoT environment are inevitable. Intrusions or attacks in the IoT and other networks can be caused in three ways:

1. Attacks by external attackers who gain access to the network, and then the systems explore various attacks against the network.

[3] IoTAnalyticsResearch 2020

2. Attacks by the internal attackers who are authorized to certain level of privileges but attempt to use additional unauthorized access.
3. The authorised internal attackers who misuse the privileges given to them.

Security threats and attacks are widely classified as Internal Attacks and External Attacks. In another way, these attacks are classified as Active Attacks and Passive Attacks.

- **Internal Attacks:** Internal attacks are initiated by the authorised people of the IoT network. They misuse their given privileges as well as pretend that they have other privileges which they might not have been granted. The attacker tries to execute malicious codes without the knowledge of the user.
- **External Attacks:** External attackers may pretend to be insiders and may execute malicious code on the IoT network. The attackers access the smart devices of the IoT devices remotely and attempt various types of attacks against the IoT networks.
- **Active/Passive Attacks:** All possible attacks in the IoT atmosphere are either passive or active. Passive attacks simply monitor the system activities, data traffic, and eavesdrop to expose information. Passive attacks are less harmful and create less damage to IoT devices and networks. Contrary to passive attacks, active attacks are dangerous and directly affect the communication systems in IoT environments. Active attacks can circumvent or destroy smart devices and can discard information or data.

In their work, [Ras, 18] presented an Attack Graph which could detect vulnerabilities in the Rank property. By exploiting these vulnerabilities, an attacker could invoke several attacks, leading to traffic compromise, network optimization, network isolation, and excessive resource consumption. [Hez, 18] underwent a

comprehensive study on the attacks in the IoT environment based on a building-blocked reference model. They classified the security attacks on IoT networks as physical-based attacks, Protocol-based attacks, network layer-based attacks, and IoT software-based attacks. An overview of all possible attacks in these categories was discussed in detail.

Based on the IoT architecture and layers, [Jyo, 17] classified all possible attacks related to IoT into four categories, such as physical attacks, network attacks, software attacks, and encryption attacks. They also compared these four attacks by considering parameters like damage level, detection chances, vulnerability, attacker location, possibility of detection, existing solution and limitation, etc. The layer-wise attacks were highlighted in this paper.

[Yan, 18] presented a review of recent research related to attacks in IoT networks. They provided a taxonomy of the cyber security attacks on IoT networks based on device, location, access level, information damage level, host promise, strategy, and protocol-based. In this work, the authors also explain the major attacks on IoT related networks.

[Ram, 18] conducted a survey on various types of IoT network attacks and countermeasures. In their survey, the IoT security attacks were classified as physical attacks, side-channel attacks, cryptanalysis attacks, software attacks, and network attacks. They explained the layered architecture of Radio Frequency Identifiers (RFID) and Wireless Sensor Networks (WSN). Denial of Service (DoS) attacks, wormhole attacks, spoofing attacks, man-in-the-middle attacks, replay attacks, and Sybil attacks are the major attacks identified on the IoT networks.

2.3.2. RPL-based Security Attacks

There are three modes in RPL to impose a secure route. They are authenticated, preinstalled, and unsecured modes. The data is transmitted without any security mechanisms in an unsecured mode. The preinstalled mode guards the RPL messages using cryptographic solutions, and the keys are already present in all nodes during the initialization time. In the preinstalled mode, RPL messages are protected by security keys that are assumed to be present in each node at boot time. In authenticated mode, the nodes receive the key after a successful authentication process. Though there are some security mechanisms, they are left unspecified in the RPL standard. Hence, RPL is susceptible to several security threats and attacks [Ant, 20].

RPL is a delicate protocol that is unprotected and vulnerable to various security threats. The majority of RPL attacks increase the negative impact on IoT networks and consume more network resources like energy, memory, and processing in the constrained RPL nodes [Sem, 20]. The ICMPv6 control messages of RPL are accountable for various functions like router discovery, neighbour discovery, packet sending, and error reporting. On the other hand, this protocol is vulnerable to various attacks, including Denial of Service (DoS), Distributed Denial of Service (DDoS), scan attacks, man-in-the-middle attacks, and protocol exploitation [Oma, 17].

2.3.3. RPL Resource Attacks

Anthea Mayzaud et al. [Ant, 16] classified the RPL-based attacks into three categories: attacks targeting network resources, attacks modifying the network topologies, and attacks related to network traffic. The RPL control messages of the ICMPv6 protocol are manipulated and misrepresented by the attackers in order to create attacks on the resources, such as attacks targeting the network and traffic.

Among all these attacks, RPL resource attacks are the predominant ones that lead to DoS and DDos attacks. This research focuses on RPL resource attacks in the Internet of Things. The classification of RPL resource attacks is depicted in Fig. 2.5.

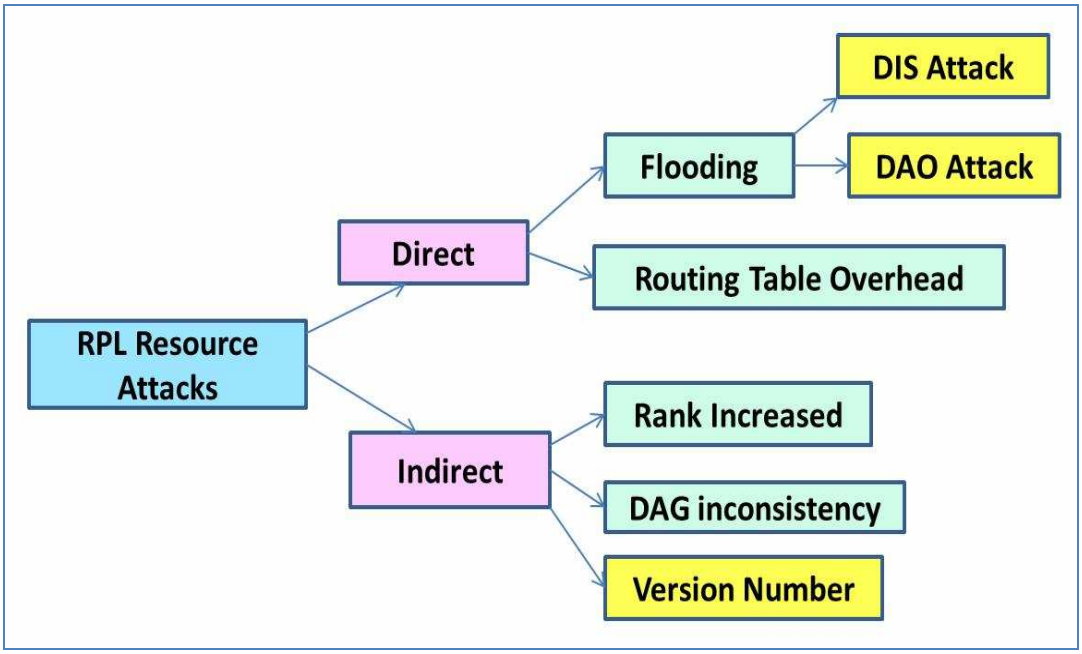


Fig. 2.5. RPL Resource Attacks

Direct and indirect resource attacks are two forms of RPL resource attacks. Flooding attack and routing table overhead are examples of direct resource attacks that consume resources directly. The resources are consumed indirectly by the Rank Increased Attack, DAG inconsistency and Version Number Attacks.

The RPL-based IoT network has been subjected to several flooding attacks. For this study, the DIS flooding attack and DAO flooding attack from the direct attack category are chosen, as well as the Version Number Attack from the indirect resource attacks category.

When the Version Number attack, DAO attack, or DIS flooding attack is initiated by the attacker, the normal nodes are forced to execute unneeded work in order to deplete their resources.

- **Version Number Attack:** Each DIO control message has an important field called the version number. It is propagated unaltered down the DODAG graph and is only updated by the root whenever the DODAG has to be reconstructed. This rebuilding of DODAG is also known as global repair. When it transmits DIO messages to its neighbors, an attacker can modify the version number by fraudulently incrementing this field. This attack forces the entire DODAG graph to be rebuilt, which isn't necessary. Due to the presence of this attack in RPL-based IoT networks, resource consumption increases significantly. The communication links also have availability problems, as the resources are depleted in the nodes [Zah, 20].
- **DIS Attack:** DIS flooding attack is created by deluging the neighbor nodes with multiple DIS messages. A node either waits for a DODAG Information Object (DIO) or sends a DODAG Information Solicitation (DIS) control message to solicit DIOs from adjacent nodes in order to join the DODAG. Sending Multicast DIS messages, on the other hand, resets the timer that controls the transmission rate of DIOs to its minimal value, causing network congestion with control messages. Malicious nodes can utilize the Multicast DIS solicitation method to launch the DIS attack, due to the resource-constrained nature of RPL-LLNs, their lack of tamper resistance, and the

security flaws in RPL [Shu, 21]. The DIS attack has the potential to harm RPL networks severely. Control packet overhead and power consumption are the predominant issues [Fai, 21].

- **DAO Attack:** DAO attack is another type of flooding attack by overloading the parent nodes with DODAG Advertisement Object (DAO) control messages. The DAO message is used to create the routes in a descending manner. It was not specified in the RPL standard about when and how the DAO messages are transmitted. To induce network overhead, the attacker sends the DAO message repeatedly. In three scenarios, the child node unicasts the DAO message to its parents: When it gets a DIO message from its preferred parents, when the parents are altered, and when certain problems are found [Nan, 19].

These resource attacks seek to exhaust the energy, memory, or processing power of the node. This may have an influence on the network's availability through congested available links, and hence on the network's lifespan may reduce severely [Ant, 16].

2.4. Intrusion Detection Systems (IDSs) for IoT

Intrusion detection is the act of monitoring and possibly preventing the malicious activities of intruders. It is a network security tool that consists of software or a combination of both hardware and software to protect traditional networks. It can be used to monitor all sorts of activities in the network. If there is any malicious activity in the network, the Intrusion Detection System (IDS) detects the intrusions, alerts the administrator, logs the attacks for forensic activities, isolates the intruder and also disconnects the connection path of the intruder. The various functions of IDS are depicted in Fig.2.6.

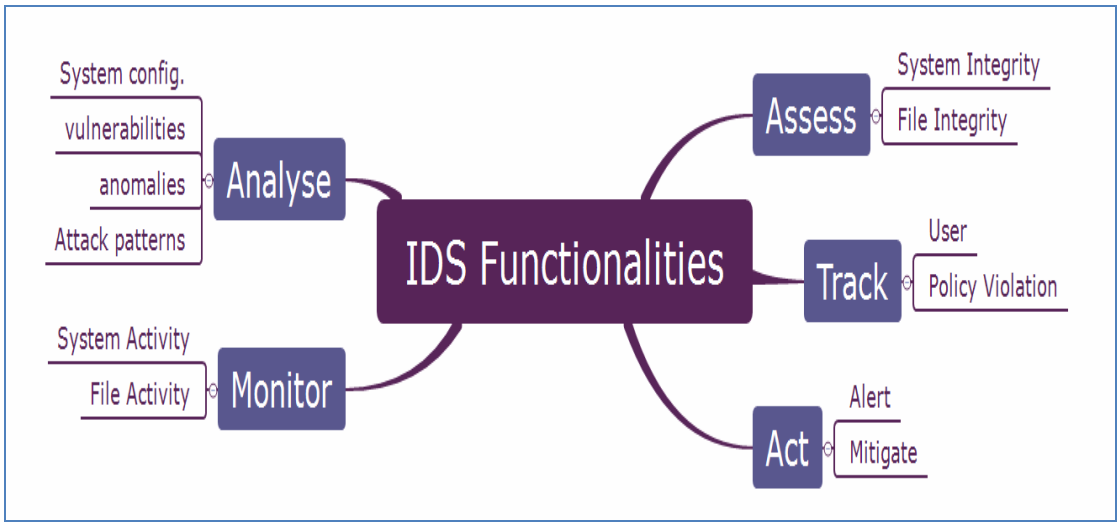


Fig. 2.6. Functionalities of Intrusion Detection Systems

Fig. 2.6 shows that IDS can monitor, analyse, assess, track, alert and mitigate attacks in IoT networks. According to [Bru, 17], IDSs are at a mature level in the traditional networks. The IDSs that are developed for traditional and wireless networks are not suitable for IoT, since IDSs consume more memory, processing capability, and energy. Because of these constraints, finding IoT nodes with higher computing capabilities to support IDS agents is very difficult. So, there is a need for lightweight IDS models to adapt to the IoT constraints. The authors elaborately described the taxonomy of Intrusion Detection Systems using different strategies, including placement, detection method, security threat, and validation strategy. They also narrated the research issues and challenges in all strategies.

To detect intrusions, attacks and malicious activities in the IoT environment, Intrusion Detection Systems are used. IDSs are networking security components that are widely used to protect network environments from attacks and malicious activities. They normally monitor the behaviour of the individual device or the network. Hence, normal and abnormal behaviour should be defined to differentiate them.

2.4.1. Types of IDS based on its Position

IDS may also be found in different forms and classified in different ways. In the IoT environment, the IDS can be installed in the border router, specific nodes, or in every physical object in the IoT networks. According to the placement strategy, the Intrusion Detection Systems are divided into three categories, such as Distributed IDS, Centralized IDS and Hybrid IDS.

a. Distributed IDS (Host-based IDS)

According to this placement strategy, each node in the IoT network is responsible for monitoring and detecting attacks. So, the IDSs are placed on almost all the nodes in the network. The responsibilities of detecting attacks are shared amongst the distributed IDS [Elh, 18]. Because the IDS deployed in each node, this strategy must be optimised due to the resource constrained characteristics of the IoT. Many techniques have been developed to address this issue.

[Oh, 14] defined a lightweight distributed algorithm for malicious pattern detection which was efficient in matching the attack signatures and packet payloads. The authors suggested using auxiliary shifting and early decision techniques to reduce the number of matches needed for detecting attacks. [Lee, 14] suggested a lightweight distributed IDS based on energy consumption in 6LowPAN networks.

[Cer, 15] recommended a distributed solution called INTI (Intrusion Detection of Sinkhole Attacks on 6LoWPAN for InterneT of Things) by combining notions of trust and reputation with watchdogs to monitor, detect, and mitigate the attacks. They classified the nodes as leader, associated, or member nodes and formed a hierarchical structure. According to the change in the network, like network reconfiguration or an attack occurrence, the role of each node can change over time. After that, each node

monitors a superior node by estimating its incoming and outgoing traffic. Whenever an attack is detected by any node, it alerts other nodes and isolates the attacker. The authors did not test the solution's effectiveness in low-capacity nodes. Since the distributed IDS has a hierarchy among themselves, this type of distributed IDS can also be termed a hierarchical intrusion detection system.

[Osa, 21] recommended distributed IDS using Deep Blockchain technology and Bidirectional Long Short-Term Memory (BiLSTM). This system detected DoS, DDoS, port scanning, and other attacks from UNSW-NB15 and BoT-IoT datasets effectively. It is suitable for IoT and cloud architectures. For real-world implementation, it requires further fine-tuning.

[Muh, 20] introduced a Blockchain-based distributed IDS for IoT using Machine Learning Algorithms. Here, spectral partitioning is used to divide the IoT network into Autonomous Systems (AS). Selected AS nodes are responsible for traffic monitoring in a distributed manner. The Support Vector Machine (SVM) algorithm is applied for training the dataset. This system detects botnets and routing attacks. [Gon, 20] suggested a cloud-based distributed attack detection technique for the IoT using Deep Learning algorithms. It comprises two security mechanisms, such as a Distributed Convolutional Neural Network (DCNN) and a cloud-based temporal Long-Short Term Memory (LSTM) model. The proposed mechanism detects phishing attacks, DDoS attacks, and botnets. This method can detect the attack at both the node level and the cloud level.

[Amj, 18] proposed a multi-agent-based intrusion detection system (IDS) based on the Naive Bayesian algorithm for detecting Distributed Denial of Service (DDoS) attacks in an IoT layered architecture. In this work, the multi-agents along

with Naive Bayesian algorithm were implemented in selected IoT devices throughout the network. The agents were classified as system monitoring, communicating, collector, and actuator agents. The distributed multi-agents in this approach share the responsibility of intrusion detection and reduce the workload of the individual nodes. The agent nodes could communicate with other agents too, whenever required. The authors used sensors to gather the information, and the collected information was analysed to check whether there were any attacks on the network. Malicious nodes and their activities were monitored and reported to the administrator or to the IoT objects. The authors did not consider a solution for the low-capacity systems.

b. Centralized IDS (Network based IDS)

In this centralised IDS placement strategy, the Intrusion Detection Systems are placed in a centralized component like a border router or a dedicated system. As the border router connects the IoT network to the Internet, it is very simple to implement centralised IDS in the IoT. External intruders can be easily detected by the centralized IDS, since all outside data packets enter into the IoT environment through the border router. Hence, when the Intrusion Detection System is deployed in the border router, it can easily monitor, analyse and drop the malicious data packets when it detects any attacks. Contrarily, internal attack detection is difficult in this approach since it necessitates thorough monitoring and analysing of the low-power lossy networks (LLN). The centralized IDSs detect not only external intrusions but can also detect some internal attacks like selective forwarding attacks. A typical centralized IDS tool is illustrated in Fig. 2.7.

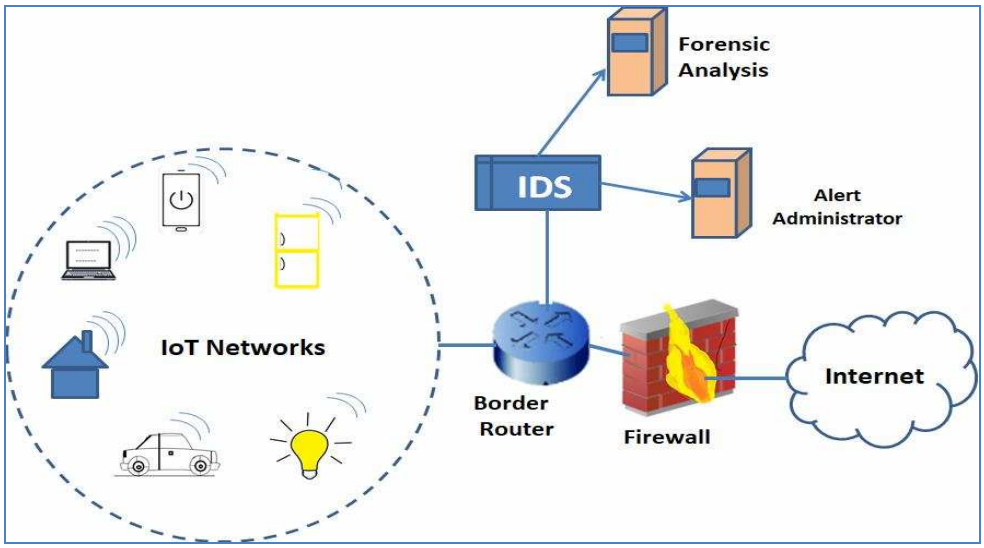


Fig. 2.7. Centralized IDS for Internet of Things

In Fig. 2.7, the smart devices are connected to the Internet via a border router. The IDS is implemented on the border router. It monitors all IoT network related activities and, whenever an intrusion arises, the IDS alerts the administrator. It also logs the events for forensic analysis.

[Mid, 17] suggested a centralized intrusion detection system for the IoT environment called the Knowledge-driven Adaptable Lightweight Intrusion Detection System (KALIS). The authors deployed KALIS in a centralized device like a border router. KALIS autonomously collects knowledge about the features of the monitored network and entities and leverages such knowledge to configure dynamically the most effective set of detection techniques. According to the authors, KALIS outperformed well in detecting DoS, routing, and conventional attacks when compared to the traditional Intrusion Detection Systems.

[Azk, 18] recommended an innovative IDS based on Software Defined Networking (SDN) that separates the data and control planes, resulting in a programmable network architecture with centralized control. SDN is programmable,

so it makes the network flexible. Here, the control mechanism is shifted to a centralized controller of the system. In this research work, Mininet 2.0 was used for implementation, and 99% attack detection accuracy was obtained. This Intrusion Detection System is capable of detecting flooding attacks only.

c. Hybrid IDS

By analysing the pros and cons of the centralised and distributed placement strategies, the hybrid placement strategy is developed. In this hybrid IDS, the strengths of both centralised and distributed approaches are included, and the drawbacks are excluded. Using this strategy, [Ama, 14] proposed a specification-based IDS. In this work, selected nodes act as watchdogs (Distributed IDS) that aim to identify intrusions by eavesdropping on the exchanged packets in their neighbourhood. This watchdog decides whether a node is compromised according to the given specification or rules. Due to the different behaviours of the network components, each watchdog has a particular set of rules. The monitored messages are matched against the set of rules that are configured in each centralised IDS agent. Thus, a hybrid approach is used in this work. The flexibility of using different sets of rules is the main advantage of this system.

[Nan, 18] developed a Hybrid Internal Anomaly Detection system which was used to monitor and to evaluate their neighbours within one-hop distance and reported their parent, only when it detected any anomaly. The monitoring node would be isolated when an intrusion is encountered and the data packets were discarded in the data link layer itself to avoid the unnecessary overhead in the network. The system also had a network fingerprinting feature to allow the edge-router to be aware of the changes in the network and to approximate the threat locations. The authors

implemented their work by allocating different responsibilities to the border router and the network nodes and making them work cooperatively. This system is capable of detecting and banning flooding attacks, selective forwarding attacks, and clone attacks. This system is very complex to handle, and it mainly focuses on limited types of attacks only.

2.4.2. Types of IDS based on its Technique

Many algorithms and techniques have been implemented to enhance the performance of the Intrusion Detection Systems. These algorithms and techniques can be applied in various stages of intrusion detection. Based on the techniques and methods implemented along with it, the IDSs are also grouped into four types, such as signature-based IDS, anomaly-based IDS, specification-based IDS, and hybrid IDS.

a. Signature-based IDS

This type of IDS is also known as a 'misuse-based intrusion detection system'. All possible known attack patterns are stored in the IDS database. These IDSs analyze the generated information and find out whether there is any match with the known attack. This type of IDS is very effective against known attacks. It needs periodic updating because the efficiency of this system depends on attack signatures available in the database [Okw, 18]. However, these systems give a higher true-positive rate, but they are unable to detect new patterns of attack, which is the drawback of these systems.

b. Anomaly-based IDS

This type of IDS is able to classify the behaviour of a system as normal or anomalous. This categorization is based on rules or heuristics rather than patterns or

signatures. First, the IDS should be trained to understand the normal behaviour of the system. If there is any activity that violates the normal behaviour, then, the IDS can identify it as an attack. This type of IDS detects unknown attacks effectively. However, it considers everything an intrusion, which does not match normal behaviour. Therefore, anomaly-based Intrusion Detection System normally have higher false positive rates than other types of IDSs [Leo, 18]. In general, machine-learning algorithms are used to train the systems. But, implementing machine learning for the resource-constrained IoT is a challenging research issue.

Based on the security attacks targeting the IoT network, [Moj, 20] recommended an anomaly-based IDS named Passban IDS for detecting intrusions at the edge level. Real-time network traffic was collected to detect the attacks, and the iForest ensemble technique was used in this methodology. This Passban IDS detected port scanning, brute force attacks, and SYNchronise (SYN) flooding attacks. The attacks during the training phase were not considered in this research. The SYN Flood attacks consumed more resources and reduced the detection accuracy of the Passban IDS.

[Imt, 21] proposed a deep learning system for anomaly detection in IoT networks. The Convolutional Neural Network (CNN) algorithm was the backbone of this research. The proposed IDS model was evaluated using IoT-related IDS datasets such as MQTT-IoT-IDS2020, IoT-23, and BoT-IoT. This multiclass model detects various attacks like DoS, DDoS, flooding attacks, OS Scan, Port Scan, Mirai, etc. efficiently in terms of accuracy and other metrics.

c. Specification-based IDS

Specification means a set of rules and thresholds defined by the experts regarding the normal behaviour of the network components and protocols. These IDSs are somewhat similar to the anomaly-based IDSs in that they detect attacks whenever there is a deviation from the specified thresholds and rules. The difference between these two techniques is that in specification-based IDS, the rules and thresholds are set by the human experts, but in anomaly-based IDS, the system should be trained. Since there is human involvement in these IDSs, the false-positive rate is lower, compared to the anomaly-based IDS [Mit, 14]. However, these IDSs are not flexible and may not adapt to different environments and are error-prone due to the manually defined specifications.

[Vik, 21] proposed a specification based Unified IDS (UIDS) for detecting DoS attacks, probe attacks, generic attacks, and exploit attacks. The decision tree algorithm is applied to the UNSW-NB15 dataset for developing this system. Various forms of rule sets are defined in this UIDS. This signature-based IDS detects the attacks more effectively than the existing research work. It needs further refinement to detect new attacks.

[Phi, 21] recommended a specification-based system to detect the malicious acts of an implanted Artificial Pancreas System (APS) which maintains the blood glucose level of the human body. In this research, the security challenges and associated risks related to patients' health and safety of APS were studied. Then, the behaviour-rules of the APS were defined. The UVa/Padova simulator was used to emulate the functionalities of APS. SVM and K-Nearest Neighbour (KNN) were the classifiers used in this research to validate the proposed model. The recommended

system monitors the components of the APS continuously, and abnormal glucose levels were identified with better accuracy. Since it is related to human life, better refinements should be required.

d. Hybrid IDS

A Hybrid IDS is the combination of the aforementioned Signature-based, Anomaly-based, and Specification-based IDSs. They are developed to optimize the performance of the hybrid IDS by minimizing the drawbacks and maximizing the advantages of these IDSs. [Ham, 17] developed a hybrid IDS based on anomaly and specification-based IDS for the IoT using the unsupervised Optimum-Path Forest (OPF) algorithm and the Map Reduce approach. It is suitable for sinkhole and selective forwarding attacks. This system has its own limitations in unsupervised learning and in the Map Reduce approach.

[Yul, 17] proposed an innovative idea of an IDS based on an input/output labelled transition system called Automata. This automata-based IDS was evaluated on a Raspberry Pi device with the help of an Android mobile phone and successfully detected the jam-attack, false-attack, and reply-attack. This intrusion detection system detected only these three types of attacks.

[Phi, 18] offered a signature-based hybrid IDS for the IoT architecture. A Denial of Service (DoS) attack was implemented using the version number modification and a "hello flood" attack. The impact of the attacks was analysed in terms of energy consumption and the reachability of nodes. The intrusion detection functionalities are not considered in this research work.

Based on the analysis of the security attacks targeting the IoT network, [Rav, 18] proposed an architecture by using the snort tool and the Raspberry Pi device. The

Prospective Backward Oracle Matching Algorithm (PBOM) was used in this work. Though PBOM reduces the memory and processing requirements, it is very difficult to perform well in an IoT environment due to the heavyweight snort tool.

[Raz, 13] proposed a hybrid IDS suitable for the IoT environment to detect sinkhole and selective forwarding attacks in real-time. It was named as 'SVELTE'. In their work, the authors tried to enhance the IDS by balancing the computing costs of the anomaly-based approach and minimising the storage costs of the signature-based approach. In SVELTE, the border router processes intensive IDS modules like analysing the RPL network data. Here, network nodes were responsible for lightweight tasks such as sending RPL network data to the border router and notifying the border router about the malicious data they received. [Shr, 17] extended this research by including an intrusion detection module that uses the link reliability metric called Expected Transmission Count (ETX), which was used in RPL networks. They suggested that by monitoring the ETX metric, the intruders' activities in the 6LoWPAN network were prevented, and they were also able to identify the location of the attacker nodes. The true-positive rate was increased in their work by combining the ETX-based rank mechanism with the rank-only approaches.

[Bac, 20] presented an Enhanced Network IDS (ENIDS) for Internet of Things to detect the clone attack. This protocol was evaluated with the performance of SVELTE model proposed by [Raz, 13] for detecting sinkhole and selective forwarding attacks in IoT. The ENIDS outperformed in terms of detection probability and energy consumption. This ENIDS is limited to clone attacks, and in the normal scenario, it consumes more energy.

2.4.3. Machine Learning-based IDS for IoT

[Hon,18] suggested a lightweight Intrusion Detection System based on fuzzy clustering algorithms for Wireless Sensor Networks having limited resources. The authors mapped the network status into the sensor measurements received at the base stations. The fuzzy c-means algorithm, one-class SVM, and sliding window procedure were merged to create this Intrusion Detection System, which detects attacks efficiently. The EXata Network simulator was used to check the effectiveness of the system. Although it is suitable for detecting communication destructive attacks, detection of multiple attacks should be improved further.

In the comparative study, [Sar, 18] explained various feature selection and classifier machine learning techniques used in Intrusion Detection Systems. The author implemented Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Neural Networks (NN) and K-Nearest Neighbour (KNN) classifiers against the Correlation-based Feature Selection method (CFS), Information Gain Ratio (IGR) based feature selection, Principal Component Analysis (PCA) and Minimum Redundancy Maximum Relevance method. The NSL-KDD dataset with 10,000 tuples and 40 attributes was used for this analysis. The authors found out that among the classifiers, the K-NN algorithm and among the feature selection techniques, Information Gain Ratio based feature selection provided the best results.

[Nou, 18] developed an AdaBoost ensemble method by using three machine learning algorithms, namely Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network (ANN), to detect intrusions in the Internet of Things environment. This method mainly detected the botnet against the Domain Name Systems (DNS), Hyper Text Transport Protocol (HTTP), and Message Queue Telemetry Transport (MQTT). The system is limited to these three application layer protocols.

[Nas, 16] presented a machine learning approach to detect anomalous activity in the network by classifying normal and abnormal behaviour. In this work, a set of three randomly selected features was used to evaluate the performance of the system. More features should be included to enhance the detection accuracy.

[Sya, 17] suggested that the training dataset should have enough samples of the different intrusions in order to enhance the detection accuracy of the machine learning algorithm. According to the authors, the KNN algorithm outperformed all the machine learning approaches in terms of reducing false positives. [Kan, 19] proposed a flexible network intrusion detection system using the Naive Bayes classifier and deep neural networks. The results and experiments of this research showed better performance in terms of accuracy and precision in both binary and multiclass.

[Moh, 18b] assessed the challenges of IoT security by considering various machine learning techniques in smart cities. The taxonomy of machine learning algorithms and the issues and challenges regarding the data analytics of machine learning algorithms were also discussed. They suggested some machine learning algorithms like ANN for IoT security and fraud detection.

[San, 19] suggested a Support Vector Machine (SVM) based lightweight algorithm to detect malicious code injection into IoT networks. The Poisson distribution of the packet arrival rate was used to differentiate the packets as benign or intrusive. A subset of the CICID2017 dataset was selected, obtaining a synchronized beget dataset from that subset, which was further utilized in this research. The packet arrival rate is the only attribute considered in this experiment. It supports the lightweight aspect of IDS, but only a single attribute from a huge dataset will not detect all possible attacks.

[Ale, 18] suggested an IDS to analyze the data packets and to detect malicious shell code. In their work, integer values were obtained by converting the byte-level data retrieved from the data transmission of the nodes and fed into the ANN. Their best classifier identified 100% of the malicious file contents in the test set. This ANN model is useful for detecting script attacks and Structured Query Language (SQL) injections.

[Ahm, 16a] developed an Intrusion Detection Mechanism (IDM) to detect a variety of anomalous activities in the IoT networks. This technique was trained with the normal behaviour of the system using Random Neural Network (RNN) with valid and anomalous cases as input parameters. The trained RNN model can be used as the network IDS to detect any anomalous activities in the network and to prevent their propagation. The proposed solution is implemented in an IoT environment, and 97.23% intrusion detection accuracy was obtained.

[Pra, 21] proposed an ensemble-based distributed IDS to safeguard the IoT network from different types of security attacks. The Gaussian Naive Bayes, KNN, Random Forest, and XGBoost algorithms were applied to develop the ensemble model. The UNSW-NB15 and DS2OS datasets were used to evaluate the performance of the IDS. Though there is much ongoing research and development in the security of IoT by implementing Intrusion Detection Systems, it is still needed to enhance the security level further by using innovative tools and techniques.

2.5. Analytical Survey

Table 2.2 shows the summary of the reviewed literature. Here, IDS research work, the type of IDS it belongs to, techniques used in the IDS, advantages, and the research gaps of these IDSs are briefly given.

Table 2.2. Analytical Survey on IDS for IoT

Research	IDS Type	Techniques/Tools	Attack Detection	Required Refinements
[Yul, 17]	Centralized	Automata	Jam-attack, false-attack and replay-attack	State-space problem
[Raz, 13]	Hybrid	SVELTE	Sink-hole attacks	Additional Control overhead due to 6Mapper module
[Shr, 17]	Hybrid	Extension to SVELTE using ETX metric	Rank attack	Maximum 8 nodes only used.
[Bac, 20]	Centralized	ENIDS protocol	Clone attacks	Consumes more energy in normal scenario
[Phi, 18]	Hybrid	Cooja Simulator, Pattern Matching Algorithm	DoS	IDS functionalities are not considered
[Hon, 18]	Hybrid	Fuzzy C-Means, One-Class SVM, Sliding Window	Anomalies and network attacks	Refinement is needed to detect the diversity of attacks
[Nou, 18]	Centralized	AdaBoost ensemble method	Botnet attacks	Limited to three IoT application layer protocols
[San, 19]	Centralized	SVM classifier	DDoS attacks	Single attribute (Packet arrival rate) only used
[Moj, 20]	Centralized, Anomaly based	Passban IDS, iForest	Port Scanning, Brute force, and flooding attack	Not considering the attacks in the training phase, flooding attack reduces the detection rate
[Osa, 21]	Distributed	Blockchain, Bidirectional Long Short-Term Memory (BiLSTM)	DoS, DDoS, and Port Scanning	Need further refinement for real-time implementation

Research	IDS Type	Techniques/Tools	Attack Detection	Required Refinements
[Muh, 20]	Distributed	Blockchain, Spectral Partitioning	Routing attacks and Botnet	Real-world conditions should be addressed
[Gon, 20]	Distributed	Deep Learning Algorithms	Phishing, DDoS, Botnet	Restricted to specific attacks
[Pra, 21]	Distributed	Ensemble	Backdoor, Reconnaissance, and DoS	Real-time deployment requires lightweight mechanisms for IoT nodes
[Oh, 14]	Distributed	auxiliary shifting, early decision	Conventional attacks using signatures	Single device only
[Lee, 14]	Distributed	Energy consumption models	Routing attacks, and DoS	Single device only
[Amj, 18]	Distributed	Naive Bayes Algorithm, Multi-agent	DDoS Attack	Low-capacity systems are not considered
[Cer, 15]	Hierarchical - Distributed	INTI	Sinkhole attacks	Low-capacity systems are not considered
[Mid, 17]	Centralized	KALIS	DoS, and Routing attacks	Complex functionalities
[Azk, 18]	Centralized	Software-Defined Networking (SDN)	Flooding attacks	Only flooding attack is considered
[Ama, 14]	Hybrid	Watchdogs	Routing attacks based on a different set of rules	Requires optimization in enforcing and storing new security rules
[Nan, 18]	Hybrid-Anomaly based	Network fingerprinting	Flooding, selective forwarding, and clone attacks	Complex to handle

Research	IDS Type	Techniques/Tools	Attack Detection	Required Refinements
[Vik, 21]	Centralized Specification-based IDS	Decision Tree	Exploit, DoS, Probe, and Generic	Requires refinement for detecting new attacks.
[Imt, 21]	Anomaly-based	Convolutional Neural Networks	Dos, DDoS, Mirai, Flooding, Port Scan, OS Scan, etc.	Training takes more time
[Phi, 21]	Centralized Specification-based	UVa/Padova simulator, SVM, KNN	The abnormal blood glucose level	Human life related. So further refinements are required.
[Ham, 17]	Hybrid	Optimum-Path Forest (OPF), Map Reduce Algorithm	Sinkhole, wormhole, and selective forward attack	Simultaneous different types of attacks reduce the performance

According to this review, when machine learning algorithms are deployed, the performance and efficiency of the intrusion detection systems are increased, and the hybrid IDS provides better accuracy, which reduces false positives and improves the true positives.

2.6. Research Challenges and Directions

Traditional network infrastructure has paved the way to the Internet of Things. As a result, it encompasses all of the conventional network's security risks and dangers. Because the Internet of Things is linked to the Internet, any security vulnerabilities that exist on the Internet also affect the IoT environment. The reasons for numerous security-related vulnerabilities in the IoT ecosystem are as follows:

- Memory, computing power, and energy are limited for the devices in IoT networks. The Internet connects a large number of such resource-constraint devices from many sources, making the IoT increasingly susceptible

- Different technologies and platforms are used by IoT devices. As a result, ensuring compatibility across these devices is a difficult task
- Cryptographic solutions are undesirable for IoT due to their heavyweight. If the cryptographic keys and nodes are hacked, the security risk increases significantly
- These vulnerabilities render the IoT insecure and can result in catastrophic consequences such as data breaches and IoT node tampering. IDSs are a savior in this situation for ensuring protection to IoT networks

As a result, an intrusion detection system is required to monitor the IoT network and detect attackers and compromised IoT devices. The existing IDS solutions for IoT are insufficient. The following are the research gaps for deploying detection mechanism in IoT networks:

- Conventional network intrusion detection solutions are heavyweights that are inadequate for resource-constrained IoT networks. When designing IDS for the IoT, lightweight factors like processing, storage, and battery energy consumption should be considered
- When implementing IDS for the IoT, accessibility and channel stability concerns should be kept in mind
- The IoT incorporates complex protocols and technologies, each with its own set of network security flaws. As a result, IDS originally created for traditional networks is not effective in an IoT context

- The sensors create a large amount of data. Handling such a large amount of data, as well as the security issues associated with such voluminous of data, tend to many research challenges.
- The resource-based RPL attacks consume more power, memory, and Processing time. As a result, innovative strategies and intrusion detection systems (IDS) are required to protect IoT gadgets and networks from such attacks

The risks and constraints of deploying IDS in IoT networks are highlighted in the facts above.

2.7. Research Roadmap

After performing the literature survey, based on the focused issues, the research roadmap is drawn. This research flow diagram, or roadmap, specifies the scope of the research work. The flow diagram of this research is portrayed in Fig. 2.8.

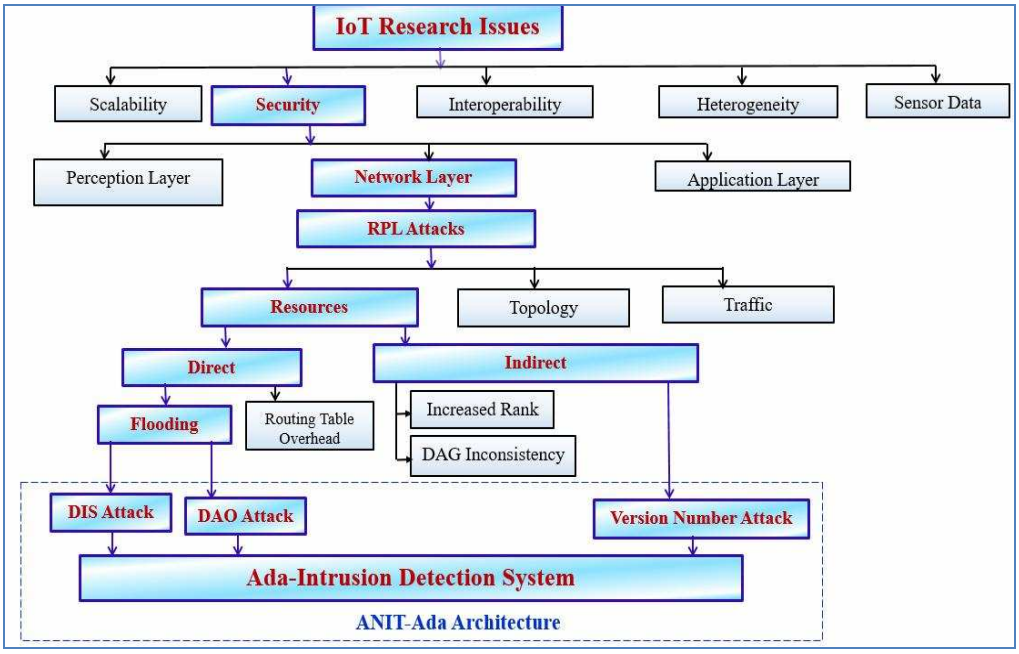


Fig. 2.8. Research Roadmap

As Fig. 2.8 explains, among all the research challenges, security issues are taken into consideration for this research. Though there are different architectures used in the IoT, the most commonly used three-layered architecture is focused. The network security aspects are at the core of this research. The Routing Protocol for Low Power Lossy Networks (RPL) is the most widely used routing protocol for the IoT. RPL protocol is exposed to several security attacks. Three RPL resource attacks which lead to Denial of Service (DoS) such as the DAO attack, Version Number Attack, and DIS attack, are selected for this research. Three intrusion detection techniques, such as VeNADet, DISDet, and DADTec, are proposed. Finally, a Network Intrusion Detection System based on the AdaBoost Algorithm called Ada-IDS is developed using the network traces collected from the simulated environment having these three attacks and a normal scenario. The three techniques are installed in the root node and the Ada-IDS is implemented in the Border Router, which provides an additional layer of security to the IoT nodes in the network. The three techniques and the Ada-IDS are the components of the ANIT-Ada Architecture. This architecture acts as an intelligent system to detect RPL resource attacks and safeguards the IoT networks.

2.8. Chapter Summary

The Intrusion Detection System adds another layer of security to the IoT environment. Different types of RPL-based attacks and IDSs are reviewed in this chapter. According to the literature, when machine learning algorithms are implemented, the performance and efficiency of the intrusion detection systems are improved and the hybrid IDS provides better accuracy by reducing the false alarm

rate and improving the true positive value. In this research work, three types of RPL resource attacks such as Version Number Attack, DIS flooding attack, and DAO attack are taken and elaborately discussed in the following III, IV, and V Chapters. The detection techniques of the respective attacks are also deliberated in detail.

Chapter – III

VeNADet: Version Number Attack Detection in RPL-based Internet of Things

Chapter - III

VeNADet: Version Number Attack Detection in RPL-based Internet of Things

3.1. Background

In our everyday lives, the Internet of Things (IoT) includes technical and industrial upgrades and advances. Though the IoT makes our lives simpler, it also presents a number of barriers when used in large-scale applications. One of the most significant topics to address with the IoT is security. Because of the limited resources, IoT networks are more susceptible. Traditional heavyweight strategies for detecting and preventing security risks and attacks are ineffective for IoT [Mah, 19].

Many IoT devices are connected to the global network without any basic security measures. Because of the limited capacities and the large number of connected devices, the IoT network is more susceptible than the conventional network. RPL is the potential network layer protocol for routing in IoT. It is prone to several types of security threats and attacks during DODAG building [Ami, 11].

Attackers generate a number of attacks by altering the RPL control messages. Version Number Attack (VNA) is one of the RPL-based resource attacks that is initiated by modifying the Version Number (VN) field of the DIO control message. Indirectly, VNA consumes additional network resources like memory and power. The VNA changes the Version Number on a regular basis, resulting in a global repair process that repeatedly creates the DODAG. As a result, the VNA is a difficult problem for RPL security which leads to Denial of Service (DoS) in legitimate nodes [Ari, 18].

In this chapter, a version number attack (VNA) detection mechanism called VeNADet is proposed and implemented in the Cooja Simulator of the Contiki Operating System. The outcomes of this research work illustrate that the proposed method detects Version Number Attacks efficiently.

3.2. Related Works

[May, 14] used static nodes in a grid architecture to investigate the effects and repercussions of version number attacks. Control packet overhead, packet delivery ratio, delay, irregularities, and loops were used to measure the performance. The study did not take into account critical characteristics of limited nodes, such as power usage and its consequences. It doesn't fit into any probabilistic attack model and dynamic environments.

[Ahm, 16b] suggested a technique for detecting VNA in the IoT network using a distributed and cooperative verification strategy, which considerably minimizes the overhead of control packets. If a node gets a DIO message from a higher version node, then only it updates the version number. For that it requires a thorough verification and analysis of its neighbours' and their parents' activities within a two-hop count range. As a result, the neighbours' trustworthiness is assured, and fraudulent version number updates are efficiently avoided.

[Moh, 18c] suggested a lightweight scheme to mitigate VNA and Rank attacks. The identity based offline/online signature (IBOOS) used in this approach provided non-repudiation and security from these attacks. According to the authors, their proposed scheme is lightweight in terms of computational cost and power consumption. The version number and rank value were protected using security keys. It is a scalable and distributed approach as it doesn’t require any backbone infrastructure.

3.3. Objectives

This chapter focuses on the mechanism called VeNADet, for detecting Version Number Attacks in IoT networks. The objectives of the chapter are listed below:

- To simulate different IoT scenarios without any attackers by increasing the number of nodes gradually and analysing the packet delivery ratio (PDR), energy consumption, and control overheads.
- To simulate the attacker scenarios with 10% of attackers by increasing the nodes and attackers, analysing the PDR, energy consumption, and control overheads, and comparing the network performance with the normal scenarios.
- To implement the VeNADet technique in the attacker scenarios and measure the PDR, energy consumption, and control overhead, and to measure the detection accuracy of the proposed technique.
- Isolate the attackers by broadcasting the attacker's details and disconnecting the communication links.

3.4. RPL DODAG and Version Number

3.4.1. RPL DODAG

RPL is a routing protocol for low-power and lossy networks that uses IPv6. The smart devices in the IoT network are connected in a special way without any cycle in the topology construction. For this specification, a Destination Oriented Directed Acyclic Graph (DODAG) is built, which is routed to a single destination called the root node. The Objective Function (OF) defines the routing metrics for constructing the DODAG. When the router is configured, an RPL instance is generated which defines the OF for the DODAGs that belong to the RPL instance.

The irregularities and loops in the DODAGs are eliminated by computing the node's rank. The rank of a node in the DODAG is its position with respect to the root node. Each node in the DODAG has a rank value which is determined by the rank value of its parent and other parameters like Hop Count, Energy, and Estimation Transmission Count (ETX) [Tsv, 11].

The IoT network's root node is connected to the Internet through an IPv6 Border Router (6BR). According to the RPL protocol, by broadcasting DODAG Information Object (DIO) messages, the root node commences the formation of the DODAG, a tree-like structure. The recipient nodes respond with a Destination Advertisement Object (DAO) to their parent nodes via which they got the DIO message in order to join the DODAG. By issuing a DAO-ACK message, the parent nodes allow the child nodes to join the network. In order to join a DODAG topology, new node multicasts DODAG Information Solicitation (DIS) messages to its neighbors for receiving DIO message from anyone of its adjacent nodes [Pav, 15]. The DODAG construction process is explained elaborately in Chapter 2, section 2.2.2.

3.4.2. Version Number

The network layer is accountable for all routing choices in RPL. As a result, all nodes, including the root node, must be aware of the topological information. The DIO message is the main concept for creating the DODAG topology. The DIO message format is shown in the Fig. 3.1.

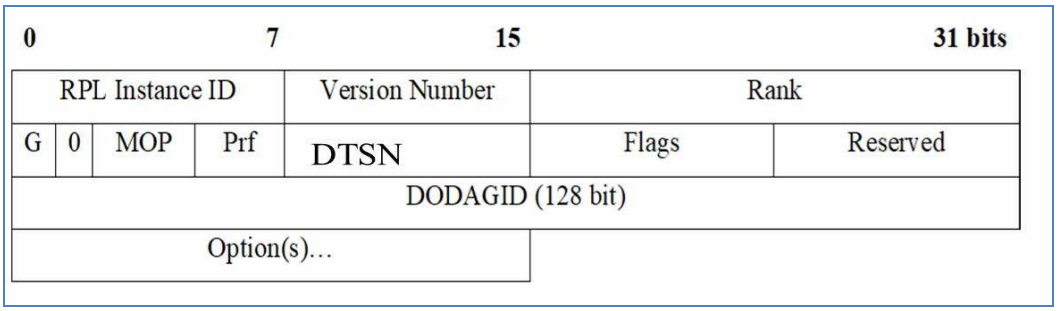


Fig. 3.1. DIO message Structure[4]

As it is given in the Fig.3.1, there are number of routing metrics and DODAG related information available in the DIO message. The important aspects of the DIO message are detailed below:

- **RPL Instance:** An RPL Instance consists of more than one DODAG sharing same RPL Instance ID. A node can be part of a single DODAG only.
- **DODAGID:** A DODAGID is the 128-bit IPv6 address of a DODAG root. The combination of RPL Instance ID and DODAGID is used to identify the DODAG uniquely.
- **Version Number:** Version Number (VN) is an unsigned 8-bit number. Whenever a new version of the DODAG is created, the version number field is incremented by the root node. The combination of RPL Instance ID, DODAGID, and Version Number are used to identity a unique version of the DOADG.
- **MOP:** The Mode of Operation (MOP) is used in the DOADG. It represents the storing and non-storing modes.
- **DTSN:** DTSN stands for Destination Trigger Sequence Number. When this field is triggered, DAO messages are transmitted from a node to its parent node.

[4] IETF- Request for Comments (RFC) 6550 [Win, 12]

Following the RPL instance value, there is an 8-bit unsigned number in the DIO message. The VN is used to identify a network topology uniquely within an RPL instance. It represents the current version of the DODAG.

Whenever an inconsistency arises due to loops, it should be instantly addressed by updating the version number and reconstructing the DODAG topology. The rebuilding DODAG process is known as global repair. All nodes in the network receive this version number when the version number of the topology is changed. By using the incorrect version number, an attacker node can disrupt the normal flow of network traffic, increasing the network's security challenges [Thu, 20] [Vas, 20]. The DODAG topologies in two versions are illustrated in Fig. 3.2.

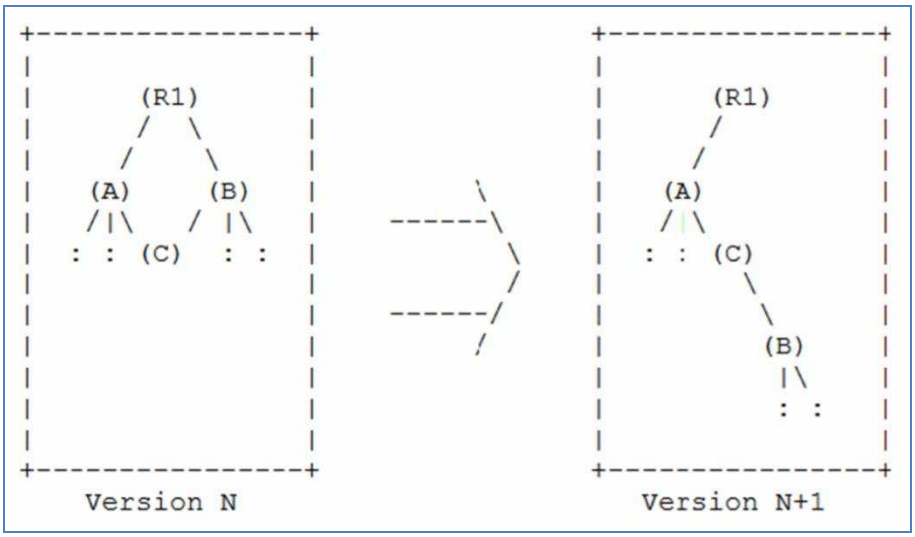


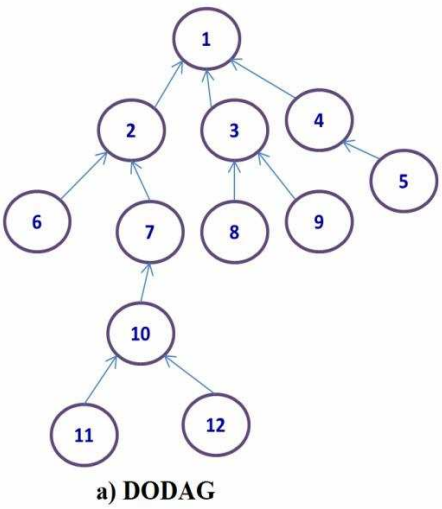
Fig. 3.2. DODAG in Different Versions[5]

Since there is a loop in Version N, the global repair mechanism is initiated and the DODAG ‘Version N+1’ is constructed. 'Version N+1’ has the same nodes as in the ‘Version N’ but has a different structure without any loops.

[5]IETF- Request for Comments (RFC) 6550 [Win, 12]

3.5. Version Number Attack: An Overview

The DODAG root node has a version number, as described by the RPL specification. The RPL DODAG is built as a tree, and the version number is used to guarantee that loop-free pathways towards the root node without any abnormalities. The root node executes a global repair to ensure DODAG integrity [Ari, 18]. An attacker node may broadcast a fake version number in its DIO control message to induce a global repair on a regular basis. The regular scenario, in which there is no Version Number Attack, is depicted in Fig.3.3.



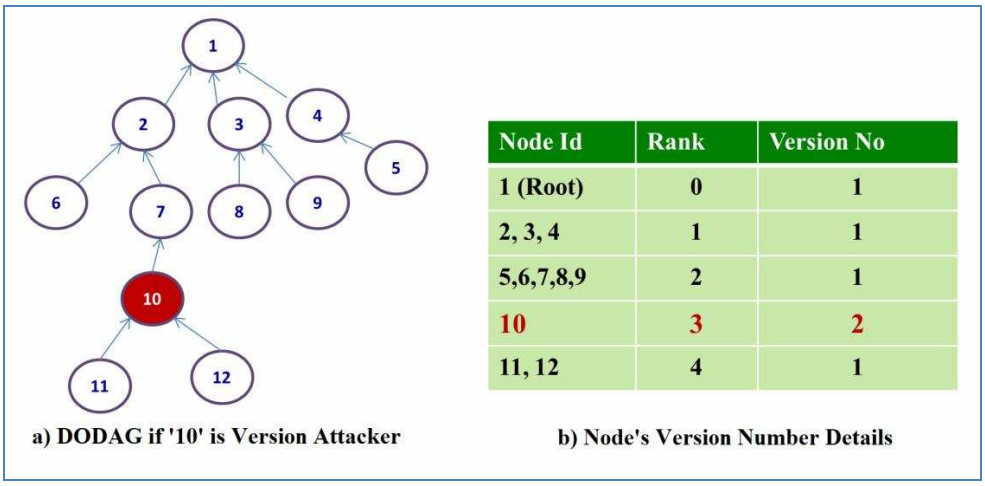
a) DODAG

Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,7,8,9	2	1
10	3	1
11, 12	4	1

b) Version Number details of the Nodes

Fig. 3.3. Normal Scenario without Version Attack

There are 12 nodes in Fig. 3.3, and the hop count is chosen as the rank measure to build the DODAG. The VN value is fixed to one. Fig.3.3 (b) shows the features of this DODAG. The DODAG has no abnormalities or contradictions in terms of loops or version changes. As a result, it's a typical DODAG without any Version Number Attack (VNA). Assume that node '10' is a Version Attacker; it changes its version number from '1' to '2', and the VN '2' appears in node 10's DIO message instead of '1'. Fig.3.4 depicts the revised information of node 10.

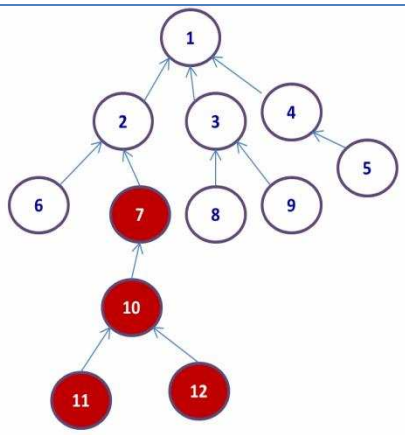


Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,7,8,9	2	1
10	3	2
11, 12	4	1

b) Node's Version Number Details

Fig. 3.4. DODAG with Version Attacker Node '10'

The attacker node '10' is indicated using red color. When node '10' delivers the DIO message to transmit the modified version to its neighbors' '7', '11', and '12', they also alter their version number. The nodes' information with version modification is shown in Fig.3.5

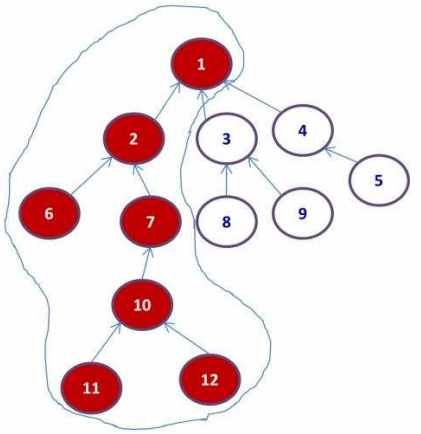


Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,8,9	2	1
7	2	2
10	3	2
11, 12	4	2

b) Version Number details after Sending DIO Messages from Node '10'

Fig. 3.5. Node '10' sends DIO Message to its Neighbors

During this procedure, the root node also receives a DIO message with a new version number, which causes a global repair. Hence, the root node initiates the DODAG creation process. Figure 3.6 depicts this problem.



a) Root Node receives updated Version Number

Node Id	Rank	Version No
1 (Root)	0	2
2	1	2
3, 4	1	1
5,6,8,9	2	1
7	2	2
10	3	2
11, 12	4	2

b) Version Number details when Root Node received updated Version Number

Fig. 3.6. Global Repair Scenario

Fig.3.6 illustrates all the nodes affected by VNA in red color and it means global repair is required. So, the root node initiates the DODAG reconstruction process. Because the '10' is an attacker, it modifies its version number on a regular basis, disrupting the DODAG topology and consuming more resources indirectly. As the global repair mechanism is initiated again and again, the network performance in terms of throughput and packet delivery ratio (PDR) is degraded.

As a result, VNA can deplete node resources indirectly and it can degrade the network, as well as degrade application performance [Ras, 20]. These detrimental effects in the IoT network leads to Denial of Service (DoS) attacks in the legitimate nodes. As the above illustration specifies, when the Version Number attacker is placed in the higher level rank position, like a neighbour to the root node, it causes more damage than in the lower levels, or as a leaf node.

3.6. Version Number Attack Model

Let N represents a set which contains all nodes in the IoT network Z, including the root node R. Let X is the parent node which has a set Y of child nodes. Let R_{vn} is

the version number of the root node R, X_{vn} is the version number of the parent node and $Y_{vn(i)}$ is the version number of i^{th} child node of X. Hence, $R \in N$, $X \in N$, $\forall Y_i \in N$. In this network Z, only the version update is allowed from higher order to lower order

Case 1: Attack occurs in fixed interval time

If an illegitimate version update e occurs from a parent node X to its lower level nodes at a particular time t, and any node Y_i updates it version number $Y_{vn(i),}$ and the child node Y_i also behaves similarly as the attacker node X. The Eq.3.1 can be used to check whether there is a version number attack in the network Z.

$$T\left(R_{vn}, X_{vn}, e, t, Z\right) = \log \frac{\left(f\left(R_{vn}, e, t, Z\right)\right)}{f\left(X_{vn}, e, t, Z\right)} \qquad (3.1)$$

where, f (R_{vn}, e, t, Z) and f (X_{vn}, e, t, Z) gives the behavior of the root node and other nodes in the network Z respectively when an event e occurs at the time t. The value of the function T(R_{vn}, X_{vn}, e, t, Z) determines the version attack in the network Z. It is given in Eq.3.2.

$$T\left(R_{vn}, X_{vn}, e, t, Z\right) = \begin{cases} 0; & X \text{ is legitimate node} \\ \text{otherwise}; & X \text{ is attac ker} \end{cases} \qquad (3.2)$$

When f (R_{vn}, e, t, Z) and f (X_{vn}, e, t, Z) are same, then the function T (R_{vn}, X_{vn}, e, t, Z) gives '0'. Hence the node X can be treated as a legitimate node.

Case 2: Attack occurs in a continuous interval time Δt

If an illegitimate event e occurs during a continuous interval time 'Δt' and any node updates it version number as X_{vn}, then to check whether there is any version number attack in the network, Eq.3.3 can be used.

$$T\left(R_{vn}, X_{vn}, e, t, Z\right) = \log \frac{\int_{t=lo}^{t=l1} f\left(R_{vn}, e, t, Z\right) dt}{\int_{t=lo}^{t=l1} f\left(X_{vn}, e, t, Z\right) dt} \quad (3.3)$$

Where $\int_{t=l0}^{t=l1} f(R_{vn}, e, t, Z)\, dt$ and $\int_{t=lo}^{t=l1} f(X_{vn}, e, t, Z)\, dt$ represent the behaviors of the root node R and other node X according to their version numbers R_{vn} and X_{vn} respectively during the time interval $\Delta t=l_1-l_0$. Now the same condition as it is given in Eq.3.2 is applied to check whether there is any version number attack occurred or not in the IoT network Z. If T (R_{vn}, X_{vn}, e, t, Z) = 0, then there is no attack, otherwise the node X is a Version Number attacker.

To overcome the negative consequences of VNA, the VeNADet technique is proposed in this research work.

3.7. The VeNADet Technique

Whenever there is an inconsistency in the DODAG, the root node initiates a global repair mechanism and increments the version number to make the DODAG in a consistent state. It is a legitimate update. An attacker also illegitimately updates the current version number with a higher version number and transmits the DIO message with the updated version number. It can also cause a similar problem by wrongly advertising a version number that is increased at regular intervals. The Version Number Attack Detection (VeNADet) technique has been proposed to counteract this attack. Fig.3.7 shows the proposed approach to detect VNAs called VeNADet.

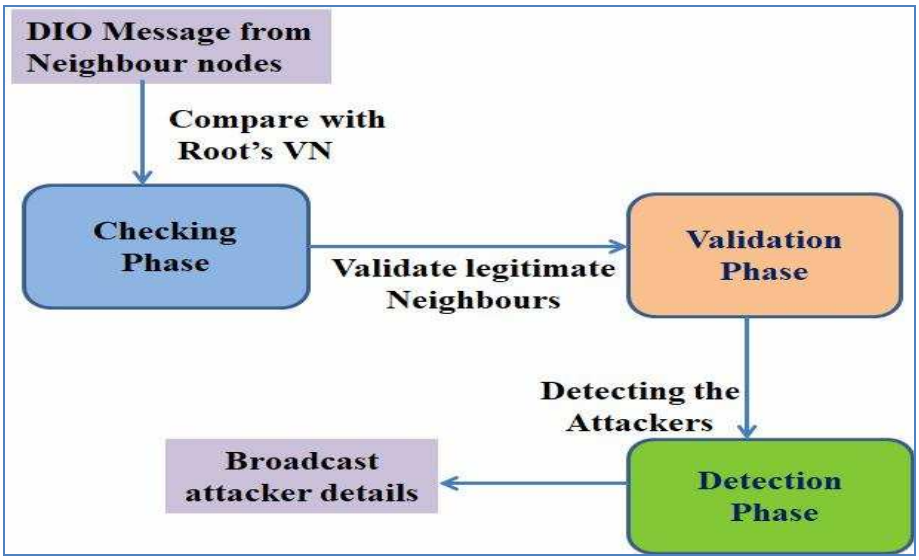


Fig. 3.7. The VeNADet Technique

As shown in Fig.3.7, the VeNADet technique is divided into three phases: checking, validation, and detection. When a node gets a DIO message from its adjacent node, it checks the VN of the DIO message with the VN of the root node. The neighbor node is then confirmed as authentic or not based on the result. Once the node is detected as a VN attacker, the root node declares the particulars of the attacker to all nodes in the DODAG tree. The VeNADet phases are explained below.

- **Checking Phase:** When a node Y_i gets a new DIO message from a neighbor X, if the VNs of $Y_{vn\ (i)}$ and X are the same, the neighbor is considered a legitimate node. If both VNs are not identical, Eq. 3.1, Eq.3.2, and Eq. 3.3 are used to compare the VN of X with the root node's VN that is R_{vn}. If both version numbers are the same, the node X is considered as a legitimate node. In such case, the current VN of node Y_i is updated to match the new VN of node X. If the two VNs disagree, the node with new VN is transmitted to the detection phase for additional investigation. Fig.3.8 elucidates the algorithm for the checking phase.

Algorithm 3.1 for Checking the Version Number

1. function **Version_Check**(R_{vn}, X_{vn}, Y_{vn});

   Input: VN of the root node R, (R_{vn}) and X, (X_{vn})

   Output: Legitimate or Malicious

   Initialization:

   $R_{vn} \leftarrow$R.DIO.VN *// VN of root R*

   $Y_{vn(i)} \leftarrow$ Y_i.DIO.VN *// VN of node Y_i*

   $X_{vn} \leftarrow$X.DIO.VN *// VN of sender X*

2. **if** ($Y_{vn(i)}$=X_{vn}) then
3. print "**Node X is Legitimate**"
4. **else if**(X_{vn}=R_{vn}) then *// use Eq.3.1.to Eq.3.3.*
5. print "**Node X is Legitimate**"
6. $Y_{vn(i)}$=X_{vn}*// receiver updates its version number*
7. **else**
8. print "**Node X is Malicious**"
9. function **Validate_Neighbor**
10. **end if**
11. end **Version_Check**

Fig. 3.8. Algorithm for Checking Phase

- **Validation Phase:** If the version numbers of nodes X and R do not match, then X is deemed a malicious node. In this scenario, this phase looks for the source of the attack by comparing the VN of neighbors with a rank value higher than its own. Let W=w_1, w_2,....., w_n denote the set of 'n' nodes within one hop of the node Y_i. The VN of all neighbor nodes 'w_i' is compared to the VN of node X. If 80% of high-ranking neighbors have the same VN as node X and root node R, then X is a genuine node, and Y_i upgrades its VN to X_{vn}. Or else, the sender of DIO message, node X is considered as an attacker node and

this node is sent to the detection phase for confirming that node X is an attacker. In such case, Y_i keeps its original version number Y_{vn} (i).Fig.3.9 shows the approach for this validation step.

Algorithm 3.2 for Validating Neighbour Nodes

1. function **Validate_Neighbor**(malicious_node X);
   Input: VN of the root node R, (R_{vn}) and VN of DIO sender X, (X_{vn})
   Output: Legitimate or Malicious
   Initialization:
   $vn_{count} \leftarrow 0$
   high_rank_nodes$\leftarrow$0
   W= {w_1, w_2…….w_n} *// neighbours of receiver Y_i*
2. **for** (i= 1 to n) do
3. **if** (w_i.rank>Y_i.rank) then // *nodes having higher rank*
4. high_rank_nodes= high_rank_nodes +1
5. **end if**
6. **if** (w_i.rank>Y_i.rank and wi_{vn}= X_{vn}) then *//checking VN*
7. vn_{count}=vn_{count}+ 1
8. **end if**
9. **end for**
10. t=high_rank_nodes * 0.8 *// 80% of higher rank nodes of Y_i*
11. **if** (vn_{count}>= t and X_{vn}=R_{vn}) then *// 80% higher rank nodes accept X_{vn}*
12. update $Y_{vn(i)}$=X_{vn}*// update version number*
13. **else**
14. print (“Node X is malicious”)
15. function **Detect_Attack***// Call detection phase*
16. **end if**
17. end **Validate_Neighbor**

Fig. 3.9. Algorithm for Validating Neighbors

- **Detection Phase:** If a malignant node is discovered, its origin must be traced. When an attacker is found as a leaf node or any intermediary node in the DODAG, the attacker may be readily recognized and identified using the aforementioned two approaches. When an attacker node 'a' is placed as the neighbor node of the root node 'R', then by verifying the VN of the higher order node 'a'. Hence, global repair is frequently initiated. Detecting the attacker is difficult in such conditions.

If the root node R's neighbour X tries to launch the global repair process very often, it checks with the other neighbour node to see whether the global repair process is required or not. It does so by using the GR_{count} variable, which is set to '0' during the DODAG creation. The GR_{count} value is increased if there is a need for global repair on a node. Only the root node starts the global repair if the GR_{count} crosses the threshold value (80%) and the DIO. delay of the node X is greater than or equal to the DIO. delay of other neighbor nodes.

Otherwise, the node X is confirmed as an attacker node, and connection channels of node X are blocked. The root node 'R' also broadcasts the address and other particulars of the attacker node X. This broadcasting allows other nodes to recognize the attack and its origin. Fig.3.10 shows the algorithm for this detecting step.

Algorithm 3.3. for Detecting and Confirming the Attack

1. function **Detect_Attack**(malicious_node X);

Input: R_{vn}, X_{vn}, DIO.delay

Output: Legitimate or Attack

Initialization:

K= {k_1, k_2……., k_n} // nodes within one hop count of R

$GR_{count} \leftarrow 0$

d= DIO_timer delay

t =n*.8 *// 80% neighbour nodes of R*

T=N*.8 *// 80% nodes in DODAG*

2. **for** (i= 1 to n) do *// All neighbours of R*
3. **if** (k_i.delay<d) then *// attacker initiate global repair often*
4. GR_{count}= GR_{count}-1
5. broadcast "**k_i is malicious**"
6. **else if** (k_i.delay>=d and X_{vn}=R_{vn}) then *// legitimate VN update*
7. GR_{count}= GR_{count}+1
8. **end for**
9. **if** (n>=t) then *// 80% neighbours of R have updated VN*
10. $Y_{vn(i)}$=X_{vn}
11. **end if**
12. **if** (GR_{count}>=T) then *// DODAG nodes require global repair*
13. broadcast "**Initiate Global Repair**"
14. R_{vn}=R_{vn}+ 1 *//update the version Number of root*
15. **end if**
16. end **Detect_Attack**

Fig. 3.10. Algorithm for Attack Detection

Thus, by implementing these three algorithms of the VeNADet technique, the node having the higher version number is checked, validated with the neighbors, and finally confirmed whether it is an attack or not. The VeNADet technique allows legitimate global repair and version updates only. For that, VeNADet maintains 80% trust among the neighbors and all nodes present in the DODAG.

3.8. Experiments and Results

3.8.1. Simulation Setup

The Contiki OS JAva simulator (Cooja) is used in this experiment for deploying the attacks and VeNADet technique. The Contiki Operating System is useful for devices with low resources, such as sensor nodes, and it is based on an event-driven kernel. The Tmote sky is deployed as the mote in all experiments of the research. An 8 MHz Texas Instruments MSP430 low-power microprocessor, 10 KB of RAM, and 48 KB of flash memory are integrated into the Sky Mote. It has a Chipcon Wireless Transceiver with a 250 Kbps, 2.4 GHz, IEEE 802.15.4, and sensors for humidity, temperature, and light, as well as 16-pin extension support and an optional Sub Miniature Version A (SMA) antenna interface [Vel, 16]. The Table 3.1 shows the simulation setup and settings adopted in this research.

Table 3.1. Simulation Settings

Legitimate nodes	Maximum 50
Version Number Attacker	10%
Mote Type	Tmote Sky
Operating System	Contiki 3.0.
Simulator	Cooja
Topology	Random
Radio Medium	Unit Disk Graph Medium (UDGM): Distance Loss
Topology Dimension	150m x 150m
Transmission Range	50m
Interference Range	100m
Tx Ratio	100%
Rx Ratio	100%
Simulation Duration	30 minutes per simulation
Number of Simulation	5 per three different scenarios

A topology is constructed by considering the basic important characteristics of the IoT applications. This experiment takes into account network performance parameters such as power consumption, control traffic overhead and packet delivery ratio. The simulation setup's performance was examined both without and with VNAs. In this experiment, the random topology is implemented. As a result, the nodes are deployed at random locations. All nodes communicate the root node via the intermediate nodes on a regular basis. This communication is based on hop-by-hop manner. The consequences of VNA attack are assessed, as well as how the VNA influences these metrics and performance deterioration. There was no external interruption in this setting except for the distance-based loss. Each simulation takes 30 minutes to complete.

3.8.2. Version Number Attacker and Normal Scenarios

Using the parameters listed in Table 3.1, the simulation environment is set up. Fig.3.11 shows the simulation for normal contexts with 10 nodes to 50 nodes by incrementing 10 nodes each time. Each simulation type also includes a root node.

In the Fig. 3.11, there are five normal simulations. The network performance of each simulation in terms of Packet Delivery Ratio (PDR), power consumption and control overhead are analyzed. 10% of version number attackers are added in all simulations given in Fig. 3.11 for implementing the attacker scenario. The VNA nodes are deployed over the network in random locations. The VNAs launch attacks when the network has been established and the DODAG has been built. The VNA node modifies the version number of the DIO message on a regular basis and sends it to the neighbor nodes. Fig. 3.12 depicts the attacker simulation.

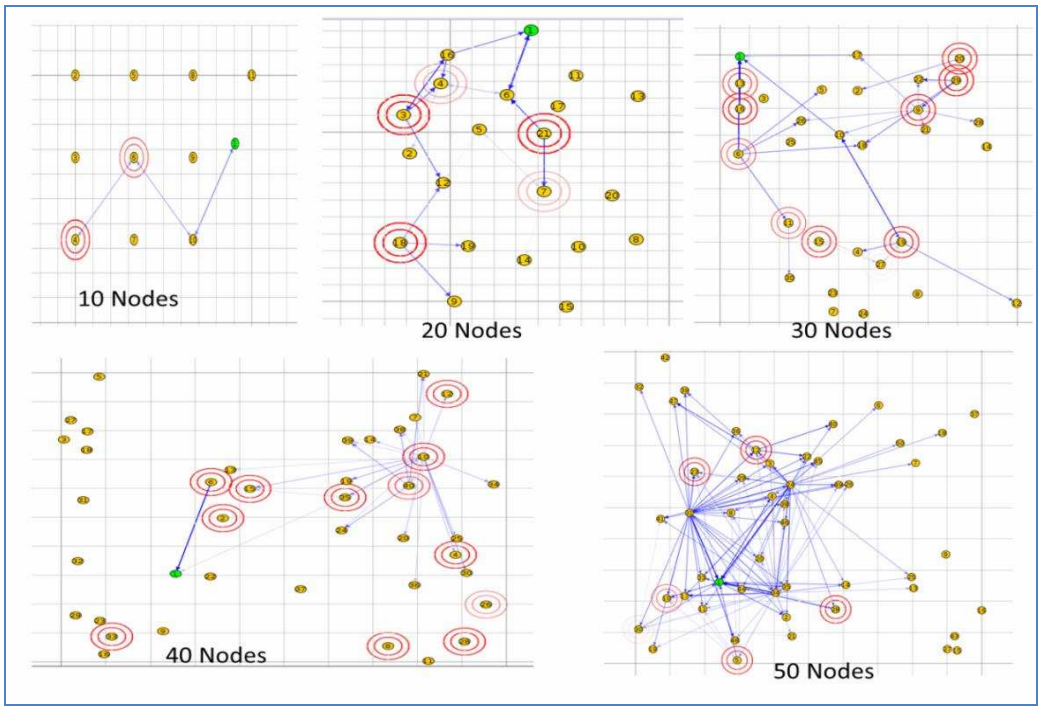


Fig. 3.11. Normal Simulations

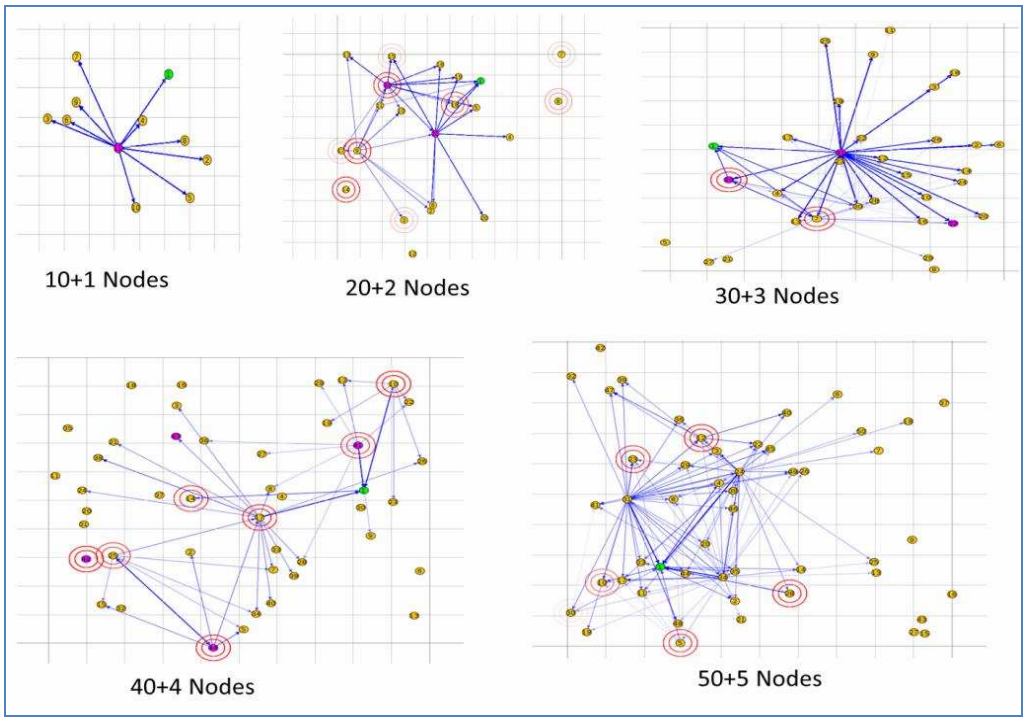


Fig. 3.12. Attacker Scenarios with 10% VNAs

The PDR, power consumption and control traffic of all attacker scenarios are also analyzed.

3.8.3. Packet Delivery Ratio (PDR)

The PDR metric is a ratio between the packets received at the destination and the packets transmitted from the sender. The formula for calculating the PDR is given in Eq.3.4.

$$PDR = \frac{Packets\ Received\ at\ Destination}{Packets\ Sent\ \ from\ Source} X 100 \qquad (3.4)$$

The IoT network works effectively when the PDR is greater than 90%. In Fig. 3.13, the PDR value achieved in the normal case and with 10% of attacker nodes is shown.

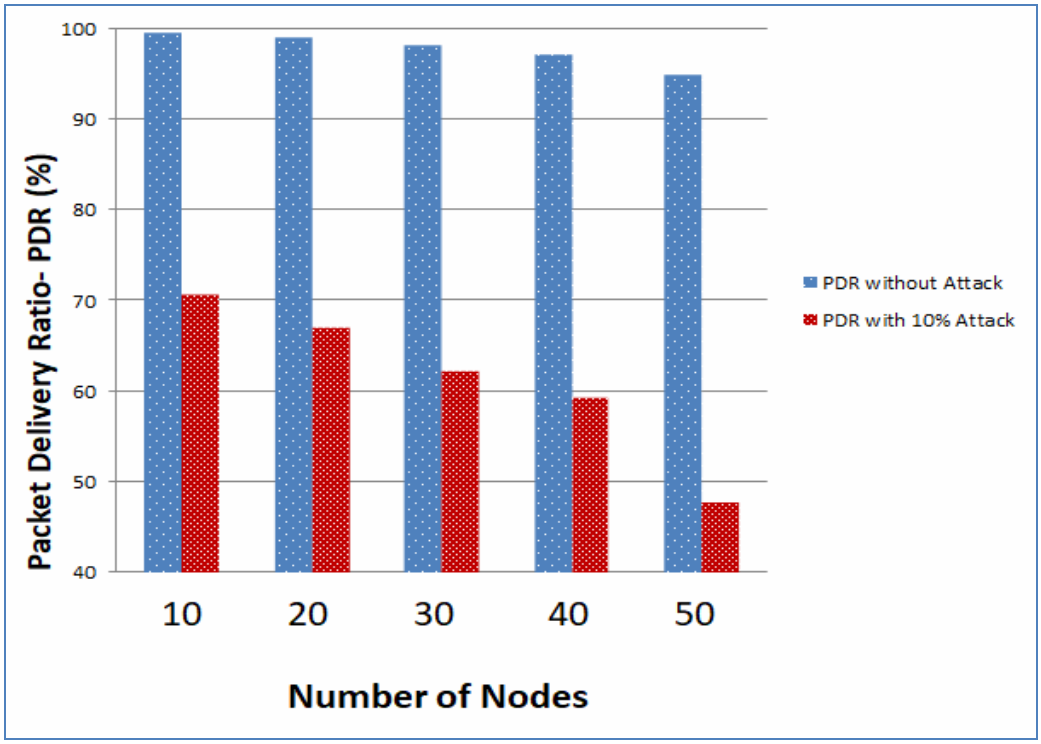


Fig. 3.13. PDR in Normal and Attacker Simulations

When there is no VNA, the PDR value is high, as shown in Fig. 3.13. The PDR value drops substantially when there are 10% attacker nodes. In the attacker scenario, network performance is deteriorated in terms of PDR.

3.8.4. Power Consumption

For each scenario, the energy consumption of each node is computed, and the average energy consumption in the VNA and normal settings is taken into account. The energy spent by each node in the network is computed. It's the sum of the power used in Low Power Mode (LPM), the CPU, radio listening, and radio broadcasting. The energy required for a node to be idle is expressed by Radio LPM; the power needed for processing is represented by CPU; the power required for receiving data is indicated by Radio Listen; and the energy required for transmitting packets is represented by Radio Transmit. Eq.3.5 [Abd, 19] is the formula for computing the energy consumed by a node during the lifespan of a network.

*Energy (mJ) = (Transmit * 19.5 mA) + (Listen * 21.5 mA) + (CPU time * 1.8 mA) + (LPM * 0.0545 mA) * 3 V / (32768)* (3.5)

Let X be the set of nodes in the DODAG. Hence, X = x_1, x_2, x_3..x_n. The energy consumption of each node, Ex_i, is calculated using Eq.3.6.

$$Ex_i = EI - ER \quad (3.6)$$

where, EI is the initial energy and ER is the residual energy of the node, Ex_i. The total energy consumed by all nodes in the DODAG is computed by using Eq. 3.7.

$$TE = \sum_{i=1}^{n} EI - ER \quad (3.7)$$

The average energy consumption of the DODAG is calculated by using Eq. 3.8.

$$AE = \frac{TE}{n} \quad (3.8)$$

where TE is the total energy consumed by all the nodes in the DODAG, and ‘n’ is the number of nodes in the DODAG. Fig.3.14 shows the average power usage for several normal and 10% attacker simulations.

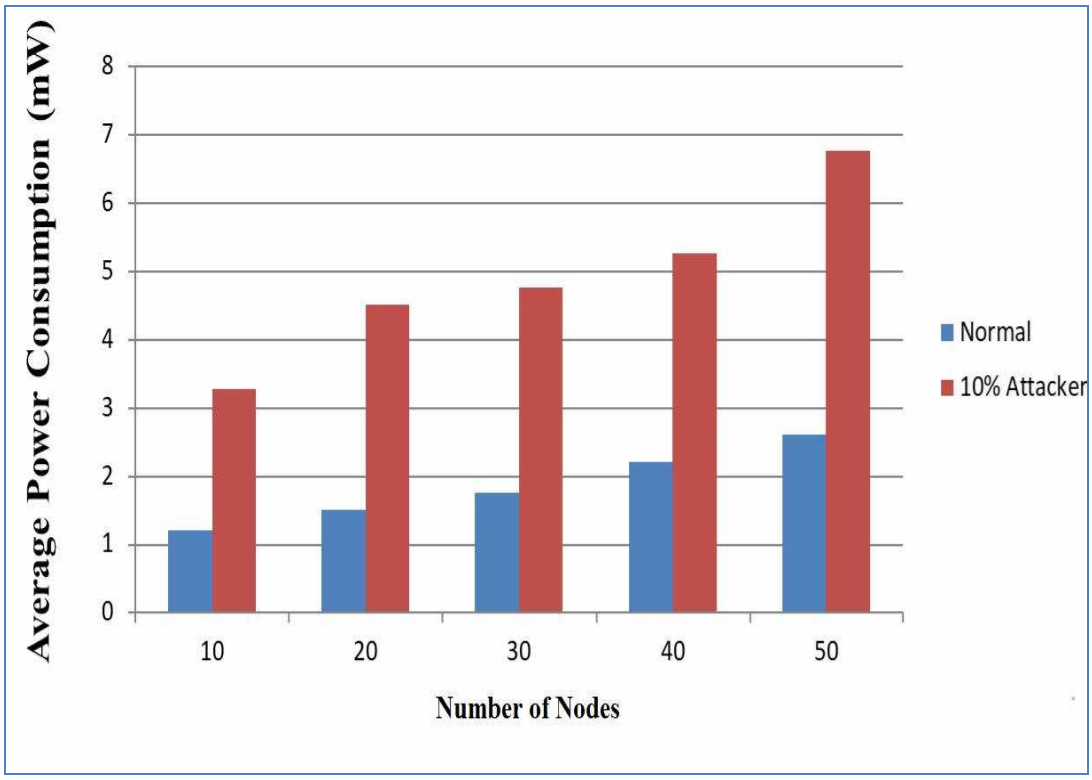


Fig. 3.14. Power Consumption in Normal and with 10% VNAs

As Fig. 3.14 indicates, the power consumption in all attacker scenarios is higher than the power consumption of the normal simulation. Due to the presence of the attacker, the energy of the nodes in the attacker's environment is quickly depleted, and the lifetime of the node is reduced.

3.8.5. Control Overhead

The control overhead is measured by the number of control messages transmitted over the network during the simulation time. Hence, in RPL networks, the control overhead is the amount of traffic caused by the number of control messages like DIO, DAO, DAO-ACK, and DIS. Eq. 3.9 is the formula for calculating the control overhead when there are 'n' nodes in the IoT network.

$$\text{Control Overhead} = \sum_{i=1}^{n} \text{DIO}(i) + \sum_{i=1}^{n} \text{DAO}(i) + \sum_{i=1}^{n} \text{DIS}(i) + \sum_{i=1}^{n} \text{DAO} - \text{ACK}(i) \quad (3.9)$$

The network's performance degrades as the quantity of control packets increases. Fig.3.15 presents the number of control packets transmitted in the normal and 10% attacker scenarios.

Nodes	With 10% Attacker	Normal
10	88341	83719
20	130677	123113
30	180521	147235
40	231871	187789
50	290379	220576

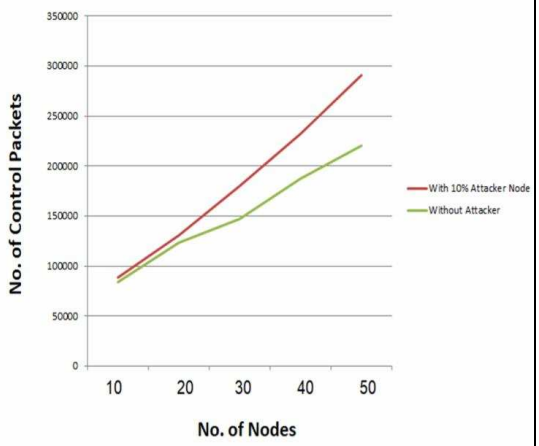


Fig. 3.15. Control Traffic in VNA and Non-Attacker Simulations

The network resources and routing parameters in the attacker environment are severely deteriorated as compared to the non-attacker scenario. As the number of attacker nodes increases, this grows worse. The VNA imposes the topology construction very often. The overwhelming number of control traffic consumes the RPL resources like memory, CPU time, and energy. The network's performance in terms of PDR is greatly reduced. To solve this problem, version number attack detection techniques are needed to secure the IoT ecosystem.

3.8.6. Implementing VeNADet

Various attacker scenarios are simulated, as shown in Fig. 3.12. In the root node, the VeNADet technique is implemented to evaluate the performance of the technique. The attack detection rate and the efficiency of the VeNADet technique are evaluated. Table 3.2 demonstrates the number of attacks launched in various simulations.

Table 3.2. Attacks Initiated in Different Simulations

Nodes	10% Attacker Nodes	No. of Attacks
10	1	113
20	2	257
30	3	413
40	4	728
50	5	1878

The recommended VeNADet scheme is deployed in the DODAG's root node. The VeNADet detects attacks when a node launches a VNA. The root node then announces the attacker node's information. In different settings, the attack detection rate achieved with 10% of version attack nodes is depicted in Fig.3.16.

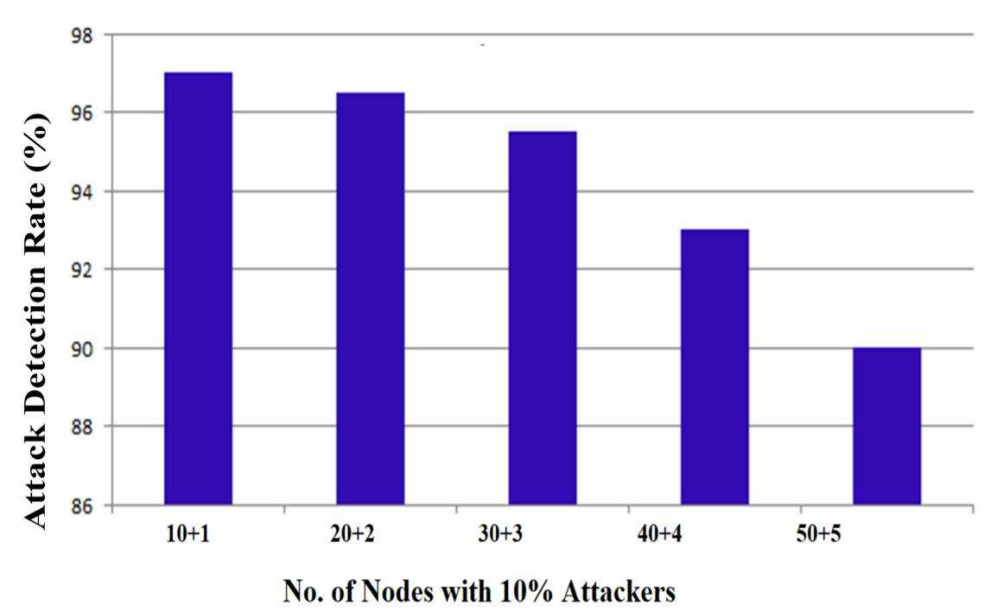


Fig. 3.16. Attacks detected by VeNADet

According to Fig. 3.16, the VeNADet mechanism effectively detects the VNAs in different attacker scenarios. When there are a lower number of attackers in the IoT networks, the VeNADet performs well. On average, VeNADet detects 94.4% of attacks in the attacker simulations.

3.9. Chapter Summary

The version number attack is one of the resource attacks that target the RPL resources indirectly and causes damage to the network. In this chapter, the consequences of the version number attacks are analysed by adding 10% of attackers in different scenarios. The PDR, power usage, and control traffic measures are evaluated in attacker and non-attacker simulations. To counteract the impact of the attack, a mechanism called VeNADet is recommended. The VeNADet technique detects 94.4% of attacks as an average. The findings of the simulations reveal that PDR, energy and control overhead are strongly associated to the number of attackers in the simulation.

When the VeNADet technique is implemented, a node updates its VN only if certain constraints are satisfied. 80% of trust is required among the neighbours and the nodes in the DODAG to modify the VN. Thus, the unnecessary VN modifications are eliminated by implementing VeNADet. The needless resource consumption is prevented by the VeNADet technique.

The next chapter deals with another type of resource attack called 'DIS Attack'. It is a direct resource attack that consumes more resources and creates detrimental effects in the IoT network.

Chapter – IV

DISDet: Detection Technique for DODAG Information Solicitation Attacks in Internet of Things

Chapter – IV

DISDet: Detection Technique for DODAG Information Solicitation Attacks in Internet of Things

4.1. Background

The Internet of Things (IoT) is a promising technology that consists of heterogeneous devices and networks. It includes the conventional Internet and networks of constrained devices connected together using the IP protocol [Lin, 13]. RPL is the routing protocol used in low-power lossy networks (LLNs). RPL was proposed by the Internet Engineering Task Force (IETF) working group [Win, 12]. The nodes in LLNs are resource constrained, so it is a challenging task to implement the security solutions of the traditional networks in the IoT. This makes RPL prone to several security threats and attacks. An attacker may modify, insert, rerun, and generate data or control messages that will affect the normal operations of the RPL [Ahm, 20]. The routing topology for the RPL is like a tree structure that is called a Destination Oriented Directed Acyclic Graph (DODAG). The DODAG is constructed by using a set of control messages.

DODAG Information Solicitation (DIS) is a control message that is sent by a node when it wants to join the existing DODAG topology. Whenever a node in the DODAG receives a DIS message, it has to send the DODAG Information Object (DIO) message to invite the sender to join the network. In a DIS attack, the attacker node unicasts or multicasts a large volume of DIS messages to its neighbour nodes. This tends to cause the legitimate nodes to restart the Trickle algorithm [Lev, 11] and broadcast a large number of DIO messages. The large volume of such messages

consumes energy and other resources of the legitimate nodes and leads them to be unavailable. So, this DIS attack also leads to Denial of Service (DoS) attacks on the IoT network [Con, 19]. This DIS flooding attack targets the resources directly and exhausts them very quickly, reducing the lifetime of the neighbour nodes of the attacker in the networks.

To overcome the negative impacts of the DIS flooding attack, in this Chapter, a novel detection technique named DISDet is proposed. DISDet avoids the overwhelming DIS request from the malicious node and discards the duplicate messages sent by the same node again and again. Hence, the DISDet mechanism reduces packet loss, energy consumption issues, and control overhead and increases the packet delivery ratio. The attacker node is also identified and quarantined by the DISDet technique. The proposed DISDet technique is implemented in the Cooja simulator and compared with some recent techniques. DISDet performs well in terms of detection accuracy rate.

4.2. Related Works

Only a few works have been published on the DIS flooding attack. In this section, some of the important works related to this research are highlighted.

Cong Pu [Con, 19] investigated the DIS spam attack, which sent large volumes of DIS messages with different identifiers and led to a DoS attack in an RPL-based environment. The research was implemented in the OMNet++ simulator. The researchers also evaluated the negative impacts of the DIS flooding attack on the lossy networks in terms of energy consumption and node lifetime. The attack detection phase was not considered in this work.

Abhishek et al. [Abh, 19] conducted an experiment on DIS flooding attack. The memory and energy consumption of such an attack were evaluated, and they proposed a lightweight technique called Secure-RPL for mitigating such attacks in order to improve the performance of the lossy networks. The attack detection rate of the proposed technique and the impacts on network performance after implementing the technique were not given in their work.

Faiza et al. [Fai, 21] investigated the consequences of the DIS flooding attack both on dynamic and static PRL networks. They presented RPL-Maximum Response Code (MRC), a technique for improving RPL resistance against DIS Multicast attacks. The goal of RPL-MRC is to shorten the time it takes for DIS Multicast messages to be responded to. According to their experiment, the attack degraded network performance by considerably increasing control packet overhead and power consumption. For several scenarios, the RPL-MRC mechanism showed a considerable improvement in lowering control overhead and energy usage. This technique only delays the response to the DIS message.

4.3. Motivation and Problem Definition

4.3.1. Motivation

A node that is not connected to the DODAG requests the routing information by transferring a DIS message to its neighboring nodes. On receiving this message, the neighbours reset their trickle timer algorithm and sent back the DIO message. An attacker node sends multiple DIS messages in order to get back multiple DIO messages frequently. This increases the number of control messages and makes the network resources unavailable. To overcome this issue, the DISDet technique is proposed in this Chapter. The DISDet technique detects the malicious node and safeguards the IoT networks.

4.3.2. Problem Definition

Due to the resource constrained nature and the inclusion of voluminous devices, the IoT network tends to be more vulnerable and is prone to several attacks. The control messages of the RPL protocol construct the topology for data transmission in the network. The DIS message is one of the control messages that facilitate a new node joining the DODAG. The DIS flooding attack is launched by flooding a lot of DIS messages. The nodes that are receiving the DIS messages have to send the DIO messages in turn, which increases the control message overhead in the DODAG. On account of this control traffic, the network resources are unavailable for other functionalities. Therefore, the harmful impacts of the DIS flooding attacks should be addressed to reduce resource consumption.

4.4. Objectives

This Chapter focuses on the detection mechanism called DISDet, for detecting DODAG Information Solicitation Attacks in IoT networks. The objectives of the Chapter are listed below:

- To simulate the IoT nodes without any DIS attacker and with a DIS attacker and analyse the PDR, energy consumption, and control overheads of the two scenarios.
- To implement the DISDet technique in the attacker scenario and measure the PDR, energy consumption, and control overhead, and to measure the detection accuracy of the proposed technique.
- To discard the large number of DIS messages sent by the malicious node
- To isolate the DIS attacker node by broadcasting the attacker's details and disconnecting the communication channels.

4.5. DIS Attack

4.5.1. DODAG Information Solicitation (DIS)

When a new node wants to join the DODAG, it requests the routing details from its neighbor nodes by sending the DIS message. While receiving the new node's request, the neighbours have to respond using the DIO message. In order to join the network, a new node can continuously transmit the DIS message until it receives a DIO message [Abh, 19]. The DIS message sent by a node to join the DODAG is illustrated in Fig. 4.1.

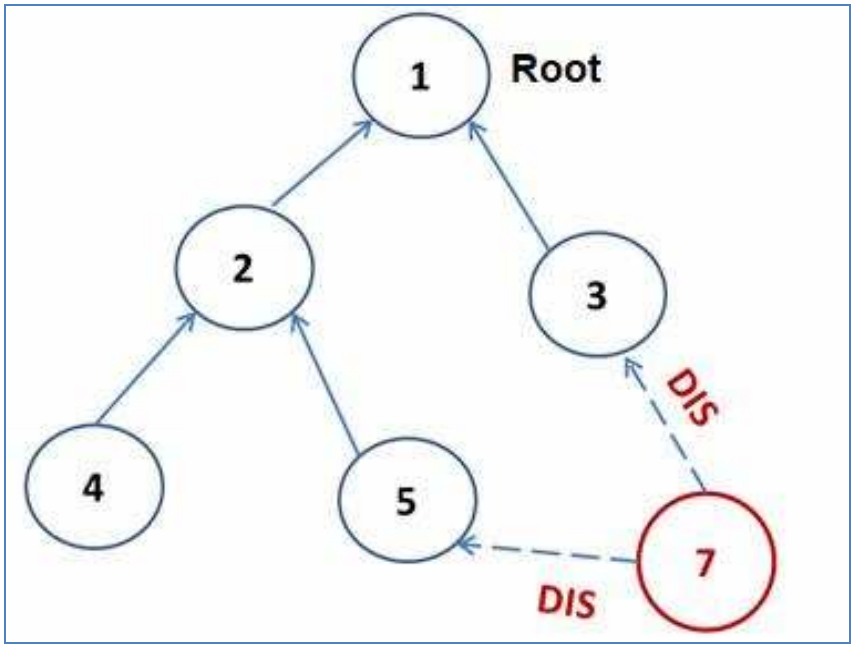


Fig. 4.1. New Node Joining DODAG using DIS Message

As it is given in Fig.4.1, the node '7' is not in the DODAG, so it sends DIS messages to its neighbor nodes '5' and '3'. The DIS message is transmitted periodically until the node '7' gets a DIO message from any of its neighbors. The DIS message is represented using the dotted line since the node '7' is not yet connected to the DODAG.

4.5.2. DODAG Information Solicitation (DIS) Attack

Whenever a legitimate node receives the DIS message from its adjacent nodes, it sends the DIO message to invite the node to join the DODAG. In order to send the network routing information periodically, the trickle algorithm is set for each node. As per the network stability of the DODAG, the duration for initiating the Trickle Algorithm differs [Con, 20]. When a node gets a DIS message from its adjacent nodes, it ends the current DIO transmission and reruns the Trickle Algorithm, and the time is set as a minimum.

There are two variables available for DIS messages, which are DIS_DELAY and DIS_INTERVAL. The DIS_DELAY is the delay to initiating the first DIS message by the new node. The DIS_INTERVAL is the waiting time for the new node to send the second and so on DIS messages when it may not receive any DIO messages from its neighbours during that interval. So, a node has to wait for the DIS_INTERVAL time to send each DIS message. The default value defined in RPL for DIS_DELAY is 5 seconds, and DIS_INTERVAL is 60 seconds. An attacker doesn't wait for the DIS_INTERVAL time to send the next DIS message. It continuously sends the DIS message to its neighbours though it also receives the DIO message from the neighbours. This misbehaviour of the malicious node increases the control message overhead in the network, disrupts the network topology, and consumes more resources like memory, processing time, and energy. This attack is also known as a "DIS spam flooding attack". It causes a Denial of Service (DoS) attack by disrupting subsequent data communication in the DODAG. Fig. 4.2 depicts the DIS flooding attack concept.

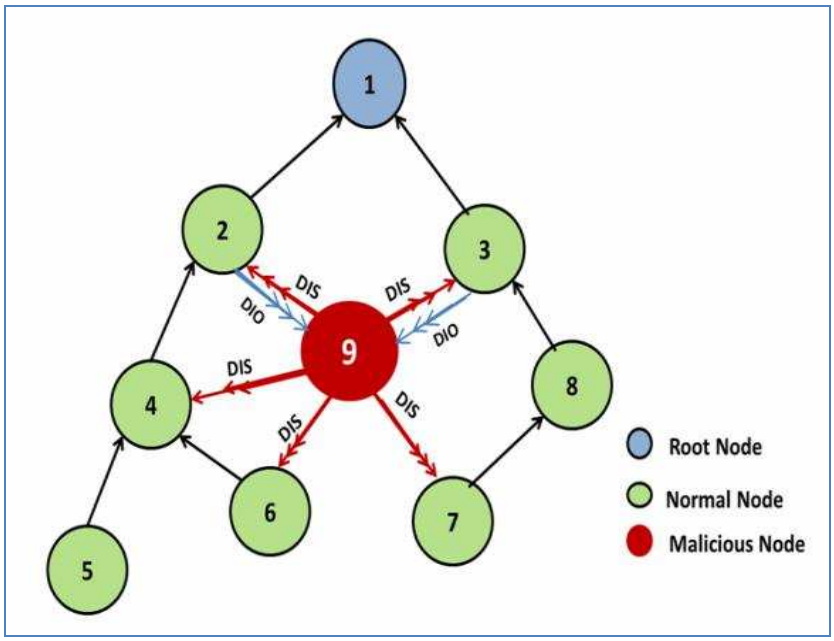


Fig. 4.2. DIS Flooding Attack

As it is given in Fig.4.2, node '9' is the malicious node which sends multiple DIS messages to its nearby nodes. The multiple arrows from the node '9' indicate the multi-requests initiated by the attacker node. On receiving these requests, the adjacent nodes '2', '3', '4', '6' and node '7' restart the Trickle Timer Algorithm to send the DIO message to the node '9'. This malicious act of the node '9' consumes the resources of the IoT nodes like memory, processing, and energy and also jam the routing process. This unnecessary resetting of the trickle timer and the increase in the number of control messages make the network and communication links unavailable for the resource-limited IoT devices.

4.5.3. DIS Attack Model

Let $X = \{x_1, x_2, x_3, \ldots\ldots\ldots, x_n\}$ is a set of new nodes that want to join the DODAG, where $|X|=n$. The neighbors of each new node are represented by using $Y = \{y_1, y_2, y_3, \ldots, y_m\}$ where $|Y|=m$. Here, X and Y are distinct sets. Hence, $X \cap Y = \Phi$. The sets X and Y form n xm matrix.

After a small interval Δt (DIS_DELAY), the new nodes send first DIS messages to their neighbors to get back DIO message. The DIS flooding attack may occur during the time t (DIS_INTERVAL). The DIS attacks for the new nodes can be occurred in I different values, x_i $(i = 1, 2, \ldots, I)$, and the probability of the value x_i being taken is $P(x_i)$. The set of numbers $P(x_i)$ is said to be a probability mass function. The variable $P(x_i)$ takes one of the values as it is given in Eq. 4.1.

$$\sum_{i=1}^{I} P\left(X_i\right) = 1 \tag{4.1}$$

It takes any value in this $0 \leq P_X(x_i) \leq 1$ range for all x_i. The probability mass function can be defined as in Eq.4.2.

$$P(x_i) = m_i / Z \tag{4.2}$$

Where, Z is the total number of DIS attacks and m_i is the node from which the DIS attack is initiated. To represent the number of attacks, whenever new nodes enter the network,an $n \times m$ matrix M_{ij} is constructed. The row elements are the new nodes and the column elements are their neighbors. Whenever there is a DIS message sent from a node to its neighbors after the DIS_DELAY and before the completion of the DIS_INTERVAL, the corresponding M_{ij} value is incremented.

4.6. Proposed DISDet Technique

The DISDet technique has been proposed in this section as a countermeasure to address the DIS attacks on RPL-based IoT networks. Keeping in mind the characteristics of the DIS message and the DIS flooding attacker, the DISDet technique has been deployed. The unnecessary reset of the trickle timer and the overwhelming DIS message generation are circumvented using this DISDet. The steps involved for detecting the DIS flooding attack are explained using the Fig. 4.3.

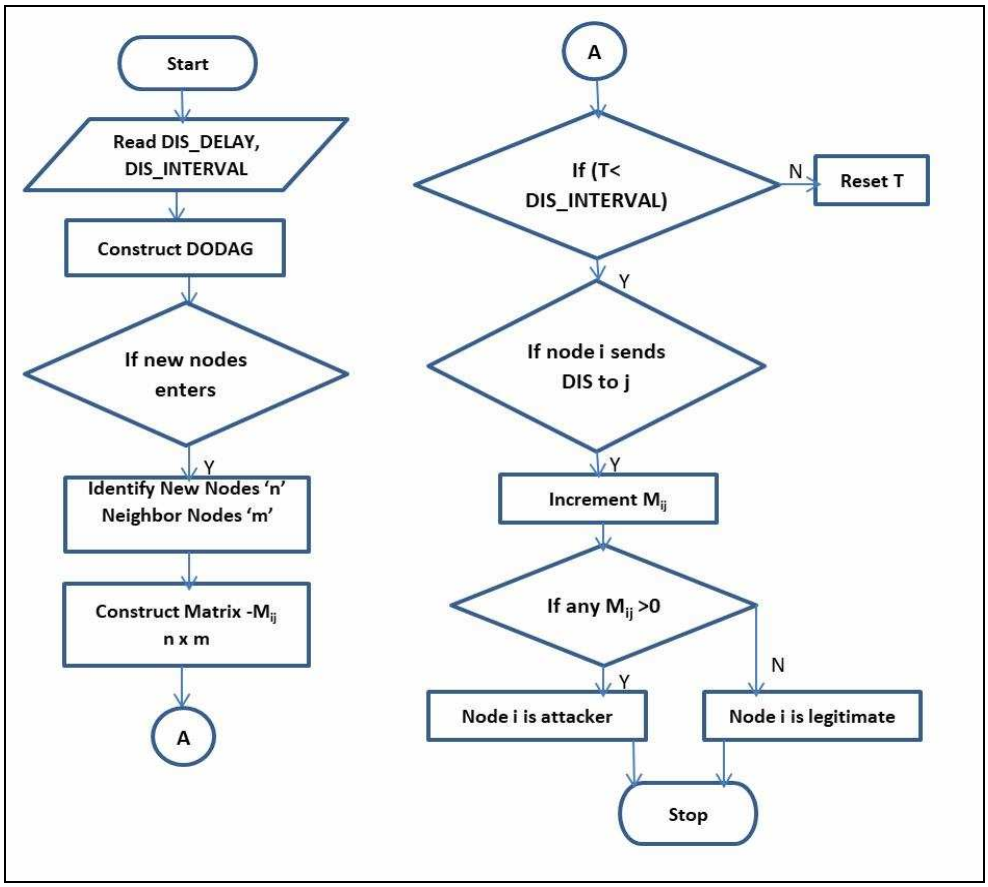


Fig. 4.3. DISDet Technique

As it is shown in Fig.4.3, the DODAG is constructed using the normal procedure. When a new node enters the DODAG, it has to send DIS messages to its neighbors to get back the DIO messages. An authentic node sends the DIS message only after the DIS_INTERVAL. But an attacker sends multiple DIS messages continuously without considering the DIS_INTERVAL time using unicast or multicast strategies. This overwhelming of DIS messages from the attacker increases the control overhead and affects the performance of the DODAG. To address this DIS attack, a matrix M (n x m) is created, where 'n' is the number of new nodes and 'm' is the corresponding neighbors. If there is a single new node, then the matrix will be like a list or array. The n x m matrix M is given in Fig. 4.4.

	1	**2**	**3**	**..**	**m**
1	M_{11}	M_{12}	M_{13}	..	M_{1m}
2	M_{21}	M_{22}	M_{23}	..	M_{2m}
3	M_{31}	M_{32}	M_{33}	..	M_{3m}
…	..	..	..	..	..
n	M_{n1}	M_{n2}	M_{n3}	..	M_{nm}

Fig. 4.4. n x m Matrix to address DIS attack

In Fig.4.4, ‘n’ is the number of new nodes wishing to join the DODAG. The row is denoted by using ‘i’ and column is represented using the variable ‘j’. After the DIS_DELAY (5 seconds), the matrix is created and initialized with zero. The matrix records the number of DIS messages sent from node ‘i' to the node ‘j’ during the DIS_INTERVAL (60 seconds). Whenever a DIS message is generated from a node ‘i’ to the node ‘j’, at the time the corresponding value of M_{ij} is incremented. After the DIS_INTERVAL time, the values in the matrix ‘M’ are analyzed and the attacker is detected using the M_{ij} values as it is given in the Eq.4.3.

$$M_{ij} = \begin{cases} = 0 & if \quad i = normal\ node \\ > 0 & if \quad i = attacker\ node \end{cases} \tag{4.3}$$

Hence, if any cell M_{ij} has a value greater than zero, then the particular ‘i' node is treated as the attacker node. All the messages generated during the DIS_INTERVAL by the node ‘i' are discarded, and the node ‘i' is declared as an attacker and removed from the DODAG. Then the DODAG with the remaining nodes is reconstructed. The timer and the matrix values are reset after the DIS_INTERVAL and the process is continued. The symbols used in the DISDet algorithm are described in Table 4.1.

Table 4.1. Symbols and their Descriptions used in DISDet

Symbol	Description
D_D	DIS_DELAY; 5 Seconds
D_I	DIS_INTERVAL; 60 Seconds
t	Counter clock (60 to 0)
M	n x m matrix
(Row in M) i	A new node i
(Column in M) j	Neighbours of the new node i

The Algorithm used for implementing the DISDet technique used in this research work is explained in Fig. 4.5.

Algorithm 4.1 for Detecting DIS Attack

1. function **DISDet** (New_Node);
   Input: New_Node, DIS_DELAY, DIS_INTERVAL
   Output: Legitimate or Malicious
   Initialization:
   $D_D \leftarrow$ DIS_DELAY *// 5 Seconds*
   $D_I \leftarrow$ DIS_INTERVAL *// 60 Seconds*
2. construct the DODAG
3. **if** (n nodes enters the DODAG)then
4. wait for D_D time
5. construct **(n x m)** matrix
6. matrix $M_{ij} \leftarrow 0$ *// initialize (n x m) matrix*
7. $t \leftarrow D_I$ //*set counter clock t as* D_I
8. **if** ($t \neq 0$ and node_i sends DIS message to node j) then
   //check node i sends DIS within DI
9. increment corresponding M_{ij} value

```
10.     end if
11.     for (i =1 to n) do
12.     for (j= 1 to m) do
13.         if (Mij>0) then  // check attacker node
14.             declare "Node i is an Attacker"   // detect attacker
15.             discard all DIS messages from node i
16.             disconnect communication links for node i//isolate attacker
17.          end if
18.     end for
19.     end for
20.     reconstruct the DODAG
21. end if
22. end DISDet
```

Fig. 4.5. DISDet Algorithm for Detecting DIS attack

The output of the algorithm is to make a decision on whether there is a DIS flooding attack on the IoT network or not and disconnect the attacker. If all the M_{ij} values of the matrix M are zero, then there is no attack in the DODAG. Otherwise, there is a DIS attack and the corresponding i^{th} node is detected as the source of the attack and is eliminated from the DODAG. By implementing this algorithm, the DIS flooding attacks on the RPL-based lossy networks can be detected.

4.7. Experimental Setup

In this experiment, a maximum of 52 nodes with unique identifiers are included to form a 6LoWPAN network using random topology. The border router (6BR) acts as the root node. 51 nodes, including the root, act as normal nodes, and one node is a malicious node that sends DIS messages very often. The sample snapshot taken from the simulation experiment is shown in Fig.4.6.

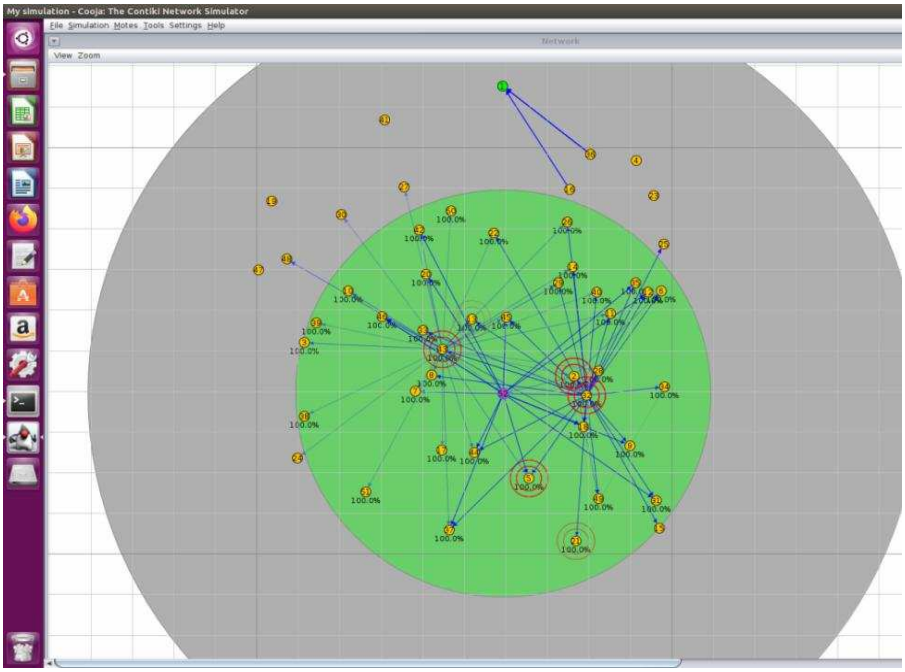


Fig. 4.6. Screenshot with Attacker Scenario

The node represented in green is the root node. The yellow nodes are the normal nodes, and the purple node is the attacker node. The lines between the nodes represent the communication among the nodes. The transmission range (50 meters) is given as the green color region, and the interference range (100 meters) is shown using the grey color region. A simulation grid of size 200 m x 200 m is used in this experiment. The other simulation parameters are as in Chapter 3 (Table 3.1).

4.8. Simulation Results and Discussion

4.8.1. Network Graph

First, the normal scenario with 50 nodes and a root node was implemented in the Cooja Simulator, and the simulation was performed for 30 minutes. The network graph captured from the normal simulation environment is given in Fig.4.7.

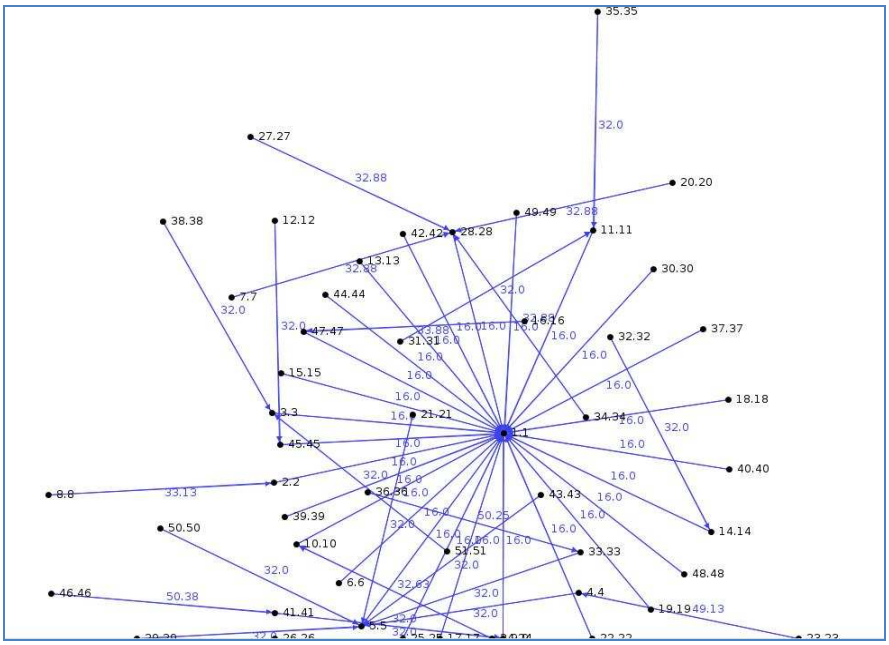


Fig. 4.7. Network Graph in Normal Scenario

As Fig. 4.7 denotes, the root node is connected to all child nodes without any loop in the DODAG. All nodes are able to communicate with one another, and there is no inconsistency in the normal situation. Then a DIS attacker is included in the normal scenario and simulated for 30 minutes. The network graph obtained during the attacker simulation is given in Fig.4.8.

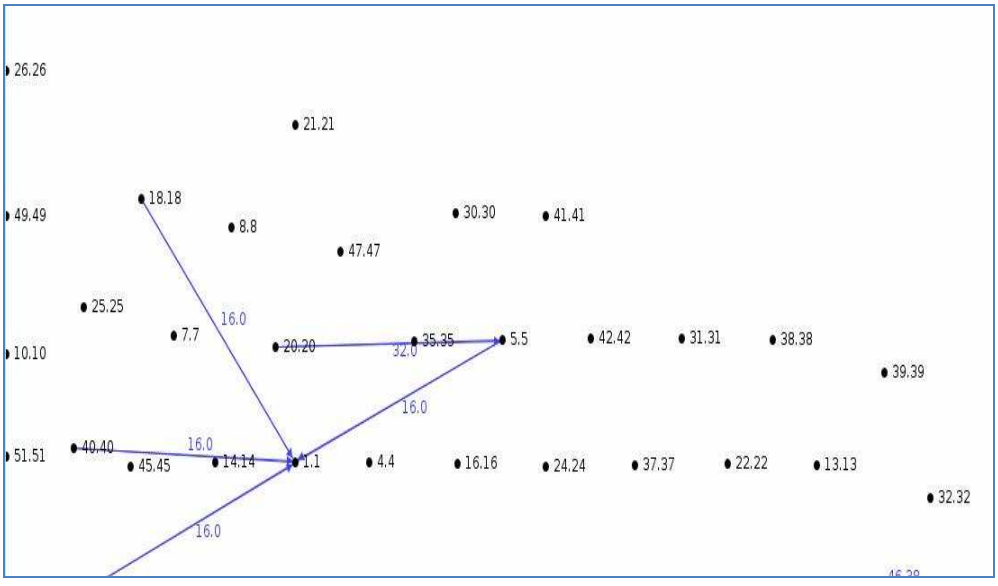


Fig. 4.8. Network Graph in Attacker Simulation

As Fig.4.8 denotes, because of the presence of the attacker node, there is a difficulty in constructing the DODAG and the nodes are unable to communicate. The performance of the network in each scenario is analyzed and compared in terms of the control overhead, power consumption, and PDR.

4.8.2. Control Overhead

To construct the DODAG, a number of control messages such as DIO, DAO, DAO-ACK, and DIS are generated. The increase in the control message in LLNs reduces the performance. In Fig. 4.9, the number of control messages obtained in the normal and attacker environments is shown. The number of control message after 5 minutes, 15 minutes, and 30 minutes of simulation is clearly depicted.

Simulation Time	No. of Control Messages	
	Normal	Attacker
5 minutes	5875	9218
15 minutes	13998	25678
30 minutes	28889	52314

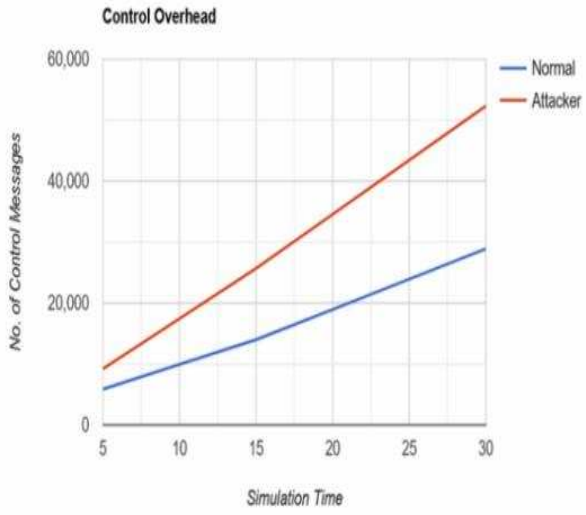


Fig. 4.9. Control overhead in Normal and Attacker Scenarios

As it is given in Fig.4.9, the control overhead is higher in the attacker scenario. Because the attacker continuously sent DIS messages without considering the DIS_INTERVAL, for each DIS message, the Trickle Timer is restarted and the DIO messages are regulated. This increased the control overhead in this scenario.

4.8.3. Power Consumption

Next, the power consumption of each node in normal and attacker scenarios is monitored using the collect view of the cooja simulator. The power consumed by the nodes in the normal scenario is depicted in Fig. 4.10.

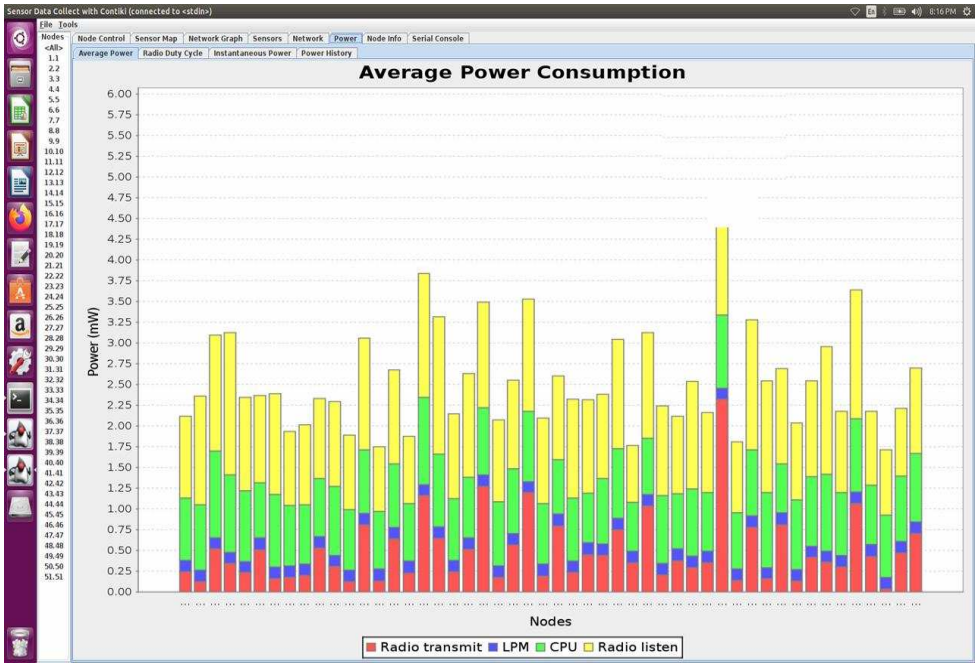


Fig. 4.10. Power Consumption in Normal Simulation

The power consumption for the attacker scenario was also monitored using the collect view. Due to the presence of the attacker node, the root node could not collect the power consumption for all nodes. The power consumption in the attacker scenario is shown in Fig.4.11.

According to Fig.4.10 and Fig.4.11, the nodes in the attacker scenario consumed morepower than the normal scenario. The average power consumed by the normal nodes is 2.5 mW, whereas the attacker simulation nodes consume 6 mW. Hence, the nodes in the attacker environment drop their energy very quickly compared to the normal environment. The lifetime of the nodes is also decreased due to this issue.

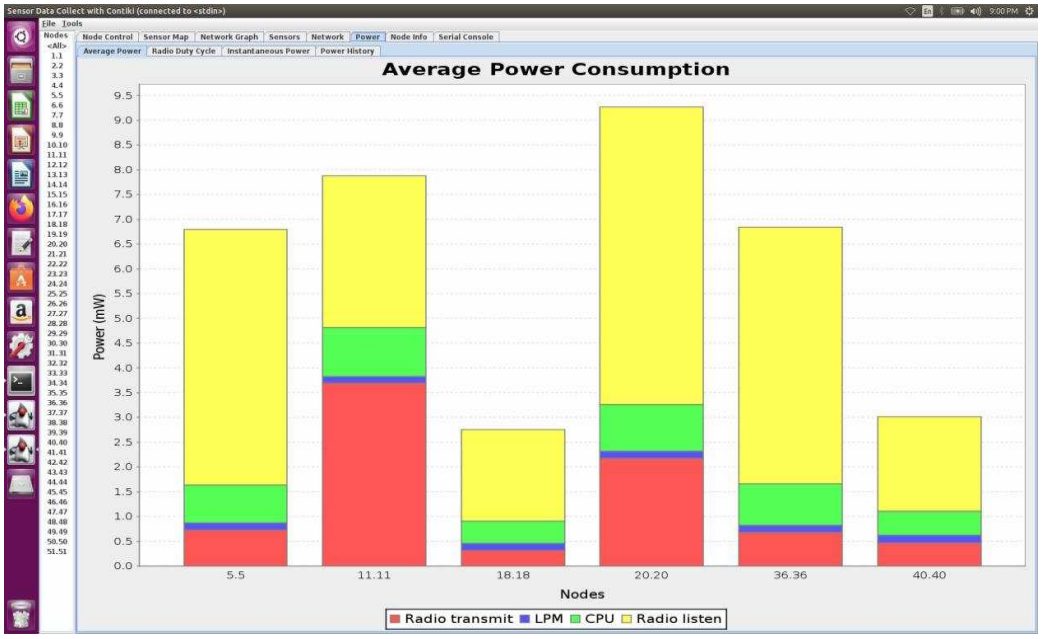


Fig. 4.11. Power Consumption in Attacker Simulation

4.8.4. Packet Delivery Ratio (PDR)

PDR is the ratio between the packets received at the root node and the packets sent from the client nodes. The formula for calculating the PDR is given in Eq.4.4.

$$PDR = \frac{\sum No.of\ \ packets\ received}{\sum No.of\ \ packets\ sent} \times 100 \qquad (4.4)$$

The packets in the normal and attacker scenarios are captured and analysed using the Wireshark tool. The PDR is calculated in the root node based on received and sent packets. The PDR values in both scenarios are given in Table 4.2.

Table 4.2. PDR in Normal and Attacker Scenarios

Simulation Time	Normal	Attacker
5 minutes	99.70%	90.07%
15 minutes	97.97%	84.21%
30 minutes	96.87%	73.58%

As it is shown in Table 4.2, the PDR value is higher in the normal scenario. But it is low in the attacker scenario due to the heavy packet loss in the attack simulation.

4.8.5. Implementing DISDet Technique

The proposed DISDet technique is installed in the border router in order to safeguard the IoT networks from DIS flooding attacks. This technique was implemented by carefully analysing the nature of new nodes and the DIS attacker. The sample screenshot after implementing the DISDet in the attacker scenario is given in Fig. 4.12.

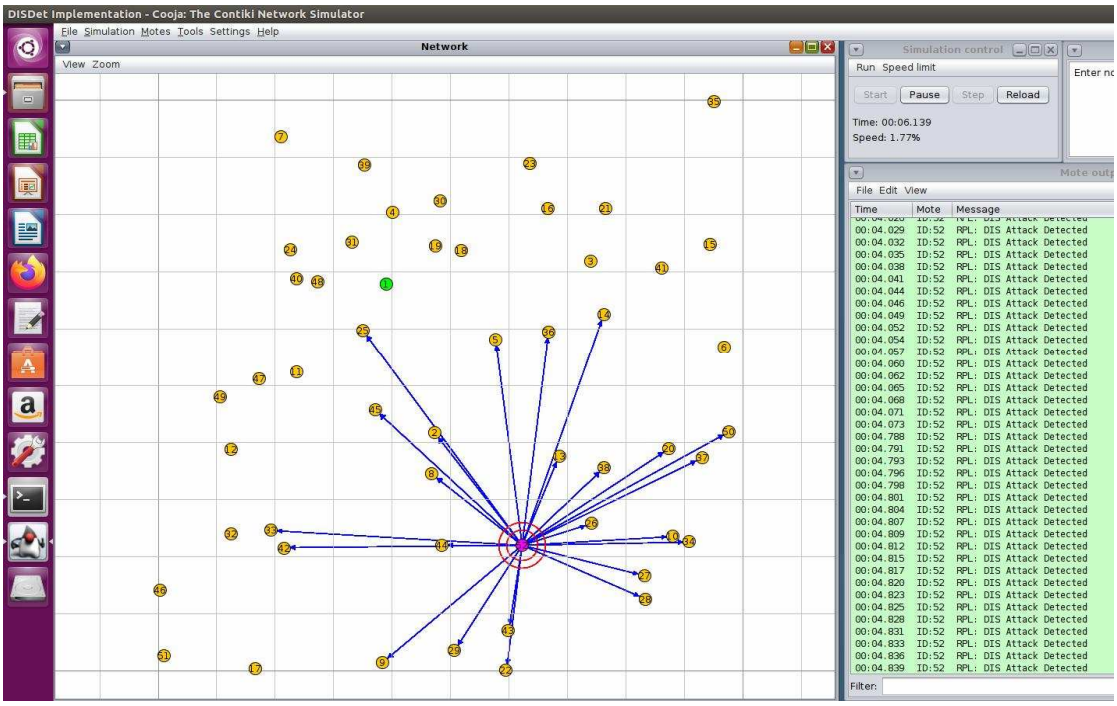


Fig. 4.12. Implementing the DISDet in Attacker Scenario

Fig. 4.12 depicts the DIS attacks detected by the proposed method. The attacks initiated by the DIS attacker (Node_52) are immediately detected by the DISDet system. The DISDet approach almost detects all initiated attacks by the DIS flooding

attackers. The **n x m** matrix created by the DISDet easily identified the attacker and the number of attacks entered the networks from a particular node. The attacks initiated and detected are listed in Table 4.3.

Table 4.3. Attack Detection Rate of DISDet

Simulation Time	Attacks	Detected	Detection Rate
5 minutes	187	185	98.93%
15 minutes	463	459	99.14%
30 minutes	922	908	98.48%
Average Detection Rate			**98.85%**

The DISDet detection technique detected **98.85%** of attacks during the simulation period. When a node is detected as an attacker, its details are declared and isolated from the DODAG, and the root node initiates the global repair process in which the attacker is excluded from the new DODAG. Fig. 4.13 illustrates the DODAGs before and after implementing the DISDet.

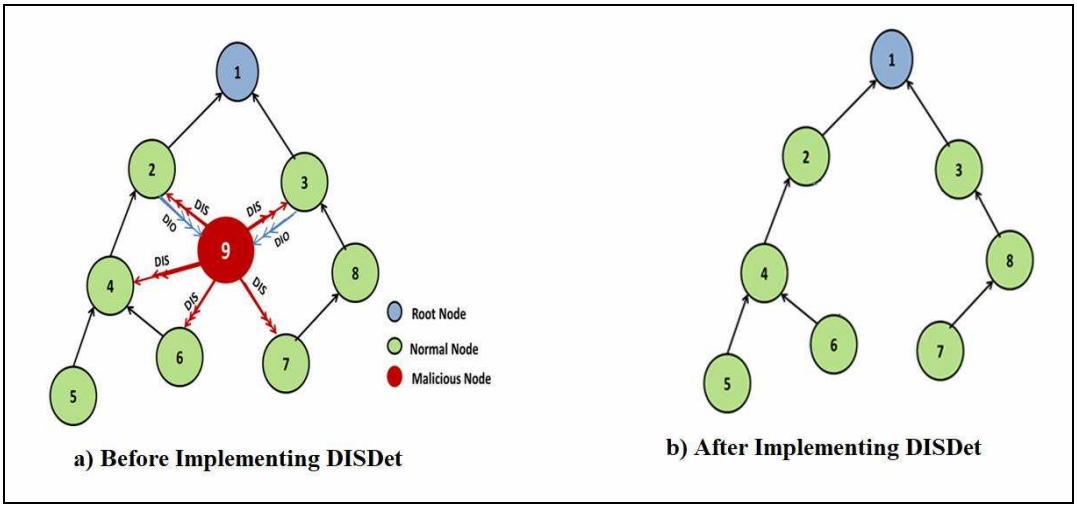


Fig. 4.13. Before and After Implementing DISDet

As Fig. 4.13 depicts, the attacker node '9' is removed from the DODAG after the implementation of the DISDet. The elimination of the attacker node from the

topology saves the resources of the legitimate nodes and makes them available for their normal routing responsibilities. Thus, the negative impacts of the DIS flooding attacks are minimized.

4.9. Comparison of DISDet with Existing Techniques

The DISDet technique was compared with the existing DIS attack research work [Con, 19] and [Abh, 19]. The various parameters used in their research work are compared, and the outcome of the comparative analysis is listed in Table 4.4.

Table 4.4. Comparative Analysis with Existing Research

Parameters	Cong Pu [Con, 19]	Secure-RPL [Abh, 19]	DISDet
Maximum Nodes	30	16	52
Attacker Nodes	1	1	1
Simulator	OMNeT ++	Cooja	Cooja
Performance Metrics	Energy, Node Lifetime	Energy, Control Overhead	Energy, PDR, Control Overhead
Simulation Time	5000 seconds	900 seconds	1800 seconds
Node Type	CC2420	Z1 Mote	Sky mote
Communication Range	30 meters	50 meters	50 meters
Detection	No	No	Yes

In the existing work, the detection rate is not given in terms of quantitative measures. Hence, it was not included in the table. As it is given in Table 4.4, the proposed DISDet technique is implemented using 51 legitimate nodes and one attacker. The PDR and detection accuracy are also considered in the proposed approach.

4.10. Chapter Summary

In this Chapter, the characteristics and the negative impacts of the DIS flooding attacks were analyzed. The DIS flooding attacks consume more network resources like memory, processing and energy. It is a direct resource attacks which increases the control overhead in the LLNs. The proposed DISDet technique detects the DIS flooding attacks efficiently. After implementing the DISDet technique in the attacker scenario, the network performance is improved. The proposed technique was also compared with the existing recent research works. According to the experiment, the DISDet technique achieved **98.85%** of detection accuracy.

Another RPL resource attack called Destination Advertisement Object (DAO) Attack and a novel technique to detect the DAO attack are addressed in the next Chapter.

Chapter – V

DADTec: DAO Attack Detection Technique for RPL-based Internet of Things

Chapter – V

DADTec: DAO Attack Detection Technique for RPL-based Internet of Things

5.1. Background

RPL is susceptible to a large number of security threats and attacks. The nodes in the RPL are resource-constrained, and the security mechanisms like cryptographic algorithms are not possible due to their heavyweight nature in terms of memory and processing requirements. This allows RPL to be vulnerable to several attacks in the IoT environment [Anu, 14]. In RPL-based LLN, the routing is performed after forming the Destination Oriented Directed Acyclic Graph (DODAG). By altering the details of the control messages, a number of attacks are created. These attacks target resources, network, and traffic. The attacks that target the resources, consume more energy, memory, and CPU time than the normal RPL nodes. DAO attack is one of the resource attacks that consume more resources that are constrained to the IoT devices.

When a parent node sends a DODAG Information Object (DIO) message to its children, the receiver nodes have to respond by sending back a DODAG Advertisement Object (DAO) message. The DAO attack is initiated by sending a large number of DAO control messages from an RPL child node to its parent node, and the parent node forwards the same until the DAO message reaches the DODAG root node. The malicious node generates more control messages, which increases the control overhead in the network, consumes energy and memory, and degrades the DODAG performance in terms of PDR and throughput. In this chapter, the impacts of the DAO attacks and the DADTech technique to detect them are discussed in detail.

5.2. Related Works

In this section, the state of art for DAO attacks is discussed elaborately. The DAO attacker node sends voluminous DAO packets to its parent node which increases the control overhead and decreases the network performance. In some cases, the attacker node drops the packets and replies with an error packet to its parent node, which causes the parent node to discard the valid downward routes.

To limit the harmful impact of DAO attacks on RPL networks, Cong Pu [Con, 18] suggested a Dynamic Threshold Mechanism (DTM). For each parent node, a dynamic threshold for tolerating forwarding error packets is determined throughout a time period according to this DTM method. In this approach, the DAO attacks were identified and the malicious nodes were eliminated.

Isam Wadhaj et al. [Isa, 20] examined the effects of the DAO attack on the RPL-based IoT network and provided a mitigating strategy for evaluating the recommended technique's performance. The simulation results indicated that the proposed strategy performed better in different scenarios. It showed better performance in terms of control traffic, delay, power usage, and packet delivery ratio.

Baraq Ghaleb et al. [Bar, 19] discussed the DAO insider attack using 50 nodes and proposed a strategy to mitigate its negative consequences. According to their experiment, the DAO attacker node not only creates trouble for the immediate parent node, but also for all ancestor nodes, since the DAO message traverses all intermediate nodes in the DODAG until it reaches the root node. The impact of such an attack on control traffic and power usage was studied. The mitigating approach entails setting a limit on the number of DAO messages sent to each destination.

Ahmed et al. [Ahm, 20] performed an investigation into the DAO induction attack. The attacker's performance using network metrics such as power usage, delay, and PDR was evaluated in both storing and non-storing mode. A lightweight security solution was suggested by the authors to identify such DAO induction attacks. As per the obtained result, the DAO attack causes more destructive effects on non-storing operations.

5.3. Objectives

This Chapter analyses the DAO attacks and their effects on the RPL-based IoT network. The proposed detection mechanism called DADTec for detecting the DAO attacks is also elaborated in detail. The objectives of the Chapter are listed below:

- To simulate the IoT nodes without any DAO attacker and with a DAO attacker and to analyse the PDR, energy consumption, and control overheads of the two scenarios.
- To propose a technique called DADTec for detecting DAO attacks.
- To implement the DADTec technique in the attacker scenarios and measure the PDR, energy consumption, and control overhead, and to measure the detection accuracy and false alarm rate of the proposed method.

5.4. DAO Attacks in RPL

5.4.1. DAO Attack Scenario

The DAO attacker node sends intermittent DAO messages to the set of parent nodes. The same messages are transmitted until they reach the root node. This nefarious conduct dumps the upstream channel towards the root with excessive DAO messages, lowering the packet delivery ratio and increasing the network traffic. The

malicious activity's eventual result is a Denial of Service (DoS) attack on the normal parent nodes. The attacker's devastation has a broader scope as the DAO message traverses from the malicious node to all the ancestor nodes on its route to the root node [Con, 18].

The DAO messages have to pass through a number of intermediary nodes to reach the root node. If the DAO attack is triggered by the leaf node, it causes substantial network disruption. Fig. 5.1 illustrates the DAO attack in several scenarios.

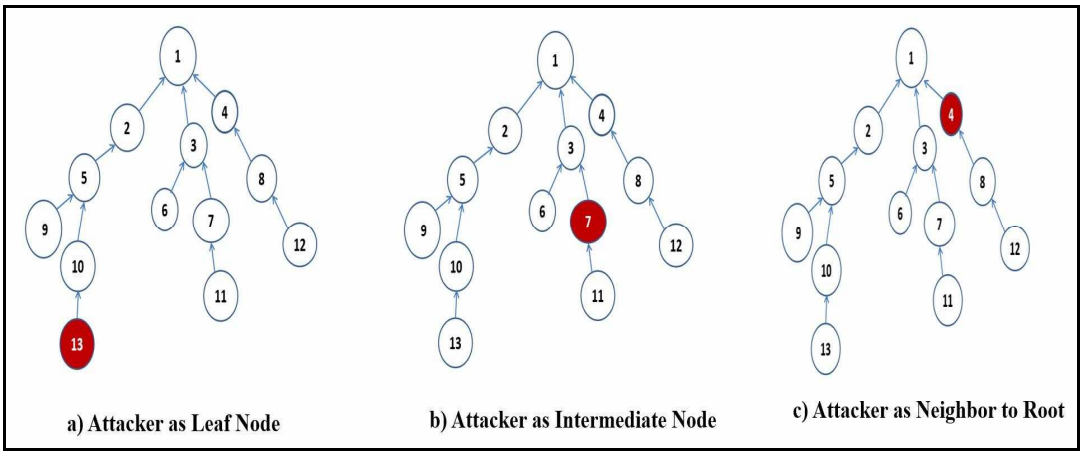


Fig. 5.1. DAO Attacker in Different Locations

As it is given in Fig.5.1, the attacker node is placed as a leaf, intermediate, and neighbor node to the root node. There are thirteen nodes in each scenario, including the root node and an attacker. The root node in this case is ‘node 1’. The attacker is represented by the color red, while all other nodes are considered normal nodes.

Take a look at Fig.5.1 (a), where the attacker node is positioned as a leaf node. To reach the root node, it must make four hops. Nodes 10, 5, 2, and 1 get a significant number of DAO messages from the attacker node '13.' All intermediary nodes are affected by the existence of the attacker node. It takes more time for computation and consumes a lot of network resources like memory and power. The resource exhaustion is higher in this scenario than in any other.

Two hop counts are required to reach the destination (root) from the malicious 'node 7' in Fig.5.1 (b). The destructive effects of the DAO attacks are less severe in this scenario than in the prior one. The attacker 'node 4' in Fig.5.1 (c) sends its messages to the root in a single hop. When the attacker node sends a lot of DAO messages, this scenario is equally dangerous. However, when compared to the alternatives, it is a lesser evil. The rank is calculated using the hop count in this case. According to Fig.5.1 (a), the attacker node takes four hops to reach the root node, giving it a rank of four. In Fig.5.1 (b) and Fig.5.1 (c), the attacker nodes have a rank of 2 and 1, respectively. If there are five DAO messages sent by the attacker nodes in a second, then the control overhead caused by the DAO and DAO-ACK messages is listed in Table 5.1.

Table 5.1. The Negative Impacts of DAO Attacks

Scenario (Attacker as)	Rank of the attacker	DAO messages from attacker per second	DAO-ACK for each DAO message	DAO, DAO-Ack messages
Leaf node '13'	4	20	20	**40**
Intermediate Node '7'	2	10	10	**20**
Neighbour to Root Node '4'	1	5	5	**10**

In Table 5.1, only the DAO and DAO-ACK messages are taken into account. The control traffic will be higher if other control messages such as DIO and DIS are also included. As shown in Table 5.1, when the DAO attacker acts as a leaf node, it increases the network traffic in the DODAG, causing increased power usage and degrading the network's performance. Compared to other circumstances, when the attacker node is placed as the leaf node, the attacker's detrimental effect is increased.

5.4.2. DAO Attack Model

Let D = (N, L) denote a DODAG with N nodes and L node-to-node connections. Each DAO attacker node $n \in N$ affects its parent node *m* at a time t, with a probability profile *p (t, μ, n, m).* This probability depends on the attacker node *n,* parent node *m,* time *t,* and level of harmfulness μ, due to the DAO attack. The attack probability profile increases according to the level of virulence of the attacker nodes. Hence,

$$p(t,\mu_2, n, m) > p(t,\mu_1, n, m)$$

$$when\ \mu_2 > \mu_1$$

When the level of virulence approaches maximum or infinity, the probability equals one, ensuring the occurrence of a DAO attack on the network. This can be represented using Eq. 5.1.

$$\lim_{\mu \to \infty} p(t, \mu, n, m) = 1 \qquad (5.1)$$

When this occurs, the parent node of the attacker node is not able to carry out its legitimate responsibilities, which leads to a Denial of Service (DoS) attack. Hence, the following things are obtained at time t.

- A specific set *U C N* is isolated as it is infected by the DAO attack and after the global repair mechanism, the new DODAG D' is constructed, which is induced by Eq.5.2.

$$N' = N - U \qquad (5.2)$$

- A dynamic threshold T is set for each parent node according to the number of children it has. This threshold value limits the DAO messages from the child node. A DAO counter (D_Cnt) is also assigned to the parent node to count the number of DAO messages it receives from the child node for each DIO message.

- Each node $x_0 \in N'$ is chosen as a legitimate node if the DAO counter of the parent node is less than the dynamic threshold (D_Cnt < T)
- The remaining nodes in $N' - \{x_0\}$ are treated as vulnerable nodes.

Using this DAO attack model, the DAO attack can be defined and detected.

5.5. The DADTec Technique

The DAO attacker node floods the parent node with multiple DAO messages, and the same are transmitted towards the root node. The RPL network suffers much greater harm as a result of this. In this section, a strategy named 'DADTec' is presented to counteract the negative consequences of DAO attacks. In this DADTec, a dynamic threshold T is assigned to each parent node to restrict the DAO messages from its child node. A child node can send at most one DAO message for each DIO message. Hence, a subordinate node can send up to T messages when it receives a DIO message.

Let C_Node be the child node, P_Node be the parent node, and D_Cnt (x_i) represent the DAO message counter for node x_i. The counter value is increased by one if C_Node sends a DAO message to P_Node. The DADTec is designed to identify DAO attacks in the IoT context using these terms. This approach can be used in both the RPL's storing and non-storing modes. The routing information is recorded in the router node for the storing mode of operations. However, in the non-storing mode, each node is responsible for its own routing details. By using Fig. 5.2, the steps of the DADTec are discussed.

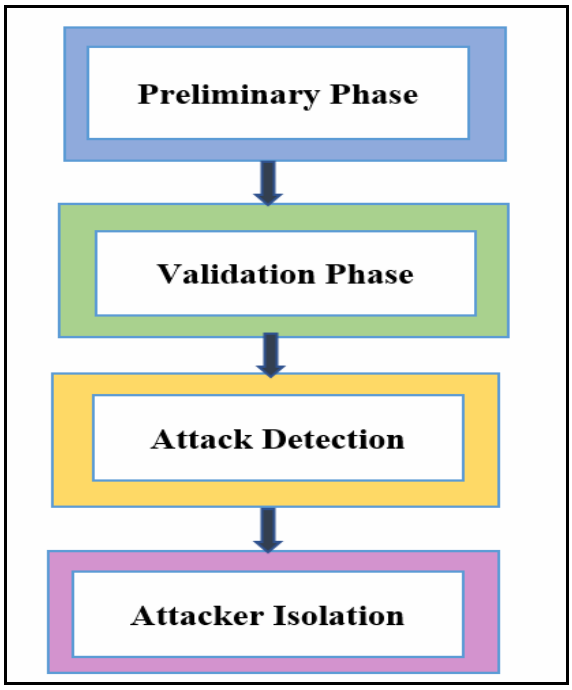


Fig. 5.2. Different Steps of DADTec

- **Preliminary Phase:** The preliminary phase is used to build the DODAG topology and assign the threshold T for each parent node. This threshold T is a dynamic threshold which varies according to the number of child nodes of each parent node. A counter, D_Cnt(x_i), is also assigned to each parent node. Initially, the counter value is set to zero. Whenever the parent node sends a DIO message, the child node replies with a DAO message. For each DAO message, the D_Cnt(x_i) value is incremented.
- **Validation Phase:** The validation step begins when the DODAG has been constructed. This phase determines if the DODAG has any DAO attackers. The D_Cnt(x_i) is incremented by one whenever a P_Node receives a DAO message from the C_Node. Each counter value is compared to the threshold value 'T' in this validation step. The attack detection phase receives the outcome of the comparison.

- **Attack Detection:** In this attack detection phase, from the leaf node to the root node, all paths are identified and the D_Cnt (x_i) is examined for each node. The node is determined as an attacker or legitimate node by checking the following conditions.

  if (D_Cnt (x_i) > T) then, node x_i is an attacker

  else, node x_i is legitimate

  Based on the outcome of the above condition, a node's status is determined as whether it is an attacker or not. If 'Node x_i' is an attacker, then it is declared as a DAO attacker and its details are given to the next phase. This procedure is carried out until all nodes have been verified.

- **Attacker Isolation:** The isolation step of the DADTec begins when the DAO attacker is detected. The root node broadcasts the discovered attacker details in this DADTec approach. The genuine nodes seize their communication channels with the attacker nodes when they hear this. By removing the attacker node, the DODAG is rebuilt. Table 5.2 lists the symbols used in the DADTec algorithm.

Table 5.2. DADTec Symbols and Descriptions

Symbol	Description
X= {x_1, x_2, ..x_n}	Set of all nodes in the DODAG
T	Dynamic threshold for each parent
R_node	Root Node
P_node	Parent Node
C_node	Child Node
L_node	Leaf Node
D_Cnt	Counter: contains the number of DAO messages
Φ	Null value
P	No. of paths from leaf to root node

The different types of symbols used for the DADTec technique are listed in Table 5.2. DADTec is a simple technique that may be applied to a border router to successfully identify DAO attacks. DADTec classifies a node as malicious when its counter value exceeds the permitted threshold value, T. As it is a lightweight technique, it can be used both in storing mode and in non-storing mode. But the non-storing mode is addressed in this study.

The procedure starts from the leaf node and ends at the root node. All paths from a leaf node in the DODAG are considered. For each DIO message from the parent node, according to the D_cnt value, it is determined whether the node 'x_i' is an attacker node or not. The DADTec algorithm is expounded using Fig.5.3.

Algorithm 5.1 for Detecting DAO Attacks

1. function **DADTec** ( );
   Input: X= {x_1, x_2, … , x_n} *// All nodes in the network*
   Threshold T *// Dynamic Threshold*
   Output: Legitimate or Malicious
   Initialization:
   D_Cnt(x_i)← 0 *// Initialize DAO counter to zero*
2. construct the DODAG
3. **for** (k =1 to p) **do** *// each path in the DODAG*
4. **while** (P_Node[x_i]) ≠ Φ then **do** *// checking whether it is root*
5. **for** (i= L_Node to R_Node) **do** *// from Leaf node to Root*
6. **if** (x_i.MessageType== "DAO") **then** *// check DAO message*
7. increment D_Cnt (x_i) *// increment the counter*
8. **end if**
9. **end for**
10. **end while**
11. **end for**
12. **for** (i= 1 to n) **do**

```
13.     if (D_Cnt(x_i)>T) then   // check the counter exceeds T
14.             declare "Node x_i is an attacker"      // detect attacker
15.             initiate global repair                 //reconstruct DODAG
16.             disconnect communication links for node x_i    //isolate attacker
17.     else
18.             print "Node x_i is legitimate"          //x_i is legitimate node
19.     end if
20. end for
21. end DADTec
```

Fig. 5.3. The DADTec Algorithm

The DADTec algorithm is a simple technique that is deployed in the border router for the purpose of detecting DAO attacks successfully.

5.6. Experimental Setup

The DAO attack and its countermeasure, known as the DADTec approach, are implemented in the open-source Contiki [Vik, 20] operating system, which was designed by the Swedish Computer Science Institute for developing sensor networks. The high-level network simulator for WSN, Cooja [Ang, 21], was created for the Contiki Operating System. The IoT devices in this experiment are Tmote Sky [Sop, 21] nodes, which are low-power wireless modules.

The DODAG creation is started by using the Border router as the root/sink node. The DAO attack is carried out by altering the RPL protocol stack. The attacker node, like the other nodes, takes part in the DODAG creation. After a while, the fraudulent node sends DAO messages to the parent list incessantly. Each DAO message delivered by the fraudulent node is routed through its parent list to the root. The simulation setup and parameters of this experiment are listed in Table 5.3.

Table 5.3. Simulation Parameters

No. of normal nodes	50
No. of attacker nodes	1
Mote Type	Tmote Sky
Operating System	Contiki 3.0.
Simulator	Contiki Cooja
Topology	Random
Radio Medium	Unit Disk Graph Medium (UGDM): Distance Loss
Topology Dimension	150m x 150m
Transmission Range	50m
Interference Range	100m
Tx Ratio	100%
Rx Ratio	100%
Simulation Duration	30 minutes per simulation

In the attacker scenario, a high volume of DAO messages increases the control traffic in the RPL-based network. IoT network performance is measured in terms of energy usage, packet delivery rate, and network traffic. This experiment includes 50 nodes, including the root and a DAO attacker. The attacker and normal simulations are executed in four separate scenarios. They are: simulation without attack; an attacker placed as a leaf; an attacker in an intermediary location; and an attacker as a child of the root node.

5.7. Results and Discussion

The control overhead, PDR, and energy consumption levels were assessed during the simulation under normal and attacker simulations. The network performance in both scenarios is evaluated. Fig.5.4 is an example snapshot with 50 normal nodes.

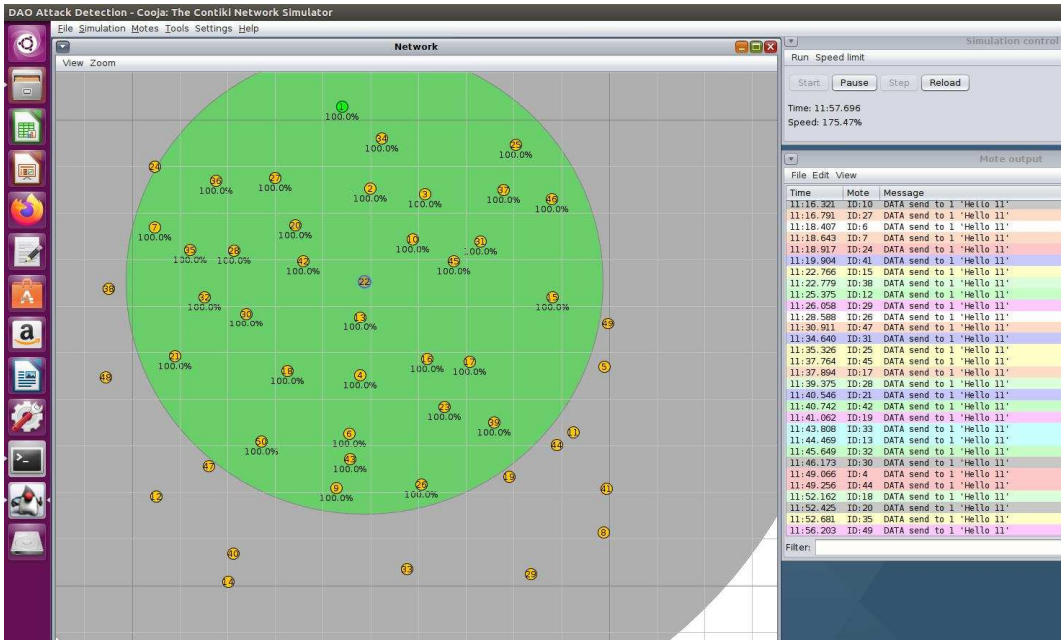


Fig. 5.4. Sample Screenshot for Normal Simulation

The border router is the green-colored node in Fig.5.4, whereas the yellow-colored nodes are client nodes. The transmission range of the given node is shown by the green-colored circular region, while the inference range is represented by the grey-colored circular region. For the performance evaluation, a normal situation is replicated without the attacker, and the PDR, control traffic, and power consumption are observed.

A malicious node is deployed in three distinct locations in the normal environment, and the destructiveness of the node is quantified in terms of control overhead, PDR, and power consumption. Fig.5.5 depicts the attacker scenario and the deployment of the attacker node in three distinct places.

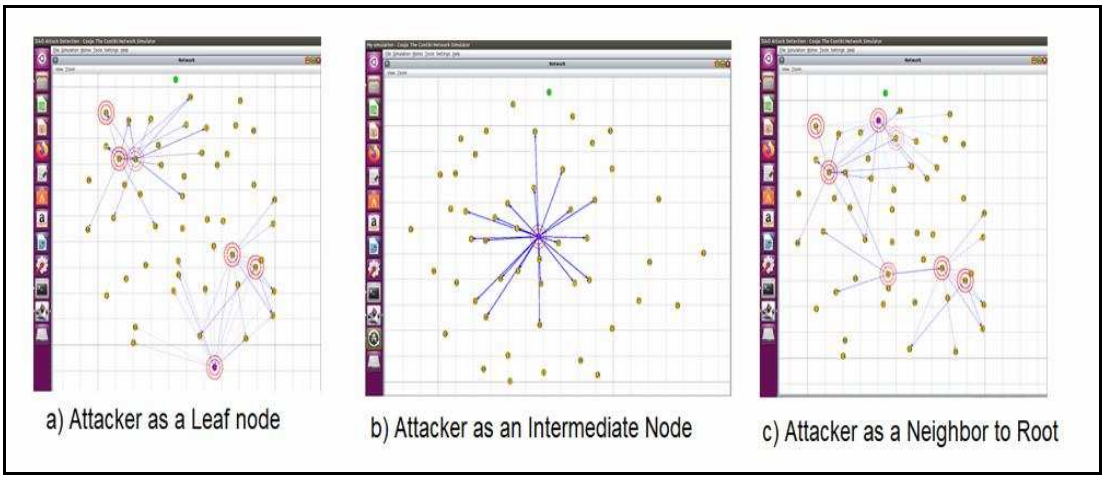


Fig. 5.5. Attacker in Three Locations

The attacker is positioned in different places as shown in Fig.5.5, and the detrimental effects caused by the malicious node in terms of control overhead, PDR, and power consumption are documented for study. In this subsection, the observed results are described.

5.7.1. Power Consumption

For each scenario, the power consumption of each node is computed. In both the attacker and normal settings, the average power usage is taken into account. The power utilization is calculated using the same method for power consumption as in Chapter 3 (3.8.4). The average energy of several simulations with and without attacker nodes is calculated, and the results are displayed in Fig.5.6.

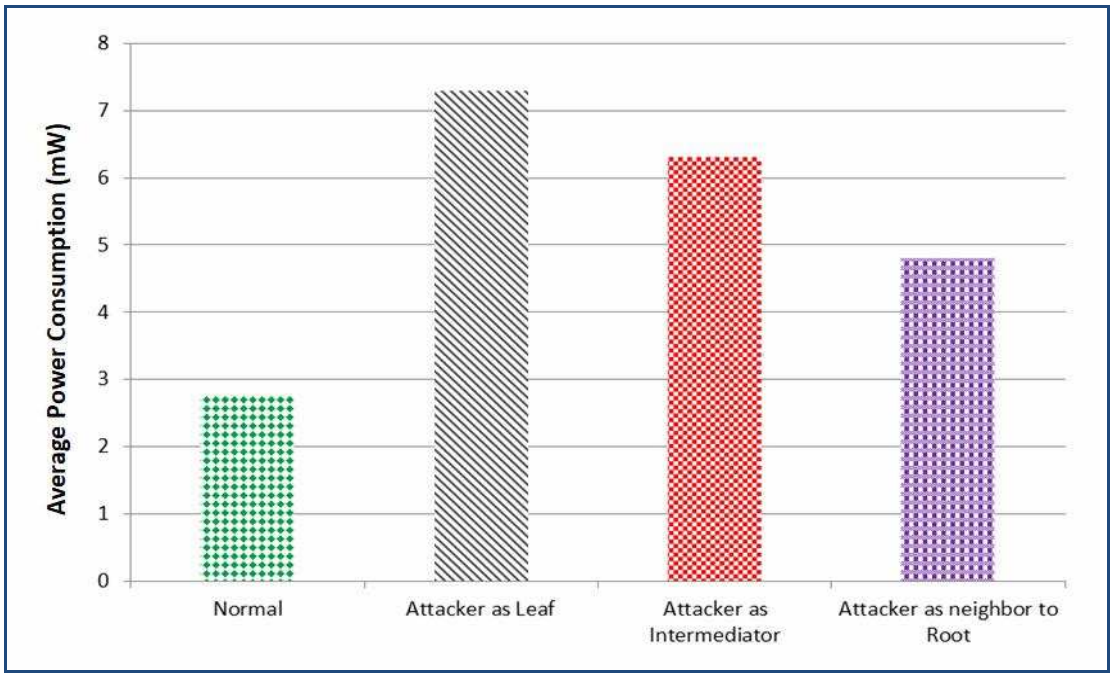


Fig. 5.6. Power Utilization in Different Scenarios

The power usage of the attacker simulations is greater than the normal scenario, as shown in Fig.5.6. The 'Attacker as a Leaf Node' scenario spent more energy than the other attacker simulations. This is due to the fact that DAO messages issued from the leaf node are sent to all intermediate nodes and then to the root node. In comparison to other attacker simulations, the volume of DAO messages transmitted to the root node is lesser when the attacker is situated as a neighbor to the root node. As a result, the power usage in this simulation is reduced. When a DAO attacker is present, the nodes' lifetime is shorter than in a normal scenario.

5.7.2. Control Traffic

The traffic created by the quantity of ICMPv6 control messages in the network such as DIO, DAO, DAO-ACK, and DIS is known as Control Traffic. Eq. 5.3. shows the formula for computing the control message traffic in the network.

$$\text{Control Traffic} = \sum \text{ICMPv6 Control Messages} \quad (5.3)$$

The number of control packets obtained in the Normal and the different attacker scenarios are given in Fig. 5.7.

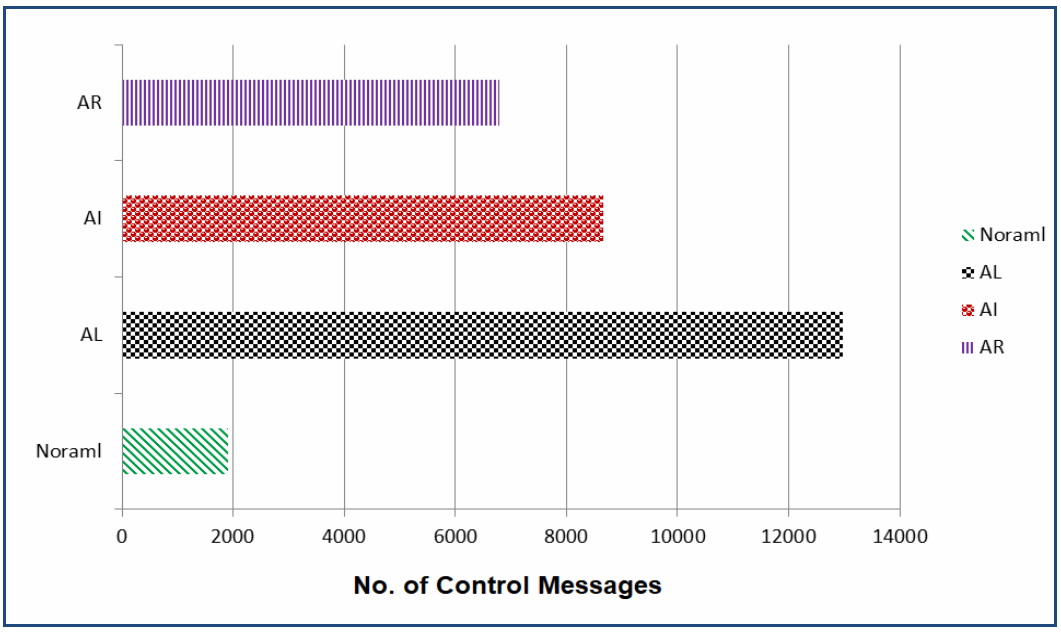


Fig. 5.7. Control Traffic in Different Simulations

In this Fig.5.7, the 'Attacker as a Leaf Node', 'Attacker as an Intermediate Node', and 'Attacker as the Neighbor to the Root Node' are represented by AL, AI, and AR, respectively. As shown in Fig.5.7, the control traffic in the attacker cases is higher than in the ordinary simulation. When the attacker is put into the leaf node, the volume of control traffic is greater than in the other attacker scenarios. It uses additional network resources, which are limited to the IoT devices. As a result, the DAO attacker node degrades the performance of the RPL-based IoT networks.

5.7.3. Packet Delivery Ratio (PDR)

It represents the ratio of the received packets at the root node to the total packets sent to the root node. Eq.5.4 is the formula for computing the PDR. Let the number of packets received at the root node be represented as PR and the number of packets sent to the root node as PS.

$$PDR = (\sum PR) / (\sum PS) \times 100 \quad (5.4)$$

In both ordinary and attacker simulations, the root node's packet delivery ratio is computed. Fig. 5.8 depicts the results.

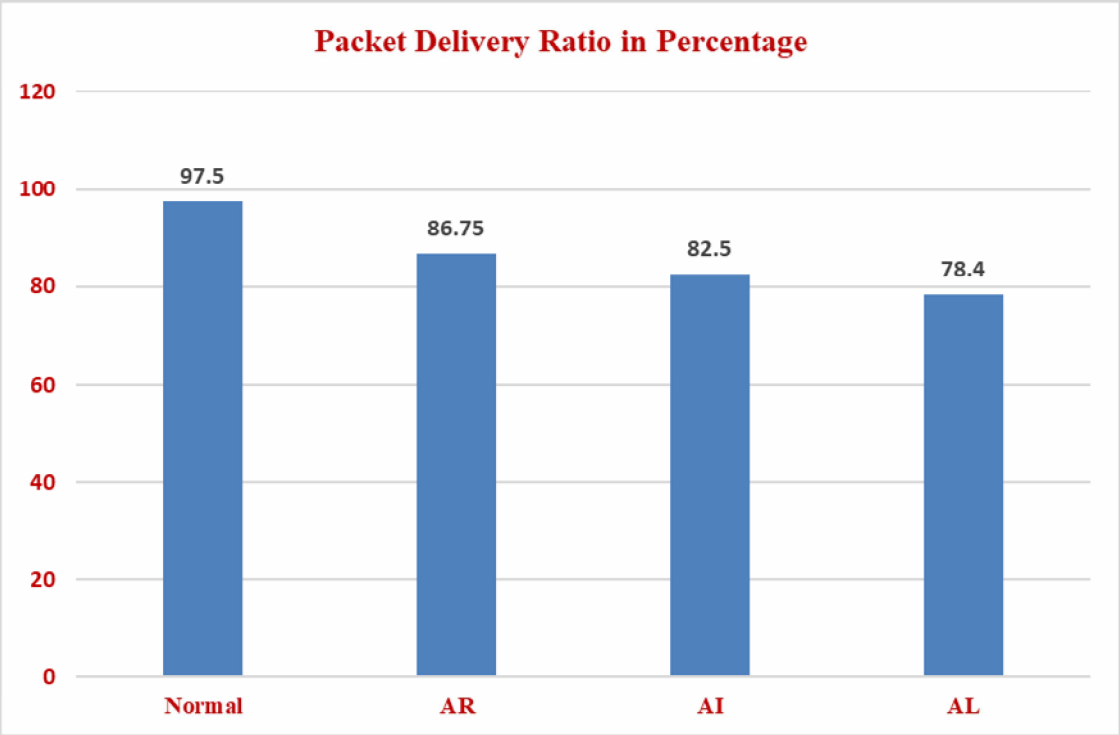


Fig. 5.8. PDR in Different Simulations

As shown in Fig.5.8, the PDR in the normal simulation is 97.5 percent, but it is reduced in all attacker situations and is below 80 percent when the attacker is placed as a leaf node. This demonstrates the RPL network's poor performance in a DAO attacker setting. The packet delivery ratio is lower in the 'Attacker as a Leaf Node' scenario than in other instances because the leaf node must transmit the DAO message to the root node through several parents.

5.7.4. Implementing DADTec

The DADTec mechanism is implemented in the Cooja Simulator for identifying DAO attacks by taking into account the resource-constrained characteristics of the

RPL protocol. Following the implementation, whenever a DAO attack occurs in the RPL network, DADTec identifies it and, once it exceeds a certain threshold, the attacker is removed from the network. The detection system has a lower overhead. Almost all security threats posed by attackers are detected by DADTec. Table 5.4 shows the detection results after using the DADTec approach.

Table 5.4. Performance of DADTec

	Detected As Attack	**Detected As Normal**	**Total**
Attack	**917 (TP)**	**33 (FN)**	**950**
Normal	**27 (FP)**	**8190 (TN)**	**8217**
Total	**944**	**8233**	**9167**

There are 9167 events as shown in Table 5.4. There are 917 security attacks and 8190 normal occurrences that are accurately recognized among them. Thirty-three attacks are undetected, while twenty-seven regular events are misidentified as attacks. Eq. 5.5 is used to calculate the detection accuracy.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.5}$$

Using the formula with the values in Table 4, the detection accuracy is 99.34 percent. The DADTec approach's detection accuracy is compared to an existing technique presented by [Ahm, 20]. When compared to the existing approach, the DADTec technique worked admirably. The average detection accuracy of the two approaches is considered, and the outcome of the comparison is displayed in Fig. 5.9.

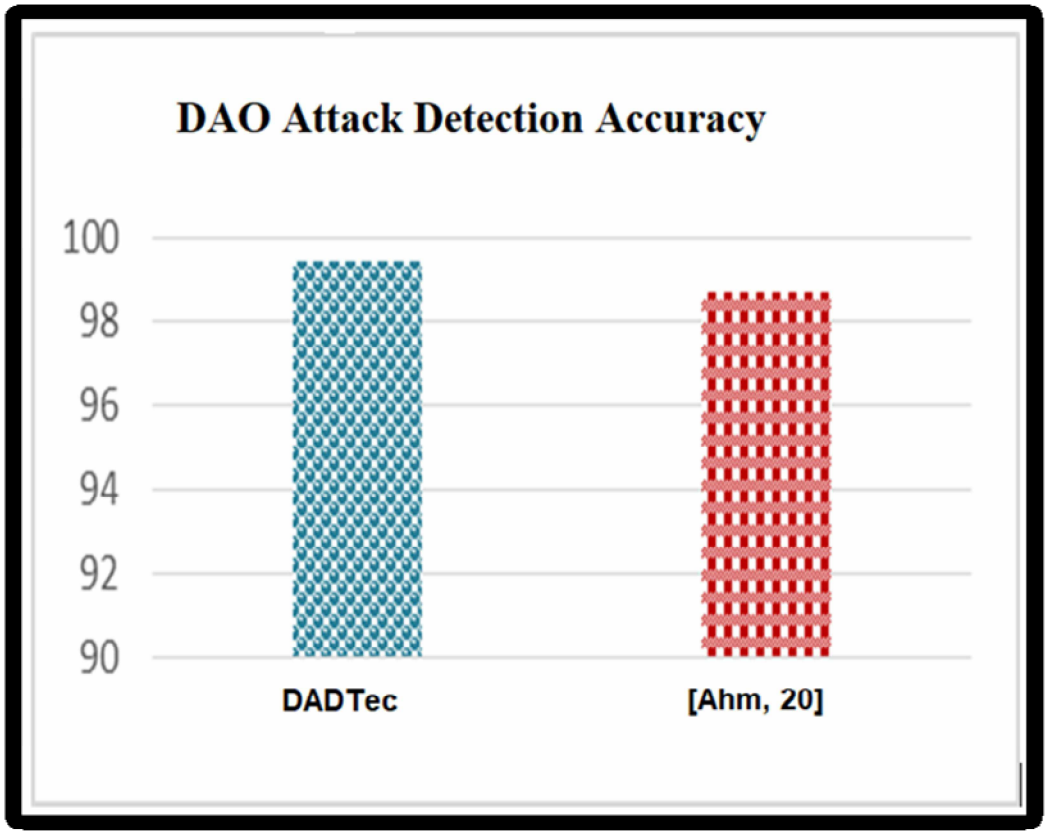


Fig. 5.9. Comparison of DADTec with Existing Work

As shown in Fig.5.9, the DADTec detects 99.34% of DAO attacks, whereas [Ahm, 20] detects 98.7% attacks only. As a result, the recommended technique of this Chapter has a greater detection rate than the present method. This might be owing to the suggested technique's low threshold value, which allows for a lesser amount of DAO messages from the child nodes to the parents, as well as the attacker node's different positioning. The proposed approach identifies DAO attacks and minimizes the negative consequences of the DAO attacker node's actions. The DADTec technique proposed in this Chapter safeguards the RPL-based IoT networks and the IoT devices against the DAO attacks.

5.8. Chapter Summary

The RPL protocol is susceptible to several security threats. The DAO attack is one of the attacks based on the DAO control message. The flooding of the DAO messages consumes the network resources. In this experiment, the attacker node is placed in different locations, and the detrimental effects of the DAO attacks are analysed in terms of power consumption, control overhead, and packet delivery ratio. In this Chapter, a technique called DADTec is proposed to detect and to minimize the impacts of such attacks. The simulation results show that the DADTec detects almost all the DAO attacks with a lower false alarm rate and reduces the negative impacts caused by these attacks.

In the next Chapter, the log files of normal scenario packets and the RPL resource attacks such as Version Number attack, DIS flooding attack, and DAO attack packets are captured, and an AdaBoost Ensemble-based Intrusion Detection System is developed in order to enhance the detection accuracy of the system. This system functions as an intelligent system, which provides an extra layer of security to IoT networks.

Chapter – VI

ANIT-Ada: An Intelligent AdaBoost Architecture for Detecting RPL Resource Attacks in IoT

Chapter - VI

ANIT-Ada: An Intelligent AdaBoost Architecture for Detecting RPL Resource Attacks in IoT

6.1. Background

The Internet Protocol version 6 (IPv6) inherited several features from its predecessor, the IPv4 protocol. So, it has the associated vulnerabilities of IPv4 and the specific security challenges of IPv6. These security threats have to be addressed to improve the security scheme. In RPL, the routing topology is constructed by the control messages of the Internet Control Message Protocol version 6 (ICMPv6). The ICMPv6 messages are grouped as error messages and informational messages. The ICMPv6 protocol is entirely responsible for communication between IPv6 nodes. It is also responsible for router and node configuration. The error messages have a preceding '0' in the high-order bit of the 'Type' field, and the informational message contains a preceding '1' in the ICMPv6 protocol. ICMPv6 is the backbone of IPv6 and RPL as it has the building blocks such as DIO, DAO, DIS, and DAO-Ack informational messages for constructing the DODAG for routing. ICMPv6 is an insecure protocol and it is prone to several security threats and attacks [Oma, 16]. Using the unicast and multicast mechanisms, several ICMPv6-based RPL resource attacks are created by the adversary, and these RPL resource attacks cause significant damage to the networks. It also leads to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks in RPL network. Version Number attacks, DIS flooding attacks, and DAO attacks are some of the RPL resource attacks that lead to harmful effects in the IoT environment.

In this Chapter, the IoT network communication traces are collected from the normal simulation environment and RPL resource attack scenarios such as Version Number attack, DIS flooding attack, and DAO attack. By using the collected data, an AdaBoost ensemble model-based intrusion detection system was developed. For that, the preprocessing and feature engineering processes are carried out on the collected data. Finally, an ensemble of AdaBoost machine learning algorithms is applied to the collected dataset to build an ensemble model called Ada-IDS to detect the RPL resource attacks. This ensemble model is deployed in the Border Router (6BR) of the ANIT-Ada architecture, which detects the attacks effectively and protects the IoT network as an additional layer of security mechanism.

6.2. Related Works

Adnan Hasan Bdair et al. [Adn, 20] critically assessed the recent ICMPv6-based Intrusion Detection systems, with a specific focus on DoS and distributed DoS threats. The researchers looked at three forms of ICMPv6-based attacks: ICMPv6 flood, ICMPv6 amplification, and ICMPv6 protocol exploitation. This study also discussed several types of IDSs for ICMPv6-based threats.

Arul Anitha et al. [Aru, 19] recommended an Artificial Neural Network-based IDS System (ANNIDS) for the RPL-based IoT networks. In this work, the dataset was collected from normal scenarios, Version Number Attacks, and DIS Attacks scenarios using the Cooja Simulator. The proposed method correctly classified the attacks and normal packets.

Emre Aydogan et al. [Emr, 19] used the Genetic Programming principle to implement a Centralized IDS Model for Industrial IoT. Using a Genetic Algorithm

technique with 50 populations and various default parameters, this system detected 'Hello Flood Attacks' and 'Version Number Attacks.' This study did not examine the network traces.

For detecting low-rate Denial of Service (LDoS) attacks, Dan Tang et al. [Dan, 20] suggested a multi-feature-based AdaBoost system. The network traffic was collected at predetermined intervals and the resulting samples were examined using different statistical techniques. The best feature set was chosen as a result of the correlation scores between the features and the class labels. The AdaBoost ensemble model was created using the best features. The model's performance was evaluated using the NS2 simulator and a testbed, which yielded 94.05 percent and 97.06 percent attack detection accuracy respectively. This system is not specific to IoT.

Using the Decision Tree (DT), Naive Bayes (NB), and Artificial Neural Network (ANN) algorithms, Nour Mustafa et al. [Nou, 18] built an AdaBoost ensemble Network Intrusion Detection System (NIDS). This technique identifies IoT threats at the application layer. This ensemble model was built using the UNSW-NB15 and NIMS botnet datasets. In the UNSW-NB15 dataset, the proposed model detects intrusions with 99.54 percent accuracy, and in the NIMS botnet dataset, it detects intrusions with 98.29 percent accuracy. This research considers the application layer related attacks only.

6.3. Objectives

This Chapter proposes an intelligent architecture called ANIT-Ada using the AdaBoost ensemble algorithm to detect ICMPv6-based RPL resource attacks. For that, 50 normal nodes, three attackers, namely Version Number Attacker, DIS Attacker,

and DAO Attacker, and a border router were included. The objectives of the Chapter are detailed below:

- The Version Number Attack, DIS Attack, and DAO Attack are deployed in the Contiki Cooja Simulator.
- The network traces, including the malicious as well as normal packets, are captured using the 6LowPAN analyser tool for further analysis.
- The captured packets are pre-processed for building the Ada-IDS model.
- The Ada-IDS Model is deployed in the border router of the ANIT-Ada architecture for developing a centralized intrusion detection system.
- The VeNADet, DISDet, and DADTec techniques are also implemented on the root node in order to provide better security to the IoT environment.

The AdaBoost ensemble model, Ada-IDS, is implemented in the border router of the ANIT-Ada Architecture and acts as a centralised intrusion detection system. All three techniques are installed in the root node. Hence, this architecture provides double-layer security to the IoT environment.

6.4. Developing Ada-IDS Model

The two main types of IDS are centralised and distributed IDS. The IDS is implemented on the border router or on a dedicated server in the centralised approach. It is installed on the client nodes in Distributed IDS. Because IoT nodes are resource limited, the Distributed IDS approach is ineffective for devices with low resources.

The overhead on individual nodes like communication and computation is reduced as the centralized type of IDS is followed in this proposed model. So, an Intelligent AdaBoost Ensemble based centralized IDS (Ada-IDS) model is deployed

on the border router. It monitors the incoming and outgoing data packets of the IoT network. When the Ada-IDS encounters any attacks or intrusions, it alerts the admin about the issue. Fig. 6.1 depicts the various phases of developing the Ada-IDS model.

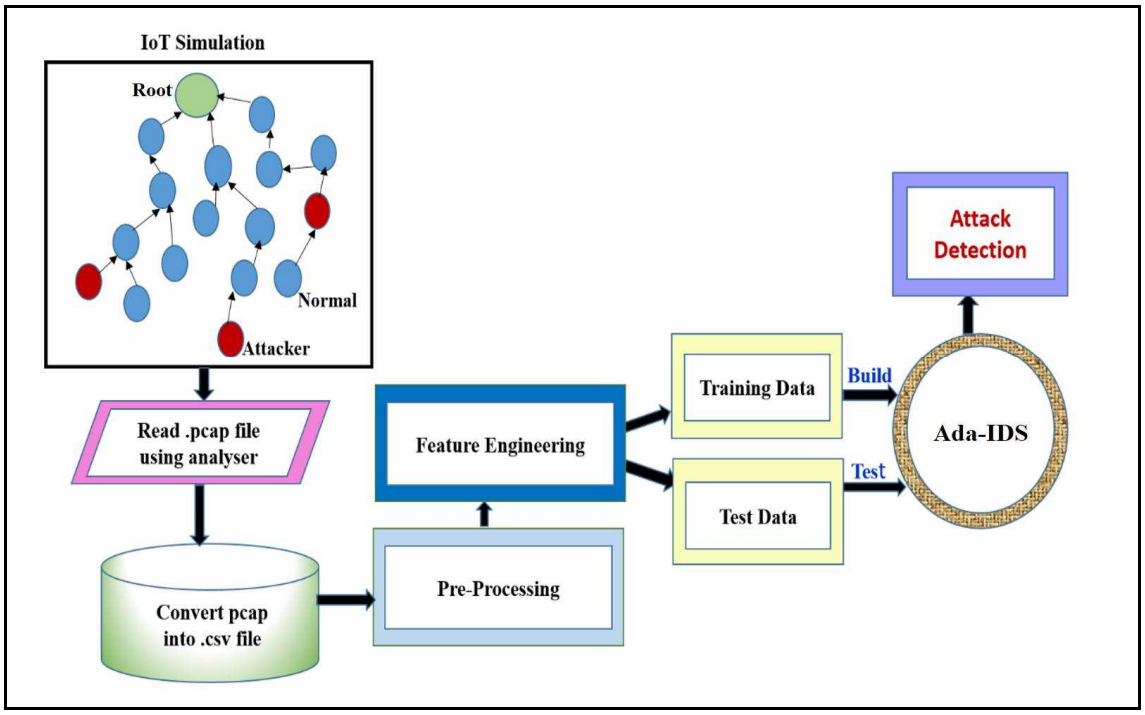


Fig. 6.1. Proposed Ada-IDS

The data collection, preprocessing, feature engineering, model building, and deployment are the five phases of developing the Ada-IDS model, as illustrated in Fig.6.1.

6.4.1. Data Collection

The Cooja Simulator is used to gather the data. The simulation includes 50 normal nodes, one root node, and attacker nodes. The DIS attack, DAO attack, Version number attack, and a simulation without an attacker are implemented, and the log files from all of these experimental settings are gathered using the 6LoWPAN Analyzer tool. In each scenario, the simulation runs for 30 minutes. WireShark software is used to analyze the collected packets, and the packet capture (.pcap) files

are converted to '.csv' files. The .csv files are named as ‘icmpv6.csv’. The file 'icmpv6.csv' is utilized in the Ada-IDS model construction. Normal packets, DIS Attacks, Version Number Attacks, and DAO Attacks are all included in the dataset. Table 6.1 shows the normal and attack occurrences.

Table 6.1. Attack and Normal Packets

S.No.	IDS Type	No. of Packets
1.	Normal	125184
2.	DIS Attacks	325
3.	DAO Attacks	1193
4.	Version Number Attacks	982
	Total	**127684**

The dataset contains 127684 samples, comprising 125184 normal, 325 DIS Attacks, 1193 DAO Attacks, and 982 Version Attacks, as shown in Table 6.1. The dataset has nine features. Table 6.2 has a description of the dataset.

Table 6.2. Icmpv6 Dataset Description

S.No.	Attribute Name	Data Type	Description
1.	No.	Integer	Packet No.
2.	Source	String	Source IPv6 Address of a Packet
3.	Time	Float	Time is represented in milliseconds
4.	Destination	String	Destination IPv6 Address of a Packet
5.	Protocol	String	Protocol used for Communication
6.	Length	Integer	Length of a packet in no.of Bytes
7.	Info	String	Description about the protocol
8.	Class	String	The packet is attack or normal
9.	Type	String	The type of attacks (Version, DIS, DAO)

The features of the Icmpv6 dataset are described in Table 6.2. Figure 6.2 shows a snapshot of example records obtained with Python code.

	No	Source	Time	Destination	Protocol	Length	Info	Class	Type
0	1	fe80::212:742f:2f:2f2f	0	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
1	2	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
2	3	fe80::212:741d:1d:1d1d	0.114	fe80::212:7421:21:2121	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
3	4	fe80::212:742f:2f:2f2f	0.114	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
4	5	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
...	...	...	...	...	...	...	...	...	...
127679	127680	fe80::212:740b:b:b0b	431.709	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127680	127681	fe80::212:741f:1f:1f1f	431.71	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal
127681	127682	fe80::212:7432:32:3232	431.71	fe80::212:7424:24:2424	ICMPv6	76	RPL Control (Destination Advertisement Object)	Attack	Version Attack
127682	127683	fe80::212:740b:b:b0b	431.711	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127683	127684	fe80::212:741f:1f:1f1f	431.712	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal

127684 rows × 9 columns

Fig. 6.2. Screenshot with Sample Data

The Class and Type fields, as shown in Fig.6.2, indicate whether a packet is an attack or a normal packet. The Type field also includes information about the attack, such as DIS, DAO, or Version Number Attacks. A brief description of these attacks is given below.

- **Version Number Attacks:** An unsigned 8-bit number in the DIO message is the Version Number (VN). The parent nodes use the DIO control message to multicast VN. The global repair process is triggered if there is a discrepancy in the DODAG, and the root node updates the Version Number. A DIO control message is sent from the root node to propagate this updated information. Without the knowledge of the root node, a Version Number Attacker modifies the version number on a regular basis and distributes the revised version

number to its neighbors via DIO messages. When neighboring nodes get this DIO message, they join the global repair procedure, and the DODAG is rebuilt repeatedly. This fraudulent behavior interferes with the legitimate nodes' usual functions and drains the IoT nodes' limited resources. In the long run, the malicious behavior of the Version Number Attacker increases network control traffic and exposes the network to DoS attacks [Ari, 18].

- **DIS flooding Attacks:** The DIS messages' header information is altered to perform this attack. In order to join the DODAG, the DIS messages are multicast to probe its neighbors. Neighbor nodes respond with DIO messages to the sender after receiving the DIS message. The Trickle Timer determines how long it takes to send DIO messages. Even though it has previously received DIO messages, a DIS flooding attacker continues to multicast DIS messages to its neighbors. This massive influx of DIS messages on the network impairs network performance and results in a DoS attack [Con, 19].
- **DAO Attacks:** Manipulating the DAO control message results in a DAO attack. To preserve the reverse root, a child node must reply with a DAO message when it gets a DIO message from its parent. The DAO message delivered by the child node passes via several ancestors before arriving at the root node. A DAO attacker sends the DAO message to its parent list on a regular basis. All such unnecessary network messages must be routed to the root node. It uses up more network resources and restricts authentic nodes from carrying out their normal functions. Consequently, the network would be in inconsistent condition, making it prone to DoS attacks [Isa, 20] [Bar, 19].

By manipulating the ICMPv6 control packets, these three RPL-based resource attacks are initiated. These attacks deplete the IoT network's resources and degrade its performance. Finally, all three attacks result in a DoS attack, which triggers greater network damage.

6.4.2. Pre-Processing

In order to be efficient in generating the ensemble model, the dataset obtained from the simulation environment must be pre-processed. In the Source and Destination fields, there are 394 missing values. The missing values cannot be replaced by mean, median, or mode values because these two variables represent the nodes' IPv6 addresses. The missing values in the Source and Destination Address fields are replaced by new values.

6.4.3. Feature Engineering

To make categorical features meaningful for ML algorithms, one hot encoding and label encoding are applied. For the 'Time' feature, the frequency encoding approach is used. The 'Class' feature distinguishes between normal and attack data samples. The Type feature categories the type of attacks as DIS Attack, DAO Attack, or Version Number Attack. It also identifies the normal packets. Label encoding is performed on these 'Type' and 'Class' fields. The attribute 'No.' denotes a packet number that has no influence on forecasting the target and is thus removed from the dataset. The dummy values 'a' and 'b' are used to substitute null values in the 'Source' and 'Destination' fields, respectively. The dataset appears like Fig.6.3 once the pre-processing and feature engineering operations are completed.

	Source	Time	Destination	Protocol	Length	Info	Class	Type
0	58	3	29	0	76	4	0	0
1	21	4	15	0	76	4	0	0
2	40	4	25	0	76	4	0	0
3	58	4	29	0	76	4	0	0
4	21	4	15	0	76	4	0	0
...	...	...	...	...	...	...	...	...
127679	22	1	2	0	76	4	0	0
127680	42	2	37	0	97	2	0	0
127681	61	2	28	0	76	4	0	0
127682	22	1	2	0	76	4	0	0
127683	42	1	37	0	97	2	0	0

127684 rows × 8 columns

Fig. 6.3. Dataset after Pre-processing

All of the dataset's categorical values are transformed to numerical values, as illustrated in Fig.6.3. The dataset is now suitable for developing the proposed model.

6.4.4. Model Building

This experiment uses the pre-processed 'Icmpv6.csv' dataset with eight features. There are 127684 data samples in the dataset. The 20% of data samples are considered as the test set, which comprises 25537 data packets, and 80% of data samples are partitioned into a training set, which has 102147 instances.

AdaBoost Ensemble Model: In this experiment, an Ada-Boost (Adaptive Boosting) model is designed to identify these three forms of attacks, such as Version Number Attack, DIS Attack, and DAO Attack. AdaBoost model was proposed in 1996 by Yoav Freund and Robert Schapire. By merging multiple classifiers, the classifier accuracy is improved [Avi, 18]. The AdaBoost classifier combines many

weak classifiers to produce a strong classifier with maximum accuracy. Adaboost's core principle is to train the data sample and update the classifier weights in each iteration. This principle detects uncommon occurrences of the dataset properly [Abd, 20].

To fine-tune the classifier, interactive training on a range of weighted training samples should be implemented. In each cycle, it seeks to minimise training error in order to produce the best possible fit for the training instances. The following are the procedures for developing the ensemble model:

1. Adaboost starts by randomly selecting a training subset.
2. It is trained repeatedly by choosing the training set based on the previous training accuracy.
3. More weightage is given to the incorrectly classified samples in order to classify them correctly in the next iteration.
4. In addition, the trained classifier is given greater weight in each iteration based on its classification accuracy.
5. More credit is awarded to classifiers that are more exact.
6. The training data is iterated in this step until it fits exactly or the predefined maximum number of estimators is achieved.

There are three basic parameters in the AdaBoost classifier: base_estimator, n_estimator, and learning_rate. The following are the parameters that have been applied in this study:

- **base_estimator:** To train the model, a weak learner is utilised. The default Decision Tree Classifier is chosen to build the ensemble model in this study.
- **n_estimator:** It defines the number of weak learners that are used to iteratively train the model. There are ten estimators in this model. The results are assessed. Then increase by ten until the estimators reach 100.

- **Learning-rate:** It indicates the learning rate of the weak learner. The default learning rate ('1') is adopted in this ensemble model.

Using the procedure for AdaBoost ensemble algorithm and the AdaBoost parameters, the Ada-IDS ensemble model is developed.

6.4.5. Deployment

The proposed Ada-IDS is placed in the Gateway device. The Ada-IDS model detects the resource attacks like DIS attacks, DAO attacks and Version Number attacks in RPL-based IoT networks. The full design of the ANIT-Ada architecture, including the Ada-IDS model, is described in detail in the next section.

6.5. ANIT-Ada Architecture for Attack Detection

The Ada-IDS ensemble model, as well as techniques like VeNADet, DISDet, and DADTec, are incorporated into the ANIT-Ada architecture. These techniques detect the Version Number Attack, DIS Attack, and DAO Attack, respectively. Fig. 6.4 illustrates the ANIT-Ada architecture and its components clearly.

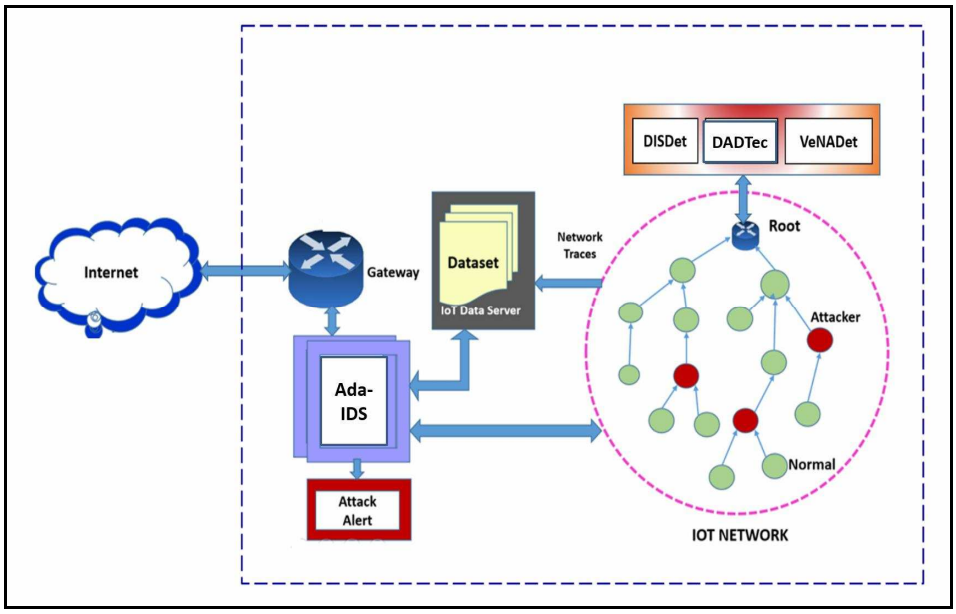


Fig. 6.4. ANIT-Ada Architecture

6.5.1. Gateway

The Border Router (6BR) is used as the gateway. Here, the IoT network communicates with the outside world (Internet) using this gateway device. The external devices also connect to the local IoT network using this device. It is also known as an edge device.

6.5.2. IoT Data Server

The communication traces from the IoT network are collected and stored on the IoT Data Server. The network log files are captured and updated periodically. Whenever there is a new pattern for the Version Attack, DIS Attack, or DAO Attack, the corresponding events are stored in the IoT Data Server. The new normal events are also stored similarly. The new attack patterns from the external devices are also captured via the gateway device.

6.5.3. Ada-IDS

The log files collected from internal and external sources are retrieved from the IoT Data Server to develop the Ada-IDS Ensemble model for attack detection. The proposed Ada-IDS is installed on the Border Router. It provides an additional layer of security to the IoT network. Incoming and outgoing data and communications are monitored by the Ada-IDS. As the proposed ensemble model is deployed in the Border Router, the intrusions and malicious activities caused by the Version Number Attacks, DIS Attacks, and DAO Attacks initiated from the IoT network and the outsider network are detected by Ada-IDS. The alarm is raised whenever there is an intrusion caused by these events.

6.5.4. Attack Detection Techniques

The VeNADet, DISDet, and DADTec techniques are implemented at the root node of the IoT network. These techniques detect Version Number Attacks, DIS Attacks, and DAO Attacks, respectively. Whenever there is an attack or intrusion in the IoT network, the corresponding technique is activated to detect the attacks. The Ada-IDS in the gateway also detected the attack. These techniques detect the three attacks and provide a supplementary layer of security to the network.

6.5.5. IoT Network

The IoT network comprises a root node with a high configuration setup and a number of client nodes. Whenever the root node is configured, the network topology (DODAG) is constructed. There may or may not be any attackers on the network. If there is any attack on the network, the corresponding technique is activated in order to detect the attack. The pseudo code for the ANIT-Ada architecture is given in Fig.6.5.

Algorithm 6.1. Pseudo Code for ANIT-Ada Architecture

Input: Network Traffic

Output: Attack- DAO, DIS, Version or Normal

1. implement Normal and Attack Scenarios in Cooja Simulator
2. collect the packets from 6LowPAN Analyser tool
3. analyse the packets using WireShark tool
4. convert the packets into .csv format
5. extract the features from the .csv file
6. pre-process the features
7. perform feature encoding
8. select the relevant features
9. split the Dataset into two parts:
   - 80% Training data

```
    - 20% Testing data
10. learning_rate=1, base_estimator=DecisionTree Classifier
11. for i=10 to 100 do:  // Build AdaBoost Ensemble Model
12.     n_estimator=i
13.     build AdaBoost(learning_rate, base_estimator,n_estimator)
14.     calculate training_time
15.     test AdaBoost(learning_rate, base_estimator,n_estimator)
16.     calculate testing_time
17.     evaluate confusion_matrix, accuracy
18.     evaluate precision, recall, f-Score
19.     increment i by 10
20. end for
21. implement AdaBoost Model in the Gateway
22. install VeNADet, DISDet and DADTec in the root node
23. return output
```

Fig. 6.5. Pseudo Code for ANIT-Ada Architecture

This intelligent architecture ANIT-Ada provides two levels of security using the three techniques and the Ada-IDS ensemble model. It acts as a countermeasure for the Version Number Attacks, DIS flooding attacks and DAO attacks.

6.6. Results and Discussions

6.6.1. Training and Testing

The AdaBoost ensemble model's findings are discussed in this section. After the preprocessing and feature engineering steps are completed, the dataset is partitioned into two sets: training and testing. The training set includes 80% of the total instances, whereas the testing set contains 20% of the data samples. Table 6.3 shows the number of data packets in each category.

Table 6.3. Training and Testing Samples

Type of Instance	Training Samples (80%)	Testing Samples (20%)	Total
Normal	100169	25015	125184
DAO Attack	79	246	325
DIS Attack	1115	78	1193
Version Attack	784	198	982
Total Samples	**102147**	**25537**	**127684**

The AdaBoost ensemble model is built using the training samples. The Decision Tree Classifier is chosen as the weak classifier to repeatedly fine tune the model. The default value for the learning rate parameter is 1. With ten base estimators, the training time, testing time, and detection accuracy are examined. The base estimator is increased by 10 until it reaches 100 to see if there is any change in accuracy as the number of estimators increases. Surprisingly, the AdaBoost classifier's accuracy is 99.6%, and it is unaffected by the number of estimators employed in its development. Table 6.4 lists the AdaBoost ensemble model's parameters and accuracy.

Table 6.4. AdaBoost Parameters and Accuracy

n_Estimator	Learning Rate	Training Time (in Seconds)	Testing Time (in Sec.)	Accuracy
10	1	0.62	0.069	0.996
20	1	1.77	0.092	0.996
30	1	1.662	0.163	0.996
40	1	2.406	0.355	0.996
50	1	2.937	0.272	0.996
60	1	4.881	0.363	0.996
70	1	5.21	0.357	0.996
80	1	6.627	0.428	0.996
90	1	5.561	0.786	0.996
100	1	6.923	0.872	0.996

The learning rate is the same throughout all experiments, as given in Table 6.4. For each experiment, the number of decision trees used to generate the AdaBoost ensemble model increases from 10 to 100 by incrementing 10 in each iteration. In all experiments, the accuracy gained is the same. The amount of time spent on training and testing varies depending on the number of base estimators employed in each experiment. The training and testing time for this experiment are depicted in Fig. 6.6.

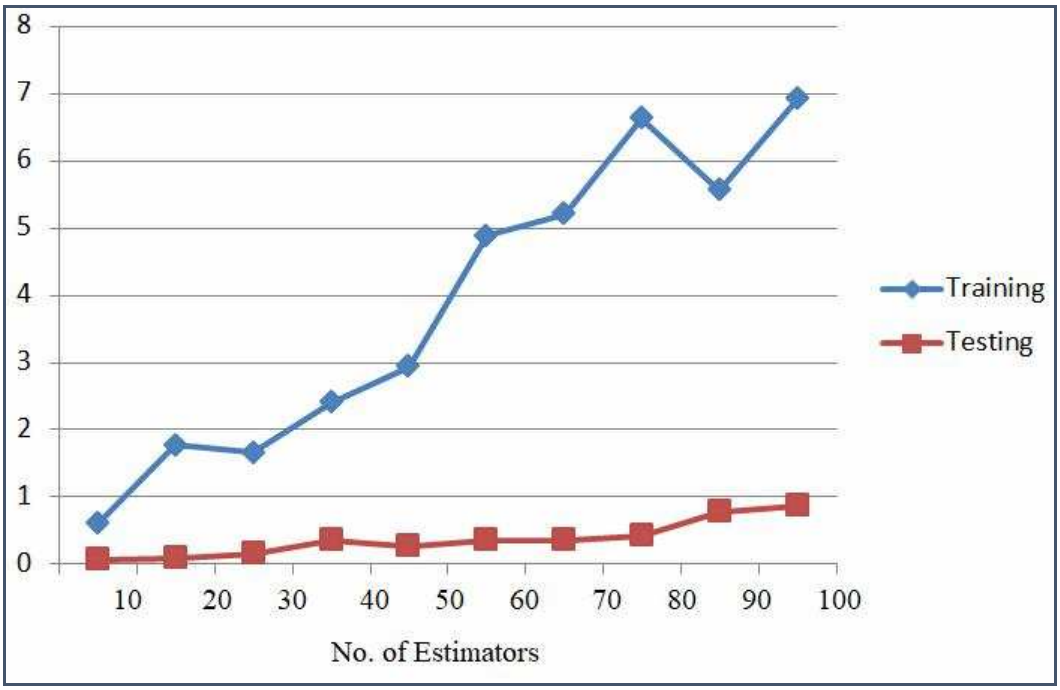


Fig. 6.6. Training and Testing Time for AdaBoost Model

As shown in Fig.6.6, the training period for developing the model is longer than the testing time. It is due to the fact that there are more samples in the training dataset (80%). As the number of Decison Tree Classifiers grows, so does the training time. As a result, the number of samples, the number of estimators, and the training duration all have a positive correlation. When the number of estimators changes in each experiment, the testing duration varies as well. The testing duration grows as more Decision Tree Classifiers are added.

6.6.2. Evaluation Metrics

The dataset contains three types of attacks. For each experiment, confusion matrices are produced, which display the actual and predicted class labels for each sample. Metrics such as accuracy, precision, recall, and F-Score are also derived from the confusion matrix to evaluate the performance of the Ada-IDS model [Moh, 17].

- **True Positive (TP):** The accurate categorization of an attack packet as an attack is denoted by TP.
- **True Negative (TN):** The correct categorization of normal packets as normal is defined by TN.
- **False Negative (FN):** The incorrect categorization of an attack packet as normal is represented as FN. When this value rises, it has an impact on the availability and confidentiality considerations.
- **False Positive (FP):** FP denotes an inaccurate classifying, in which a normal packet is labelled as an attack.
- **Accuracy:** It is the ratio of the total cases to the sum of correctly categorized samples as normal and attack. Eq. 6.1 contains the formula for calculating Accuracy.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \qquad (6.1)$$

- **Recall (Sensitivity):** The number of accurate positive predictions generated out of all accurate classifications is measured by recall. The formula for computing sensitivity or recall is given in Eq. 6.2.

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \qquad (6.2)$$

- **Precision:** It is calculated by dividing the total number of samples that are accurately categorized as attacks by the total number of samples detected as attacks. By using Eq. 6.3, the precision is computed.

$$\Pr ecision = \frac{TP}{TP + FP} \tag{6.3}$$

- **F-Score:** The F-Score combines the characteristics of accuracy and recall into a single metric. Eq.6.4 is the formula for calculating the F-Score.

$$F - Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{6.4}$$

Table 6.5 shows the confusion matrix derived using these evaluation metrics. It is extremely similar in all experiments.

Table 6.5. Confusion Matrix based on Evaluation Metrics

	Classified As			
	Normal	**DAO Attack**	**DIS Attack**	**Version Attack**
Normal	**25015**	**0**	**0**	**0**
DAO Attack	**0**	**214**	**32**	**0**
DIS Attack	**0**	**21**	**57**	**0**
Version Attack	**0**	**0**	**38**	**160**

The accurately categorized samples in the testing set are marked in blue in Table 6.5, whereas the misclassified samples are denoted in red. As indicated in the table, all normal occurrences are correctly classified. Other categories have extremely few misclassifications. The accuracy, precision, recall, and f1-score values are computed using the confusion matrix and the equations Eq.6.1 through Eq.6.4. The obtained results are displayed in Table 6.6.

Table 6.6. Results Obtained from the Confusion Matrix

n_Estimator	Accuracy	Precision	Recall	F1-Score
10	0.996	0.99	1.00	1.00
20	0.996	0.99	1.00	1.00
30	0.996	0.99	1.00	1.00
40	0.996	0.99	1.00	1.00
50	0.996	0.99	1.00	1.00
60	0.996	0.99	1.00	1.00
70	0.996	0.99	1.00	1.00
80	0.996	0.99	1.00	1.00
90	0.996	0.99	1.00	1.00
100	0.996	0.99	1.00	1.00

The Ada-IDS model with Decision Tree Classifier performs better in terms of accuracy, precision, recall, and f-score, as shown in Table 6.6. For all observations, the resultant confusion matrix is the same, resulting in the same accuracy, precision, recall, and f1-score values. The proposed Ada-IDS detects **99.6%** attacks accurately. Since it doesn't have any false alarm-rate, it is suitable for anomaly detection.

The proposed Ada-IDS ensemble model is deployed in the gateway device of the ANIT-Ada architecture to provide an additional layer of security to the system. The VeNADet, DISDet, and DADTec techniques are also installed in the root node, which provides an inner layer of security to the nodes that are connected to the IoT network. Thus, the ANIT-Ada architecture safeguards the nodes and IoT networks from resource attacks such as Version Number Attacks, DIS Attacks, and DAO Attacks.

6.7. Chapter Summary

Security attacks are inevitable on the RPL-based Internet of Things as they have limited resources compared to other networks. In this Chapter, an AdaBoost ensemble-based ANIT-Ada architecture is developed using the Ada-IDS model and the countermeasures such as VeNADet, DISDet, and DADTec to overcome the RPL resource attacks like Version Number Attack, DIS flooding Attack, and DAO Attack, respectively.

These three techniques are implemented in the root node of the IoT network to provide the inner layer security to overcome these attacks. The Ada-IDS ensemble model is deployed in the Border Router of the ANIT-Ada architecture. According to the experiments, the Ada-IDS model of the ANIT-Ada architecture detected these three types of attacks with no false alarm rate. Hence, it is suitable for anomaly-based IDSs that provide an additional layer of security to the IoT network.

This ANIT-Ada architecture is suitable for all IoT domains and acts as a shield to protect the nodes from flooding of DIS messages, unnecessary version updates, and bulk sending of the DAO message in the RPL-based IoT network. The security requirements like availability and reliability are also ensured by implementing this ANIT-Ada architecture.

The next chapter briefly summarizes the overall research work and concludes the thesis with future directions in this area.

Chapter – VII

Conclusion

Chapter – VII

Conclusion

7.1. Overview

This research has been performed to ensure the security requirements and to safeguard the smart devices in the IoT networks from RPL resource attacks. As the number of cyber-physical systems on the global network continues to increase, the need for security tools and techniques also increases. This thesis introduced an intelligent architecture (ANIT-Ada) to act as a centralised intrusion detection system that monitors the network activities of the incoming and outgoing packets in an IoT network. The AdaBoost ensemble system called Ada-IDS in this architecture detects three ICMPv6-based RPL resource attacks, such as Version Number Attack, DIS Attack, and DAO Attack. Three techniques, such as VeNADet, DISDet, and DADTec, are also proposed to safeguard the IoT nodes from these three RPL resource attacks.

This Chapter consolidates the major findings in connection to the research objectives, as well as discusses their importance and contributions. The limitations of the research work and recommendations for further research are also deliberated at the end.

7.2. Importance of the Proposed Techniques

This section briefly narrates the three techniques proposed in this thesis for detecting resource attacks and elucidates the ANIT-Ada architecture, which combines the three techniques and an AdaBoost ensemble IDS, which provides an additional layer of security to the IoT network. The three mechanisms to secure the IoT nodes and the functionalities of the proposed ANIT-Ada architecture are explained below.

7.2.1. VeNADet Technique

The VeNADet technique is recommended to identify the Version Number Attack (VNA) in the RPL-based Internet of Things. The destructive impacts of the Version Number Attacks in terms of power consumption, control overhead, and PDR are analyzed. The unnecessary updates to the Version Number (VN) are detected and prevented by the VeNADet technique.

This mechanism allows only the root node to update the VN. When a node receives a DIO message with a modified VN from its neighbor, first it has to check and validate the VN of the root node. If both Version Numbers are the same, then it accepts the VN. The receiver node also verifies the VN with other neighbor nodes. If 80% of neighboring nodes have the updated VN, then only the current node will update its VN. If the conditions are not satisfied, then the node that has an updated VN is declared as a malicious node. The malicious node is also disconnected from the DODAG.

The VeNADet technique identifies the VNAs initiated by the malicious nodes and safeguards the IoT network by disconnecting them from the IoT network. VeNADet maintains 80% trust among the neighbors and all the nodes in the DODAG. It saves the network resources of the low-power devices and increases the lifetime of the nodes.

7.2.2. DISDet Technique

The DISDet technique is proposed to detect the DIS flooding attacks in an RPL-based IoT environment. The DIS attack and non-attack scenarios were analyzed by using metrics such as energy consumption, control traffic, and packet delivery

ratio. The flooding of DIS messages into the network is circumvented using this DISDet technique.

The DIS messages are used to discover the neighbor nodes that are already in the DODAG in order to get a DIO message and join the network. Though the attacker already received the DIO message from its neighbor, it dumps the neighbor nodes with DIS messages. This malicious behavior of the DIS attacker increases traffic in the network, excessive power usage, and reduces the PDR. Hence, the network performance is also degraded.

In RPL, before joining the DODAG, a node has to wait for a DIS_DELAY (5 seconds) and sends its first DIS message. At the same time, the DISDet technique constructs an 'n x m' matrix M, where 'n' is the number of new nodes and 'm' is the number of their neighbors. After sending the first DIS message, the new nodes have to wait for DIS_INTERVAL time (60 seconds) to send the second and subsequent DIS messages. During this interval, a node is not allowed to send a DIS message. The attacker continuously transmits the DIS messages without considering the DIS_INTERVAL time. During this interval, the DISDet technique counts the number of DIS messages transmitted from a node 'i' to a node 'j' and stores them in a matrix M_{ij}. If any M_{ij} value in the matrix has a value other than zero, then the particular node 'i' is an attacker and should be removed from the DODAG. While reconstructing the DODAG, the attacker is isolated.

The DIS attacker causes detrimental effects to its neighbors and reduces the lifetime of the constrained nodes. The DISDet technique prevents such attacks.

7.2.3. DADTec Technique

The DADTec mechanism is proposed to handle DAO attacks on IoT networks. The harmful effects of the DAO attacker are studied by placing it as a leaf node, as an intermediate node, or as a neighbor to the root node.

The DAO messages are sent by a child node to its parent node after receiving the DIO message. A DAO message is used for recording the reverse route. The DAO attacker sends a large volume of DAO messages to its parent. The same has to be traversed to all the intermediate nodes until reaching the root node. The excessive DAO messages cause resource depletion of the constrained node and degradation of its performance. To overcome the DAO attacks and their impact, DADTec is proposed.

After constructing the DODAG, the DADTec assigns a dynamic threshold value for the maximum permissible DAO message from a node to its parents. A counter is also assigned for each parent node to count the DAO messages it receives from the child node. From all the leaf nodes to the root node, the DADTec checks whether there is an attacker. Whenever a node sends a DAO message, the counter is incremented. If the DAO counter value exceeds the threshold value, then the corresponding node is identified as an attacker. When the attacker is placed as the leaf node, the destructive effects are greater than in other attacker scenarios. The experiment shows the better performance of the DADTec technique than the existing one.

7.2.4. ANIT-Ada Architecture

The intelligent AdaBoost architecture (ANIT-Ada) is the main contribution of this research work. It comprises the VeNADet, DISDet, and DADTec techniques, and

an AdaBoost ensemble model (Ada-IDS). To protect RPL-based IoT networks from Version Number Attack, DIS Flood Attack, and DAO Attack, the three aforementioned techniques are installed on the root node. The network traffic from the simulation environment is collected that contains the three types of attacked packets and normal packets. To develop this model, an AdaBoost ensemble model with decision tree classifiers is utilized. The proposed Ada-IDS can be installed on the Border Router or on a dedicated server. In this research, it is deployed in the Border Router to provide an additional layer of security to the IoT environment. The three techniques and the Ada-IDS model of the ANIT-Ada architecture protect IoT networks from ICMPv6-based RPL resource attacks like Version Number Attack, DIS flooding attack, and DAO attack.

7.3. Significance of the Research Findings

The security-related research is the most promising area throughout the evolution of networks and IoT. The proposed Ada-IDS in the ANIT-Ada architecture is a Centralized IDS that plays a crucial role in detecting and responding the abnormal network traffic in an IoT environment. The inbound and outbound network traffic is monitored and analysed for detecting the attacks. The data communication among the nodes in the IoT networks is also monitored and triggered alerts when it encounters Version Number attack, DIS flooding attack, or DAO attacks. These RPL resource attacks and their negative impacts on the IoT network and the harmfulness caused by such attacks according to the location of the attacker are explained in Table 7.1.

Table 7.1. Attacks and their Impacts

Attack	Effects	Attacker Location and Harmfulness		
		Leaf	**Intermediate**	**Child to Root**
Version Number Attack	Global Repair	Low	Medium	High
DIS Attack	Floods neighbours	It doesn't depend on the location		
DAO Attack	Floods the Parent	High	Medium	Low

As it is shown in Table 7.1, the three attacks cause harmful effects on the IoT network. The DIS attack causes damage to their neighbours, and its impact doesn't depend on the location. When the DAO attacker placed as the leaf node, it has more harmful effects than other locations. For the Version Number Attack, the attacker causes more damage when it is near to the root node. Because, all the descendent nodes receive DIO messages from the ancestor nodes and the children consider the DIO messages from the parents with updated version Number are reliable.

All three types of attacks consume more power, storage space, processing requirements, and increase network traffic. These three attacks make the legitimate nodes unavailable for their normal responsibilities and lead to DoS attacks. Three techniques are suggested in this research to overcome the destructive effects of these three attacks. The Ada-IDS ensemble model monitors the network traffic to detect attacks. The importance of the proposed security mechanisms is listed in Table 7.2.

Table 7.2. Importance of the Proposed Security Solutions

Attack	Technique	Accuracy	Importance
Version Number	VeNADet	94.4%	▪ Unnecessary VN update and global repair are prevented ▪ Prevent the wastage of resources
DIS Attack	DISDet	98.85%	▪ Nodes are protected from DIS Attack ▪ Network resources are saved
DAO Attack	DADTec	99.34%	▪ Parent nodes are safeguarded from the DAO Attacks ▪ Constrained resources are preserved
Version Number, DIS and DAO Attack	Ada-IDS	99.6%	▪ Ada-IDS detects three types of attacks: Version Number Attack, DIS Attack and DAO Attacks ▪ Provide an additional layer of security ▪ Node's lifetime is increased

As it is shown in Table 7.2, the proposed Ada-IDS and three techniques detect the three types of RPL resource attacks and safeguard the resource-constrained nodes from the negative impacts of such attacks. The lifetime of the nodes is increased by deploying the ANIT-Ada architecture. The proposed techniques are lightweight solutions, so they are suitable for the IoT environment. As the number of cyber-physical systems keeps on increasing, the security threats and challenges are also increasing. Thus, this ANIT-Ada architecture, comprising these components, provides two levels of security to the constrained nodes in the IoT networks.

7.4. Limitations and Future Directions

There are different types of security attacks being launched on the Internet of Things. This research work is limited to ICMPv6-based RPL attacks, which target the constrained resources. In the RPL protocol, there are several ICMPv6 control

messages that are used to construct the routing path. In this investigation, three ICMPv6 control message-related RPL resource attacks are considered. Thus, this research work has a boundary with the Version Number Attack, DIS Attack, and DAO Attack only. By using the network log files captured from the normal and attack scenarios, an AdaBoost IDS is developed. As it is an intrusion detection system, the false alarm rate is unavoidable in some circumstances. Hence, the IDS should be fine-tuned frequently. The patterns for attacks and normal packets should be up-to-date to enhance detection accuracy. These are the limitations of the proposed system. The future research directions are listed below.

- Three techniques, such as VeNaDet, DISDet, and DADTec, are proposed to detect the Version Number Attack, DIS Attack, and DAO Attack, respectively. These techniques can be enhanced further by including variant methodologies to improve detection accuracy.
- The 'icmpv6.csv' dataset can be updated to include new attack patterns and normal traces in order to enhance the Ada-IDS further.
- In this existing AdaBoost ensemble model, the Decision Tree Classifier is used. To improve the performance of the IDS, the system can be developed with other classifiers.
- As Deep Learning techniques improve detection accuracy, a Deep Learning-based Intrusion Detection System could be created in the future.
- The 'icmpv6.csv' dataset used in this research is unbalanced. More attack patterns can be included to make the dataset balanced, so that the performance of the Ada-IDS will be improved.

These issues can be considered for future research. Thus, the scope and limitations of the system pave the way for the proposed system for future research and investigation.

7.5. Thesis Summary

This thesis elaborately discussed the detection techniques for three types of RPL resource attacks, such as Version Number Attacks, DAO Attacks, and DIS Attacks. An intelligent ANIT-Ada architecture has been developed which incorporates the detection strategies of these attacks and also an AdaBoost ensemble-based intrusion detection system called Ada-IDS. The Ada-IDS utilizes the network traces of the normal, Version Attack, DIS Attack, and DAO Attack scenarios. This system acts as an additional level of security to the RPL-based IoT networks.

Furthermore, this study will aid future investigations into similar attacks as well as detection approaches related to ICMPv6-based RPL attacks. To improve the performance of the Ada-IDS, more attack traces can be included, and various classifiers can be added.

References

REFERENCES

[Abd, 19] Abdelhadi EloudrhiriHassani, Aicha Sahel and Abdelmajid Badri, "Impact of RPL objective functions on energy consumption in Ipv6 based wireless sensor networks", Colloque sur les Objets et systemes Connectes, Ecole Superieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologied Aix-Marseille (France), CASABLANCA, Morocco, 2019, hal-02298879.

[Abd, 20] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas and Xuan Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles", Transactions on Emerging Telecommunications Technologies, Wiley, 2020, DOI:10.1002/ett.4088.

[Abh, 19] Abhishek Verma and Virender Ranga, "Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks", Trans Emerging Tel Tech., Wiley, 2019, DOI: 10.1002/ett.3802.

[Adn, 20] Adnan Hasan Bdair, Rosni Abdullah, Selvakumar Manickam and Ahmed K. Al Ani, "Brief of Intrusion Detection Systems in Detecting ICMPv6 Attacks", Computational Science and Technology, Lecture Notes in Electrical Engineering 603, Springer Nature, 2020, DOI: 10.1007/978-981-15-0058-9_20.

[Ahm, 16a] Ahmed Saeed, Ali Ahmadinia, Abbas Javed and Hadi Larijani, "Random Neural Network based Intelligent Intrusion Detection for Wireless Sensor Networks", Procedia Computer Science, Volume 80, pp. 2372-2376, 2016.

[Ahm, 16b] F. Ahmed and Y. B. Ko, "A Distributed and Cooperative Verification Mechanism to Defend against DODAG Version Number Attack in RPL", In Proceedings of the 6th International Joint Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2016), pp.55-62 ISBN: 978-989-758-195-3, 2016, DOI: 10.5220/ 0005930000550062.

[Ahm, 20] Ahmad Shabani Baghani, Sonbol Rahimpour and Majid Khabbazian, "The DAO Induction Attack Against the RPL-based Internet of Things", 2020, arXiv:2003.11061v1 [cs.CR], 2020.

[Ale, 18] Alex Shenfield, David Day and Aladdin Ayesh, "Intelligent Intrusion Detection Systems using Artificial Neural Networks", ICT Express, Volume 4, pp.95-99, 2018. DOI: 10.1016/ j.icte.2018.04.003.

[Ama, 14] J. Amaral, L. Oliveira, J. Rodrigues, G. Han and L. Shu, "Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks", IEEE International Conference on Communications (ICC-2014), pp. 1796-1801, 2014.

[Ami, 11] Amit Dvir, T. Holczer and L. Buttyan, "VeRA - Version Number and Rank Authentication in RPL", IEEE 8th International Conference Mobile Adhoc and Sensor Systems (MASS), pp. 709 -714, 2011.

[Amj, 18] Amjad Mehmood, Mithun Mukherjee, Syed Hassan Ahmed, Houbing Song and Khalid Malik, "NBC-MAIDS: Naive Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks", The Journal of Supercomputing, Springer, Volume 74, Issue 11, 2018, DOI: 10.1007/s11227-018-2413-7.

[Ana, 18] Anatol Badach, "RPL messages and their Structure", Internet of Things-Technologies, Protocols and Applications, 2018, https:// www. researchgate.net/ publication/ 326960497_RPL_messages_and_ their_ structure.

[Ang, 21] ANRG, "Contiki Tutorials", University of Southern California, http:// anrg.usc.edu/contiki/index.php/Contiki_tutorials, [Accessed Online: 21 June, 2021]

[Ant, 16] Anthea Mayzaud, Remi Badonnel and Isabelle Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things", International Journal of Network Security, Volume 18, Issue 3, pp. 459-473, 2016.

[Ant, 20] Antonio Arena, Pericle Perazzo, Carlo Vallati, Gianluca Dini and Giuseppe Anastasi, "Evaluating and Improving the Scalability of RPL Security in the Internet of Things", Computer Communications, Volume 151, pp. 119-132, 2020, DOI: 10.1016/j.comcom.2019.12.062.

[Anu, 14] Anuj Sehgal, Anthea Mayzaud, Remi Badonnel, Isabelle Chrisment and Jurgen Sconwalde, "Addressing DODAG Inconsistency Attacks in RPL Networks", Global Information Infrastructure and Networking Symposium (GIIS), Canada, IEEE Xplore,e-ISSN: 2150-329X, 2014, DOI: 10.1109/GIIS.2014.6934253.

[Ari, 18] A. Aris, S. F. Oktugand S. Berna Ors Yalcin, "New Lightweight Mitigation Techniques for RPL Version Number Attacks", Ad hoc Networks, 2018, DOI:10.1016/j.adhoc.2018.10.022.

[Aru, 19] A. Arul Anitha, L. Arockiam, "ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things", International Journal of Innovative Technology and Exploring Engineering, Volume 8, Issue 11, pp. 2278-3075, 2019.

[Ash, 21] Ashwin Karale, "The Challenges of IoT Addressing Security, Ethics, Privacy, and Laws", Internet of Things, Volume 15, No. 00420, 2021, DOI:10.1016/j.iot.2021.100420

[Avi, 18] Avinash Navlani, "AdaBoost Classifier in Python", DataCamp Tutorials, 20th November, 2018, https://www.datacamp.com/ community/ tutorials/ adaboost-classifier-python, [Accessed online: 15 October, 2021].

[Azk, 18] Azka Wani and S. Revathi, "Analyzing Threats of IoT Networks Using SDN Based Intrusion Detection System (SDIoT-IDS)", Smart and Innovative Trends in Next Generation Computing Technologies (NGCT-2017), Springer, CCIS 828, pp. 536-542, 2018.

[Bac, 20] Bacem Mbarek, Mouzhi Ge and Tomas Pitner, "Enhanced Network Intrusion Detection System Protocol for Internet of Things", Proceedings of the 35th Annual ACM Symposium on Applied Computing-SAC'20, pp.1156-1163, 2020, DOI: 10.1145/3341105.3373867.

[Bar,19] Baraq Ghaleb, Ahmed Al-Dubai, Elias Ekonomou, Mamoun Qasem, Imed Romdhani and Lewis Mackenzie, "Addressing the DAO Insider Attack in RPL's Internet of Things Networks", IEEE Communications Letters, Volume 23, Issue 1, pp. 68-71, 2019, http://eprints.gla.ac.uk/ 180885/

[Bil, 17] Biljana L. Risteska Stojkoska and Kire V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions", Journal of Cleaner Production, Volume 140, pp.1454-1464, 2017, DOI:10.1016/j.jclepro.2016.10.006.

[Bor, 14] E. Borgia, "The Internet of Things vision: Key Features, Applications and Open Issues", Computer Communications, 2014, DOI: 10.1016/j.comcom.2014.09.008.

[Bri, 21] Bricata, "Suricata, Snort and Zeek: 3 Open Source Technologies for Securing Modern Networks", https://bricata.com/blog/snort-suricata-bro-ids, [Accessed Online: 25 October, 2021].

[Bru, 17] Bruno Bogaz Zarpaelo, Rodrigo Sanches Miani, Claudio Toshio Kawakani and Sean Carlisto de Alverenga, "A Survey of Intrusion Detection in Internet of Things", Journal of Network and Computer Applications, 2017, DOI: 10.1016/j.jnca.2017.02.009.

[Car, 21] Carrie Mac Gillivray and David Reinsel, "Worldwide Global DataSphere IoT Device and Data Forecast 2021–2025", Market Forecast - Doc # US48087621, https://www.idc.com/getdoc.jsp? containerId= US48087621, [Accessed Online: 21 August, 2021].

[Cer, 15] C. Cervantes, D. Poplade, M. Nogueira and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things", IFIP/IEEE International Symposium on Integrated Network Management (IM-2015), pp. 606–611, 2015.

[Con, 18] Cong Pu, "Mitigating DAO Inconsistency Attack in RPL-based Low Power and Lossy Networks", IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, DOI: 10.1109/CCWC.2018.8301614.

[Con, 19] Cong Pu, "Spam DIS Attack Against Routing Protocol in the Internet of Things", International Conference on Computing, Networking and Communications (ICNC), IEEE, 2019, DOI:10.1109/iccnc. 2019. 8685628.

[Con, 20] Cong Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defences", IEEE Internet of Things Journal, Volume 7, Issue 6, pp. 4937 – 4949, 2020, DOI: 10.1109/JIOT.2020.2971463.

[Dal, 21] Dalvi Abhishek Vishwanath, "Blockchain Technology for Security of IoT devices", 30 September, 2021, https://www.hcltech.com/ blogs/ blockchain-technology-security-iot-devices, [Accessed online: 24 October, 2021].

[Dan, 18] Dan-Radu Berte, "Defining the IoT", Proceedings of the 12th International Conference on Business Excellence, ISSN 2558-9652, pp. 118-128, 2018, DOI:10.2478/picbe-2018-0013.

[Dan, 20] Dan Tang, Liu Tang, Rui Dai, Jingwen Chen, Xiong Li and Joel J.P.C. Rodrigues, "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost", Future Generation Computer Systems, Volume 106, pp. 347–359, 2020, DOI: 10.1016/j.future.2019.12.034

[Elh, 18] Elhadj Benkhelifa, Thomas Welsh and Walaa Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for

IoT: Towards Universal and Resilient Systems", IEEE Communications Surveys & Tutorials, Volume 20, Issue 4, pp. 3496 – 3509, 2018, DOI: 10.1109/COMST.2018.2844742.

[Emr, 19] Emre Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsstrom and M. Gidlund, "A Central Intrusion Detection System for RPL-Based Industrial Internet of Things," 15th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1-5, 2019, DOI: 10.1109/WFCS.2019.8758024.

[Fai, 21] Faiza Medjek, Djamel Tandjaoui, Nabil Djedjig and Imed Romdhani, "Multicast DIS attack mitigation in RPL-based IoT-LLNs", Journal of Information Security and Applications, Volume 61, No 102939, 2021, DOI: 10.1016/j.jisa.2021.102939

[Fat, 17] Fatima Hussain, "Internet of Everything", Springer Briefs in Electrical and Computer Engineering, ISSN: 2191-8120, 2017, DOI:10.1007/978-3-319-55405-1

[Fot, 19] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos and Dionisis Kandris, "A Review of Machine Learning and IoT in Smart Transportation", Future Internet, Volume 11, Issue 94, 2019, DOI: 10.3390/fi11040094.

[Gar, 21] Gartner Glossary, "Information Technology/Internet of Things", https://www.gartner.com/en/information-technology/glossary/internet-of-things, [Accessed Online: 18October, 2021.

[Gon, 20] Gonzalo De La Torre Parra, Paul Rad, Kim-Kwang Raymond Choo and Nicole Beebe, "Detecting Internet of Things attacks using distributed deep learning", Journal of Network and Computer Applications, Volume 163, Issue 1, 2020, DOI: /10.1016/j.jnca.2020.102662.

[Ham, 17] Hamid Bostani and Mansour Sheikhan, "Hybrid of Anomaly-Based and Specification-Based IDS for Internet of Things Using Unsupervised OPF Based on Map Reduce Approach", Computer Communications, Elsevier, Science Direct, Volume 98, pp. 52-71, 2017, DOI: 10.1016/j.comcom.2016.12.001.

[Hez, 18] Hezam Akram Abdul-Ghani, Dimitri Konstantas and Mohammed Mahyoub, "A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Model", International Journal of Advanced Computer Science and Applications, Springer, Volume 9, Issue 3, 2018.

[Hon, 17] Hongliang Tian, Zhihong Qian, Xue Wang and Xiao Liang, "QoI-Aware DODAG Construction in RPL-Based Event Detection Wireless Sensor Networks", Journal of Sensors, Volume 2017, Hindawi, Article ID: 1603713, 9 pages, DOI: 10.1155/2017/160371

[Hon, 18] Hongchun Qu, Libiao Lei, Xiaoming Tang and Ping Wang, "A Lightweight Intrusion Detection Method Based on Fuzzy Clustering Algorithm for Wireless Sensor Networks", Advances in Fuzzy Systems, Volume 2018, Article ID: 4071851, DOI: 10.1155/2018/4071851.

[Imt, 21] Imtiaz Ullah and Qusay H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks", IEEE Access, e-ISSN: 2169-3536, Volume 9, 2021, DOI: 10.1109/ACCESS.2021.3094024.

[Isa, 20] Isam Wadhaj, Baraq Ghaleb, Craig Thomson, Ahmed Al-Dubai and William J. Buchanan, "Mitigation Mechanisms against the DAO Attack on the Routing Protocol for Low Power and Lossy Networks (RPL)", Green Internet of Things, IEEE Access, Volume 8, 2020, DOI: 10.1109/ACCESS.2020.2977476.

[Ivi, 21] Ivica Dodig, Davor Cafuta, Tin Kramberger and Ivan Cesar, "A Novel Software Architecture Solution with a Focus on Long-Term IoT Device Security Support", Applied Sciences, Volume 11, 4955, 2021, DOI:10.3390/ app11114955.

[Jai, 09] A. K. Jain, L. Hong and S. Pankanti, "Internet of Things - Strategic Research Roadmap, Tech. rep., Cluster of European Research projects on the Internet of Things", 2009, http://www.internet-of-things-research. eu/pdf/IoT_Cluster_ Strategic _Research_Agenda_2009.pdf

[Jay, 21] Jayaram Hariharakrishnan and N Bhalaji, "Adaptability Analysis of 6LoWPAN and RPL for Healthcare applications of Internet-of-Things", Journal of ISMAC, Volume 3, Issue 02, pp. 69-81, 2021, DOI:10.36548/jismac.2021.2.001

[Jen, 16] Jen Clark, "What is Internet of Things?", IBM, 2016, https://www.ibm.com/blogs/internet-of-things/what-is-the-iot, [Accessed Online:17October, 2021].

[Jer, 21] Jerry Chun-Wei Lin and Kuo-Hui Yeh, "Security and Privacy Techniques in IoT Environment", Sensors, Volume 21, Issue 1, 2021, DOI:10.3390/s21010001.

[Jes, 19] Jesus Pacheco, Victor H. Benitez and Zhiwen Pan, "Security Framework for IoT End Nodes with Neural Networks", International Journal of Machine Learning and Computing, Volume 9, Issue 4, 2019, DOI:10.18178/ijmlc.2019.9.4.814

[Jis, 21] Jisi Chandroth, Navrati Saxena, Abhishek Roy and Eshita Rastogi, "A New Design and Analysis of Power Saving for IoT Gateway", IETE Technical Review, 2021, DOI: 10.1080/02564602.2021.1880343

[Jos, 20] Jose Costa Sapalo Sicato, Sushil Kumar Singh, Shailendra Rathore and Jong Hyuk Park, "A Comprehensive Analyses of Intrusion Detection System for IoT Environment", Journal of Information Processing Systems, Volume 16, Issue 4, pp.975-990, 2020, DOI:10.3745/ JIPS.03. 0144

[Jyo, 17] Jyoti Deogirikar and Amarsinh Vidhate, "Security Attacks in IoT: A Survey", International Conference on IoT in Social, Mobile, Analytical and Cloud (I-SMAC 2017), IEEE, 2017.

[Kan, 19] Kangyi Wang, "Network Data Management Model based on Naive Bayes Classifier and Deep Neural Networks in Heterogeneous Wireless Networks", Computer and Electrical Engineering, Volume 75, pp.135-145, 2019, DOI:10.1016/j.compeleceng 2019.02.015.

[Keh, 21] Kehinde Lawal and Hamed Nabizadeh Rafsanjani, "Trends, benefits, risks, and challenges of IoT implementation in residential and commercial buildings" Energy and Built Environment, ISSN:2666-1233, 2021, DOI:10.1016/j.enbenv. 2021.01.009

[Key, 16] Keyur K. Patel and Sunil M. Patel, "Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application and Future Challenges", International Journal of Engineering Science and Computing, ISSN: 2321-3361, Volume 6, Issue 5, 2016.

[Lat, 21] Latika Kakkar, Deepali Gupta, Sapna Saxena and Sarvesh Tanwar, "IoT Architectures and Its Security: A Review", Proceedings of the Second International Conference on Information Management and Machine Intelligence, Lecture Notes in Networks and Systems 166, 2021, DOI: 10.1007/978-981-15-9689-6_10

[Lee, 14] T.H. Lee, C.H. Wen, L.H. Chang, H.S. Chiang and M.C. Hsieh, "A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN", Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Lecture Notes in Electrical Engineering, Springer, Volume 260, pp. 1205–1213, 2014.

[Leo, 18] Leonel Santos, Carlos Rabadao and Ramiro Goncalves, "Intrusion Detection Systems in Internet of Things: A Literature Review", 13th Iberian Conference on Information Systems and Technologies (CISTI), 2018, DOI:10.23919/CISTI.2018.8399291.

[Lev, 11] P. Levis, T. Clausen, J. Hui, O. Gnawali and J. Ko, "Trickle Algorithm", RFC 6206, Internet Engineering Task Force, ISSN:2070-1721, 2011.

[Lev, 18] Leverage, "User Interface and User Experience (Chapter 5)", An Introduction to the Internet of Things, Leverage LLC, First Edition, 2018, https://www.leverege.com/iot-ebook/ui-and-ux-design-iot, [Accessed Online: 21 October, 2021].

[Li, 21] Li Da Xu, Yang Lu and Ling Li, "Embedding Blockchain Technology into IoT for Security: A Survey", IEEE Internet of Things Journal, Volume 8, Issue 13, pp: 10452-10473, 2021, DOI: 10.1109/ JIOT.2021. 3060508

[Lil, 20] Lily Hay Newman, "An open Source Effort to Encrypt the Internet of Things", 20 January 2020, https://www.wired.com/story/e4-iot-encryption, [Accessed Online: 23 October, 2021].

[Lin, 13] Linus Wallgren, Shahid Raza and Thiemo Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things", International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, Volume 2013, Article ID 794326, 11 pages, 2013, DOI: 10.1155/2013/794326.

[Mah, 19] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif and M.M.A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches", Internet of Things, Elsevier publication, 2019.

[May, 14] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment and J. Schonwalder, "A Study of RPL DODAG Version Attacks", Monitoring and Securing Virtualized Networks and Services Lecture Notes in Computer Science, Springer, Volume 8508, pp. 92– 104, 2014.

[Mee, 21] Meenakshi Srivatsava and Rakesh Kumar, "Smart Environmental Monitoring Based on IoT: Architecture, Issues, and Challenges", Advances in Computational Intelligence and Communication Technology, Springer Nature, pp.349-358, 2021, DOI:10.1007/978-981-15-1275-9_28.

[Mid, 17] D. Midi, A. Rullo, A. Mudgerikar and E. Bertino, "Kalis: A system for knowledge-driven adaptable intrusion detection for the Internet of Things", Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17), 2017.

[Min, 18] Minhaj Ahmad Khan and Khaled Salah, "IoT security: Review, blockchain solutions, and open challenges", Future Generation Computer Systems, Volume 82, pp. 395-411, 2018, DOI:10.1016/j .future.2017. 11.022.

[Mit, 14] R. Mitchell and I. Chen, "A Survey of Intrusion Detection Techniques for Cyber-Physical Systems", ACM Computing Surveys (CSUR), Volume 46, Issue 4, 2014.

[Moh, 18a] Mohamed Faisal Elrawy, Ali Ismail Awad and Hesham F.A. Hamed, "Intrusion detection systems for IoT-based smart environments: A Survey", Journal of Cloud Computing: Advances, Systems and Applications, Volume 7, Issue 21, 2018, DOI: 10.1186/s13677-018-0123-6

[Moh, 18b] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi and Amit P. Sheth, "Machine Learning for Internet of Things Data Analysis: A Survey", Journal of Digital Communication and Networks, Volume 4, pp.161-175, 2018, DOI:10.1109/COMST.2019.2896380.

[Moh, 18c] Mohammad Nikravan, Ali Movaghar and Mehdi Hosseinzadeh, "A Lightweight Defense Approach to Mitigate Version Number and Rank Attacks in Low-Power and Lossy Networks", Wireless Pers. Commun. 99, Springer Nature, pp. 1035–1059, 2018, DOI: 10.1007/s11277-017-5165-4.

[Moh,17] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh, “Evaluation of Machine Learning Algorithms for Intrusion Detection System”, IEEE Explore, ISSN: 1949-0488, 2017, DOI: 10.1109/SISY.2017.8080566.

[Moj, 20] Mojtaba Eskandari, Z. H. Janjua and Fabio Antonelli, “Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices”, IEEE Internet of Things Journal, 2020, DOI: 10.1109/JIOT.2020.2970501.

[Muh, 18] Muhammad Mahtab Alam, Hassan Malik, Muhidul Islam Khan, Tamas Pardy, Alar Kuusik and Yannick Le Moullec, “A Survey on the Roles of Communication Technologies in IoT-Based Personalized Healthcare Applications”, Special Section on Wearable and Implantable Devices and Systems, IEEE Access, Volume 6, 2018, DOI:10.1109/ACCESS.2018.2853148.

[Muh, 20] Muhammad Asaad Cheema, Hassaan Khaliq Qureshi, Chrysostomos Chrysostomou and MariosLestas, “Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things”, 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE Xplore, 2020, DOI: 10.1109/DCOSS49796.2020.00074.

[Nan, 18] Nanda Kumar Thanigaivelan, Ethiopia Nigussie, Seppo Virtanen and JouniIsoaho, “Hybrid Internal Anomaly Detection System for IoT: Reactive Nodes with Cross-Layer Operation”, Security and Communication Networks, Hindawi, Volume 2018, Article ID 3672698, 15 pages, 2018, DOI: 10.1155/2018/3672698.

[Nan, 19] P.S. Nandhini and B.M. Mehtre, "Directed Acyclic Graph Inherited Attacks and Mitigation Methods in RPL: A Review", Sustainable Communication Networks and Application, ICSCN 2019, Lecture Notes on Data Engineering and Communications Technologies, Springer, Volume 39, pp.242-252, 2019, DOI: 10.1007/978-3-030-34515-0_25

[Nas, 16] M.Nasimuzzaman Chowdhary, Ken Ferens and Mike Forens, "Network Intrusion Detection using Machine Learning", International Conference of Security and Management SAM-16, 2016.

[Nat, 21] National Institute of Standards and Technology-Computer Security Resource Centre, "glossary-Sandbox", https://csrc.nist.gov/glossary/term/sandbox, [Accessed Online: 11October, 2021].

[Nav, 18] Navinkumar Maheshwari and Haresh Dagal, "Secure communication and firewall architecture for IoT applications", 10th International Conference on Communication Systems & Networks (COMSNETS), IEEE Xplore, IEEE, e-ISSN: 2155-2509, 2018, DOI: 10.1109/COMSNETS.2018.8328215

[Nou, 18] Nour Moustafa, Benjamin Turnbull and Kim-Kwang Raymond Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things", IEEE Internet of Things Journal, 2018, DOI 10.1109/JIOT.2018.2871719

[Oh, 14] D. Oh, D. Kim and W. W. Ro, "A malicious pattern detection engine for embedded security systems in the Internet of Things", Sensors, ISSN: 24188–24211, Volume 14, Issue 12, 2014.

[Oka, 17] Okamura Toshihiko, "Lightweight Cryptography Applicable to Various IoT Devices", NEC Technical Journal, Special Issue on IoT That Supports Digital Businesses, Volume 12, Issue 1, 2017.

[Okw, 18] Okwori Anthony Okpe, Odey Adinya John and Siman Emmanuel, "Intrusion Detection in Internet of Things", International Journal of Advanced Research in Computer Science", ISSN: 0976-5697, Volume 9, Issue 1, 2018, DOI:10.26483/ijarcs.v9i1.5429.

[Oma, 16] Omar E. Elejla, Bahari Belaton, Mohammed Anbar and Ahmad Alnajjar, "A Reference Dataset for ICMPv6 based Flooding Attacks", Journal of Engineering and Applied Sciences, Volume 11, Issue 3, pp. 476-481, 2016.

[Oma, 17] Omar E. Elejla, Mohammed Anbar and BahariBelaton, "ICMPv6-Based DoS and DDoS Attacks and Defense Mechanisms: Review", IETE Technical Review, Volume 34, Issue 4, 2017, DOI: 10.1080/02564602. 2016.1192964.

[Osa, 21] Osama Alkadi, Nour Moustafa, Benjamin Turnbull and Kim-Kwang Raymond Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", IEEE Internet of Things Journal, Volume 8, Issue 12, pp. 9463 – 9472, 2021, DOI: 10.1109/JIOT.2020.2996590.

[Pav, 15] Pavan Pongale and Gurunath Chavan, "Real Time Intrusion and Wormhole Attack Detection in Internet of Things", International Journal of Computer Applications, ISSN: 0975-8887, Volume 121, Issue 9, 2015.

[Phi, 18] Philokypros P. Ioulianou, Vassilios G. Vassilakis, Ioannis D. Moscholios and Michael D. Logothetis, "A Signature-based Intrusion Detection System for the Internet of Things", International Conference on Information and Communication Technology Forum (ICTF-2018), Austria, 2018.

[Phi, 21] Philip Virgil Astillo, Jaemin Jeong, Wei-Che Chien, Bonam Kim, Joung Soon Jang and Ilsun You, "SMDAPS: A Specification-based Misbehavior Detection System for implantable devices in Artificial Pancreas System", Journal of Internet Technology, e-ISSN 2079-4029, Volume 22, Issue 1, pp. 1-11, 2021.

[Pra, 21] Prabhat Kumar, Govind P. Gupta and Rakesh Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks", Journal of Ambient Intelligence and Humanized Computing, Volume 12, pp. 9555–9572, 2021, DOI: 10.1007/s12652-020-02696-3.

[Rac, 21] Rachit, Shobha Bhatt and Prakash Rao Ragiri, "Security trends in Internet of Things: A Survey", SN Applied Sciences, Springer Nature, Volume 3, Issue 121, 2021, DOI: 10.1007/s42452-021-04156-9.

[Rad, 21] Radu Tyrsina, "5+ best IoT antivirus software and antimalware", 31st March, 2021, https://windowsreport.com/iot-antivirus-anti-malware [Accessed Online: 21 October, 2021].

[Ram, 18] C. Ramakrishna, G. Kiran Kumar, A. Mallikarjuna Reddy, Pallam Ravi, "A Survey on various IoT Attacks and its Countermeasures", International Journal of Engineering Research in Computer Science and Engineering, ISSN: 2394-2320, Volume 5, Issue 4, 2018.

[Ras, 18] Rashmi Sahay, G. Geethakumari and Koushik Modugu, "Attack Graph based Vulnerability Assessment of Rank property in RPL-6LowPAN in IoT", IEEE Explore, 2018, DOI: 10.1109/WF-IoT.2018.8355171.

[Ras, 20] Rashmi Sahay, G. Geethakumari, Barsha Mitra and Ipsit Sahoo, "Efficient Framework for Detection of Version Number Attack in Internet of Things", Springer Nature Switzerland AG 2020, AISC 941, pp. 480–492, 2020, DOI: 10.1007/978-3-030-16660-1_47.

[Rav, 18] Ravi Teja Gaddam and M. Nandhini, "An Analytical Approach to enhance the Intrusion Detection in Internet of Things Network", International Journal of Latest Trends in Engineering and Technology, e-ISSN: 2278-621X, Volume 9, Issue 3, pp. 258-267, 2018, DOI: 10.21172/1.93.43.

[Raz, 13] S. Raza, L. Wallgren and T. Voigt, "SVELTE: real-time intrusion detection in the Internet of Things," Ad Hoc Network, Volume 11, Issue 8, pp. 2661-2674, 2013.

[Saf, 18] Safa A. Ahmed, Nahla Fadhil Alwan and Ammar Mohamed Ali, "Overview for Internet of Things: Basics, Components and Applications", Journal of University of Anbar for Pure Science, ISSN: 1991-8941, Volume 12, Issue 3, 2018.

[Sam, 21] Sam Daley, "Blockchain and IoT: 8 Examples Making Our Future Smarter", 9 May 2021, https://builtin.com/blockchain/blockchain-iot–examples, [Accessed Online: 28 October, 2021].

[San, 19] Sana Ullah Jan, Saeed Ahmed, Vladimir Shakhov and Insoo Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things", IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2907965.

[Sar, 18] Saroj Kr. Biswas, "Intrusion Detection Using Machine Learning: A Comparison Study", International Journal of pure and Applied Mathematics, ISSN: 1311-8080, e-ISSN: 1314-3395, Volume 118, Issue 19, pp.101-114, 2018.

[Sem, 20] Semih Cakir, Sinan Toklu and Nesibe Yalcin, "RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning", IEEE Access, Volume 8, 2020, DOI: 10.1109/ACCESS.2020.3029191.

[Shr, 17] D. Shreenivas, S. Raza and T. Voigt, "Intrusion Detection in the RPL connected 6LoWPAN Networks", Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, Abu Dhabi, United Arab Emirates, 2017.

[Shu, 21] Shubair Abdullah, "IPv6 Multicast Vulnerability–An Overview", Applied Computing Journal, ISSN: 2788-9688, Volume 1, Issue 1, pp 1-9, 2021, DOI: 1052098/acj.202112

[Sop, 21] Sophie Moore, "Wireless Sensor Networks: Tmote Sky", https://wirelesssensornetworks.weebly.com/blog/tmote-sky, [Accessed Online: 07 June, 2021]

[Ste, 17] Stephabie B. Baker, Wei Xiang and Ian Atkinson, "Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities", IEEE Access, Volume 5, 2017, DOI: 10.1109/ACCESS.2017.2775180,

[Sya, 17] Syam Akhil Repalle and Venkata Ratnam Kolluru, "Intrusion Detection System using AI and Machine Learning Algorithm", International Research Journal of Engineering and Technology, e-ISSN: 2395-0056, Volume 4, Issue 12, 2017.

[Sye, 18] Syed Rizvi, Joseph Pfeffer, Andrew Kurtz and Mohammad Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT", 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering, IEEE, 2018, DOI 10.1109/ TrustCom/BigDataSE.2018.0003

[Thu, 20] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (RPL)", https://tools.ietf.org/html/rfc6552, [Accessed Online: 20 April 2020].

[Tia, 20] Tian Wang, Yaxin Mei, Xuxun Liu, Jin Wang, Hong-Ning Dai and Zhijian Wang, "Edge-based auditing method for data security in resource-constrained Internet of Things", Journal of Systems Architecture, Elsevier, 2020, DOI:10.1016/j.sysarc.2020.101971

[Tra, 19] Tran Anh Khoa, Mai Minh Man, Tan-Y Nguyen, Van Dung Nguyen and Nguyen Hoang Nam, "Smart Agriculture Using IoT Multi-Sensors: A Novel Watering Management System", Journal of Sensor and Actuator Networks, Volume 8, Issue 45, 2019, DOI:10.3390/jsan8030045.

[Tsv, 11] Tsvetko Tsvetkov, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks", Network Architectures and Services, pp. 59-66, 2011, DOI: 10.2313/NET-2011-07-1_09.

[Vas, 20] J.P. Vasseur, M. Kim, K. Pister, N. Dejean and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks", https://tools.ietf.org/html/rfc6551, [Accessed Online: 14 April 2020].

[Vel, 16] A. Velinov and A. Mileva, "Running and Testing Applications for Contiki OS Using Cooja Simulator", International Conference on Information Technology and Development of Education–ITRO 2016, Republic of Serbia, pp. 279-285, June 2016, https://core.ac.uk/ download/pdf/80817534.pdf

[Vik, 20] S. N. Vikram Simha, Reema Mathew, Shubhashisa Sahoo and Rajashekhar C. Biradar, "A Review of RPL Protocol Using Contiki Operating System", Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI-2020) (48184), IEEE Xplore, ISBN: 978-1-7281-5518-0, 2020, DOI:10.1109/ icoei48184. 2020.9142903.

[Vik, 21] Vikash Kumar, Ayan Kumar Das and Ditipriya Sinha, "UIDS: a Unified intrusion detection system for IoT environment", Evolutionary Intelligence, Springer, Volume 14, Issue 1, pp. 47-59, 2021, DOI:10.1007/s12065-019-00291-w

[Vip, 17] Vipindev Adat and B.B.Gupta, "Security in Internet of Things: issues, challenges, taxonomy, and architecture", Telecommunication System, Springer, 2017, DOI: 10.1007/s11235-017-0345-9.

[Vuk, 21] Vuk Lesi, Zivana Jakovljevic, and Miroslav Pajic, "Security Analysis for Distributed IoT-Based Industrial Automation", IEEE Transactions on Automation Science and Engineering, pp.1-16, 2021, DOI: 10.1109/TASE.2021.3106335.

[Win,12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", Internet Engineering Task Force (IETF), RFC 6550, ISSN: 2070-1721, 2012.

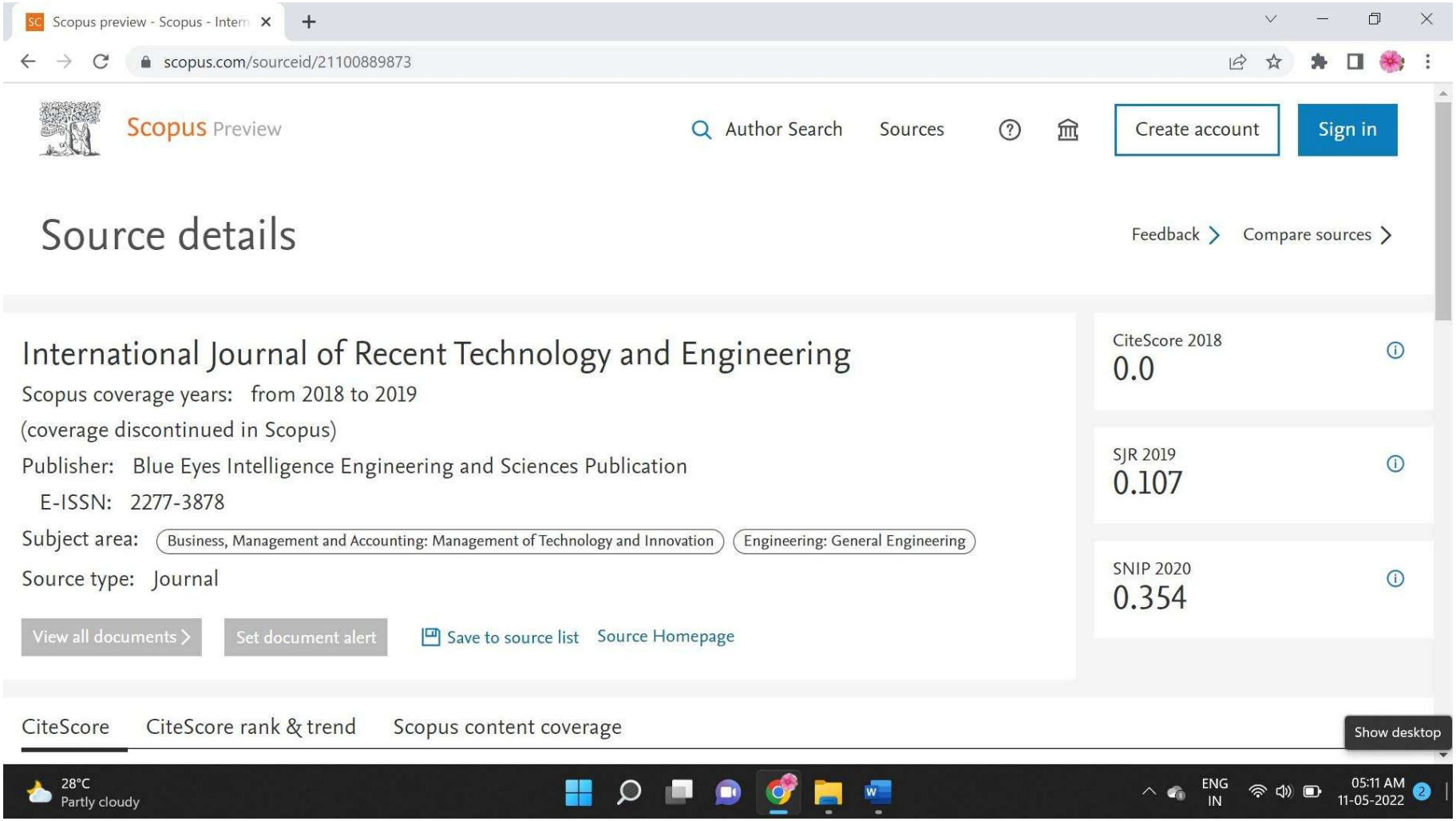
[Yan, 18] Yang Lu and Li Da Xu, "Internet of Things (IoT) Cyber Security Research: A Review of Current Research Topics", IEEE Internet of Things Journal, 2018, DOI 10.1109/JIOT.2018.2869847.

[Yul, 17] Yulong Fu, Cheng Yan, Jin Cao, Ousmane Kore and Xuefei Cao, "An Automata based Intrusion Detection method for Internet of Things", Mobile Information Systems, Hindawi, Volume 2017, Article ID 1750637, DOI: 10.1155/2017/1750637

[Zah, 20] Zahrah A. Almusaylim, NZ Jhanjhi and AbdulazizAlhumam, "Detection and Mitigation of RPL Rank and Version Number Attacks in the Internet of Things: SRPL-RP", Sensors, Volume 20, Issue 21, 2020, DOI:10.3390/s20215997

[Zia, 21] Zia A. Sardar, Stewart Merkel and Aaron Arellano, "Why Hardware-Based Cryptography offers stronger IoT Design Protection, https://www.maximintegrated.com/en/design/blog/why-hardware-based cryptography-offers-stronger-iot-design-protection.html, [Accessed online: 10 September, 2021].

Photocopies of Papers Published in the International Journals


Scopus preview - Scopus - Intern
scopus.com/sourceid/21100889873
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
International Journal of Recent Technology and Engineering
Scopus coverage years: from 2018 to 2019
(coverage discontinued in Scopus)
Publisher: Blue Eyes Intelligence Engineering and Sciences Publication
E-ISSN: 2277-3878
Subject area: Business, Management and Accounting: Management of Technology and Innovation
Engineering: General Engineering
Source type: Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2018
0.0
SJR 2019
0.107
SNIP 2020
0.354
CiteScore
CiteScore rank & trend
Scopus content coverage
Show desktop
28°C
Partly cloudy
ENG
IN
05:11 AM
11-05-2022

A Hybrid Method for Smart Irrigation System

A. Arul Anitha, A. Stephen, L. Arockiam

Abstract*: Internet of Things (IoT) is a boon to the technological developments during the past decade. Though the adoption of this technology in agriculture has gone up immensely in recent years, the implementation of the smart irrigation system remains its initial stage in this agricultural setup. The sprinkler or dripper irrigation methods are widely used in the smart irrigation environment. In this paper a hybrid method is proposed to select the irrigation method automatically based on the climate changes and soil moisture level. By enhancing this method using the rapid growing technologies and IoT enabled smart irrigation controllers, the agriculture sector will be improved over the foreseeable future.

***Keywords*: Smart Irrigation, Sprinkler, Dripper, Hybrid Method.**

I. INTRODUCTION

Agriculture plays a vital role in countries like India. As Mahatma Gandhi said, the development of our country depends on the economic status of the villages which are mainly depending upon agriculture. Water is the core element for agriculture [1]. Figure 1 explains the need of water resource for agriculture. In India 80% of water resource is used for agriculture. Nowadays, climate changes reflect in the time and duration of monsoons which are the main water source.

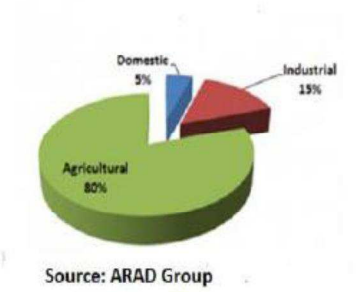


Figure 1: Water usage in India

To overcome this water scarcity issue, the smart irrigation system is deployed in agriculture field. The smart irrigation system monitors the weather, soil type and its moisture level, evaporation and water usage of the plants and automatically adjusts the watering schedule [2]. It helps the farmers to optimize the water usage, enrich the quality of crop growth and quantity of yields in their fields. Smart irrigation systems are easy to implement and it has a straight forward approach [3]. In this paper, the background study related to smart irrigation, the issues and challenges in implementing the smart irrigation and IoT-based smart irrigation methods are discussed. A hybrid irrigation framework is proposed and some research issues in the smart irrigation systems are also highlighted

II. RELATED WORKS

Yuthika et. al [4] proposed an Intelligent IoT based irrigation system. For analysing and predicting KNN (K-Nearest Neighbour) classification machine learning algorithm was used in this approach. Machine to Machine (M2M) technology was implemented for communication among the devices and a prototype model was developed to test the efficiency. The security and water source issues were ignored in their work. Alauddin et al [5] proposed a Cloud based IoT for Smart Garden Watering System using Arduino Uno which was used to monitor and to maintain the soil moisture and light intensity. The monitored data was sent to ThingSpeak IoT cloud. The data gathered in the cloud was analysed and when it reached the threshold value, an action was sent accordingly from the cloud to the irrigation system. It needs further refinement like including temperature sensor and controlling the system using smart phone. Harishankar et al [6] suggested an automatic sprinkler irrigation system using solar power for automating the irrigation process using solar power and to optimize the use of water. When implemented for bore holes, the system was found to be successful. Solar pumps also offered clean solutions with no danger of borehole contamination. Maroufpoor et al [7] recommended three artificial intelligence methods such as Artificial Neural Network (ANN), Adaptive Neuro-fuzzy Inference Systems (ANFIS) and Gene Expression Programming (GEP) for estimating wind drift and evaporation losses from sprinkler irrigation systems. According to the authors, Gene Expression Programming method provided the best result. Fabrizio et al [8] explained a machine learning technique to manage heterogeneous datasets which include physical, biological and sensory values collected from real-time agricultural sector. Weather, humidity, wind speed and soil types were the factors considered in their approach. The supervised machine learning algorithms such as decision tree, K-nearest neighbours, Neural Network and polynomial predictive models were used in this research. According to the authors, effective implementation of their work will increase productivity and will save the environmental resources and will pursue economic profits.

Revised Manuscript Received on September 25, 2019.

A. Arul Anitha*, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli 620002, India. Email: arulanita@gmail.com.

A. Stephen*, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli 620002, Tamilnadu India. Email: stephena003@gmail.com.

Dr. L. Arockiam*, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli 620002, Tamilnadu, India. Email: larockiam@yahoo.co.in

III. ISSUES AND CHALLENGES

To adopt and implement the technologies in agricultural sector, the developing countries like India have to face many issues and challenges.

- Lack of knowledge and fear of implementing and upgrading the technology in higher levels among large number of farmers in the country.
- The solution must have the customization facilities for different languages, so that it could be easy to understand for the ordinary people.
- Interoperability is another issue, due to lot of platforms and vendors for IoT tools and techniques.
- The farms own by the farmers are varying in its size. Hence, the solution related to smart irrigation should be scalable and flexible.
- Security is another big issue. If one of the sensors is hacked it will collapse the entire system. The security tools have to be updated frequently and it leads to additional headache to the poor farmers.

To find out a solution having all these requirement is not easy. These challenges and issues lead to further research and developments in the smart irrigation field.

IV. IOT BASED IRRIGATION METHODS

IoT based smart irrigation system is capable of automating the irrigation process by analyzing the moisture of soil and the climate condition. When the power supply is given to the microcontroller, it will check the soil moisture content [9]. If the moisture content is not up to the threshold then it makes the motor to get on automatically and turns off automatically if it reaches to the threshold level. The need of water for any crop is also reduced drastically. Remote monitoring is also possible in IoT based smart irrigation system.

A. Smart Irrigation System Requirements:

The core components for deploying the smart irrigation system are: Node MCU, Soil moisture sensor, temperature sensor, humidity sensor, 5v Relay, Sprinkler, Dripper, Solenoid valve and Water tank [10]. There are two types of irrigation methods such as dripper and sprinkler can be used according to the season. Dripper irrigation method can be used in the windy season, whereas sprinkler irrigation method can be adopted in the summer season.

B. Sprinkler Irrigation System:

Sprinkler irrigation system allows application of water under high pressure with the help of a pump. Small diameter nozzle is placed in the pipes, it releases rainfall like water through the distributed system of pipes and sprays into air and irrigates [11]. Thus, it is not suitable for the windy season. Figure 4.1 depicts the sprinkler irrigation system with solenoid valve and other required components

In summer, the leaves of the plants easily wither; since this sprinkler irrigation method sprays water like rainfall, it is suitable for the summer season

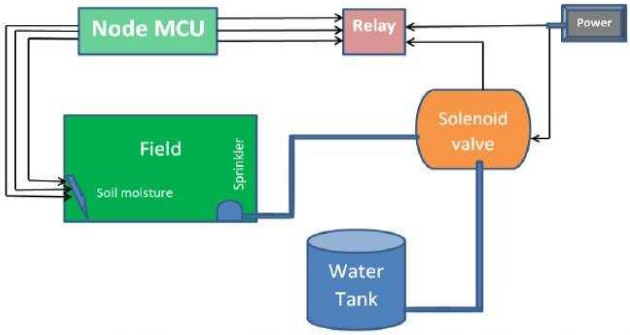


Figure 4.1 Design of Sprinkler Irrigation Method

C. Dripper Irrigation System:

Drip irrigation systems distribute water through a network of valves, pipes, tubing and emitters. Depending on how well designed, installed, maintained, and operated it is, a drip irrigation system can be more efficient than sprinkler irrigation. This system with dripper component is explained in figure 4.2.

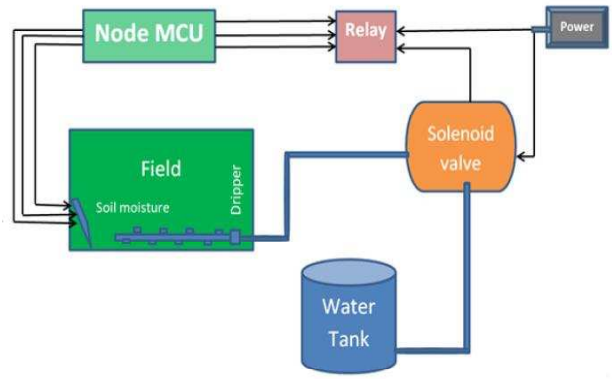


Figure 4.2 Design of Dripper Irrigation Method

This method is useful for all seasons, but sprinkler outperforms this drip system during summer season. There is a need for a better irrigation method which adapts all weather.

V. PROPOSED HYBRID METHOD FOR SMART IRRIGATION

In some situation both sprinkler and dripper irrigation methods can be used when the crop is needed to spray water on leaves of the crop as well as to be fed water to the root of the crop. According to the weather and climate condition either sprinkler or dripper method can be adopted. It is called hybrid irrigation method. This system can be controlled from anywhere through the User Interfaces such as mobile phone or laptop. The sensor data sent by different sensors are stored into the Cloud like ThingSpeak through the border router. The working environment with the combination of sprinkler and dripper is shown in the figure 5.1 and the various functionalities of the smart hybrid irrigation system framework are explained below:

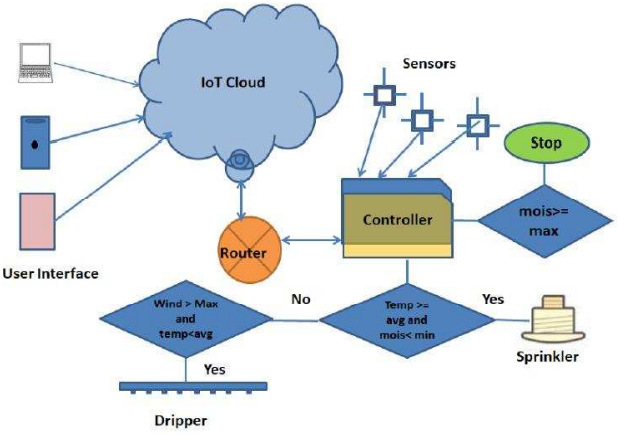


Figure 5.1. Hybrid Smart Irrigation Framework

Step1: The soil moisture sensor and weather sensors will give the details of moisture level of the soil, temperature, rainfall, wind speed and humidity information to node MCU (Microcontroller) whether water is needed to the crop or not.

Step 2: If watering is needed, the Microcontroller will trigger the relay to be switched on the power.

Step 3: Once the relay is switched on then the solenoid valve will be opened which is already connected with water tank and water is poured using sprinkler/dripper to the crop. If the temperature is high and the soil moisture level is very low then the sprinkler system is enabled to water the plants. If the wind speed is very high and also the moisture level of the soil is below the average level then the drip irrigation method is triggered.

Step 4: After irrigation process, the information will be sent to the microcontroller and the relay will be triggered to switch off the power.

Step 5: If water is not needed the irrigation system remains idle.

Mobility of the system helps the farmers to monitor the irrigation process from anywhere. Thus, by using this hybrid smart irrigation strategy, protection of the crops against various climate conditions is very easy.

VI. CONCLUSION

The Smart hybrid irrigation system is recommended to provide a valuable tool for conserving water planning and irrigation scheduling. The dripper or sprinkler method is selected automatically according to the moisture level of the soil, surrounding temperature and climate condition. This system can be used in large agricultural area where human effort needs to be minimized and the farmers can monitor and control the irrigation process from anywhere. Many aspects of the system can be customized and fine-tuned according to the requirement of a particular plant.

REFERENCES

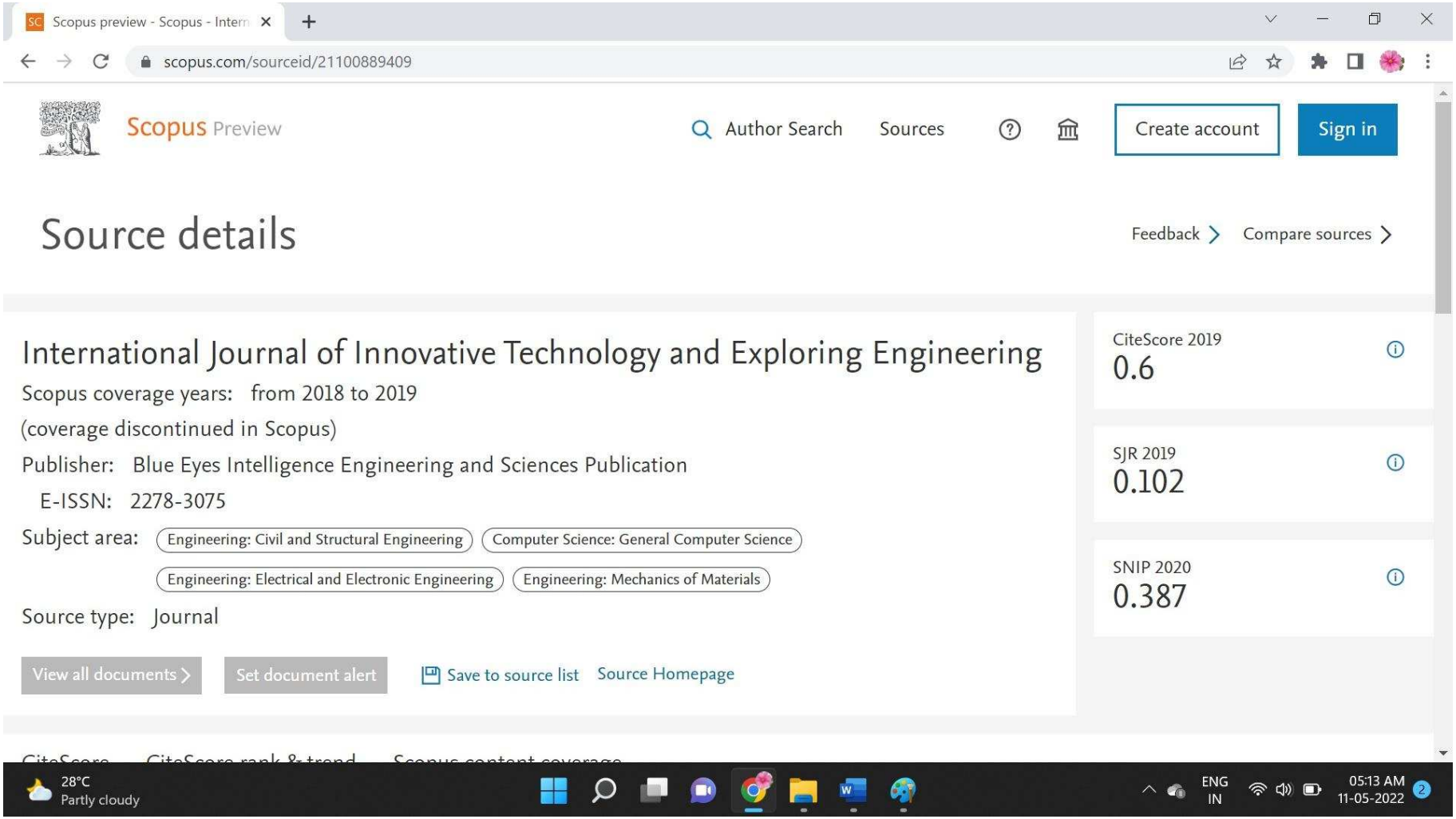
1. K K Namala, Krishna Kanth Prabhu A V, Anushree Math, Ashwini Kumari, Supraja Kulkarni, "Smart Irrigation with Embedded Systems", IEEE Bombay Section Symposium (IBSS), 2016.
2. N. Đuzić and D. Đumić, "Automatic Plant Watering System and its Applications", Coll. Antropol. 41 (2017).
3. L. Selvam and Dr. P. Kavitha, "Smart Agriculture Monitoring System Based On Internet Of Things (IoT)", Vol. 9, issue 6, 2017, pp. 1416-1426.
4. Yuthika Shekhar, Ekta Dagur and Sourabh Mishra, "Intelligent IoT Based Automated Irrigation System", International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 12, Number 18, 2017, pp. 7306-7320.
5. Alauddin Al-Omary , Haider M. AlSabbagh , Hussain Al-Rizzo, "Cloud based IoT for Smart Garden Watering System using Arduino Uno", Smart Cities Symposium 2018 (SCS'18), University of Bahrain", April 2018.
6. S. Harishankar, R. Sathish Kumar, Sudharsan K.P, U. Vignesh and T.Viveknath, "Solar Powered Smart Irrigation System", Advance in Electronic and Electric Engineering, ISSN 2231-1297, Volume 4, Number 4 (2014), pp. 341-346
7. E. Maroufpoor, H.Sanikhani, S. Emamgholizadeh and Ö. Kisi, "Estimation of wind drift and evaporation losses from Sprinkler Irrigation Systems by different Data-driven method", Irrigation and Drainage 2017, DOI: 10.1002/ird.2182.
8. Fabrizio Balducci, Donato Impedovo and Giuseppe Pirlo, "Machine learning applications on agricultural datasets for smart farm enhancement", Machines 2018, 6, 38 (Scopus), Doi:10.3390/machines6030038.
9. Aman Bafna, Anish Jain, Nisarg Shah and Rishab Parekh, "IoT Based Irrigation Using Arduino And Android On The Basis Of Weather Prediction", International Research Journal of Engineering and Technology (IRJET), Volume 05 Issue 05, 2018, pp. 433-437.
10. Meraj Ahmed, Md Kamre Alam and Imaad shafi, "A Nobel Report on Smart Irrigation System using IoT", International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 11, Nov 2018, e-ISSN: 2395-0056, p-ISSN: 2395-0072.
11. M. Rakibuzzaman, Sk. Rahul, M.R. Jahan, F.B.R. Urme and AFM Jamal Uddin, "Performance of Drip Irrigation System over Conventional Irrigation Technique for Tomato Production on Rooftop", International Journal of Business, Social and Scientific Research, ISSN: 2309-7892 (Online), 2519-5530 (Print), Volume: 7, Issue: 1, Page: 40-43, August-November 2018.

AUTHORS PROFILE

A.Arul Anitha is a Full-time Ph.D Research Scholar in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli which is affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India. She received her Master's degree in Computer Applications (MCA) from Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India and Bachelor of Science in Computer Science from Madurai Kamaraj University, Madurai, Tamilnadu, India. Her Research interest is on Network Security, Intrusion Detection Systems, Internet of Things (IoT) and Machine Learning. She has cleared the National Eligibility Test (NET) conducted by the National Testing Agency (NTA) in December, 2018.

A.Stephen is a Full-time Ph.D Research Scholar in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India. He received his Master of Philosophy (MPhil) in Computer Science from St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India. He received his Master degree in Computer Science (MSc) from Loyola College (Autonomous), Chennai, Tamilnadu, India and Bachelor of Science in Computer Science (BSc) from Loyola College, Tiruvannamalai, Tamilnadu, India. His research interests are Internet of Things (IoT) and Cloud Computing.

Dr. L. Arockiam, working as an Associate Professor in the Department of Computer Science and Dean of Computing Sciences at St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 29 years of experience in Teaching and 21 years of experience in Research. He has Published 345 Research Articles in the International / National Journals and Conferences. He has guided more than 38 M. Phil Research Scholars and 29 Ph. D. Research Scholars and at preset he is guiding 6 Ph. D Research Scholars. His research interests are Internet of Things, Cloud Computing, Big Data, Data Mining, Software Measurement, Cognitive Aspects in Programming, Web Service and Mobile Networks.


Scopus preview - Scopus - Intern
scopus.com/sourceid/21100889409
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
International Journal of Innovative Technology and Exploring Engineering
Scopus coverage years: from 2018 to 2019
(coverage discontinued in Scopus)
Publisher: Blue Eyes Intelligence Engineering and Sciences Publication
E-ISSN: 2278-3075
Subject area:
Engineering: Civil and Structural Engineering
Computer Science: General Computer Science
Engineering: Electrical and Electronic Engineering
Engineering: Mechanics of Materials
Source type: Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2019
0.6
SJR 2019
0.102
SNIP 2020
0.387
28°C
Partly cloudy
ENG
IN
05:13 AM
11-05-2022

ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things

A. Arul Anitha, L. Arockiam

Abstract*: Internet of Things (IoT) makes everything in the real world to get connected. The resource constrained characteristics and the different types of technology and protocols tend to the IoT be more vulnerable than the conventional networks. Intrusion Detection System (IDS) is a tool which monitors analyzes and detects the abnormalities in the network activities. Machine Learning techniques are implemented with the Intrusion detection systems to enhance the performance of IDS. Various studies on IoT reveals that Artificial Neural Network (ANN) provides better accuracy and detection rate than other approaches. In this paper, an Artificial Neural Network based IDS (ANNIDS) technique based on Multilayer Perceptron (MLP) is proposed to detect the attacks initiated by the Destination Oriented Direct Acyclic Graph Information Solicitation (DIS) attack and Version attack in IoT environment. Contiki O.S/Cooja Simulator 3.0 is used for the IoT simulation.

Keywords: Artificial Neural Network, IDS, IoT, Multilayer Perceptron

I. INTRODUCTION

The Internet of Things (IoT) paves way to connect large volume of real world objects to the global network. These objects communicate with other objects using their unique identifiers to perform certain tasks and for data transmission. The Low power and Lossy Networks (LLN) are deployed in large-scale to meet the high demand of this technology. Different technologies, protocols and standards used in IoT and the tremendous growth of IoT devices in the global network bring additional vulnerabilities to the IoT networks [1]. On account of the resource constrained characteristics of the IoT nodes, the conventional authentication and cryptography security mechanisms are not desirable to the IoT networks. Hence, it is mandatory to provide additional security mechanism like IDS to protect the IoT network from security threats and vulnerabilities [2].

IDS can be software/hardware or the combination of both which is used to investigate the malicious traffic in the network or a particular node. If there is any attack, the IDS monitors, detects and alerts the administrator and logs the attacks for analysis [3]. Intrusion Detection System automates the process of monitoring and detecting and alerting the administrators to take necessary actions to prevent the destructive impacts of the attacks [4].

According to Jyothi et al., physical attacks are initiated on the hardware of the system, network attacks are performed on IoT network elements and Software attacks are performed by using software like malware, virus, spyware and worms [5]. Based on the security vulnerabilities targeting on network resources, network topology and network traffic Anthea et al., proposed the taxonomy of RPL attacks. Because of the fake control messages and building of loops in the Destination Oriented Direct Acyclic Graphs (DODAGs), the attacks reduce the lifetime of the RPL network [6].

DIS (DODAG Information Solicitation) attack and Version attacks are the two RPL attacks considered in this work. To get the topology related information, a new node sends DIS message to its neighbors before it becomes the member of the network. In DIS attack, the malicious node resets the DIO timer frequently and sends DIS messages to the nodes within one-hop distance. This reduces the throughput and leads the energy of the nodes also to be exhausted. The DIS attacker sends unicast, broadcast or multicast DIS messages to its neighbors. Thus, it increases the control overhead in the network traffic.

Each DODAG tree has its own version number. It will be reconstructed when a new version number greater than the current one is published. Version number attack will occur by reconstructing the DODAG tree frequently using higher version. The nodes start the process of constructing a new DODAG when they receive the higher version number which leads to inconsistency in the network topology [7]. The inconsistencies in the networks also upturn the possibilities of generating loops and rank inconsistencies in the network. When the attacker node communicating with other attackers, DIS attack and Version attack will lead RPL network to other types of attacks and which will collapse the entire network.

Machine Learning Algorithms are used to enhance the detection accuracy of the IDS. Artificial Neural Network (ANN) provides better detection accuracy rate in terms of true and false alarm rates [8]. In neural network, the weight, the associated bias and the number of epochs given for the training phase will determine the accuracy of the classification. Multilayer Perceptron (MLP) type of Neural Network is used for off-line analysis of data and also useful for intrusion detection [9].

In this paper, an Artificial Neural Network based IDS using MLP concept is suggested to detect the RPL attacks such as DIS attack and Version attack. The section 2 elucidates the basic concepts of Artificial Neural Networks and MLP. Section 3 explains some related works in this research. Section 4 proposes the ANN based IDS for IoT.

Revised Manuscript Received on September 07, 2019
***** Correspondence Author
A. Arul Anitha*, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli-620002, Tamilnadu, India,
Dr. L. Arockiam, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli-620002, Tamilnadu, India,

Section 5 explains the results obtained by the proposed model. Finally, conclusion is given in the section 6 and this section also opens new perspectives related to this research.

II. ARTIFICIAL NEURAL NETWORKS

A set of processing units also called as neurons are interconnected according to the specified topology is termed as an Artificial Neural Network (ANN). It has the ability to learn by example and generalizes from limited, noisy and incomplete data. ANN has been successfully employed in a broad spectrum of data-intensive applications [10-11]. The neural network consists of an input layer, number of hidden layers and an output layer. Each layer has number of neurons. The information enters the neural network via the input layer, it is processed in the hidden layers and the result can be retrieved in the output layer. A typical neural network model with a hidden layer is shown in Fig.1.

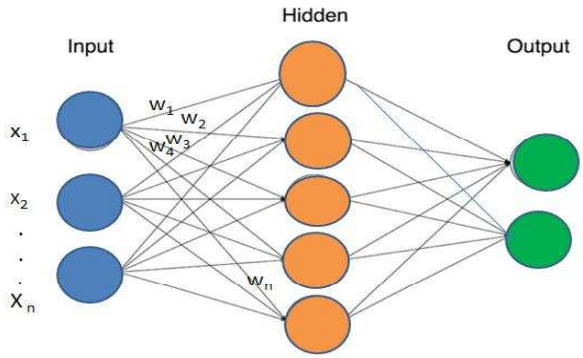


Fig.1. Neural Network Model

There are 'n' numbers of inputs available for a single neuron in this network and each input is associated with a weight on it. 'x_0' is the bias value which is added to the input of the activation function. Let x_1, x_2, x_3, ……..x_n are the inputs to a neuron and let w_1, w_2, w_3, …….w_n are weights, let 'b' is the bias and then 'a' is the out of the neuron which is calculated using the equation (1).

$$a = f(\sum_{i=0}^{n} \text{wixi} + \text{b}) \qquad (1)$$

where, f is the activation function which is used to get the output of that layer and feed it as an input to the next layer [12]. Artificial Neural Network is made up of nodes and corresponding weights which typically require learning based on the given patterns. Some examples of learning patterns include supervised learning and unsupervised learning. In supervised learning, the output has been labelled and so the network has a known expected answer. The Back-propagation algorithm and Multilayer Perceptron (MLP) belong to this category [13]. In unsupervised learning, the neural network analyses the input patterns and extract the features based on the characteristics of the given input. The Self-Organizing Map is an example of this unsupervised learning [14].

Multilayer Perceptron

Multilayer Perceptron is the widely used neural network model. It is based on supervised learning technique which uses the historical data as input to generate a labeled output. In Multilayer Perceptron, a set of input and its corresponding output are trained to learn the relationship between those input and the output. In the training phase, the parameters like weights and biases are adjusted, so that the error is minimized. This trained MLP model is used in the testing phase to classify the test dataset.

MLP is a feed forward neural network which involves forward and backward pass. The signal flow moves through the hidden layer from the input layer to the output layer. The result of the output layer is measured against the labels and the error is calculated. In order to minimize the error, the weights and bias are adjusted in the backward pass. The error is minimized in all iteration and finally it will be closer to the approximate output. Determining the optimal number of hidden layers and the hidden units in each layer is also challenging issue. It is difficult to determine the optimal hidden units than the hidden layer. Based on empirical method, the optimum number of hidden units suitable for the MLP can be assigned [15].

III. RELATED WORKS

Petteri et al. [16] performed a study on the requirement analysis of a benchmark dataset for Network and Host Intrusions Detection System (NHIDS). The requirements were finalized based on the dataset features, overall composition, and systems used to produce the datasets. Nine datasets starting from the traditional KDD CUP'99 dataset to UNSW-NB15 were reviewed. The coexistence of both Host-based and Network-based entities was rare in a single dataset. According to this study, the real-world network environment is difficult to replicate using the test-bed datasets.

Kelton et al. [17] reviewed various machine learning techniques suitable for intrusion detection in IoT environment. The recent research works related to IoT security were analysed with a special concern on the Intrusion Detection Systems using machine learning approaches. In this review the protocols, intelligent techniques like machine learning techniques and precision obtained in the recent works were highlighted. Finally, the research challenges and future directions for IoT security were also emphasised.

Ganesh et al. [18] proposed an approach for ANN based Intrusion Detection System with less number of features. Important features from KDD Cup'99 dataset were selected by using Mutual Information based feature selection method. The performance of Mutual Information with ANN was compared with Support Vector with ANN and Mutation Information approach outperformed without any false positive and less negative rates. Though there are many advantages in this method, it requires more computation in terms of number epochs to obtain the accuracy.

Mohammad et al. [19] assessed the challenges of IoT security by considering various machine learning techniques in smart cities. Taxonomy of machine learning algorithms and the issues and challenges regarding the data analytics of machine learning algorithms were

also discussed. They suggested some machine learning algorithms like ANN that are useful for the IoT security and fraud detection.

Alex et al. [20] suggested that the IDS to analyze the data packets and to detect malicious shell code. In their work, integer values were obtained by converting the byte level data retrieved from the data transmission of the nodes and fed into the ANN. Their best classifier identified 100% of malicious file contents in the test set. This ANN model is useful for detecting the script attack and SQL injection.

IV. PROPOSED ANNIDS MODEL FOR IoT

Multilayer Perceptron (MLP) is applied in the research work for detecting the attacks in IoT environment. The DIS attack and Version attack are simulated and the raw datasets are pre-processed to make them ready for detection process. The proposed ANN based IDS model for IoT environment is depicted in Fig.2. It has three phases such as simulation phase, pre-processing phase and ANNIDS model phase.

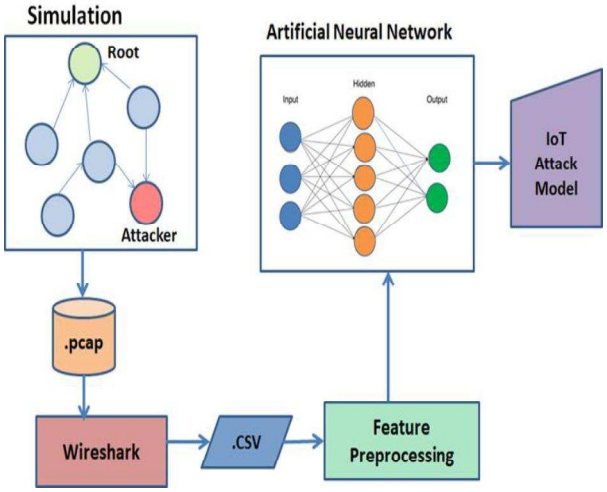


Fig.2. ANNIDS Methodology Diagram

1) **Simulation Phase:** In the simulation phase, the open source Contiki/Cooja simulator is used to generate the data packet details equivalent to the real-time data packets. At first, the packet capture file (.pcap) generated by the Cooja simulator is transformed into the CSV file format for further processing.
2) **Pre-processing Phase:** In this phase, the CSV files will undergo the pre-processing stages like feature extraction and normalization. After this step, the dataset will be ready for the ANNIDS phase.
3) **ANNIDS Phase:** In this phase, the pre-processed datasets are produced which consists of a mixture of normal packets and attack data packets. Then the dataset are fed into Artificial Neural Network system. The input, weight and bias values are adjusted and the IoT Attack Detection Model based on MLP is created.

The flowchart in Fig.3 explains the overall functionality of the proposed technique.

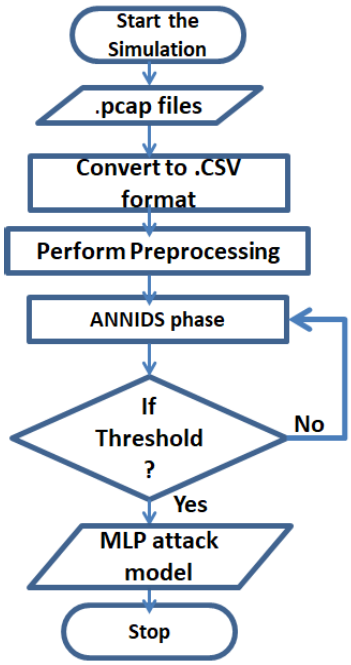


Fig.3. Flowchart for ANNIDS

IoT devices are resource constrained, so that in the proposed work minimum burden is given to the IoT network. The packet traces are only taken from IoT simulated environment. The proposed pseudo code for the ANNIDS technique is given as follows:

Procedure ANNIDS

Input: IoT Simulation Dataset
Output: MLP Model for detecting Attacks

Step 1: Start
Step 2: Perform Simulation with server node, normal nodes and attacker nodes
Step 3: Capture the .pcap file of the simulation using 6LoWPAN packet analyzer
Step 4: Open the .pcap files in Wireshark
Step 5: Import the .pcap file into .csv
Step 6: Perform pre-processing
Step 7: Use Resampling technique to get training and test set
Step 8: Use ANN to classify the attacks and benign packets
Step 9: Generate MLP model for IoT attack
Step 10: Stop

This ANNIDS pseudo code is implemented to produce a MLP model for detecting the attacks in IoT environment.

V. RESULT AND DISCUSSION

This section shows the outcomes obtained from the ANNIDS model that is used for intrusion detection. The proposed model was implemented in Contiki O.S. Cooja Simulator 3.0. The Simulation parameters used in this research is given in the Table 1.
According to the given

parameters the simulation is performed.

Table- I: The Simulation Parameters

Simulation	Cooja Simulator 3.0
Mote Type	Sky mote
Root node (node no 1)	1
Child node (node no 2-13)	12
Malicious Node (node no 14,15)	2
Radio Medium	UDGM: Distance Loss
Transmission Range	50 m
Interface Range	100 m
Mote start delay	1000 ns
Random Seed	123,456
Positioning	Random Positioning

At the beginning of the simulation, the Destination Oriented Direct Acyclic Graph (DODAG) generated by the Cooja simulator is shown in Fig.4.

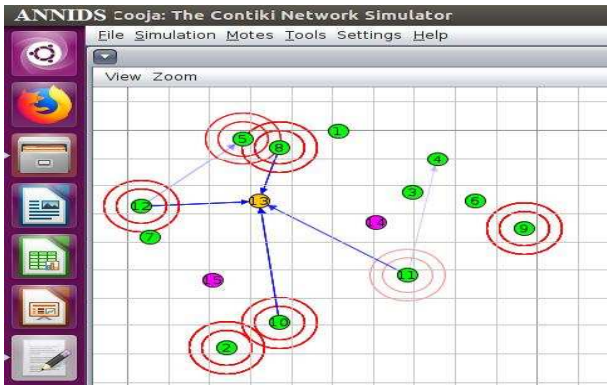


Fig.4. DODAG generated in Contiki O.S. Cooja Simulator

As it is given in the Table 1, there are 12 child nodes, one root node and two attackers used in this simulation. The mote output generated by the ANNIDS simulation environment is shown in Fig.5.

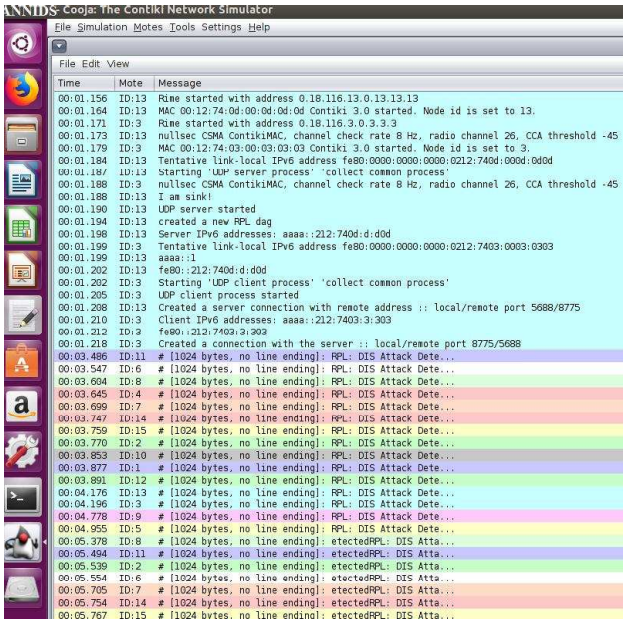


Fig.5. Mote output having attacks

In the second phase of the proposed system, the data packets generated by the Cooja simulator are captured using Wireshark tool as a .pcap file. Fig.6 shows the .pcap file captured by the simulator and also the I/O graph of this .pcap file. It contains the details of Packet_No, Time, Source_IP, destination_IP, Protocol, Length and other details about the simulated packets.

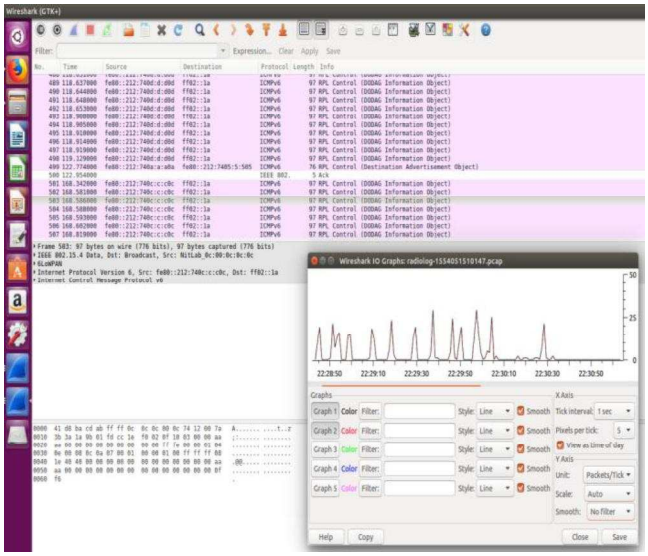


Fig.6. pcap file and I/O graph

The radio message generated by the two attacker nodes 14 and 15 are shown in the Fig.7. Next, the .pcap files are converted into .csv file for analyzing the data. Then the .csv file is fed into the Artificial Neural Network to generate the IoT Attack model.

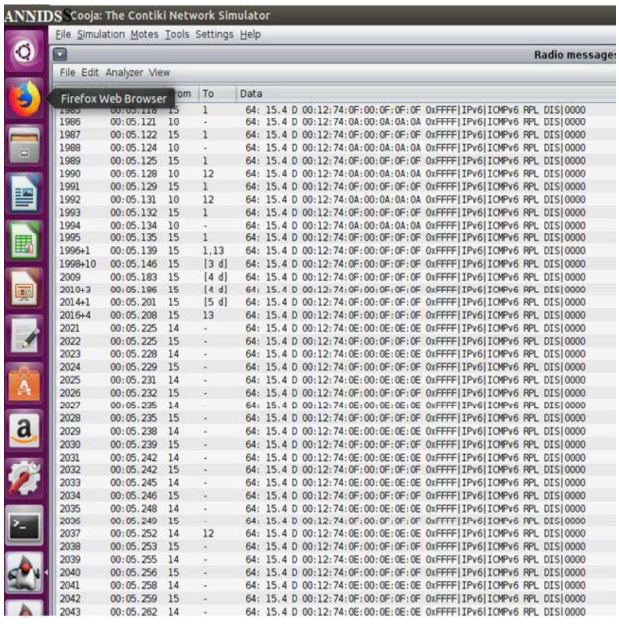


Fig.7. Radio messages Generated by the attacker nodes

Totally 73,880 data packets were captured in the IoT simulation experiment and among them attacker node 14 generated 325 malicious data packets whereas node 15 generated 982 malicious data packets. So, there are 1307

harmful data packets were initiated in this simulation and the observations are depicted in the Fig.8.

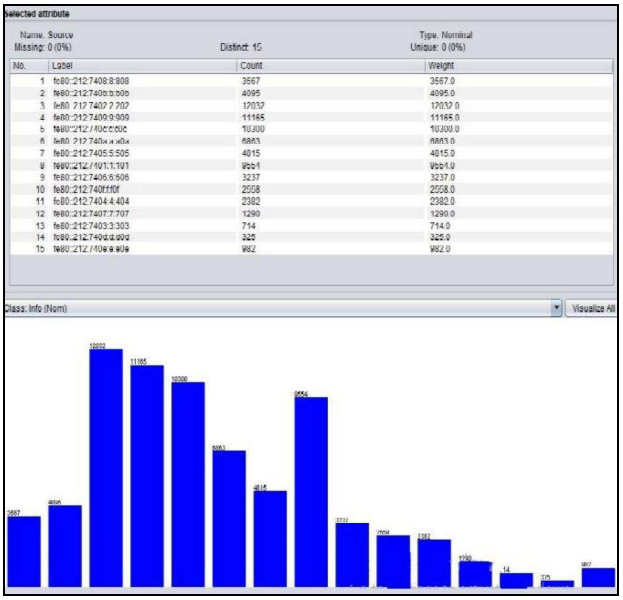


Fig.8. Data packets sent by all nodes

The pre-processed data was fed into the Multilayer Perceptron (MLP) Neural Network which has five layers including three hidden layers, input layer and output layer. This MLP representation is depicted in Fig.9.

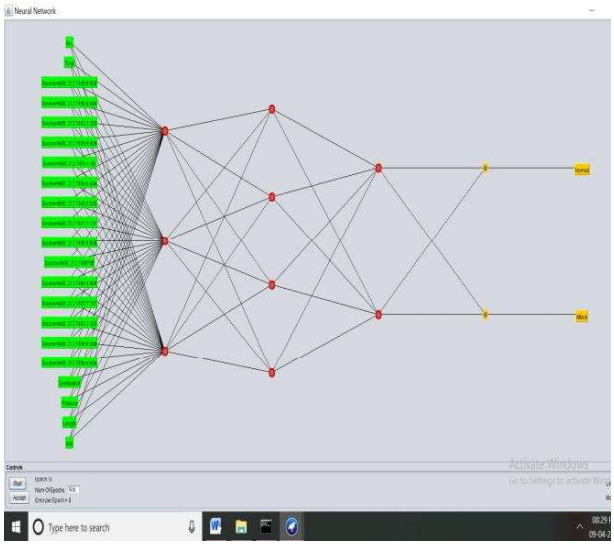


Fig.9. MLP Model for IoT Attack

Since MLP is a supervised Learning technique, it has training as well as testing phases. From the total dataset, 80% of data packets (59,104) are used to train the data model and 20% data packets (14,776) are used in the testing phase. The MLP model classified the data packets like attacks and normal packets according to the IPV6 source of the corresponding packets. Among the 14,776 data packets of the testing dataset, there are 260 attack packets and 14516 normal packets. The confusion matrix and other measures obtained during the testing phase are shown in the Fig.10.

```
=== Re-evaluation on test set ===

User supplied test set
Relation:     mycsvfiles-weka.filters.unsupervised.instance.Resample-S1-Z60.0-no-replacement-V-
weka.filters.unsupervised.instance.Resample-S1-Z50.0-no-replacement
Instances:     unknown (yet). Reading incrementally
Attributes:   8

=== Summary ===

Correctly Classified Instances       14776               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0002
Root mean squared error                  0.0003
Total Number of Instances            14776

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Normal
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Attack
Weighted Avg.  1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

     a     b   <-- classified as
 14516     0 |     a = Normal
     0   260 |     b = Attack
```

Fig.10. Classification of Attacks using test data

The best trained model correctly classified the malicious packets and normal packets in the test dataset. It has mean absolute error as 0.0002 and Root Mean Square error as 0.0003 which is very less. The True Positive Rate, Precision, Recall and F-Measure values are having the maximum values in this experiment.

VI. CONCLUSION

In this paper, an ANN based IDS is proposed to detect two RPL attacks such as DIS attack and Version attack. The simulated data is captured as a .pcap file and it is sent to the feature pre-processing phase and finally it is fed into the Multilayer Perceptron (MLP) to generate an IoT attack model. The proposed ANNIDS technique can be implemented with the Intrusion Detection System to enhance its performance. In future, instead of the simulated dataset, the real-time sensor data from smart city application can be captured and perform neural network based data analytics to detect the security attacks of the Internet of Things.

REFERENCES

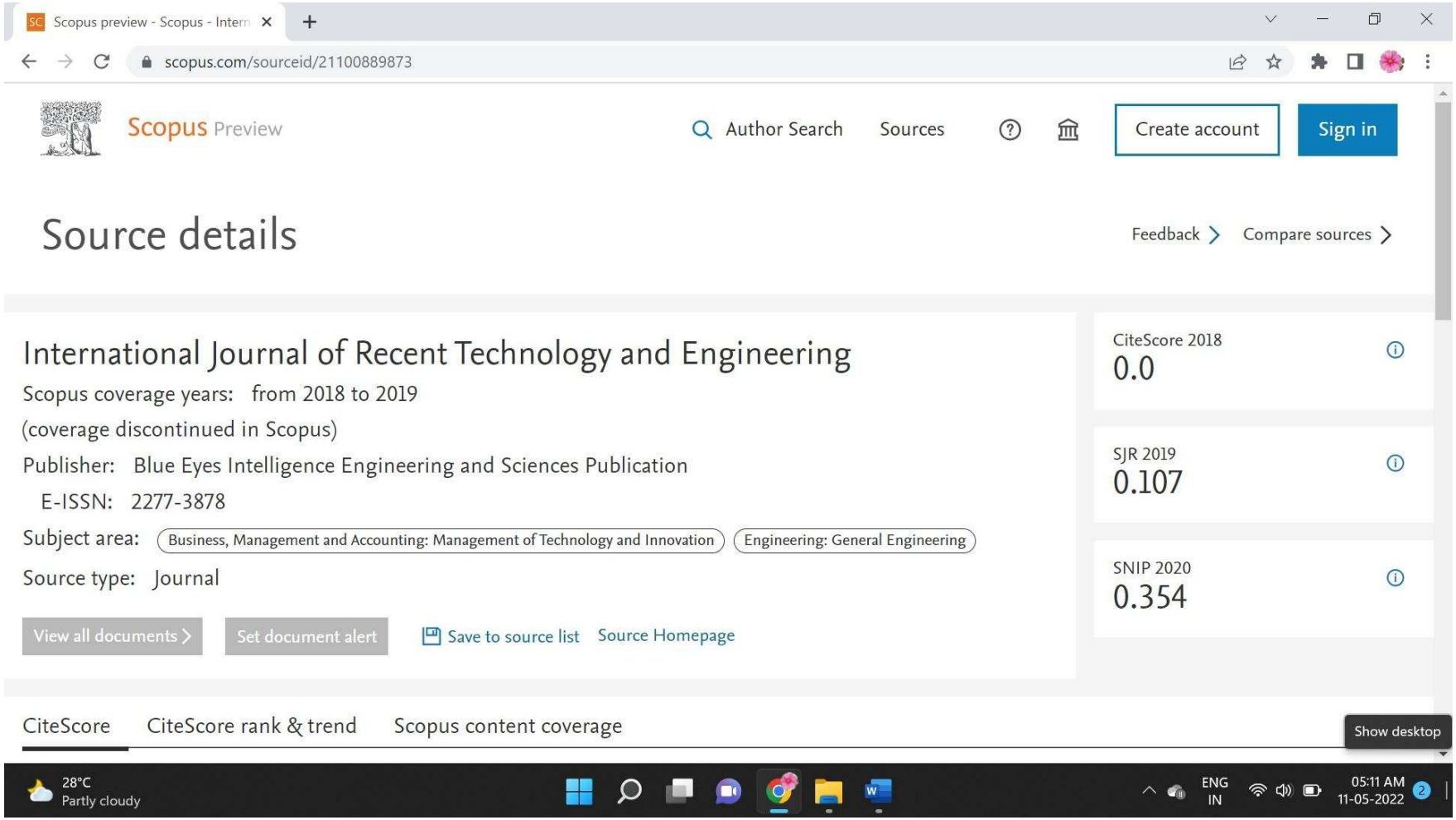
1. Bruno Bogaz Zarpaelo, Rodrigo Sanches Miani, Claudio Toshio Kawakani and Sean Carlisto de Alverenga (2017) A Survey of Intrusion Detection in Internet of Things. International Journal of Network and Computer Applications, http://dx.doi.org/10.1016/j.jnca.2017. 02. 009.
2. RaviTeja Gaddam and Nandhini (2018) An Analystical Approach to enhance the Intrusion Detection in Internet of Things Network. International Journal of Latest Trends in Engineering and Technology, Volume 9, Issue 3, pp.258-267, e-ISSN: 2278-621X, DOI: http://dx.doi.org/10.21172/1.93.43.
3. Tariqahmad Sherasiya and Hardik (2016) Intrusion Detection for Internet of Things. International Journal of Advance Research and Innovative Ideas in Education, Volume 2, Issue 3, ISSN: 2395-4396
4. A. Arul Anitha (2011) Network Security using Linux Intrusion Detection System. International Journal of Research in Computer Science, 2 (1): pp. 33-38, doi:10.7815/ijorcs.21.2011.012.
5. Jyoti Deogirikar and Amarsinh Vidhate (2017) Security Attacks in IoT: A Survey. International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC.

6. Anthea Mayzaud, Remi Badonnel and Isabelle Chrisment (2016) A Taxonomy of Attacks in RPL-based Internet of Things. International Journal of Network Security, Volume 18, Issue 3, pp. 459-473.
7. Divya Sharma, Ishani Mishra and Dr. Sanjay Jain (2017) A Detailed Classification of Routing Attacks against RPL in Internet of Things. International Journal of Advanced Research, Ideas and Innovations in Technology, Volume 3, Issue 1, ISSN: 2454-132X.
8. Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis and Robert Atkinson (2016) Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System. International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6.
9. Teik-Toe Teoh, Yok-Yen Nguwi, Yuval Elovici, Wai-Loong Ng and Soon-Yao Thiang (2018) Analyst Intuition Inspired Neural Network Based Cyber Security Anomaly Detection. International Journal of Innovative Computing, Information and Control, Volume 14, Issue 1, pp. 379–386.
10. Omar Y. Al-Jarrah , Yousof Al-Hammdi, Paul D. Yoo , Sami Muhaidat and Mahmoud Al-Qutayri (2018) Semi-supervised Multi-layered Clustering Model for Intrusion Detection. Journal of Digital Communications and Networks, Volume 4, pp.277-286, DoI: https://doi.org/10.1016/j.dcan.2017.09.009.
11. Seungwon Lee, Changbae Mun and Ook Lee (2018) A Study of Neural Network Based IoT Device Information Security System. Journal of Theoretical and applied Information Technology, Volume 96, Issue 22, ISSN: 1992-8645, E-ISSN: 1817-3195.
12. Yuntian Chen, Haibin Chang, Jin Meng and Dongxiao Zhang (2019) Ensemble Neural Networks (ENN): A gradient-free stochastic method. Neural Network Journal, Volume 110, pp. 170-185.
13. Dongwoo Lee , Sybil Derrible and Francisco Camara Pereira (2018) Comparison of Four Types of Artificial Neural Network and a Multinomial Logit Model for Travel Mode Choice Modeling. Journal of the Transportation Research Board, Volume 2672, Issue 49, pp. 101-112, DOI:10.1177/ 0361198118796971.
14. Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac and Parvez Faruki (2018) Network Intrusion Detection for IoT Security based on Learning Techniques. IEEE Communications Surveys and Tutorials, Volume. 00, Issue 0, DOI: 10.1109/COMST.2019.2896380.
15. McCarthy, R. V., McCarthy, M. M., Ceccucci, W., and Halawi, L. (2019) Predictive Models Using Neural Networks. Applying Predictive Analytics, 145–173, Springer Nature Switzerland AG 2019, doi:10.1007/978-3-030-14038-0_6
16. Petteri Nevavuori and Tero Kokkonen (2019) Requirements for Training and Evaluation Dataset of Network and Host Intrusion Detection System. Springer Nature Switzerland AG, WorldCIST'19, AISC 931, pp. 534–546, https://doi.org/10.1007/978-3-030-16184-2_51.
17. Kelton A.P. da Costa, Joao P. Papa, Celso O. Lisboa, Roberto Munoz and Victor Hugo C. de Albuquerque (2019) Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches. Computer Networks, PII: S1389-1286(18)30873-9, DOI: https://doi.org/,10.1016/j.comnet.2019.01.023.
18. P. Ganesh Kumar and D. Devaraj (2010) Intrusion Detection using Artificial Neural Network with Reduced Input Features. ICTACT Journal on Soft Computing, Volume 1, Issue 1, ISSN 2229-6956(Online), DOI: 10.21917/ijsc.2010.0005.
19. Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi and Amit P. Sheth (2018) Machine Learning for Internet of Things Data Analysis: A Survey. Journal of Digital Communication and Networks, Volume 4, pp.161-175, DOI:10.1109/COMST.2019. 2896380, IEEE: https://doi.org/10.1016 /j.dcan. 2017.10.002.
20. Alex Shenfield, David Day and Aladdin Ayesh (2018) Intelligent Intrusion Detection Systems using Artificial Neural Networks. ICT Express, Volume 4, pp.95-99, DOI: https://doi.org/10.1016/ j.icte.2018.04.003.

AUTHORS PROFILE

A. Arul Anitha is a Full-time Ph.D Research Scholar in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli which is affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India. She received her Master's degree in Computer Applications (MCA) from Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India and Bachelor of Science in Computer Science from Madurai Kamaraj University, Madurai, Tamilnadu, India. Her Research interest is on Network Security, Intrusion Detection Systems, Internet of Things (IoT) and Machine Learning. She has cleared the National Eligibility Test (NET) conducted by the National Testing Agency (NTA) in December, 2018.

Dr. L. Arockiam is working as an Associate Professor in the Department of Computer Science and Dean of Computing Sciences at St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 29 years of experience in Teaching and 21 years of experience in Research. He has Published 345 Research Articles in the International / National Journals and Conferences. He has guided more than 38 M. Phil Research Scholars and 29 Ph. D. Research Scholars and at preset he is guiding 6 Ph. D Research Scholars. His research interests are Internet of Things, Cloud Computing, Big Data, Data Mining, Software Measurement, Cognitive Aspects in Programming, Web Service and Mobile Networks.


Scopus preview - Scopus - Intern
scopus.com/sourceid/21100889873
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
International Journal of Recent Technology and Engineering
Scopus coverage years: from 2018 to 2019
(coverage discontinued in Scopus)
Publisher: Blue Eyes Intelligence Engineering and Sciences Publication
E-ISSN: 2277-3878
Subject area: Business, Management and Accounting: Management of Technology and Innovation
Engineering: General Engineering
Source type: Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2018
0.0
SJR 2019
0.107
SNIP 2020
0.354
CiteScore
CiteScore rank & trend
Scopus content coverage
Show desktop
28°C
Partly cloudy
ENG
IN
05:11 AM
11-05-2022

Promoting a Clean and Hygienic Environment using IoT

A. Arul Anitha, L. Arockiam

Abstract*: Internet of Things (IoT) based smart devices are the core elements for any smart environment. The sensors and actuators make the life easier when they are connected to one another and to the Internet. The Smart city and 'Swach Bharath Abhiyan' projects introduced by the Government of India tried to promote clean and hygienic Environment. The constant growth of population, industrialization and urbanization increase the unorganized manner of dumping the solid waste in landfills. Smart waste management is the must in all countries due to the voluminous generation of solid waste. In this paper, a methodology for monitoring the dustbins in smart cities, household or organization is proposed. The dustbins are monitored very often to check the garbage level. Whenever the dustbins reach maximum level, alert will be sent to the corresponding authorities with the bin details to dispose the waste. Additionally, the gas sensors in the dustbins detect the bad smell and alert when it reaches the threshold level though the garbage level will not reach the dustbin's maximum capacity. The areas which require emptying the dustbins very often are also identified. Large-scale implementation of the system will promote a clean and hygienic environment.

***Keywords* : Alert, garbage, IoT, smart dustbins**

I. INTRODUCTION

Cleanliness is one of the important issues in the modern society. Solid waste is the sole factor which has negative impacts on the health and hygienic aspects of people and environment. Even though many efforts have been taken to handle the trash efficiently, it is a challenging dispute for all countries. The Internet of Things (IoT) is a boon to solve this ever growing problem. To make a clean atmosphere, IoT based automated process in waste management is necessary [1]. IoT makes the real-world objects to communicate each other and also connect to the global network using various protocols and standards [2].

Whenever there is any need for disposing the trash in the dustbins, the notification will be given to the corresponding authorities. The rising population, continuous growth of industrialization and urbanization have led the country like India towards voluminous generation of garbage and polluted environment. Overflowing landfills due to the unorganized manner of dumping of waste in organizations and cities will bring serious environmental consequences [3]. Waste can be a precious asset when it is properly treated and reused. For the effective management of garbage, households and industries have to manage the waste by regularize the waste monitoring process [4]. This paper suggests a methodology for a simple and easy to use garbage monitoring system which monitors the dustbins of an organization and gives alert notifications to the waste management department to take necessary actions to dispose the waste.

The rest of this paper is organized as follows: Section 2 reviews some related works in smart garbage management. Section 3 highlights the scope and objectives of the projects. Section 4 describes the proposed methodology for garbage management system. Section 5 describes the system implementation details. Section 6 explains the results obtained by the proposed model. Finally, conclusion is given in the section 7 and this section also opens new perspectives related to this research.

II. RELATED WORKS

To understand the International and National issues related to waste management system, the related recent works for waste management system using IoT were studied and analysed. Padma et al [5] recommended a smart waste management system to notify the waste level in the dumpster. This system was composed of a microcontroller and GPRS module. The dumpster's status will be notified regularly using the sensors in the system. The sensor data is also stored in the database for future analysis.

Hassan et al [6] analysed the Solid Waste for Quweisna Industrial Zone of Egypt. According to this paper, every year the Quweisna Industrial Zone generates 170.446 ton of wastes, 32.39 ton wastes are only recovered and 81% of wastes are sent to the local municipality. To transform the zone into a green and clean area, the authors suggested recycling, recovery and reuse strategies.

A modeling study for waste management system was performed by Josiane et al [7] to evaluate the waste collection process of Itajubá, State of Minas Gerais, Brazil. Data were collected via observation, interviews and questionnaires. Simulation and modeling techniques were applied in this research. To improve the waste collection process, some operational ideas were suggested.

Revised Manuscript Received on January 15, 2020.
***** Correspondence Author
A. Arul Anitha*, Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Affiliated to Bharathidasan Univiersity, Tiruchirappalli, Tamilnadu, India. Email: arulanita@gmail.com
Dr. L. Arockiam, Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Affiliated to Bharathidasan Univiersity, Tiruchirappalli, Tamilnadu, India. Email: larockiam@yahoo.co.in

Andreasi et al [8] accomplished a comparative analysis on solid household waste and its impact on environment in seven European countries such as Germany, Denmark, France, UK, Italy, Poland and Greece. The authors considered those countries to represent the whole European Union. The collection, separation, treatment and disposal process as the waste management in this research. All countries need to update their technology periodically to meet the current challenges in the waste management process.

Shilan et al [9] from Iraq developed a smart solid waste monitoring and collection system. Ultrasonic Sensor Arduino Uno and Radio Frequency (RF) transmitter were installed on the top of the waste box for the monitoring task. A message (SMS) will be sent to the mobile phone of the truck driver about the location and ID of the dustbin whenever the waste box is full and needs for disposing the garbage.

An automatic waste segregator system was developed by which Kesthara et al [10] to separate the garbage into metal, dry and wet waste. Moisture and IR sensors were used to distinguish the dry and wet waste. The authors explained only theoretical aspects of the system. A smart waste separator was suggested by Aahash et al [11] in which Ultrasonic and Metal sensors were used to classify the waste into metallic and non-metallic wastes. The system was explained by using a block diagram and there was no experiment in their proposed system.

A garbage segregator system was implemented by Balagugan et al [12] to classify the waste at household level. PIC16F877 microcontroller was used in this segregator to control the entire process. An IR (Infrared) sensor, a moisture sensor and a metal sensor were used to detect and identify various types of waste respectively. The authors used Proteus tool to simulate their research idea to categorize the metallic and non-metallic waste efficiently. There is no real garbage segregation in their work.

To identify the metallic waste, parallel resonant impedance sensing mechanism was used by Amrutha et al [13] in an automatic garbage management system. This system can categorize one waste at a time and also it cannot segregate ceramic into dry waste. Mahajan et al [14] suggested a waste management system for municipality. In this work, to monitor the garbage level, the public dustbins would be provided with embedded device. Load sensors were used to increase the efficiency of the level of garbage whereas moisture sensor was used to segregate the wastes.

III. OBJECTIVES AND SCOPE

The primary aim of this project is to promote a clean and hygienic environment by developing and automating a cheap, easy to use garbage management system to monitor and to treat the trash at household or institutions. The sub objectives of this system are:

- To monitor the dustbins periodically.
- To give the alert to the person in-charge whenever the dustbins are full or the gas level in bins increased.
- To automate the waste management process efficiently.
- To complain the administrator of the system whenever the waste management process is not done properly.

The scope of the project is defined that it can be implemented only in household, organizations and cities where technology like internet connectivity and other technology are available. Remote areas are out of the bound as technical feasibility is not possible. Skilled and trained people are required to control, monitor and manage the whole system.

IV. PROPOSED METHODOLOGY

This project deals with the waste management systems in household or institution level through which it can promote clean city. The Methodology followed in this smart waste management project is explained by using the Fig.1.

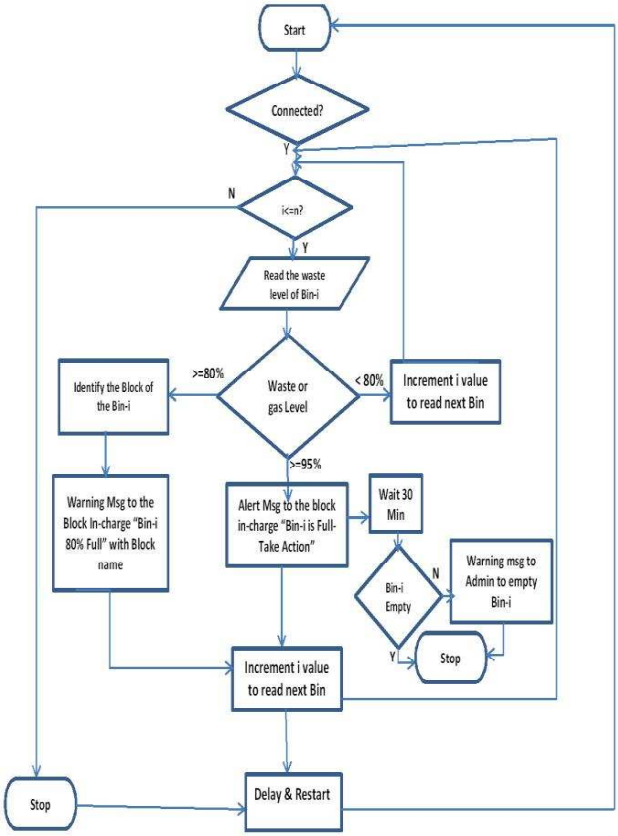


Fig.1. Flow Diagram for Garbage Management System

As it is given in the flow diagram, the dustbins in the particular area are monitored periodically. First, the connectivity is checked and the problems are rectified. Then the garbage level and bad smell level of all the dustbins are monitored, if it exceeds the threshold level, then the block in-charge who manages the particular area will be notified to take necessary actions with the block name and bin-id of the dustbin. Whenever the block in-charge receives notification, the message will be automatically forwarded to the personnel who are responsible for emptying the smart bin. After waiting for 30 minutes, again the status of the smart bin is monitored.

If the trash is not treated then the warning message will be sent to the cleanliness administrator of the system. After monitoring all the dustbins likewise, some delay will be given and again the system will restart its operation. The pseudo

code for the functionalities of the smart garbage management system is given below:

Pseudo Code for Smart Garbage Management System

Input: garbage_level, gas_level, Bin_id
Output: Alert Message with Bin_id

```
1: Start
2: Initialize no_of_dustbins, garbage_threshold,
   gas_threshold, delay
3: if (devices_not_connected==true) then
4:   rectify the Connectivity issues
5: else
6:  for i=1 to no_dustbins do
7:       get the garabge_level of Bin_i
        // get the value from Ultrasonic Sensor
8:       get the gas_level of Bin_i
         // get the value from gas Sensor
9:       if (garbage_level> =garbage_threshold  or
          gas_level>=gas_threshold) then
10:        get the Block name of the Bin_i
11.        send alert to Block In-charge with Bin-Id
12.        // wait for 30 min. and check the status
13.      if (Bin_i not empty)
14.        send the warning to Admin.
12:      else
13:  end for
14: wait for the delay go to step 3
      //after some delay restart the process
15. Stop
```

This pseudo code clearly explains various functionalities of Smart Garbage Monitoring System. Careful deployment of the system in a large scale will promote clean environment..

V. SYSTEM IMPLEMENTATION

The project was developed by keeping in mind of the need for smart garbage management system in St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India. The college campus is divided into number blocks based on the location. Each block is given a unique name and is under the control of a block in-charge and maintained by a small group of personnel. There are 'n' numbers of dustbins in each block. Each dustbin has a unique ID along with the block name for the smooth functioning of the system.

The dustbins are attached with HC04 ultrasonic sensor, MQ2 gas sensor and SG90 TowerPro Servo Motor. The Ultrasonic sensor is used to indicate the garbage level in the dustbin; MQ2 will send the presence of bad smell and level of gases and servomotor is used for the automatic movement of the dustbin's lid. The entire system is under the direct control of the Cleanliness Administrator. The system design for implementing the project is explained in Fig. 2.

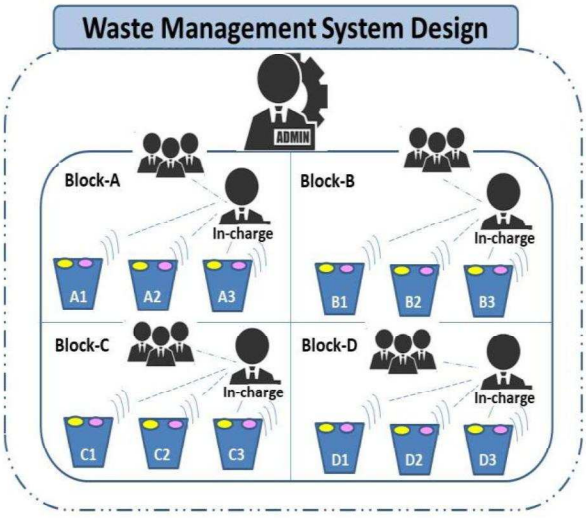


Fig.2. Waste Management System Design

When the garbage level exceeds the threshold, the ultrasonic sensor will send an alert to the block in-charge along with the bin-id. Once the message is received, automatically a message is sent to the mobile of personnel who is responsible for emptying the dustbin. The system will monitor the current status and wait some time (30 minutes), if the dust bin is emptied during the waiting period, the warning message is sent to the corresponding block in-charge and also the administrator to take immediate action. The smartbins used in this project is given in the Fig. 3.

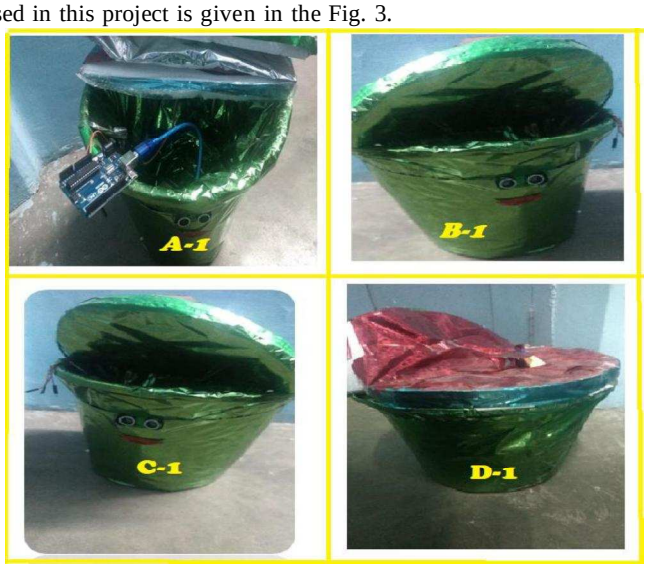


Fig.3. Dustbins used for this project

VI. RESULT AND DISCUSSION

The bins are placed in different locations. They sent the garbage level in the dustbins and gas level periodically to the block in-charge and administrator of the cleanliness department. The the output generated by the serial monitor is given in Fig. 4.

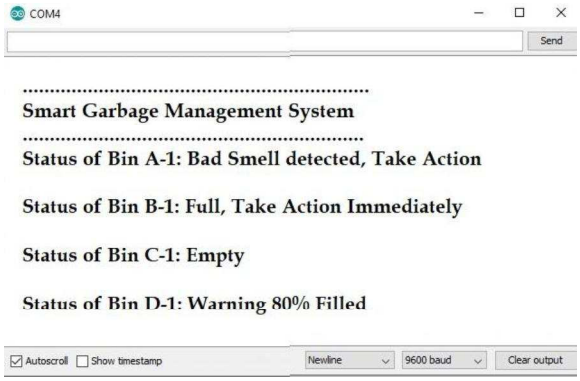


Fig.4. Output of the System in Serial Monitor

The various alert messages sent by the system according to the status of the trash bins are shown in the Table-I.

Table-I: Conditions and Corresponding Messages

S.NO.	CONDITION	MESSAGE	TO WHOM
1.	Garbage level = 0%	Bin is empty	Block in-charge
2.	Garbage level reaches 80%	Warning… 80% Filled	Block in-charge, Staff
3.	Garbage level >=95%	Bin is full. Take Action	Block in-charge, Staff, Admin
4.	30 Minutes after filled	Complaint	Admin
5.	Gas Level < 300	Nil	Nil
6.	Gas Level>=300	Bad Smell detected… Take Action	Block in-charge, Staff, Admin

The conditions and status of the dustbins are monitored. According to the current status of the bins like gaslevel and garbage level of the particular smartbins, actions are taken place. The notifications sent by the smartbins to the authorities are shown in Fig.5.

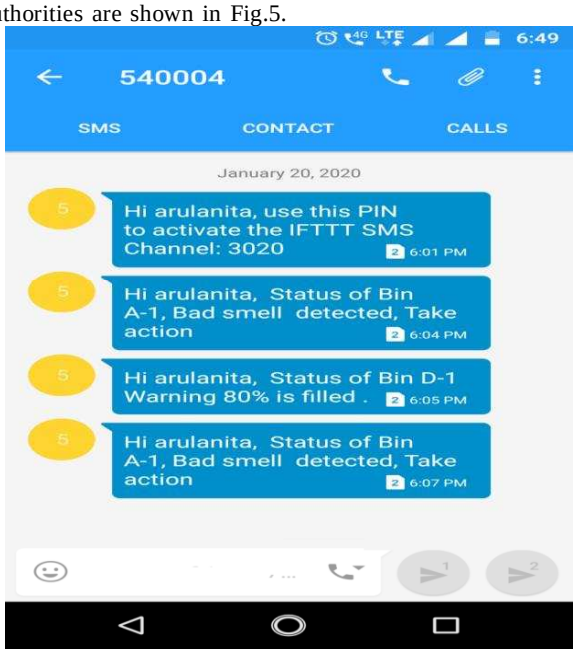


Fig.5. Notification to authorities

Actions are taken place as per the alert messages sent by the system. The located areas of the dustbins and their frequency level to empty the dustbins are listed below in Table II.

Table-II: Dustbin's area and Frequency

S.No.	Dustbin's area	Frequency
1.	Canteen	High
2.	Toilets	High
3.	Hostels	High
4.	Administrative Block	Medium
5.	Staff Room	Medium
6.	Classrooms	Low
7.	Library	Low

The above table shows the dustbins in canteen, toilets and hostel fill very often and require emptying them frequently. The garbage collection in classrooms and library are low. It is suggested to place more dustbins in the areas where the frequency is high. The system involves number of sensors and other hardware components. Implementing the system on a broad level will require lot of technical and financial investment.

VII. CONCLUSION

The project was designed to improve the waste management at an organization level. The technical aspects and constraints related to the project have been analyzed before developing the project. Large-scale implementation of the project will reduce the service cost associated with the Waste Management System significantly. It is recommended to include surveillance camera to monitor the improper usage of the garbage bins. In future, the project can be enhanced by implementing data security and device security at the edge level. The stored sensor data can also be analyzed for decision making.

ACKNOWLEDGMENT

A. Arul Anitha thanks the Management of St. Joseph's College (Autonomous), Tiruchirappalli, Tamilnadu, India for providing necessary facilities and financial support to carry out this research work in the college premises.

REFERENCES

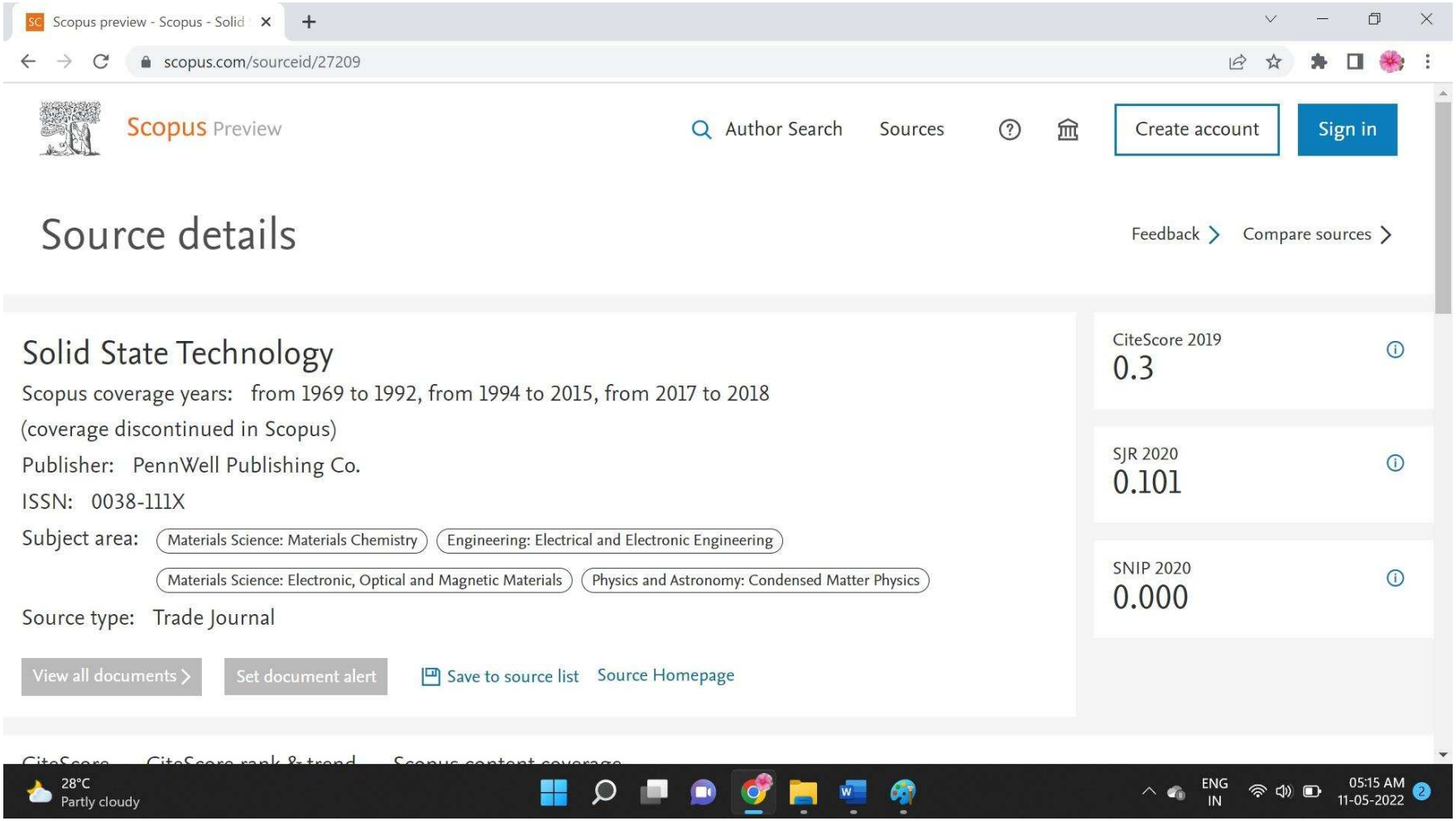
1. Fetulhak Abdurahman, Sileshi Aweke and Chera Assefa, "Automated Garbage Monitoring System using Arduino", IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 20, Issue 1, Ver. I, 2018, PP 64-76.
2. A. Arul Anitha, A. Stephen and Dr. L. Arockaim, "A Hybrid Method for Smart Irrigation System", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878 (Online), Volume-8 Issue-3, 2019, PP 2995-2998.
3. Nisarga T D, Sahana S, Saket Parashar, Suhas R, Shilpa R and Girijamaba D L, "Waste bin Monitoring System using Integrated Technology and IoT", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, 2018.
4. Patric Marques, Diogo Manfroi, Eduardo Deitos, Jonatan Cegoni, Rodrigo Castilhos, Juergen Rochol, Edison Pignaton and Rafael Kunst, "An IoT-based smart cities infrastructure architecture applied to a waste management scenario", Ad Hoc Networks, Issue 87, 2019, pp 200–208.

5. Padma Nyoman Crisnapat, Komang Agus Ady Aryanto, Made Satria Wibawa, Nyoman Kusuma Wardana, Dedy Panji Agustino, Arkav Juliandri, Ann Margareth, Ricky Aurelius Nurtanto Diaz, Naser Jawas and Made Sarjana, "STTS: IoT-based Smart Trash Tracking System for Dumpsters Monitoring using Web Technology", Journal of Physics, IOP Publishing, 1175 (2019) 012089, doi:10.1088/1742-6596/1175/1/012089.
6. Hassan, Hala, A. El- Nadi, M. E. A., Nasr, N any, A. H Badawy and Nahla M, "Solid Waste Management for Quweisna Industrial Zone" Journal of Environmental Science, Article 4, Volume 43, Issue 2, 2018, Page 61-77, ISSN: 1110-0826, DOI: 10.21608 /JES. 2018.22772
7. Josiane Palma Lima, Kelly Carla Dias Lobato, Fabiano Leal and Renato da Silva Lima, "Urban Solid Waste Management by Process Mapping and Simulation", Pesquisa Operacional, Vol.35 no.1, April 2015, ISSN 0101-7438 online version ISSN 1678-5142, DOI: 10.1590/01017438.2015.035.01.0143
8. Andreasi Bassi, S., Christensen, T. H., & Damgaard, A. (2017). Environmental performance of household waste management in Europe-an example of 7 countries. Waste Management, 69, 545-557. DOI: 10.1016/j.wasman.2017.07.042.
9. Shilan Abdullah Hassan, Noor Ghazi M. Jameel and Boran Şekeroğlu, "Smart Solid Waste Monitoring and Collection System", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 10, October 2016, ISSN: 2277 128X.
10. Kesthara .V, Nissar Khan, Praveen .S.P, Mahesha .C and Murali, "Sensor Based Smart Dustbin for Waste Segregation and Status Alert", International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), Volume VII, Issue IV, April 2018 , ISSN 2278-2540.
11. G. Aahash, V. Ajay Prasath, D. Gopinath and M. Gunasekaran, "Automatic Waste Segregator using Arduino", International Journal of Engineering Research & Technology (IJERT), Special Issue, ICONNECT - 2k18 Conference Proceedings, Volume 6, Issue 07, ISSN: 2278-0181,
12. Balagugan, Raja S, Maheswaran T and Savitha S, "Implementation of Automated Waste Segregator at Household Level", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 6, Issue 10, October 2017, ISSN (Online): 2319-8753, ISSN (Print): 2347-6710, DOI:10.15680/IJIRSET.2017.0610181.
13. Amrutha Chandramohan, Joyal Mendonca, Nikhil Ravi Shankar, Nikhil U Baheti, Nitin Kumar Krishnan and Suma M S, "Automated Waste Segregator", Texas Instruments India Educators' Conference, 2014.
14. S.A. Mahajan, Akshay Kokane, Apoorva Shewale, Mrunaya Shinde and Shivani Ingale, "Smart Waste Management System using IoT", International Journal of Advanced Engineering Research and Science (IJAERS), Vol-4, Issue-4, Apr- 2017, ISSN: 2349-6495(Print) | 2456-1908(Online), doi.org/10.22161/ijaers.4.4.12.

AUTHORS PROFILE

A. Arul Anitha is a Full-time Ph.D Research Scholar in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli which is affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India. She received her Master's degree in Computer Applications (MCA) from Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India and Bachelor of Science in Computer Science from Madurai Kamaraj University, Madurai, Tamilnadu, India. Her Research interest is on Network Security, Intrusion Detection Systems, Internet of Things (IoT) and Machine Learning. She has cleared the National Eligibility Test (NET) conducted by the National Testing Agency (NTA) in December, 2018.

Dr. L. Arockiam is working as an Associate Professor in the Department of Computer Science and Dean of Computing Sciences at St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 31 years of experience in Teaching and 23 years of experience in Research. He has Published 345 Research Articles in the International / National Journals and Conferences. He has guided more than 39 M. Phil Research Scholars and 29 Ph. D. Research Scholars and at present he is guiding 6 Ph. D Research Scholars. His research interests are Internet of Things, Cloud Computing, Big Data, Data Mining, Software Measurement, Cognitive Aspects in Programming, Web Service and Mobile Networks.


Scopus preview - Scopus - Solid
scopus.com/sourceid/27209
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
Solid State Technology
Scopus coverage years: from 1969 to 1992, from 1994 to 2015, from 2017 to 2018
(coverage discontinued in Scopus)
Publisher: PennWell Publishing Co.
ISSN: 0038-111X
Subject area:
Materials Science: Materials Chemistry
Engineering: Electrical and Electronic Engineering
Materials Science: Electronic, Optical and Magnetic Materials
Physics and Astronomy: Condensed Matter Physics
Source type: Trade Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2019
0.3
SJR 2020
0.101
SNIP 2020
0.000
28°C
Partly cloudy
ENG
IN
05:15 AM
11-05-2022

Solid State Technology
Volume: 64 Issue: 2
Publication Year: 2021

VeNADet: Version Number Attack Detection for RPL based Internet of Things

[1]A. Arul Anitha, [2]Dr. L. Arockiam

[1]Research Scholar, [2]Associate Professor, Department of Computer Science
St. Joseph's College (Autonomous), Tiruchirappalli
(Affiliated to Bharathidasan University, Tiruchirappalli)
Tamilnadu, India
[1]*arulanita@gmail.com*, [2]*larockiam@yahoo.co.in*

***Abstract*—**The Internet of Things (IoT) is one of the latest technologies in the realm of innovation. Routing Protocol for Low Power and Lossy Networks (RPL) is a promising Protocol to facilitate routing in IoT. This protocol could be exposed to specific attack during the Destination Oriented Direct Acyclic Graph (DODAG) construction. The Version Number Attack is one of the attacks initiated by the global repair mechanism and increases the control traffic in the network which in turn affects the performance. Hence, it is a challenging issue for RPL security. In this paper a new attack detection mechanism VeNADet is proposed and implemented in Cooja Simulator. The outcomes of this research work illustrate that the proposed method detects the Version Number Attacks with a very high True Positive and it raises very low false alarm rate.

Keywords- *IoT; RPL; global repair; DODAG; Version Number Attack; VeNADet*

I. INTRODUCTION

Internet of Things (IoT) is a magical buzzword which incorporates technological and industrial updates and innovations in our daily lives. Though it makes our lives easier it also has many challenges while implementing this technology in large-scale. Security is one among the important issues of Internet of Things. With the growing deployment of IoT in various fields, the security attacks and challenges are also increasing rapidly [1]. The intelligently connected IoT devices gather data, achieve the desired status via manipulating actuators in any smart environment, and monitor the networks. These devices encounter severe problems since (i) most of these devices only have constraint resources in terms of energy, computation, and memory; and (ii) many of these devices are static or mobile that can only be accessed via lossy wireless links. Low-power and Lossy Networks (LLNs) are defined by using these characteristics [2].

Many IoT devices are included into the network without any fundamental security mechanisms. The special characteristics of IoT and the voluminous inclusion of such devices tend the IoT network be more vulnerable than the traditional network. Data confidentiality, integrity, authentication and privacy are the important security requirements of IoT. Cryptography is one of the security mechanisms which cannot deal with routing attack and internal attacks. Hence, there is a need for another layer of security and Intrusion Detection System can be the right choice in such cases [3]. The limited resources in IoT, makes the implementation of IDS in IoT as a challenging issue. Many researchers are focusing on the lightweight IDS for IoT and INIT [4], RIDES [5] and SEVLTE [6] are some of the important IDSs specifically developed for IoT. Though there are some initiatives in this area, still complete security to IoT environment is not

guaranteed yet. The security issues and challenges presage the IoT infrastructure to enhance the security measures in its environment.

According to the RPL protocol, the root node in the IoT network is connected to the Internet via the IPv6 Border Router (6BR). The root node initiates the construction of the Destination Oriented Directed Acyclic Graph (DODAG) a tree like topology by broadcasting DODAG Information Object (DIO) messages. To join the DODAG, the receiving nodes reply with a Destination Advertisement Object (DAO) to their parent nodes through which they received the DIO message. The parent nodes permit the child nodes to join the network by sending DIO ACK message. Each node in the DODAG has rank value which is determined by the rank value of its parent and other parameters like Hop Count, Energy and Estimation Transmission Count (ETX). New nodes can join the DODAG by sending the DODAG Information Solicitation (DIS) message [7]. The DODAG construction is explained using the Fig.1.

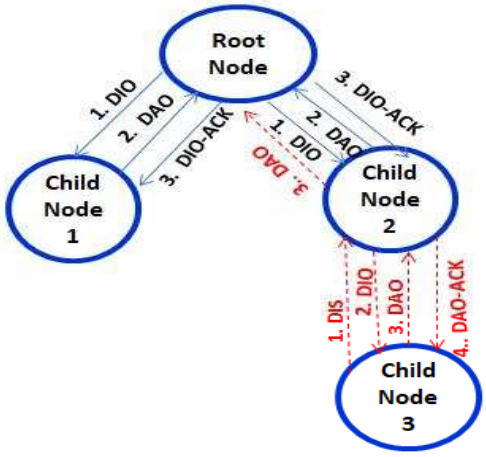


Figure 1. DODAG Construction Process

As it is given in the Figure 1, the DODAG construction is initiated by the root node by sending DIO messages to its neighbours, the nodes send a DAO message and again the sender of DIO message has to acknowledge by sending the DIO ACK message. A node which is not included in the DODAG can join the network by sending the DIS message to get the DIO message from a node which is already in the DODAG.

There are many security attacks in RPL during the DODAG construction. These attacks consume more resources, disrupt the topology and interrupt the network traffic of IoT. Version attack is one of the serious RPL attacks which devours more resources and degrades the performance of the DODAG [8]. The attacker frequently changes the version number which will lead to an enormous increase in delay and control overhead. As a result, the network lifetime and the packet delivery ratio are degraded. To overcome these issues, a version attack detection technique VeNADet is proposed.

This paper is organised as follows: Section II discusses the related works for Version Attack and the detection and mitigation techniques designed for safeguarding the IoT networks from Version Attacks, Section III explains the security issues related to Version Number Attacks. The proposed VeNADet technique is elaborated in Section IV. The outcome of the simulated environment is discussed in Section V and finally the conclusion and some future research directions are highlighted in Section VI.

II. RELATED WORKS

There are many investigations on IoT and many researchers have contributed for the security issues of IoT. In the Literature, only few studies have focused on the Version Number attacks and its impacts on the RPL networks. Arul Anitha et al. [9] proposed an Artificial Neural Network based Intrusion Detection

System for Internet of Things using Multilayer Perceptron (MLP) for detecting the Version Attacks and DIS attacks. These attacks were implemented in the Contiki Cooja Simulator and the data traffics were collected using packet analyser and performed the classification of attacks and normal packets using MLP. This method detects whether the packet is an attack or not. It doesn't classify the types of attacks.

To mitigate the Version Number Attacks (VNA), two lightweight techniques are proposed by Aris et al. [10]. In this lightweight approach, the elimination technique eliminates the VN updates initiated by the leaf nodes and the shield technique accepts the VN updates only when it comes from the majority of nodes having higher ranks. There were 36 nodes used in this work and the performance was analysed using different topologies. Only one attacker node was used in this research and mobility was not considered. It is not suitable for the heterogeneous environment. By implementing the one-way hash chain and signatures, Dvir et al. [11] suggested a technique named Version Number and Rank Authentication in RPL (VeRA) to secure the rank and version numbers from attackers. The performance of system in terms of memory, CPU, time and power consumption were not considered.

Mayzaud et al. [12] analyzed the impacts and consequences of version number attacks using static nodes in a grid topology. The performance results were provided in terms of control packet overhead, packet delivery ratio, average end-to-end delay, inconsistencies and loops. The important aspects of the constrained nodes like power consumption of nodes and its impacts are not considered in this study. It also does not fit for any probabilistic attacking model.

F. Ahmed et al. [13] suggested an algorithm for detection of version number attack using distributed and cooperative verification mechanism in IoT network which reduces the control packet overload significantly. In this approach, whenever a node receives the DIO message with higher DODAG version number, only after careful verification and analysis of its neighbours and their parents' behaviour within a two hop count range, it will update the version number and send the modified DIO message to its child node. Thus, reliability of the neighbours is ensured and the malicious version number update is prevented effectively.

Napiah et al. [14] implemented six machine learning algorithms to endorse an Intrusion Detection System called CHA-IDS. It detected the Hello Flooding, Sinkhole, and Wormhole attacks. The outcomes of CHA-IDS are evaluated using Cooja simulator which gives high detection rate and low energy and memory.

III. Version Number Attack and its Impacts

In RPL, the network layer takes all routing decisions. Therefore, it is essential for all the nodes including the root node to know about the topological information. Whenever there is an inconsistency arises in the DODAG, it should be instantly addressed. For this purpose, Version Number is used to identify a network topology uniquely within an RPL instance. All nodes in the network receive this version numbers when the version number of the topology will be changed. An attacker node can disrupt normal flow of network traffic by using a wrong version number which increases the security challenges in the network [15][16].

As the RPL specification indicates, the DODAG root node has a version number. RPL DODAG forms a tree structure and version number is the measure to make sure loop free paths to the root node and there is no irregularity in the DODAG. To maintain the DODAG integrity, the root node initiates a global repair. An attacker node may broadcast a false version number in its control message to force a global repair frequently without the prior knowledge of the root node [17] [18]. The Figure 2 and Table 1 explain the normal scenario where there is no Version Number Attack.

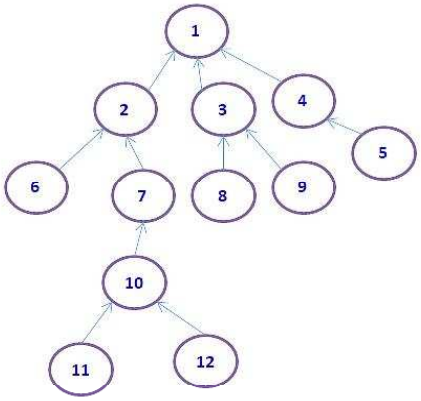


Figure 2. Normal Scenario without Version Attack

Table 1. DODAG Information of Figure 2

Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,7,8,9	2	1
10	3	1
11, 12	4	1

The hop count is used as rank metric for constructing the DODAG. The Version Number is set as 1. The details of this DODAG are given in Table 1. There is no irregularity and inconsistency in terms of loops or Version in the DODAG. Hence, it is a normal DODAG. Suppose, if the node '10' is a Version Attacker, it updates its version number from '1' to '2'and the DIO message of Node 10 contains the Version Number '2', instead of '1'. The updated details of the node 10 are shown in Figure 3 and Table 2.

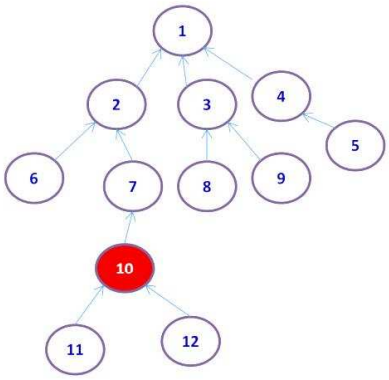


Figure 3.Version Attacker Node '10'

Table 2. Updated details of the Attacker

Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,7,8,9	2	1
10	**3**	**2**
11, 12	4	1

When the node '10' sends the updated Version using the DIO message to its neighbours '7', '11' and '12', they will also update their version Number and the nodes' details will be updated as it is given in the Fig.4 and the Table 3.

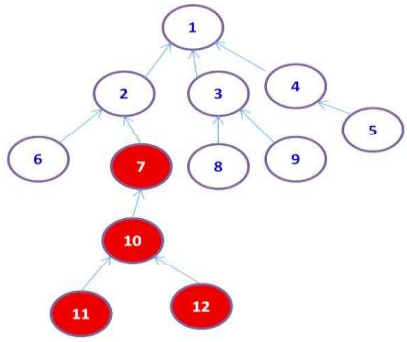


Table 3. Details of Attacker's neighbours

Node Id	Rank	Version No
1 (Root)	0	1
2, 3, 4	1	1
5,6,8,9	2	1
7	**2**	**2**
10	**3**	**2**
11, 12	**4**	**2**

Figure 4. Node 10 sends DIO Message to its neighbours

When this process is going on, the root node will also get the DIO message with an updated version number and this will lead to a global repair, so that the DODAG construction process is restarted by the root node. This situation is depicted using Figure 5 and Table 4.

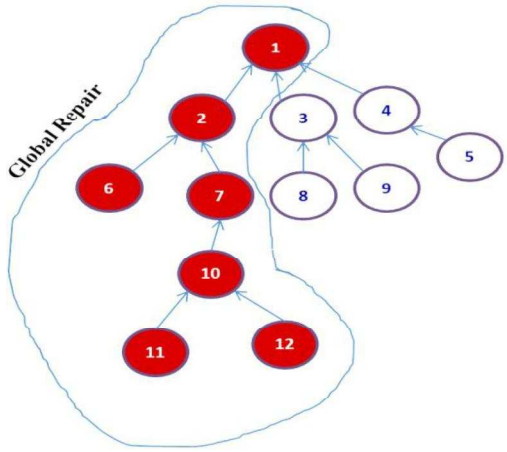


Table 4. Details of Attacker's neighbours

Node Id	Rank	Version No
1 (Root)	0	2
2	1	2
3, 4	1	1
5, 8,9	2	1
6, 7	2	2
10	3	2
11, 12	4	2

Figure 5. Global Repair Scenario

The Fig.5 denotes, there is a need for global repair and the root node starts the reconstruction process. Since the '10' is an attacker, it again and again updates its version number and it disrupts the DODAG topology frequently which will consume the resources indirectly.

Hence, the Packet Delivery Ratio, Energy and throughput are degraded. This attack can take place by changing the version number. It will lead the DODAG to reconstruct. As a result, the performance of the IoT network will be degraded due to the harmful effects of the constrained nodes [19][20]. Thus, the version number attacks can cause ruinous effects on node resources and network and also reduce application performances. The Denial of Service (DoS) attack caused by version number attacks will lead the devices to have a catastrophic effect in the IoT environment. To overcome the impacts of the Version Attack, the VeNADet technique is proposed in this work.

IV. VERSION NUMBER ATTACK DETECTION

An attacker illegitimately updates with a higher version number and initiates the nodes in a DODAG to increment their version number. The malicious node can create a similar situation by falsely advertising an incremented version number at regular intervals. To overcome this attack, the VeNADet technique is proposed.

Whenever there is a global repair, the Version number of the root 'r' is updated and a new Version Number initiated by the root. In the DODAG, the nodes communicate the root nodes via intermediate nodes and this will introduces additional computational complexity and communication overhead. Hence, whenever there is a reconstruction of DODAG, the version number of the root node 'r' will be incremented. The proposed methodology for the Version Number Attack Detection (VeNADet) is depicted in Figure 6. This methodology has three phases such as Checking phase, Validation phase and Mitigation phase.

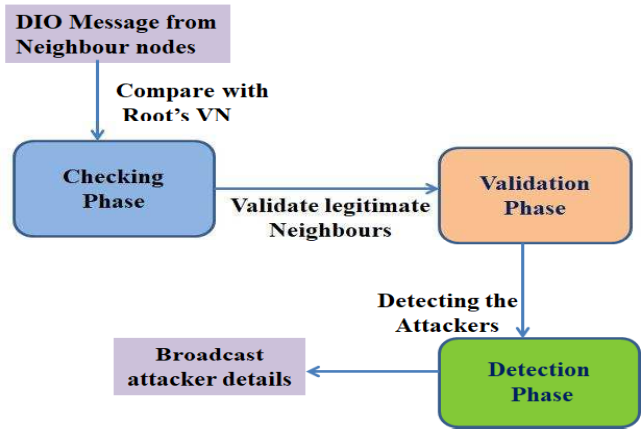


Figure 6. VeNADet Methodology

As the Figure 6 denotes, the Version Number of the DIO message received from the neighbour nodes that are compared with the root node, based on the result the neighbour node is validated as legitimate or not and then the attack is detected and the details of the attacker node is broadcasted to all the nodes in the DODAG tree. The three steps involved in the VeNADet are explained in detail as follows:

- ***Checking Phase*:** When a node 'y' receives a new DIO message from the neighbour 'x', if the version number of y and the version number of 'x' are same then the neighbour is considered as a normal node. If both are not equal then the version number of 'x' is compared with the version no of root node 'r', if they have same Version Number, then also the node 'x' is treated as the normal node and node 'y' will update its version number as the node 'y'. If both VN are different, then the Version number is sent to the detection phase for further analysis. The procedure for this phase is given in Figure 7.

Procedure Version_Check

1. r_{vn} ← version number of the root node r
2. y_{vn} ← version number of the DIO receiver node y
3. x_{vn} ← version number of the DIO sender node x
4. if ($y_{vn= =}x_{vn}$) then
5. print "DIO sender x is normal"
6. else if(x_{vn}==r_{vn}) then
7. print "DIO sender x is normal"
8. y_{vn}=x_{vn} // *receiver updates its version number*
8. else
9. print "DIO sender x is malicious"
10. procedure Detect_Attack
11. end if
12. end Version_Check

Figure 7. Procedure for Checking Phase

- ***Validation Phase*:** If there is any mismatch in the version number of the node x and r, then 'x' is considered as malicious node. In such case, this phase detects the source of the attack by analysing the version number of the neighbours who have greater than to its rank value. Let X= {x_1, x_2…….x_n} be the set of 'n' nodes located within one hop distance from the current node y. For any neighbour node x_i the version no is compared with the version number of the node x. If majority of the neighbours with high ranks having the same version no as the node x and root node r, then it is a legitimate node, and the node y updates its version number as it is given in the x_{vn}. Otherwise the

neighbour node x is treated as an attacker node and the node y retains its old version number y_{vn}. The procedure for this validation phase is given in the Figure 8.

Procedure Validate_Neighbour

```
1.  if (x is malicious) then  // DIO sender x is malicious
2.     X= {x1, x2…….xn} // neighbours of receiver y
3.     Initialize vncount ← 0
4.     for (i= 1 to n) do
5.          if (xi.rank > y.rank) then  // nodes having higher rank only
6.               if (xivn= = xvn) then
7.                    print "neighbour xi is normal"
8.                    vncount=vncount + 1
9.               else
10.                   print "neighbour xi is malicious"
11.                   vncount=vncount-1
12.              end if
13.         end if
14.    end for
15. end if
16. if (vncount> 0 and xvn=rvn) then
17.    update yvn=xvn  // update version number
18. end if
19. end Validate_Neighbour
```

Figure 8. Procedure for Validation Phase

- ***Detection Phase:*** If any malicious node is found, then the source of the malicious activity has to be identified. When an attacker is located as a leaf node, or any intermediate node in the DODAG, then using the above two methods the attacker can be easily detected and identified. When an attacker node 'a' acts as the neighbour node of the root node r, it frequently initiates the global repair and the version number is modified accordingly. In that condition, all the nodes in the DODAG also modifies the same when they receive the Version Number from the higher order nodes. It is a challenging task for the root node to detect the attacker.

When the neighbour node initiates the global repair frequently, the root node 'r' checks with other neighbour node whether it is mandatory for global repair process. For that, it uses a GR_{count} variable and during the DODAG construction, its value is set as 0. If there is any requirement of global repair for a node, the GR_{count} is incremented. If the GR_{count} reaches the threshold value (80%), then only the root node starts the global repair. Otherwise, it identifies the node 'a' is an attacker node and terminates the communication links to and from the node 'a'. It also broadcasts that the node 'a' is an attacker, so that other nodes also identify the attacker node. A node will update its version number if and only if 80% of its neighbours have the updated version number. Otherwise, the sender of the DIO message will be treated as an attacker node. The algorithm for this detection phase is given in Figure 9.

Procedure Detect_Attack

```
1.  K= {k1, k2…….kn} // nodes within one hop count
2.  Initialize GRcount ← 0
3.  d= DIO_timer delay
4.  t =n*.8     // 80% neighbour nodes
5.  T=N*.8     // 80% nodes in DODAG
6.     for (i= 1 to n) do
7.          if (ki.delay<d) then  // attacker initiate global repair often
8.               GRcount= GRcount-1
9.             broadcast "ki is malicious"
10.         else if(ki.delay=d and xvn=rvn) then
```

```
11.              GR_count= GR_count+1
12.    end for
13.    if (GR_count>=t) then// 80% neighbours have updated VN
14.         x_vn=y_vn
15.       end if
16.    if (GR_count>=T) then  // DODAG nodes require global repair
17.              broadcast "Initiate Global Repair"
18.              r_vn=r_vn + 1   //update the version Number of root
19.    end if
20. end Detect_Attack
```

Figure 9. Procedure for Detection Phase

V. SIMULATION RESULTS AND DISCUSSION

A. *Simulation Environment*: In this experiment, Cooja simulator of the Contiki Operating System is used to implement the Version Number attacks. The emulated Tmote sky sends the temperature data periodically. The simulation parameters used in this research are shown in Table 5.

Table 5. Simulation Parameters

No. of normal nodes	Maximum 50
No. of attacker nodes	10%
Mote Type	Tmote Sky
Operating System	Contiki 3.0.
Simulator	Contiki Cooja
Topology	Random
Radio Medium	Unit Disk Graph Medium: Distance Loss
Topology Dimension	150m x 150m
Transmission Range	50m
Interference Range	100m
Tx Ratio	100%
Rx Ratio	100%
Simulation Duration	30 minutes per simulation
Number of Simulation	5 per three different scenarios

Keeping in mind with some basic essential traits of IoT application, a topology is created. The performance metrics of IoT network such as control traffic overhead and packet delivery ratio are considered in this experiment. The performance of the simulation setup without any version attack and also with version attack was analysed. The random topology is used in simulation. So, the nodes are placed with random densities. All nodes send their data periodically to their root node. The temperature details sent by the Tmote sky were recorded and logged the details for 30 simulation minutes. The effects of version attack and how the version attack affects these metrics and the performance degradation were measured. Except the distance-based loss, there was no external interrupt in this environment.

B. Normal and Attacker Scenario

In the normal scenario, the simulation environment is setup with 10 normal nodes, 20 normal nodes, 30 normal nodes, 40 normal nodes and 50 normal nodes including a root node in each category which is given in Figure 10.

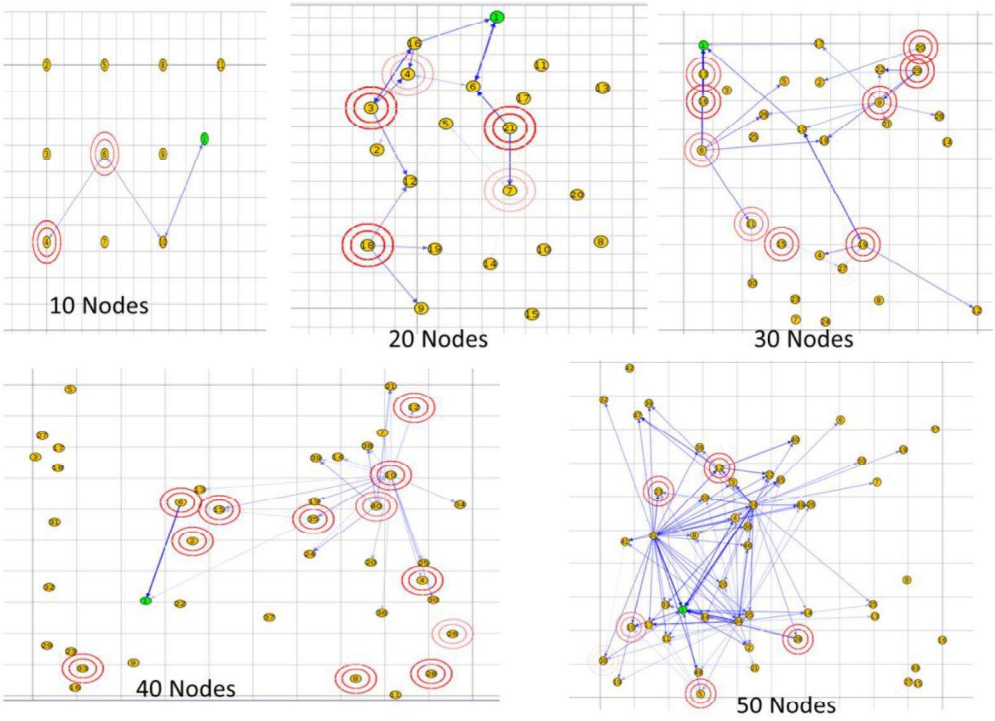


Figure 10. Normal Scenario

In the attacker scenario, 10% of attacker is included in all simulations. The attacker nodes are placed in different locations and act as a leaf node, an intermediate node and a neighbour of the root node. It initiates attacks after the network is stable and DODAG construction is over. The attacker node periodically updates the version number and sends DIO messages accordingly. As the normal scenario, in the attacker scenario also, the simulation environment is setup with 10 normal nodes, 20 normal nodes, 30 normal nodes, 40 normal nodes and 50 normal nodes including a root node and with 10% of attacker nodes in each category. The attacker scenario is shown in the Figure 11.

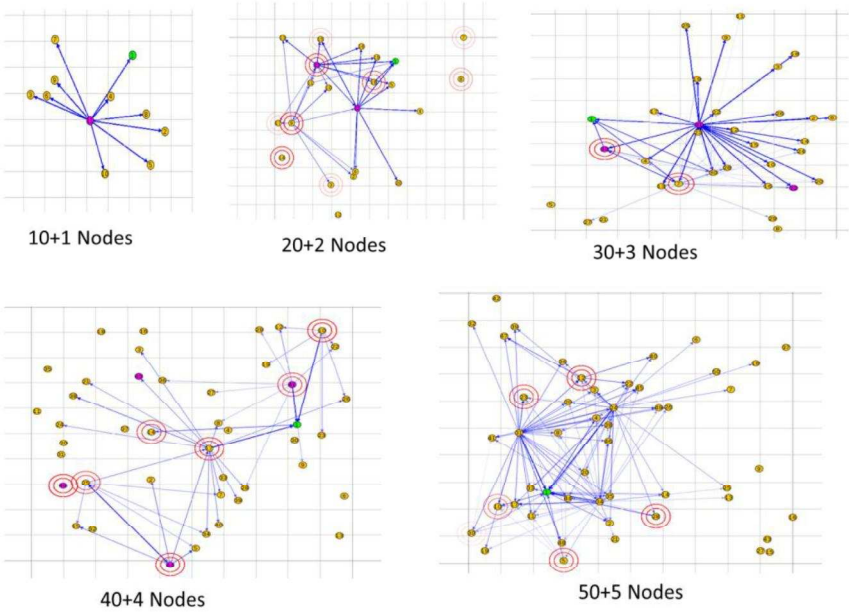


Figure 11. Simulation with 10% Attacker Nodes

The routing metrics used in this simulation are Packet Deliver Ratio and Control overhead. When packet delivery ratio (PDR) is more than 90% then the IoT network performs well. The PDR value obtained in the normal with 10% of attacker nodes are depicted by graphs and shown in Figure 12.

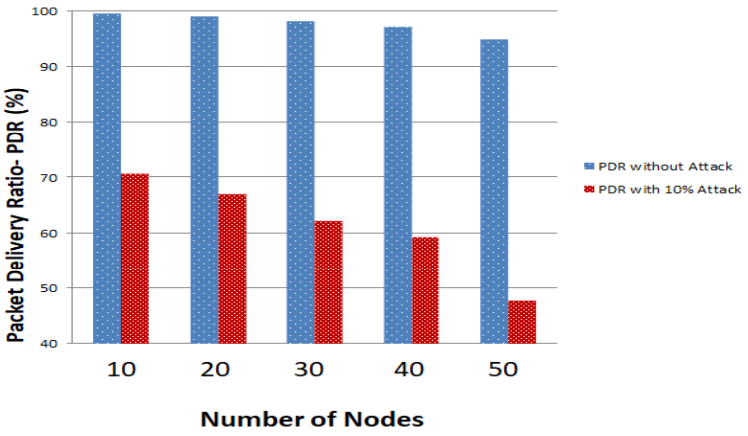


Figure 12. PDR value in normal and attacker scenario

The Figure 12 shows that the PDR value is high when there is no version attack. The PDR value is decreased drastically when there are 10% attacker nodes. Thus the performance of the network is degraded and this network consumes more energy and resources for the successful delivery of data packets. Control overhead is another routing metrics and when the number of control packets is increased, the network's performance is degraded. The number of control packets generated in the normal and 10% attacker simulation environment is portrayed in the Figure 13.

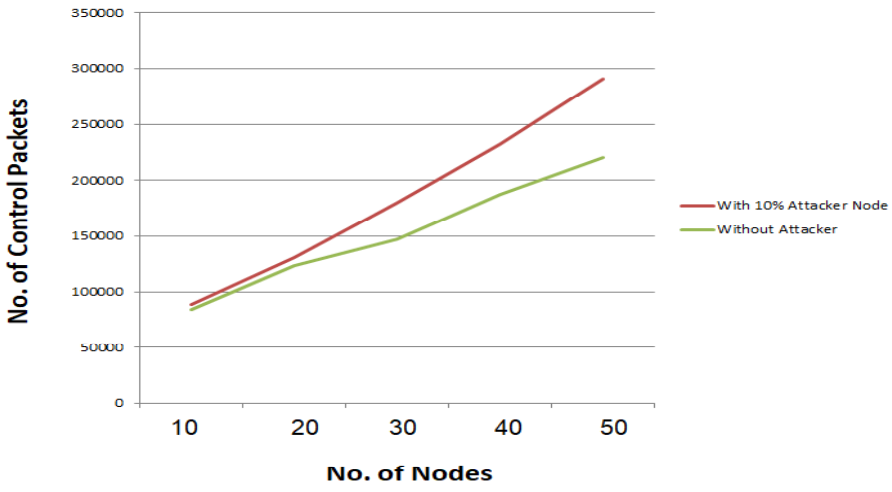


Figure 13. Control overhead in normal and attacker scenario

Comparing to the normal scenario, routing metrics in the attacker scenario are drastically degraded which will be worse when the attacker nodes are increased. The attacker affects the topology construction and due to the frequent update of the version number, control overhead is enormously increased which make nodes to consume more energy. To overcome this issue, it is necessary to protect the IoT environment with some detection and mitigation techniques.

C. Attacker with VeNADet Mechanism

The proposed VeNADet detection technique for version attack is implemented in the border router with the attack scenario and the detection accuracy of the technique and the routing metrics' performance are measured. The number of attacks initiated in different simulation environment without implementing the VeNADet technique is given in the Table 6.

Table 6. No. of Attacks initiated

Nodes	10% Attacker Nodes	No. of Attacks
10	1	113
20	2	257
30	3	413
40	4	728
50	5	1878

The proposed VeNADet detects the version attack when a single attacker exists. The attacker is identified and the details of the attacker node are broadcasted by the root node. When the number attackers increased, the performance of the VeNADet is degraded gradually. The True Positive Rate (TPR) achieved with 10% of version attack nodes in different scenarios are given in Figure 14. It shows the decrease of TPR value when number of attacker nodes are increasing.

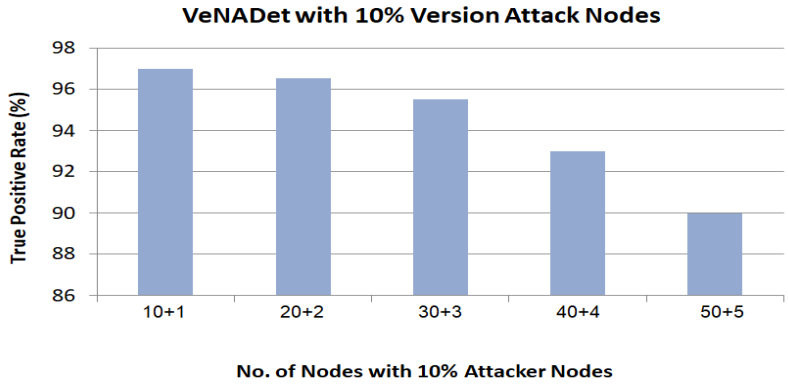


Figure 14. True Positive Rates with 10% Version Attacks

The False Positive Rates (FPR) received while implementing the VeNADet technique with 10% of version attack nodes are given in Figure 15. It shows that the FPR value is positively correlated to the no. of attacker nodes.

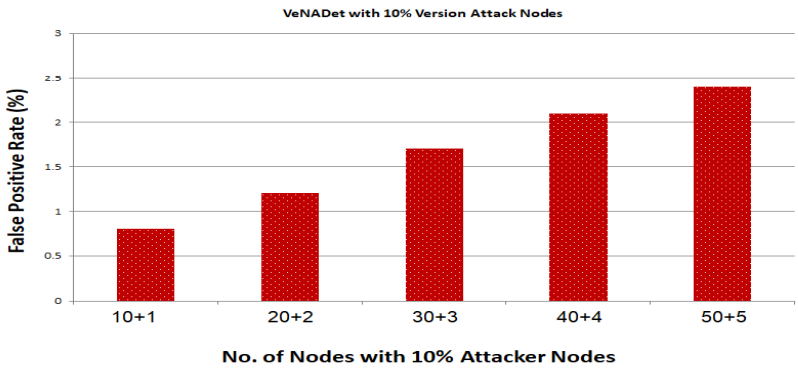


Figure 15. False Positive Rates with 10% Version Attacks

When the percentage of version attacker node increases, it can falsely detect a normal node as an attacker. Thus, the proposed VeNADet mechanism effectively detects and mitigates the Version Attacks in different IoT simulation scenarios.

VI. CONCLUSION

In this paper, a Version Number Attack Detection System (VeNADet) is proposed with three phases like checking phase, validation phase and detection phase. The simulation results clearly show that the performance metrics are highly correlated with number of attackers in the simulation. The VeNADet technique detects the attacker though it is placed in different locations like leaf node, intermediate node or neighbor node to the root. In the VeNADet approach, the node which receives a DIO message updates its Version Number if and only if certain conditions met. Otherwise, it is treated as a malicious node. Thus, the unnecessary Version updates are reduced. The attacker is also isolated from the IoT network by disconnecting the links from the DODAG. According to the simulation results, the proposed VeNADet technique detects 94.4% of Version attacks efficiently.

In this experiment, 10% attacker nodes are included and the effects are analyzed. Mobility is one of the important aspects of IoT devices and affects the behavior of the DODAG topology a lot. In future research, mobility traits and energy will also be considered.

REFERENCES

[1] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif and M.M.A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches", Internet of Things, Elsevier publication, 2019.

[2] Chao Liang, Bharanidharan Shanmugam, Sami Azam, Mirjam Jonkman, Friso De Boe and Ganthan Narayansamy, "Intrusion Detection System for Internet of Thing based on a Machine Learning approach", International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTE CoN), IEEE, 2019.

[3] Bruno Bogaz Zarpaelo, Rodrigo Sanches Miani, Claudio Toshio Kawakani and Sean Carlisto de Alverenga (2017) A Survey of Intrusion Detection in Internet of Things. International Journal of Network and Computer Applications, 2017, DOI: /10.1016/j.jnca.2017. 02.009.

[4] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)", pp. 606–611, 2015.

[5] A. S. Obaid, S. Muhammad Shoaib, H. Choong Seon, and L. Sungwon,"RIDES: Robust intrusion detection system for ip-based ubiquitous sensor networks", Sensor (Basel), Volume: 9, Issue: 5, pp.3447–3468, 2009.

[6] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Realtime intrusion detection in the internet of things", AdHoc Networks, Volume: 11, Issue: 8, pp. 2661-2674, 2013.

[7] Pavan Pongale and Gurunath Chavan, "Real Time Intrusion and Wormhole Attack Detection in Internet of Things", International Journal of Computer Applications (0975-8887), Volume: 121, Issue: 9, 2015.

[8] Anthea Mayzaud, Remi Badonnel and Isabelle Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things", International Journal of Network Security, Volume: 18, Issue: 3, pp. 459–479, 2016.

[9] A. Arul Anitha and L. Arockiam, "ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things", International Journal of Innovative Technology and Exploring Engineering, Volume: 8 Issue: 11, ISSN: 2278-3075, 2019.

[10] Aris, A., Oktug, S. F. and Berna Ors Yalcin, S., "New Lightweight Mitigation Techniques for RPL Version Number Attacks", Ad hoc Networks, 2018, DOI:10.1016/j.adhoc.2018.10.022.

[11] Amit Dvir, T. Holczer and L. Buttyan, "VeRA - Version Number and Rank Authentication in RPL", IEEE 8th International Conference Mobile Adhoc and Sensor Systems (MASS), pp. 709–714, 2011.

[12] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schonwalder, "A Study of RPL DODAG Version Attacks", Monitoring and Securing Virtualized Networks and Services Lecture Notes in Computer Science, Springer Berlin Heidelberg, Volume: 8508, pp. 92–104, 2014.

[13] Ahmed, F. and Ko, Y-B, "A Distributed and Cooperative Verification Mechanism to Defend against DODAG Version Number Attack in RPL", In Proceedings of the 6th International Joint Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2016), pp.55-62 ISBN: 978-989-758-195-3, 2016, DOI: 10.5220/ 0005930000550062.

[14] Mohamad Nazrin Napiah, Mohd Yamini Inda Bin Idris, Roziana Ramli and Ismail Ahmedi, "Compression Header Analyzer Intrusion Detection System (CHA-IDS) for 6LowPAN Communication Protocol", IEEE Access, Volume: 6, pp.16623-16638, 2018.

[15] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., Richardson, M., "A security threat analysis for the routing protocol for low-power and lossy networks (RPLs)", https://tools.ietf.org/html/rfc7416, Accessed April 2020.

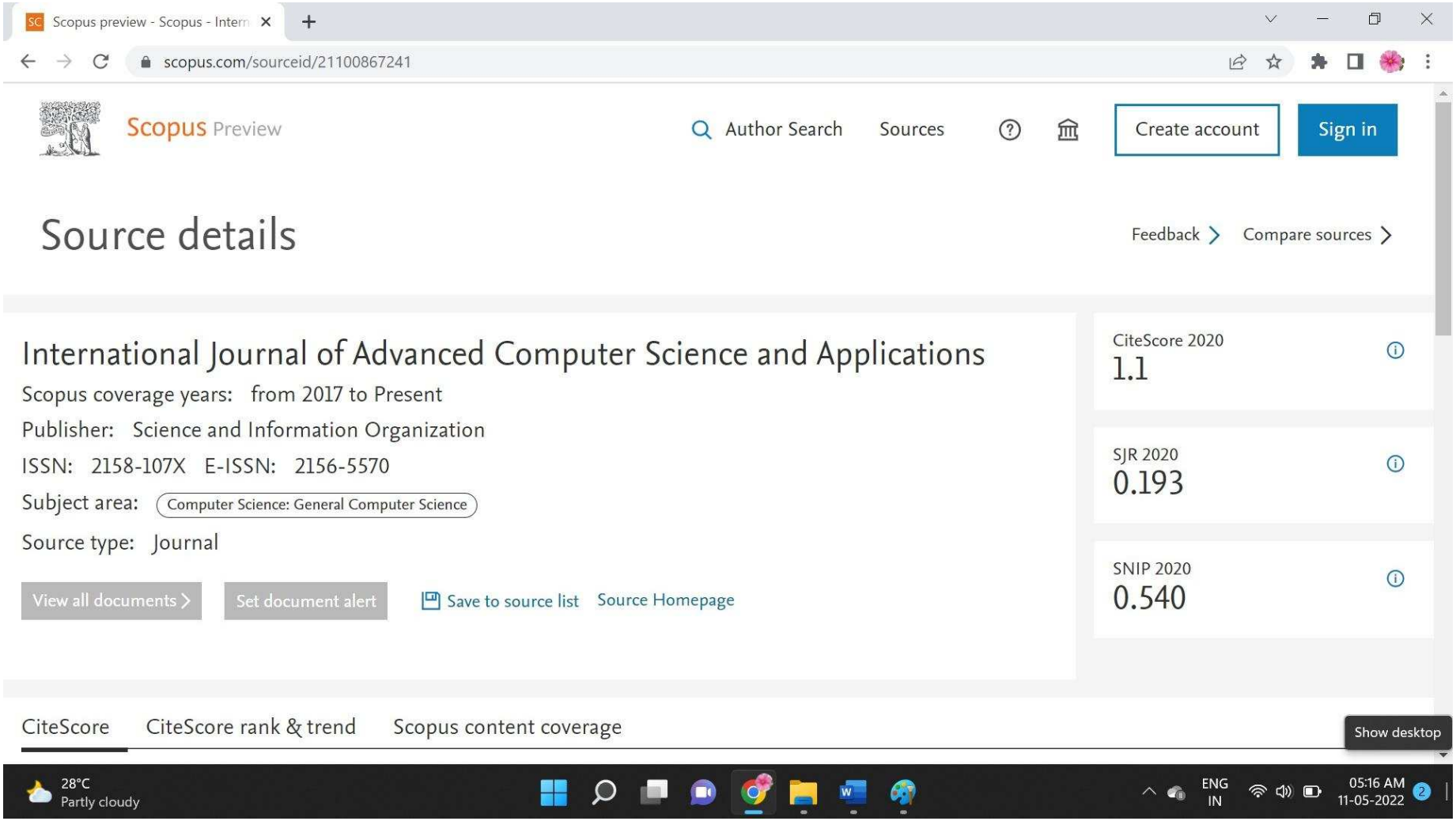
[16] Thubert, P., "Objective function zero for the routing protocol for low-power and lossy networks (RPL). https://tools.ietf.org/html/rfc6552", Accessed April 2020.

[17] Vasseur, J.P., Kim, M., Pister, K., Dejean, N., Barthel, D., "Routing metrics used or path calculation in low-power and lossy networks", https://tools.ietf.org/html/rfc6551, Accessed April 2020.

[18] Zahrah A. Almusaylim, Abdulaziz Alhumam and N.Z. Jhanjhi, "Proposing a Secure RPL based Internet of Things Routing Protocol: A Review", Ad Hoc Networks, 2020, DOI: 10.1016/j.adhoc.2020.102096.

[19] Aris, A., Oktug, S. F. and Berna Ors Yalcin, S., "RPL version number attacks: In-depth Study", IEEE/IFIP Network Operations and Management Symposium, NOMS 2016, DOI:10.1109/noms.2016.7502897.

[20] Rashmi Sahay, G. Geethakumari, Barsha Mitra and Ipsit Sahoo, "Efficient Framework for Detection of Version Number Attack in Internet of Things", Springer Nature Switzerland AG 2020, AISC 941, pp. 480–492, 2020, DOI: 10.1007/978-3-030-16660-1_47.


Scopus preview - Scopus - Intern
scopus.com/sourceid/21100867241
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
International Journal of Advanced Computer Science and Applications
Scopus coverage years: from 2017 to Present
Publisher: Science and Information Organization
ISSN: 2158-107X E-ISSN: 2156-5570
Subject area: Computer Science: General Computer Science
Source type: Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2020
1.1
SJR 2020
0.193
SNIP 2020
0.540
CiteScore
CiteScore rank & trend
Scopus content coverage
Show desktop
28°C
Partly cloudy
ENG
IN
05:16 AM
11-05-2022

Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things

A.Arul Anitha[1]
Research Scholar, Department of Computer Science
St. Joseph's College (Autonomous)
Tiruchirappalli, Tamil Nadu, India
(Affiliated to Bharathidasan University, Tiruchirappalli)

Dr. L. Arockiam[2]
Associate Professor, Department of Computer Science
St. Joseph's College (Autonomous)
Tiruchirappalli, Tamil Nadu, India
(Affiliated to Bharathidasan University, Tiruchirappalli)

***Abstract*—The magical buzzword Internet of Things (IoT) connects any objects which are diverse in nature. The restricted capacity, heterogeneity and large scale implementation of the IoT technology tend to have lot of security threats to the IoT networks. RPL is the routing protocol for the constraint devices like IoT nodes. ICMPv6 protocol plays a major role in constructing the tree-like topology called DODAG. It is vulnerable to several security attacks. Version Number Attack, DIS flooding attack and DAO attack are the ICMPv6 based attacks discussed in this paper. The network traffic is collected from the simulated environment in the normal and attacker settings. An AdaBoost ensemble model termed Ada-IDS is developed in this research to detect these three ICMPv6 based security attacks in RPL based Internet of Things. The proposed model detects the attacks with 99.6% accuracy and there is no false alarm rate. The Ada-IDS ensemble model is deployed in the Border Router of the IoT network to safeguard the IoT nodes and network.**

Keywords*—IoT; ICMPv6; version number attack; DIS attack; DAO attack; Ada-IDS

I. Introduction

Internet of Things (IoT) is a network of embedded objects having unique identifier with sensing and actuation capacities and limited resources. IoT has the ability to connect any objects in the real world to the global network. Though IoT makes the people's life easier, it has lot of security issues and challenges. The privacy and security vulnerabilities increase as the global network includes greater number of connected devices from various fields and domain [1][2]. The large volume of connected devices in IoT network are uniquely identified using IPv6 addressing. IPv6 inherited several features from its previous version IPv4. So, it has the associated vulnerabilities of IPv4 and the specific security challenges of IPv6 [3]. These security threats have to be addressed in order to enhance the IoT security schemes.

IoT resource limited devices form Low-Power Lossy Networks (LLNs). To meet the requirements of the LLNs, the Routing Protocol for Low-Power Lossy Network (RPL) is designed. This RPL protocol is exposed to several security threats [4]. In RPL, the routing is performed by the control messages of the Internet Control Message Protocol version 6 (ICMPv6). The control messages construct a Destination Oriented Directed Acyclic Graph (DODAG). It is a tree structure with hierarchy of nodes with a single root node called as Border Router which acts as a gateway to the global network [5].

The ICMPv6 messages are grouped as error messages and informational messages. The communication between the IPv6 nodes totally depends upon the ICMPv6 Protocol. It is also responsible for router and node configuration. The error messages have a preceding '0' in the high-order bit of the 'Type' field and the informational message contains a preceding '1' in the 'Type' field. ICMPv6 is the backbone of IPv6 and RPL as it has the building blocks such as DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS) and DAO-Acknowledgement (DAO-ACK) informational messages for constructing the DODAG for routing [6].

The root node initiates the DODAG formation by emitting DIO messages in a multicasting fashion. When a node receives the DIO message, based on the information available in the DIO message, it joins the DODAG and sends back the DAO message to the sender. Then it starts multicasting the DIO messages to its children. The DIO messages are regulated by the Algorithm. In order to identify the neighbors and join the DODAG, a node transmits DIS messages in a unicast or multicast manner. After receiving the DAO messages from the children, the parent node acknowledges the DAO message by sending DAO-ACK messages [7].

RPL and ICMPv6 protocols are prone to several security threats and attacks. According to Anthéa Mayzaud et al. [8], the attacks in RPL protocol are classified into three types such as attack against topology, attacks against resources and attacks against network. The attacks against the resources consumes more resources of the constrained devices, the attacks against topology cause sub-optimization and isolation in the topology and the attacks against the traffic creates security threats using the network traffic.

The ICMPv6 based attacks are created by manipulating the control messages. These attacks cause many damages to the networks. It also leads to Denial of Service (DoS) and Distributed Denial of Service (DDoS) in the resource constrained networks. Version Number attacks, DIS flooding attacks and DAO attacks are some of the ICMPv6 control message based attacks which lead to harmful effects in the IoT environment [9]. Machine Learning models are used to detect the intrusions from the network traces and log files. It is very

difficult to design IDS that performs well in terms of accuracy and less false alarm rate. Ensemble machine learning algorithms boosts the accuracy by combining many classifiers [10].

In this paper, an AdaBoost ensemble Intrusion Detection System called Ada-IDS is proposed to detect the Version Number attack, DIS flooding attack and DAO attacks in the IoT network. To develop this system, the IoT network communication traces are collected from the normal simulation environment and attack scenarios such as Version Number attack, DIS flooding attack and DAO attack. The Ada-IDS is developed by using the collected network traces. For that, the pre-processing and feature engineering processes are carried out on these collected data. Finally, an ensemble AdaBoost machine learning algorithms is applied on the collected dataset to build the Ada-IDS for detecting the ICMPv6 based attacks. The proposed Ada-IDS detects the Version Number Attack, DIS flooding attack and DAO attacks with 99.6% accuracy and with very less false alarm rate.

The rest of the paper is organized as follows: Section II explicates the related works of this research. The three ICMPv6 based attacks are explained in Section III. The Icmpv6 dataset used in this research and the proposed Ada-IDS is elaborately discussed in Section IV. The results obtained by the Ada-IDS model are presented in Section V. Finally, the Section VI concludes the paper.

II. Related Work

Adnan Hasan Bdair et al. [11] critically reviewed the latest ICMPv6 based Intrusion Detection mechanisms with a special focus on the Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Three types of ICMPv6 based attacks such as ICMPv6 flood, ICMPv6 amplification and ICMPv6 protocol exploitation were addressed. Different types of Intrusion Detection systems for ICMPv6 based attacks were also explicated in this paper.

Arul Anitha et al. [12] proposed an Artificial Neural Network based Intrusion Detection System for Internet of Things using Multilayer Perceptron for detecting the Version Attacks and DIS attacks from the dataset collected from the Cooja Simulator and the proposed method classified the attacks and normal nodes correctly.

EmreAydogan et al. [13] developed a Centralized Intrusion Detection System for RPL based Industrial IoT using Genetic Programming concept. This system detects 'Hello Flood Attacks' and 'Version Number Attacks' using the Genetic Algorithm approach with 50 population and other default parameters. Network traces are not analyzed in this work.

Nour Mustafa et al. [14] developed an AdaBoost ensemble Network Intrusion Detection System (NIDS) by using Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network (ANN) algorithm. This system detects the application layer related IoT attacks. The UNSW-NB15 and NIMS botnet dataset were used to develop this ensemble model. According to their findings, the proposed model detects the attacks in the UNSW-NB15 dataset with 99.54% accuracy and NIMS botnet dataset with 98.29% accuracy.

Dan Tang et al. [15] proposed a multi-feature based AdaBoost system for detecting the low-rate Denial of Service (LDoS) attacks. At fixed time intervals the network traffics were captured and the obtained samples were analyzed using various statistical measures. The correlation scores between the features and the class labels were attained to choose the optimal feature set. Using the optimal features, the AdaBoost ensemble model was developed. NS2 simulator and a test-bed were used to evaluate the performance of the model and achieved 94.05% and 97.06% attack detection accuracy respectively.

A.R.Javed et al. [16] proposed an AdaBoost ensemble classifier to detect botnet attacks in connected vehicles. The decision tree algorithm was used as the base estimator and the cluster size was 100 in the AdaBoost algorithm. The performance of the AdaBoost classifier was compared with the decision tree, probabilistic neural network and sequential minimal optimization. According to their findings, the AdaBoost classifier outperformed other models and achieved 99.7% true positive rate and 99.1% accuracy.

Amin Shahraki et al. [17] performed a comparative analysis on various AdaBoost algorithms like Real Adaboost, Gentle Adaboost and Modest Adaboost using the well-known Intrusion detection datasets such as KDDCUP99, NSL-KDD, CICIDS2017, UNSW-NB15 and TRAbID. In this research, the authors identified that Gentle AdaBoost and Real AdaBoost performed better than the Modest AdaBoost algorithm. At the same time, the Modest AdaBoost algorithm was faster than the other AdaBoost algorithms.

III. ICMPv6 Attacks in RPL based IoT

The ICMPv6 protocol is susceptible to various security threats and attacks. In this research, three ICMPv6 based attacks are implemented such as Version Number Attack DIS attack and DAO attack. The characteristics of these attacks are explained below:

A. Version Number Attacks

Version Number is an 8-bit number which denotes the Version of the DODAG topology. It is multicasted by the parent nodes using the DIO control message. Whenever there is an inconsistency in the DODAG, the global repair mechanism is initiated and the Version Number is updated by the root node. This updated information is multicasted from the root node via DIO control message. A Version Number Attacker without the knowledge of the root node updates the Version Number periodically and sends the updated version number using the DIO messages to its neighbors. On receiving this DIO message, the neighboring nodes join the global repair mechanism. Hence, the DODAG is reconstructed again and again. This malicious act affects the normal responsibilities of the legitimate nodes and consumes the constrained resources of the IoT devices. In the long run, it increases the control traffic while constructing the DODAG repeatedly in the network and this leads to Denial of Service (DoS) attacks [18][19].

B. DIS Flooding Attacks

This attack is created by manipulating the header details of the DIS messages. The DIS Control messages are used to probe its neighbors in order to join the DODAG. On receiving this DIS message, the neighbor nodes send back DIO messages to the sender. The Time duration for sending DIO messages is scheduled by the Trickle Timer. A DIS flooding attacker continuously multicasts DIS messages to its neighbors even though it received DIO message already. This heavy flooding of DIS messages in the network degrades the performance of the Network and leads to Denial of Service (DoS) attack [20].

C. DAO Attacks

DAO attack is generated by manipulating the DAO Control Message. When a Child node receives a DIO message from its parent, it has to send back a DAO message for maintaining the reverse root. The DAO message sent by the child node traverses multiple ancestors until it reaches the root node. A DAO attacker continuously transmits the DAO message to its parent list. All such unnecessary messages in the network have to be forwarded to the root node. It consumes more network resources and also prohibits the legitimate nodes to perform regular activities. Finally, the network will be in an inconsistent state which causes Denial of Service (DoS) attacks in the network [21].

These three attacks are created by using the ICMPv6 control messages which consumes more resources in the IoT network and reduces network performance. At last, all the three attacks lead to Denial of Service (DoS) attack which causes more damage to the RPL based IoT network.

IV. Proposed Ada-IDS Model

Network or Centralized Intrusion Detection System and Distributed Intrusion Detection System are the major two categories of IDS. In the centralized concept, the IDS is installed in the border router or a dedicated server. In the Distributed IDS, it is deployed in the client nodes. As the IoT nodes are resource constrained, the Distributed IDS concept is not suitable for limited resource devices.

The proposed Ada-IDS belongs to the Centralized IDS category. It monitors the nodes in the network and whenever there is an intrusion occurs, it raises an alarm to notify the admin about the issue. The various phases involved in developing the Ada-IDS are given in Fig. 1.

As it is given in Fig. 1, there are five phases for developing the Ada- IDS that are Data Collection Phase, Pre-Processing Phase, Feature Engineering Phase, Model Building Phase and Deployment Phase.

A. Data Collection Phase

The data is collected from the simulation environment. There are 50 normal client nodes, one root node and an attacker involved in the simulation. The Version Number Attack, DIS flooding Attack DAO attacks and a simulation without attacker are implemented in the Cooja simulator and the network traces from all these experimental setups were captured using the 6LoWPAN Analyzer tool. The simulation is performed for 30 minutes in each scenario. The captured packets are analyzed using the WireShark tool and the .pcap files were converted into .csv files. The file is named as 'Icmpv6.csv' that is used for building the Ada-IDS model. The collected dataset contains normal packets, Version Number Attacks, DIS flooding Attacks and DAO Attacks. The Normal and Attack instances are listed in Table I.

As it is given in Table I, there are 127684 samples in the dataset including 125184 Normal, 325 DIS Attacks, 1193 DAO Attacks and 982 Version Number Attacks. There are nine attributes in the dataset. The description of the dataset is given in Table II.

TABLE I. Normal and Attack Instances

S.No.	Type	No. of Packets
1.	Normal	125184
2.	DIS Attacks	325
3.	DAO Attacks	1193
4.	Version Number Attacks	982
Total		**127684**

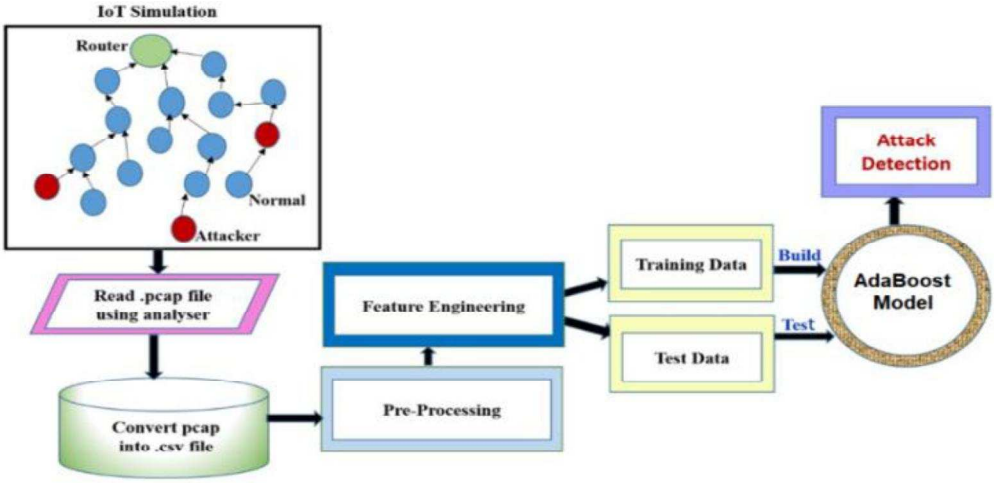


Fig. 1. Ada-IDS Model.

Table II explains the attributes of the Icmpv6 dataset. The screenshot with sample records captured using python code is shown in Fig. 2.

As it is given in Fig. 2, the Class field and Type field denote whether a packet is attack or normal. The Type field also gives the details of an attack as Version Attack, DIS Attack or DAO Attack.

TABLE II. DESCRIPTION OF THE ICMPV6 DATASET

S.No.	Attribute Name	Data Type	Description
1.	No.	Integer	Packet Number
2.	Source	String	Source Address of a packet
3.	Time	Float	Time represented in millisecond
4.	Destination	String	Destination Address of a packet
5.	Protocol	String	Protocol for Communication
6.	Length	Integer	Packet length in Bytes
7.	Info	String	Description about the protocol
8.	Class	String	The packet is Attack or Normal
9.	Type	String	Type of the Attack (Version, DIS, DAO)

B. Pre-Processing Phase

The dataset collected from the simulation environment has to undergo a pre-processing stage in order to be relevant for building the AdaBoost ensemble model. There are 394 missing values in Source and Destination fields. Since these two fields represent the IPv6 address of the nodes, the missing values cannot be replaced by mean, median or mode values. A new value is given for the Source and Destination Addresses.

C. Feature Engineering

One hot encoding and label encoding are performed on the categorical features to make them relevant for the ML algorithms. The frequency encoding is applied for the 'Time' feature. The Class feature is created which separates the Normal data samples from the Attack samples. The Type feature categorizes the different types of attacks such as DIS Attack, DAO Attack and Version Number Attack. The feature 'No.' indicates the packet number which doesn't have any significance in predicting the target and hence it is eliminated from the dataset. The null values in the 'Source' feature are replaced by a dummy value 'a'. Similarly, the null values in the 'Destination' field are replaced by a dummy value 'b'. After the accomplishment of the pre-processing and feature engineering tasks, the dataset will look like the Fig. 3.

As shown in Fig. 3, all the categorical values of the dataset are converted into numerical values. Now, the dataset is relevant for model building.

D. Model Building Phase

The pre-processed dataset with eight features is used in this experiment. The combined dataset has 127684 data samples. 80% of the data samples are split into a training set which contains 102147 instances and the remaining 20% of data samples are treated as the test set which contains 25537 instances.

No	Source	Time	Destination	Protocol	Length	Info	Class	Type
1	fe80::212:742f:2f:2f2f	0	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
2	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
3	fe80::212:741d:1d:1d1d	0.114	fe80::212:7421:21:2121	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
4	fe80::212:742f:2f:2f2f	0.114	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
5	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
...	...	...	...	...	...	...	...	...
127680	fe80::212:740b:b:b0b	431.709	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127681	fe80::212:741f:1f:1f1f	431.71	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal
127682	fe80::212:7432:32:3232	431.71	fe80::212:7424:24:2424	ICMPv6	76	RPL Control (Destination Advertisement Object)	Attack	Version Attack
127683	fe80::212:740b:b:b0b	431.711	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127684	fe80::212:741f:1f:1f1f	431.712	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal

127684 rows × 9 columns

Fig. 2. Screenshot with Sample Date.

	Source	Time	Destination	Protocol	Length	Info	Class	Type
0	58	3	29	0	76	4	0	0
1	21	4	15	0	76	4	0	0
2	40	4	25	0	76	4	0	0
3	58	4	29	0	76	4	0	0
4	21	4	15	0	76	4	0	0
...	...	...	...	...	...	...	...	...
127679	22	1	2	0	76	4	0	0
127680	42	2	37	0	97	2	0	0
127681	61	2	28	0	76	4	0	0
127682	22	1	2	0	76	4	0	0
127683	42	1	37	0	97	2	0	0

127684 rows × 8 columns

Fig. 3. Sample Data after Pre-processing.

E. AdaBoost Ensemble Model

An Ada-Boost (Adaptive Boosting) model is built to detect the Version Number Attack, DIS flooding attack and DAO attacks in the IoT environment. It was developed by Yoav Freund and Robert Schapire in 1996 as a classifier that uses ensemble boosting. Classifier accuracy is improved by combining multiple classifiers [22]. AdaBoost classifier creates a powerful classifier by combining several weak classifiers, resulting in a powerful classifier with high accuracy. The basic idea behind Adaboost is to train the data sample and adjust the classifier weights in each iteration, so that unusual observations can be accurately predicted [23]. Interactive training on a variety of weighted training examples should be used to fine-tune the classifier. It tries to minimize training error in order to provide the best fit possible for these examples in each iteration. The steps for obtaining the ensemble model are given below:

1) Adaboost begins by picking a training subset at random.

2) The AdaBoost machine learning model is trained iteratively by selecting the training set based on the accuracy of the previous training prediction.

3) It gives more weight to observations that were incorrectly classified, increasing the likelihood that these observations will be correctly classified during the next iteration.

4) Additionally, the trained classifier is given more weight in each iteration based on how accurately it classifies.

5) Classifiers that are more precise will be given more credit.

6) In this process, the training data is iterated until it fits perfectly, or until the specified maximum number of estimators has been reached.

In AdaBoost classifier, there are three basic parameters such as base_estimator, n_estmator and learning_rate. The parameters used in this research are given below:

- base_estimator: A weak learner is used to train the model. In this work, the default DecisionTreeClassifier is used to train the ensemble model.
- n_estimator: It specifies how many weak learners are used for training the model repeatedly. In this model 10 estimators are used. The performance is analyzed. Then increment by 10 until it reaches 100 estimators.
- learning_rate: The default learning rate is 1, it denotes the weights of the weak learner. In this ensemble model, the default learning rate is used.

In AdaBoost ensemble approach, weak learners are combined to improve accuracy, which is done iteratively by fixing the faults of the weak classifier. AdaBoost isn't prone to being overfit issue. Though AdaBoost has these advantages, the performance is degraded if there are outliers in the dataset.

F. Deployment Phase

The proposed Ada-IDS model is installed in the Border Router (Gateway). The Pseudo Code for the Ada-IDS is given in Fig. 4.

This Ada-IDS detects the icmpv6 based attacks such as Version Number Attacks, DIS flooding attacks and DAO attacks in RPL based IoT networks.

Pseudo Code for Ada-IDS

Input: Network Traffic

Output: Attack- DAO, DIS, Version or Normal

```
1. implement Normal and Attack Scenarios in Cooja Simulator
2. collect the packets from 6LowPAN Analyser tool
3. analyse the packets using WireShark tool
4. convert the packets into .csv format
5. extract the features from the .csv file
6. pre-process the features
7. perform feature encoding
8. select the relevant features
9. split the Dataset into two parts:
    - 80% Training data
    - 20% Testing data
10. learning_rate=1, base_estimator=DecisionTree Classifier
11. for i=10 to 100 do:  // Build AdaBoost Ensemble Model
12.     n_estimator=i
13.     build AdaBoost(learning_rate, base_estimator,n_estimator)
14.     calculate training_time
15.     test AdaBoost(learning_rate, base_estimator,n_estimator)
16.     calculate testing_time
17.     evaluate confusion_matrix, accuracy
18.     evaluate precision, recall, f-Score
19.     increment i by 10
20. end for
21. implement Ada-IDS Model in the Gateway
22. return output
```

Fig. 4. Pseudo Code for Ada-IDS.

V. Result and Discussion

This section elaborates the results obtained by the AdaBoost ensemble model. After accomplishing preprocess and feature engineering phases, the dataset is split into two sets like training and testing set. The training set contains 80% of the original data samples and the testing set consists of 20% of the dataset. The No. of samples in both categories is given in Table III.

The training samples are used to build the AdaBoost ensemble model. The DecisionTreeClassifier is selected as the weak classifier to fine tune the model iteratively. The learning rate parameter takes the default value. The no. of base_estimator is initially given as 10. The training time and testing time with 10 base estimators are analyzed. The testing accuracy for the AdaBoost Classifier with 10 base estimators is noted. To check whether there will be any change in the accuracy with respect to the number of estimators, the base estimator is incremented by 10 until it reaches 100. Surprisingly, the accuracy is 99.6% and it is not affected by the number of estimators used for building the AdaBoost classifier. The parameters and accuracy of the AdaBoost ensemble model is listed in Table IV.

As it is given in Table IV, the learning_rate is constant of all experiments. The number of Decision Trees used for building the AdaBoost ensemble model for each experiment varies from 10 to 100. The accuracy obtained is the same in all experiments. The training time and testing time varies in each experiment according to the no. of base estimators used. The relationship between the training time and the testing time is indicated by using Fig. 5.

As Fig. 5 depicts, the training time required for building the model is more compared to the testing time. Because, the training set contains 80% of data. Also when number of DecisonTreeClassifier increases, the training time also increases. So, there is a positive correlation between the number of samples, number of estimators and the training time. The testing time also varies according to the no. of estimators in each experiment. When more DecisionTreeClassifiers are included, the testing time also increases.

TABLE III. Description of the ICMPv6 Dataset

Type of Instance	Training (80%)	Testing (20%)	Total
Normal	100169	25015	125184
DAO Attack	79	246	325
DIS Attack	1115	78	1193
Version Attack	784	198	982
Total Samples	**102147**	**25537**	**127684**

TABLE IV. AdaBoost Parameters and Accuracy

n_Estimator	Learning Rate	Training Time (Sec.)	Testing Time (Sec.)	Accuracy
10	1	0.62	0.069	0.996
20	1	1.77	0.092	0.996
30	1	1.662	0.163	0.996
40	1	2.406	0.355	0.996
50	1	2.937	0.272	0.996
60	1	4.881	0.363	0.996
70	1	5.21	0.357	0.996
80	1	6.627	0.428	0.996
90	1	5.561	0.786	0.996
100	1	6.923	0.872	0.996

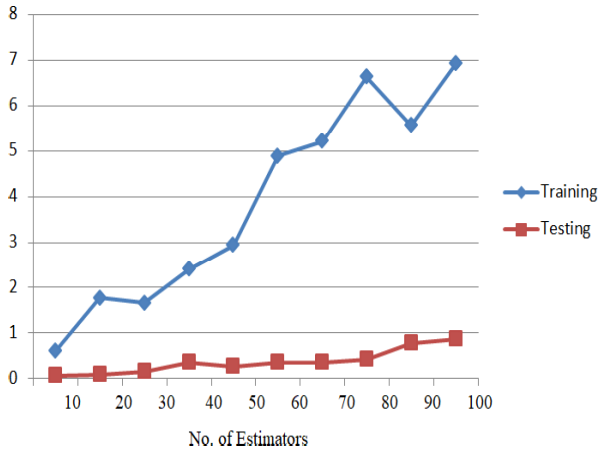


Fig. 5. Training and Testing Time Comparison.

A. Evaluation Metrics

There are three classes of attacks in the dataset. The confusion matrices are generated for each experiment which shows the actual and predicted class labels for each sample. To evaluate the performance of the models, the metrics such as accuracy, precision, Recall, F-Score are also computed [24].

- True Positive (TP): TP represents the correct classification of an attack packet as attack.
- True Negative (TN): TN specifies the correct classification of normal packets as normal.
- False Negative (FN): FN illustrates the wrong classification of an attack packet as normal. When this value increases, it affects the confidentiality and availability which are very important security concerns.
- False Positive (FP): FP signifies the incorrect classification where the normal packet is classified as attack.
- Accuracy: It denotes the ratio between the sum of correctly classified samples as normal and attack to the total instances. The formula for computing Accuracy is given in the Eq.1

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (1)$$

- Recall (Sensitivity): Recall quantifies the number of correct positive predictions made out of all correct classifications that could have been made. Eq. 2 is the formula for calculating the sensitivity or recall.

$$\text{Recall} = (TP) / (TP+FN) \quad (2)$$

- Precision: It represents the total number of records that are correctly classified as attack divided by a total number of records classified as attack. The precision can be calculated according to the Eq.3.

$$\text{Precision} = (TP) / (TP+FP) \quad (3)$$

- F-Score: F-Score combines the properties of both precision and recall and it expresses them using a single measure. The formula for computing the F-Score is given in Eq.4.

$$\text{F-Score} = 2*(\text{Recall}*\text{Precision})/(\text{Recall} +\text{Precision}) \quad (4)$$

In this work, the CPU time for training the model and testing the model are also taken into account for each experiment. The confusion matrix obtained for each experiment is almostthe same and it is given in Table V.

In Table V, the correctly classified samples in the testing set are given blue color text, but the misclassified samples are denoted by using red font color. As it is shown in the table, all normal events are identified correctly. There are very few misclassifications in other categories. Using the confusion matrix and by applying the equations Eq. 1 to Eq. 4, the accuracy, precision, recall and f1-score values are calculated and listed in Table VI.

TABLE V. CONFUSION MATRIX

	Normal	DAO Attack	DIS Attack	Version Attack
Normal	25015	0	0	0
DAO Attack	0	214	32	0
DIS Attack	0	21	57	0
Version Attack	0	0	38	160

TABLE VI. RESULTS FROM COFUSION MATRIX

n_Estimator	Accuracy	Precision	Recall	F-Score
10	0.996	0.99	1.00	1.00
20	0.996	0.99	1.00	1.00
30	0.996	0.99	1.00	1.00
40	0.996	0.99	1.00	1.00
50	0.996	0.99	1.00	1.00
60	0.996	0.99	1.00	1.00
70	0.996	0.99	1.00	1.00
80	0.996	0.99	1.00	1.00
90	0.996	0.99	1.00	1.00
100	0.996	0.99	1.00	1.00

As Table VI denotes, the Ada-IDS model, developed by using AdaBoost Ensemble model with DecisionTreeClassifier provides better results in terms of accuracy, precision, recall and f-score. The obtained confusion matrix is the same for all observations, so that it gives the same accuracy, precision, recall and f-score values. Since it doesn't have any false alarm-rate, it is suitable for anomaly detection. The Ada-IDS is implemented in the Border Router (6BR) to safeguard the connected devices in the IoT network.

VI. CONCLUSION

The security attacks are inevitable in RPL based Internet of Things as they have limited resources compared to other networks. In this paper, an ensemble IDS named Ada-IDS is developed using the AdaBoost ensemble model and it is deployed in the Border Router to protect the IoT network from Version Number Attack, DIS flooding Attack and DAO Attack. According to the experiments, this Ada-IDS ensemble model detected these three types of attacks with 99.6% accuracy and with no false alarm rate. Hence, it will act as an anomaly based Intrusion System. It is suitable for all IoT domains and it acts as a shield to protect the nodes from flooding of ICMPv6 messages, unnecessary version updates and bulk sending of the DAO message in the RPL based IoT network. Availability and reliability of the IoT nodes for their normal responsibilities are also ensured. To enhance this system further, more ICMPv6 related attacks can be included in the 'icmpv6.csv' dataset.

REFERENCES

[1] Zhihan LV, Liang Qiao, Amit Kumar Singh and Qingjun Wang, "AI-empowered IoT security for Smart Cities", ACM Trans. Internet Technol. 21, 4, Article 99 , July 2021, DOI: 10.1145/3406115.

[2] Mahmoud Ammar, Giovanni Russello and Bruno Crispoa, "Internet of Things: A survey on the security of IoT frameworks", Journal of

Information Security and Applications, Vol. 38, pp.8-27, 2018, DOI: 10.1016/j.jisa.2017.11.002.

[3] Lisandro Ubiedo, Thomas O'Hara, Maria Jose Erquiaga and Sebastian Garcia, "Current State of IPV6 Security in IoT", Stratosphere Research Laborartory, arXiv:2105.02710v1 [cs.NI] 5 May 2021.

[4] Mohammed Aman Kareem and Shahab Tayeb, "ML-based NIDS to secure RPL from Routing Attacks", IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, DOI: 10.1109/ccwc51732.2021.937584.

[5] Ge Guo, "A Lightweight Countermeasure to DIS Attack in RPL Routing Protocol", IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, DOI: 10.1109/CCWC51732. 2021.9376041.

[6] Omar E. Elejla, BahariBelaton, Mohammed Anbar and Ahmad Alnajjar, "A Reference Dataset for ICMPv6 based Flooding Attacks", Journal of Engineering and Applied Sciences, Vol.11, Issue: 3, pp: 476-481, ISSN: 1816-949x, 2016.

[7] Antonio Arena, Pericle Perazzo, Carlo Vallati, Gianluca Dini and Giuseppe Anastas, "Evaluating and Improving the Scalability of RPL Security in the Internet of Things", Computer Communications, Volume 151, pp. 119-132, 2020, DOI: 10.1016/j.comcom.2019.12.062.

[8] Anthéa Mayzaud, Rémi Badonnel, Isabelle Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things", International Journal of Network Security, ACEEE a Division of Engineers Network, Vol.18, No.3, pp.459-473, 2016, DOI: 10.6633/IJNS.201605.18(3), hal-01207859.

[9] Andrea Agiollo, Mauro Conti, Pallavi Kaliyar, Tsung-Nan Lin and Luca Pajola, "DETONAR: Detection of Routing Attacks in RPL-Based IoT", IEEE Transactions on Network and Service Management, Vol. 18, NO. 2, JUNE 2021, pp. 1178 – 1190, DOI: 10.1109/TNSM.2021.3075496.

[10] Alaa Alhowaide, Izzat Alsmaadi and Jiang Tang, "Ensemble Detection Model for IoT IDS", Internet of Things, 10035, 2021, DOI: 10.1016/j.iot.2021.100435.

[11] Adnan HasanBdair, Rosni Abdullah, SelvakumarManickam and Ahmed K. Al-Ani, "Brief of Intrusion Detection Systems in Detecting ICMPv6 Attacks", Computational Science and Technology, Lecture Notes in Electrical Engineering 603, Springer Nature, DOI: 10.1007/978-981-15-0058-9_20.

[12] A. Arul Anitha, L. Arockiam, "ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things", International Journal of Innovative Technology and Exploring Engineering, Volume: 8 Issue: 11, ISSN: 2278-3075, 2019.

[13] EmreAydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström and M. Gidlund, "A Central Intrusion Detection System for RPL-Based Industrial Internet of Things," 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), 2019, pp. 1-5, DOI: 10.1109/WFCS.2019.8758024.

[14] Nour Moustafa, Benjamin Turnbull and Kim-Kwang Raymond Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things", IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2018.2871719.

[15] Dan Tang, Liu Tang, Rui Dai, Jingwen Chen, Xiong Li and Joel J.P.C. Rodrigues, "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost", Future Generation Computer Systems, Vol. 106 (2020), pp. 347–359, DOI: 10.1016/j.future. 2019.12.034.

[16] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas and Xuan Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles", Transactions on Emerging Telecommunications Technologies, Wiley, 2020, DOI: 10.1002/ett.4088.

[17] Amin Shahraki, Mahmoud Abbasi and Øystein Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost", Engineering Applications of Artificial Intelligence, Vol. 94, 103770, 2020.

[18] Ahmet Arış, Sıddıka Berna Örs Yalçın and Sema F. Oktuğ, "New lightweight mitigation techniques for RPL version number attacks", Ad Hoc Networks, Vol. 85, pp. 81-91, 2018, DOI: 10.1016/j.adhoc.2018. 10.022.

[19] Mayzaud A., Sehgal A., Badonnel R., Chrisment I., Schönwälder J., "A Study of RPL DODAG Version Attacks", IFIP International Conference on Autonomous Infrastructure, Management and Security, AIMS 2014: Monitoring and Securing Virtualized Networks and Services pp 92-104, DOI: 10.1007/978-3-662-43862-6_12.

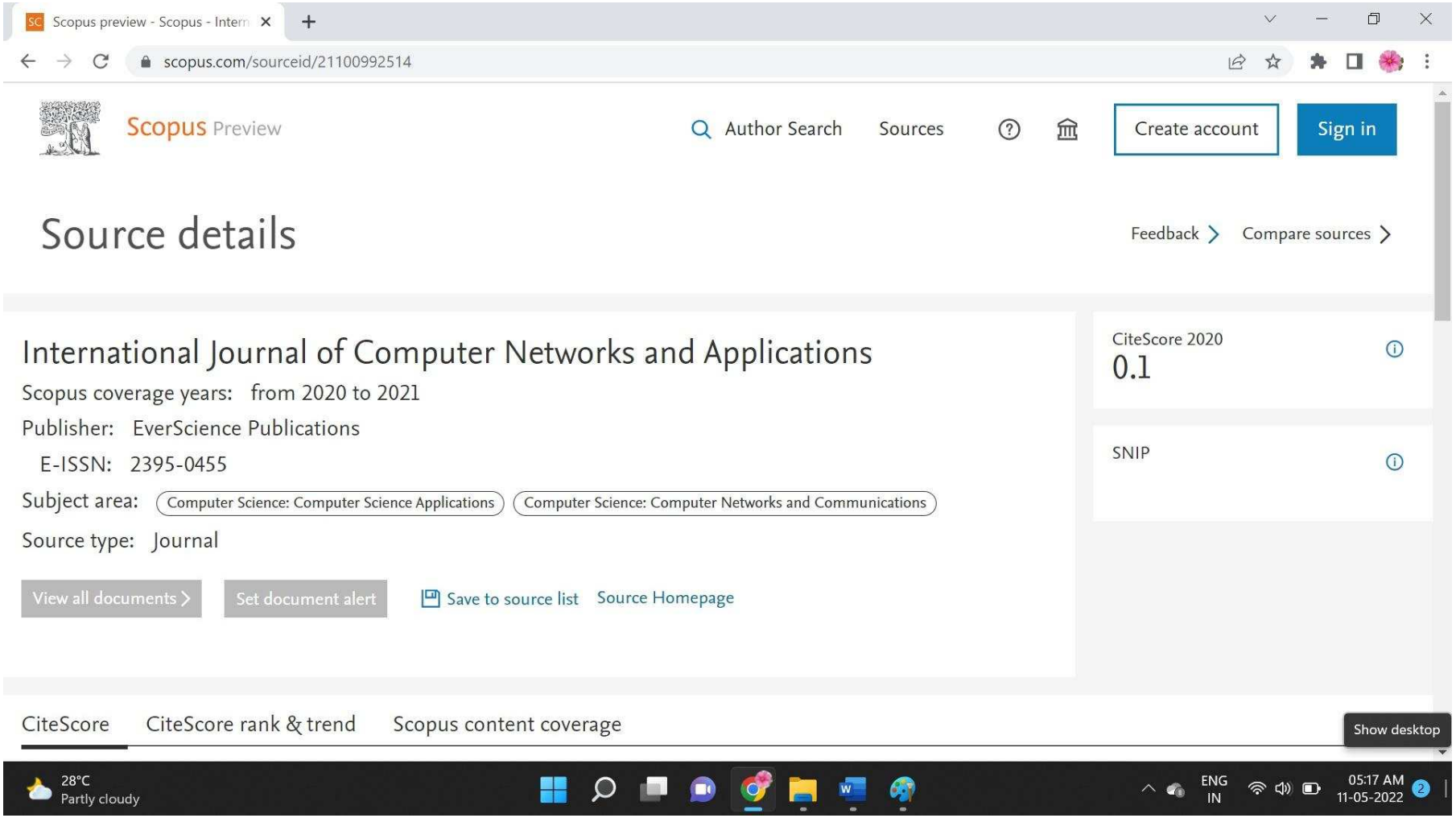
[20] Cong Pu, "Spam DIS Attack Against Routing Protocol in the Internet of Things", International Conference on Computing, Networking and Communications (ICNC), IEEE, 2019, DOI:10.1109/iccnc.2019. 8685628.

[21] Isam Wadhaj, Baraq Ghaleb, Craig Thomson, Ahmed Al-Dubai and William J. Buchanan, "Mitigation Mechanisms against the DAO Attack on the Routing Protocol for Low Power and Lossy Networks (RPL)", Green Internet of Things, IEEE Access, Volume: 8, 2020, DOI: 10.1109/ACCESS.2020.2977476.

[22] Avinash Navlani, "AdaBoost Classifier in Python", DataCampTutorials, 2018, https://www.datacamp.com/community/tutorials/adaboost-classifier -python, [Accessed on: 15th October, 2021].

[23] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas and Xuan Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles", Transactions on Emerging Telecommunications Technologies,Wiley, 2020, DOI:10.1002/ett.4088.

[24] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System", IEEE Explore, ISSN: 1949-0488, 2017, DOI:10.1109/SISY.2017.8080566.


Scopus preview - Scopus - Intern
scopus.com/sourceid/21100992514
Scopus Preview
Author Search
Sources
Create account
Sign in
Source details
Feedback
Compare sources
International Journal of Computer Networks and Applications
Scopus coverage years: from 2020 to 2021
Publisher: EverScience Publications
E-ISSN: 2395-0455
Subject area:
Computer Science: Computer Science Applications
Computer Science: Computer Networks and Communications
Source type: Journal
View all documents
Set document alert
Save to source list
Source Homepage
CiteScore 2020
0.1
SNIP
CiteScore
CiteScore rank & trend
Scopus content coverage
Show desktop
28°C
Partly cloudy
ENG
IN
05:17 AM
11-05-2022

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/211599 Volume 9, Issue 1, January – February (2022)

REVIEW ARTICLE

A Review on Intrusion Detection Systems to Secure IoT Networks

A. Arul Anitha
Department of Computer Science, St. Joseph's College (Autonomous) (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India
arulanita@gmail.com

L. Arockiam
Department of Computer Science, St. Joseph's College (Autonomous) (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India
larockiam@yahoo.co.in

Received: 23 November 2021 / Revised: 08 January 2022 / Accepted: 13 January 2022 / Published: 28 February 2022

Abstract – The Internet of Things (IoT) and its rapid advancements will lead to everything being connected in the near future. The number of devices connected to the global network is increasing every day. IoT security challenges arise as a result of the large-scale incorporation of smart devices. Security issues on the Internet of Things have been the most focused area of research over the last decade. As IoT devices have less memory, processing capacity, and power consumption, the traditional security mechanisms are not suitable for IoT. A security mechanism called an Intrusion Detection System (IDS) has a crucial role in protecting the IoT nodes and networks. The lightweight nature of IoT nodes should be considered while designing IDS for the IoT. In this paper, the types of IDS, the major attacks on IoT, the recent research, and contributions to IDS in IoT networks are discussed, and an analytical survey is given based on the study. Though it is a promising area for research, IDS still needs further refinement to ensure high security for IoT networks and devices. Hence, further research, development, and lightweight mechanisms are required for IDS to provide a higher level of security to the resource-limited IoT network.

Index Terms – Attack, IoT, Intrusion, IDS, RPL, Security.

1. INTRODUCTION

The Internet of Things (IoT) is a robustly evolving trend that incorporates technical, scientific, social, and economic implications. It is essential to all facets of human life [1]. Healthcare, logistics, smart-cities, smart-homes, and agriculture are just a few of the applications for IoT. Due to its resource-constrained characteristics, the IoT tends to have more vulnerability that can be easily exploited by an attacker. The number of connected unsecured IoT devices on the global network is rapidly increasing [2]. Researchers are mainly focusing on various encryption and authentication mechanisms to ensure data confidentiality, authentication, and privacy among users and things. Most of the IoT devices have been developed without considering the fundamental security requirements [3].

The tools and techniques available for securing the IoT are inadequate because of the large number of interconnected devices. Moreover, the security mechanisms based on cryptography are mainly used to prevent external attacks such as eavesdropping and message alternation. When the cryptographic techniques hold the valid key and are compromised by the attack, they cannot detect the vulnerable nodes. Intruders can easily access the security details from the compromised nodes and immediately launch several internal attacks. Hence, to offer an extra level of security to the IoT, the Intrusion Detection System (IDS) acts as a tool [4].

Anthea Mayzaud et al. [5] categorized the Routing Protocol for Low Power Lossy Networks (RPL) attacks into three types: attacks targeting the topology, attacks on network resources, and attacks targeting the network traffic. Attacks on resources require more of the restricted devices' resources like processing requirements, power, and memory; attacks on topology induce isolation and sub-optimization in the topology, and attacks on traffic create security risks from the network's traffic. All these types of attacks have negative impacts on the RPL based IoT network. These attacks have to be detected and mitigated to ensure the security constraints of the IoT networks.

Intrusion Detection is an act of monitoring and possibly preventing the malicious activities of the intruders. Intrusion Detection System is a network security tool that consists of software or a combination of hardware and software to protect the traditional networks. It can be used to monitor all sorts of activities in the network. If there is any attack or unwanted activity in the network, the IDS detects the intrusions, alerts the administrator, logs the attacks for forensic activities,

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/211599 Volume 9, Issue 1, January – February (2022)

REVIEW ARTICLE

isolates the intruder, and also disconnects the connection path of the intruder [6]. The functionalities of Intrusion Detection System are illustrated in Figure 1.

As it is given in Figure 1, the IDS can monitor, analyse, assess, track, alert and mitigate attacks in IoT networks. IDSs are at a mature level in the traditional networks. Since IDS consumes more memory, processing capability and energy, the IDSs that are technologically advanced for the traditional and wireless networks are not suitable for IoT. Because of these constraints, finding IoT nodes with higher computing capability to support IDS agents is very difficult. So, there is a need for modelling lightweight IDS to adapt to the IoT constraints. The Figure 2 illustrates the typical centralized IDS for IoT networks.

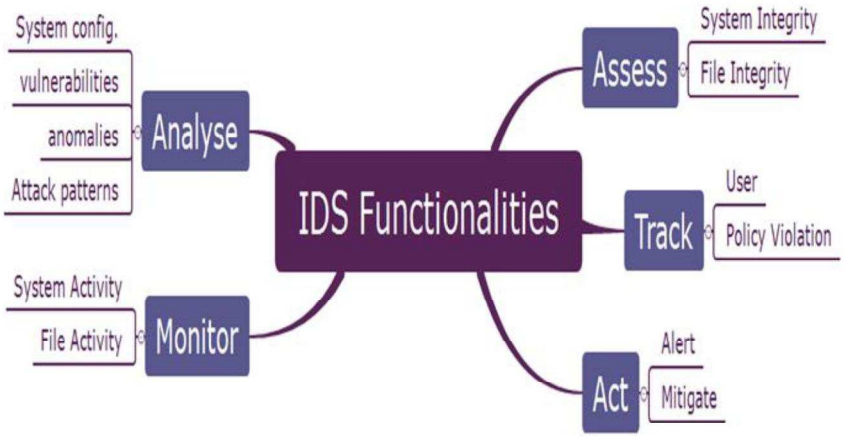


Figure 1 Functionalities of IDS

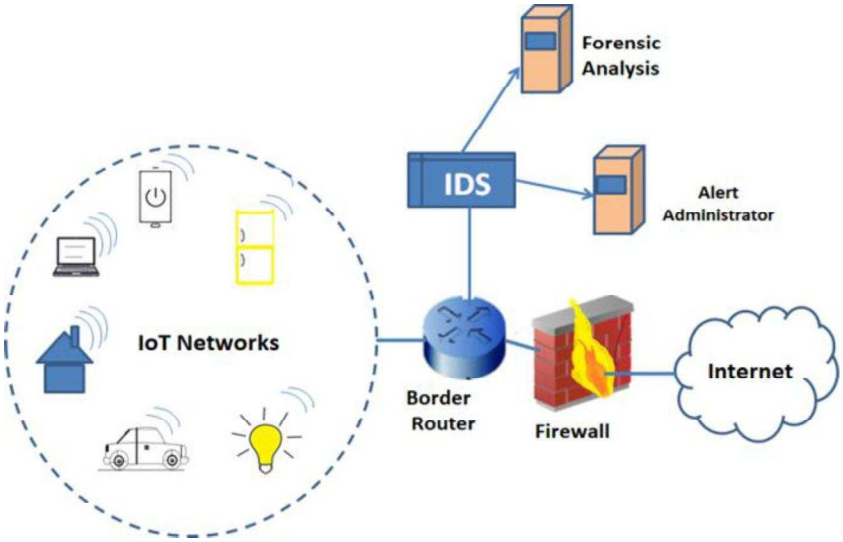


Figure 2 A Typical IDS for Internet of Things

Here, the smart gadgets are linked to the Internet through the gateway device called border router. As the Figure 2 indicates, the IDS tool is implemented in the gateway device. It monitors all IoT network-related activities and, whenever an intrusion arises, the IDS will alert the administrator. It also logs the events for forensic analysis.

1.1. Objectives

The major aim of this paper is to explore systematically the IDSs that are available for protecting the IoT networks. The objectives are listed below.

- To analyse the need for IDS in securing the IoT networks,

- To explore the different types of existing IDS for IoT,
- To discuss the issues and challenges that direct to future research,
- To provide an analytical survey of the reviewed IDSs.

This paper is structured as follows: The section 2 discusses the recent IDS research in the IoT; section 3 explains the different types of attacks in IoT environment; section 4 describes the types of IDSs for IoT based on the placement strategy and technologies implemented; section 5 summarizes the reviewed works as an analytical survey; section 6 points out some issues and challenges while implementing the IDS in IoT environment and finally conclusion is presented in section 7.

2. LITERATURE REVIEW

The literature related to the security challenges of IoT and the IDS available for detecting malicious events/and attacks are presented below.

2.1. IoT and Security

In their article, Patel and Patel [7] discussed the definition, characteristics, technologies, architecture, and applications of IoT and also highlighted research issues and challenges regarding security, interoperability, data management, and energy issues in a nutshell. According to their survey, security and privacy issues are the most challenging tasks in the IoT. Among all the security issues, secure data communication and the quality of shared data are the predominant issues to be considered for research.

Adat and Gupta [8] conducted a thorough examination of the evolution of the Internet of Things, related works, IoT statistics, IoT architecture, and security concerns. The authors provided a set of layer-wise security challenges and security requirements for the IoT architecture. They also presented a classification of security issues and existing defence mechanisms for the IoT environment. As per the paper, network security issues and attacks cause more damage to the IoT eco-system.

Tewari, and Gupta [9] provided an overview of the security challenges associated with the IoT layered architecture. The security issues in traditional networks and IoT networks are compared and discussed. Heterogeneous integration of cross layers and their associated challenges are also analysed in this paper, and some future directions are highlighted. Though the aim of the paper is to present the security and privacy issues of the IoT, they have not been given much focus in this paper.

Sahay et al. [10] suggested an Attack Graph for identifying the susceptibilities of the rank of nodes. By mistreating these vulnerabilities, an intruder could invoke several attacks, compromising network traffic, optimizing and isolating the network, and consuming more resources. The impact of the attacks was claimed only by using some qualitative measures. The results are not quantified.

Based on the IoT architecture and layers, Deogirikar and Vidhate [11] classified all possible attacks related to IoT into physical layer-related attacks, network layer-related attacks, software-related attacks, and encryption-related attacks. A comparative analysis was also performed based on the harmful effects of the attacks, possibilities for detection, vulnerability, and location of the attacks. The layer-wise attacks and advantages and disadvantages of the attack detection techniques were also discussed elaborately. Security solutions are not considered in this paper.

Sfar et al. [12] offered an overview of the IoT security roadmap based on a systematic and cognitive approach. A case study is also given to explain this approach. Various research challenges are also classified based on access control, privacy, trust, and identification. The classified elements were not explained in this paper.

2.2. IDS for Internet of Things

Hemdan and Manjaiah [13] described how IoT and IDS are useful in cybercrime investigation, as well as how to use IDS data to analyse criminal behaviour and make decisions based on the findings. Here, the authors have explained only their theoretical views and ideas.

Fu et al. [14] proposed an innovative idea for IDS using Automata. The evaluation of this IDS was performed on a Raspberry Pi device with the help of an Android mobile phone. This IDS successfully detected the jam-attack, false-attack, and replay-attack. This Intrusion Detection System detected only these three types of attacks. Some problems may also arise while running the system out of resources.

Raza et al. [15] offered Hybrid-IDS suitable for the IoT environment to detect real-time sinkhole and selective forward attacks. It was named 'SVELTE'. The authors attempted to improve performance in this study by balancing the costs associated with signature and anomaly-based IDS. In SVELTE, the border router processes intensive IDS modules by analysing the network data. The IoT devices are accountable for transmitting the data to the border router and alerting the router about the abnormal data they receive. Periodic updating of the database is required in order to make the IDS relevant to the current attack patterns.

The above work was extended by Shreenivas et al. [16] by including an IDS module that uses a metric called Expected Transmission Count (ETX) of RPL networks. They suggested the intruders' activities in the 6LoWPAN network can be prevented and the location of the attacker nodes can be identified by monitoring the ETX metric. The true-positive rate is increased in their work by combining the ETX based

REVIEW ARTICLE

rank mechanism with the rank-only approaches. Since there is an additional ETX module in this work, it requires more storage and computational overhead.

Mbarek et al. [17)] presented an Enhanced Network IDS protocol for the Internet of Things (ENIDS) to detect the clone attack. This protocol was evaluated with the performance of SVELTE and outperformed in terms of detection probability and energy consumption. This ENIDS is limited to clone attacks, and in the normal scenario it consumes more energy.

Ioulianou et al. [18] offered a Hybrid IDS using signature-based concepts for IoT architecture. Using the Version Number modification and 'hello-flood' attacks, a Denial of Service (DoS) attack was launched. The impact of the attacks was analyzed in terms of battery-power usage and reachability of nodes. The Intrusion Detection functionalities are not taken into account in this research work.

All possible attacks in the IoT environment are either passive or active. Passive attacks simply monitor the system activities and data traffic and eavesdrop to recover information. They are less dangerous and cause less damage to IoT devices and networks. Active attacks are dissimilar to passive attacks, and these attacks cause damage to the IoT infrastructure directly [19]. These attacks can circumvent smart devices and the IoT ecosystem, resulting in the loss of valuable data.

Using the IoT reference model, Abdul-Ghani et al. [20] conducted a thorough investigation on IoT attacks. Physical, protocol, data, and software attacks against IoT networks were characterised by the researchers. A detailed description of all conceivable attacks in these areas is provided. This article does not go through the security solutions. A summary of current research on security threats on IoT networks was provided by Lu and Xu [21]. Based on IoT devices, device location, access level, data damage degree, node capacity, and protocol, they created a taxonomy of cyber security attacks on IoT networks. They also eloborated the four-layer security architecture for IoT. The attacks on each layer, and the security solutions however, are not described in depth.

Ramakrishna et al. [22] conducted an analytical assessment on various forms of IoT threats and their security solutions. Physical, side-channel, cryptanalysis, software-based, and network-based attacks were all identified as IoT security attacks in this study.This paper only looked at a few attacks from each category, as well as available countermeasures.

2.3. Machine Learning and Deep Learning based IDS

For the Wireless Sensor Networks (WSN) nodes with low resources, Qu et al. [23] proposed a lightweight, fuzzy clustering-based Intrusion Detection System. The sensor data collected at the base stations were used to map the network state. To build this system, the authors combined the sliding window technique with fuzzy c-means and one-class SVM. This system was capable of quickly detecting the assaults. The EXata Network Simulator was used to test the system's efficacy. Although it is capable of identifying and detecting communication-destructive assaults, it might be enhanced in terms of recognising multiple attacks.

In a comparative study, Biswas [24] explained various feature selection techniques and machine learning classifiers for developing IDS. The classifiers used in this research are Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and Neural Networks (NN). The Correlation-based Feature Selection method (CFS), Information Gain Ratio (IGR), Minimum Redundancy Maximum Relevance method, and Principal Component Analysis (PCA) feature selection techniques were evaluated. The NSL-KDD dataset with 10,000 tuples with 40 attributes was used for this analysis. According to this study, K-NN (K-Nearest Neighbor) and information gain ratio-based feature selection (GIR) provided a better result. The NSL-KDD is one of the very old datasets for intrusion detection, so it is not suitable for IoT.

Using the AdaBoost ensemble approach, Moustafa et al. [25] created an IDS for detecting intrusions in IoT networks. To improve performance, ensemble models are created by integrating numerous classifiers. Three classifiers, namely Artificial Neural Network (ANN), Naive Bayes (NB), and Decision Tree (DT), are merged in an ensemble technique to produce this model.The botnet was mostly identified using this strategy against application layer protocols. It's also confined to the three protocols, and should be extended to include features from more IoT protocols.

Jan et al. [26] proposed a lightweight IDS based on an SVM classifier to detect attempts to inject unnecessary data into IoT networks. The packet arrival rate's Poisson distribution was used to differentiate the packets as benign or intrusive. A subset of the CICID2017 dataset was selected, obtaining a synchronized beget dataset from that subset, which was further utilized in this research. The packet arrival rate is the only attribute considered in this experiment. It supports the lightweight aspect of IDS, but only a single attribute from a huge dataset will not detect all possible attacks.

Eskandari et al. [27] suggested an anomaly-based IDS termed Passban IDS for detecting intrusions at the edge level based on security attacks. Real-time network traffic was gathered to detect the attacks, and the iForest ensemble technique was used in this methodology. This Passban IDS detected the port scanning, brute force attacks, and SYN flooding attacks. The attacks during the training phase were not considered in this research. The SYN Flood attacks in this work will consume more resources and will reduce the detection accuracy of the Passban IDS.

REVIEW ARTICLE

Alkadi et al. [28] recommended distributed IDS using Deep Blockchain technology and Bidirectional Long Short-Term Memory (BiLSTM). This system detected the DoS, DDoS, port scanning, and other attacks in UNSW-NB15 and BoT-IoT datasets effectively. It is suitable for IoT and cloud architecture. For real-world implementation, it requires further fine-tuning. The UNSW-NB15 dataset used in this research was not specific to IoT.

Cheema et al. [29] introduced a Blockchain based IDS for IoT using Machine Learning Algorithms. The IoT network is divided into number of Autonomous Systems (AS). The selected AS nodes are responsible for traffic monitoring in a distributed manner. The SVM algorithm is applied for training the dataset. This system detects the Botnets and routing attacks. Since the Blockchain module handles all attackers' associated details, it increases the computational complexity for each transaction. The lightweight features should be addressed before incorporating it into IoT networks.

Parra et al. [30] suggested a distributed attack detection technique for the IoT using Deep Learning algorithms using a cloud-based approach. It comprises two security mechanisms, such as a Distributed Convolutional Neural Network (DCNN) and a cloud-based temporal Long-Short Term Memory (LSTM) model. The proposed mechanism detects phishing attacks, DDoS attacks, and botnets. This method can detect the attack at both the node and the cloud level. The network layer-related attacks are not considered in this research.

Alsoufi et al. [31] investigated anomaly-based IDSs for the IoT using deep learning approaches. Different databases and journals having deep learning-based IDS were identified in this study. The algorithms used for anomaly-based IDS, such as supervised, unsupervised, and semi-supervised algorithms, were reviewed. Although the authors aimed to review anomaly attacks in the IoT, most of the datasets taken for the study are not specific to the Internet of Things.

Kumar et al. [32] offered an ensemble distributed IDS model to safeguard the IoT network from different types of security attacks. The Gaussian Naïve Bayes, KNN, Random Forest, and XGBoost algorithms were applied to develop the ensemble model. The UNSW-NB15 and DS2OS were the datasets used in this research to examine the IDS's performance. The model is built for detecting attacks in IoT environments. But in the experimented datasets, DS2OS is the only dataset specific to the IoT. Though there is much ongoing research and development in the security of IoT by implementing Intrusion Detection Systems, it is still needed to enhance the security level further by using innovative tools and techniques.

3. SECURITY ATTACKS IN IOT

The security related threats and vulnerabilities rise robustly as the connected devices in IoT increase. The IoT nodes create dynamic topology and the nodes perform their tasks without human intervention, so that, handling the security issues in IoT becomes more complex. The privacy and security challenges of IoT become more troublesome with the limited resources. Moreover, the enormous growth and adoption of IoT devices in all aspects of human life indicate the necessity of considering these security threats before the implementation of the countermeasures. The security market from 2019 to 2025 is given in Figure 3.

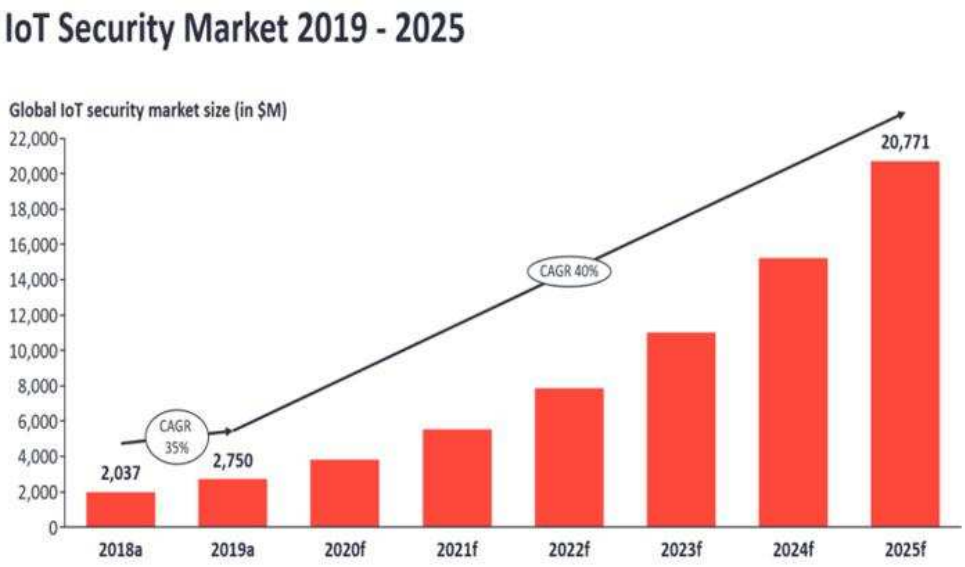


Figure 3 IoT Security Market (2019-2025) (Source: IoTAnalyticsResearch 2020)

REVIEW ARTICLE

According to IoT Analytics Research 2020, the IoT security market size was $2,750 million in 2019, and it is estimated to be the same as $20,771 million in 2025. The increase in the compound annual growth rate (CAGR) is 40% from the year 2019 to 2025. This emphasises the rapid growth of security challenges in IoT and the importance of securing the devices against various attacks. Intrusions or attacks on any network can be caused in three ways:

- Attacks are targeted by external attackers after gaining access to any network, and then the systems explore various malicious activities against the network.
- Internal attackers who have been granted a certain level of privilege but attempt to launch attacks using additional unauthorised access.
- Authorized internal attackers misuse the privileges given to them.

3.1. External Attacks

External attacks are initiated from outside of the networks. By acting as insiders, the external attackers inject malicious code during data communication. The attackers access the smart devices of the IoT devices remotely and attempt various types of attacks against the IoT networks.

3.2. Internal Attacks

Internal attacks are initiated by the authorized people of the IoT network. They misuse their given privileges as well as pretend that they have other privileges which they may not be granted. In this attack, the attacker tries to inject and run abnormal codes on the nodes without the user's awareness in this attack. IDSs protect the IoT network and devices in real-time from external and internal security threats and attacks [33].

4. TYPES OF IDS FOR IOT

Intrusion Detection Systems are used to discover intrusions, attacks, and malicious activities in the IoT environment. IDSs are networking security components that are widely used to protect network environments from attacks and malicious activities. They normally monitor the behaviour of the individual device or the network. Intrusion Detection Systems for the Internet of Things are classified into two categories:

- IDS types based on their positions
- IDS types based on their techniques

The classifications of IDS used in this review are illustrated using Figure 4.

The first category is based on where the Intrusion Detection System is located in the IoT network. The second category of classification is based on the techniques used for implementing the IDS. Each type is explained in detail.

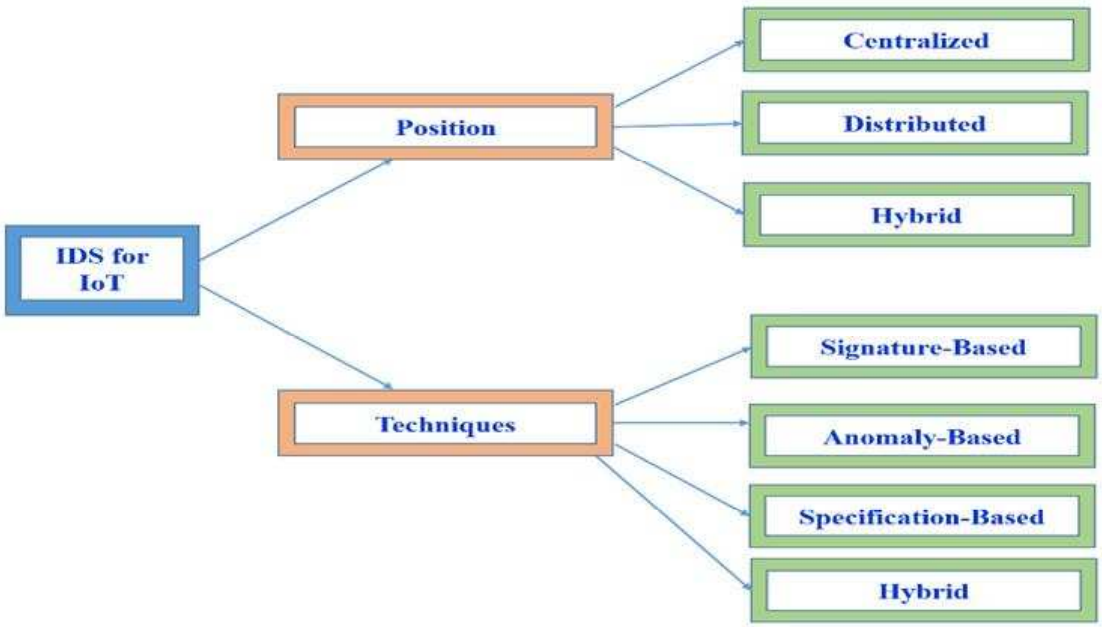


Figure 4 Types of IDS for IoT

REVIEW ARTICLE

4.1. IDS Types Based on its Position in the Network

There are several types of IDSs, each of which is classified differently. The IDS can be installed on the border router, selected nodes, or every node in the IoT network in the IoT ecosystem. Intrusion detection systems are classified into three types based on this deployment strategy: distributed IDS, centralized IDS, and hybrid IDS.

4.1.1. Distributed IDS (Host-based IDS)

Each node in the IoT network is responsible for monitoring and detecting the attacks in this distributed deployment method. As a result, the intrusion detection system is installed on nearly all nodes in the network. The attacks are detected in a distributed manner by the IDSs [34]. The resource-constrained properties of the IoT should be examined and optimised since the intrusion detection system is installed on each node. To deal with this problem, a variety of approaches have been devised.

Oh et al. [35] devised a lightweight approach for identifying assaults by comparing packet payloads and attack patterns. Auxiliary shifting and early decision, according to the authors, are required to minimize the number of matches required to identify attacks. This attack detection system skips a large volume of data that are not relevant for detecting the attacks.

The authors claim it is a lightweight system since it reduces the memory requirements and computational costs. Sometimes, the reduction of memory for pattern matching also degrades the detection accuracy of the system.

Lee et al. [36] suggested a lightweight distributed IDS for detecting Denial of Service (DoS) attacks in 6LowPAN networks. In this approach, the malicious node is identified using the battery power consumption of an IoT device. The authors considered only a single node as the parameter in their research work.

In distributed IDS settings, some nodes also have an additional responsibility to monitor their neighbours and such nodes are called watchdogs.

Mehmood et al. [37] developed a multi-agent IDS using Naïve Bayesian algorithm for detecting the probable distributed Denial of Service (DDoS) attacks in IoT layered architecture. In this work, the multi-agents along with Naïve Bayesian algorithm were implemented in selected IoT devices throughout the network. The agents were classified as system monitoring, communicating, collector, and actuator agents. The distributed multi-agents in this approach share the responsibility of intrusion detection and reduce the workload of the individual nodes. The agent nodes could communicate with other agents too, whenever required. The authors used sensors to gather the information, and the collected information was analyzed to check whether there were any attacks on the network. Malicious nodes and their activities were monitored and reported to the administrator or to the IoT objects. The authors did not consider low-capacity systems in their approach. Though the authors claim that it is suitable for IoT, it is only relevant for Wireless Sensor Networks (WSN) and Mobile Ad-hoc Networks (MANET) as they implemented these networks only in the NS2 Simulator and gave the simulated results.

Cervantes et al. [38] proposed a distributed solution named "Intrusion Detection of Sinkhole Assaults on 6LoWPAN for InterneT of Things (INTI)" that monitors, detects, and mitigates the attacks by merging the concepts of trust and status with watchdogs. Different types of nodes, such as associated, leader, and member nodes, were used to create a hierarchical structure. A change in the network, such as network reconfiguration or the occurrence of an attack, might cause the node to change its role. After then, each node keeps track of a superior node's incoming and departing traffic. When a node detects an attack, it notifies the other nodes, and the attacker node is isolated. The effectiveness of the tool in low capacity nodes is not deliberated by the authors. Since the distributed IDSs have a hierarchy among themselves, this type of IDS can be termed as Hierarchical Intrusion Detection System.

By deploying the open-source Snort tool on the Raspberry Pi device, Sforzin and Conti [39] developed a distributed IDS termed RpiDS. The Raspberry Pi is considered the core commodity for this system. It was implemented in a smart home. The performance of the Raspberry Pi is evaluated as a host of the snort tool. Though this RpiDS is capable of hosting Snort, due to its constrained nature, it is very hard to monitor and manage the attacks in a large-scale implementation.

4.1.2. Centralized IDS (Network IDS)

In this strategy, intrusion detection systems are installed on a centralized router or a dedicated server. Because of the centralized edge node, i.e., border router, which connects the IoT network to the Internet, implementing centralized IDS in IoT is very simple. Because data packets from the outside enter the IoT environment through the border router, external attackers may be quickly recognised by the centralised IDS. Hence, when the intrusion detection system is deployed in the border router, it can easily monitor, analyze, and drop the malicious data packets when it detects any attacks. Contrarily, internal attack detection is difficult in this approach since it necessitates thorough monitoring and analysis of all internal nodes connected to the border router.

Midi et al. [40] developed a centralized Intrusion Detection System for an IoT environment called “Knowledge-driven Adaptable Lightweight Intrusion Detection System (KALIS)”.

REVIEW ARTICLE

It can be deployed as a standalone tool on any specialized external device or in a centralised installation setting like a router. KALIS acquires knowledge about the characteristics of network entities on its own and uses it to dynamically create a set of detection algorithms. In compared to standard intrusion detection systems, KALIS excelled in identifying DoS, routing, and conventional attacks, according to the authors. This system is not tied to any particular protocol or architecture. Though the KALIS system outperforms traditional IDS in terms of performance, it requires more memory to deploy than the traditional IDS.

Wani and Revathi [41] recommended an innovative IDS using Software Defined Networking (SDN). It is programmable, so it makes the network flexible. Here, a centralized controller is moved to develop a global control system. The authors implemented their work in Mininet2.0. They achieved 99% accuracy in their result. In this research, the authors considered only the flooding attack. The NSL-KDD dataset is used in this research, which is a very old dataset and it is not specific to Internet of Things related attacks. The experiment and methodology are not explained in detail.

4.1.3. Hybrid IDS

By analysing the pros and cons of the centralized and distributed placement strategies, the hybrid placement strategy is developed. In this hybrid IDS, the strengths of both strategies are included and the drawbacks are excluded.

Using the hybrid strategy, Amaral et al. [42] proposed a hybrid intrusion detection system. In this work, selected nodes act as watchdogs (Distributed IDS) to detect intrusions caused by eavesdropping on their neighbours. According to the defined security rules, the watchdogs determine whether there is any attack on the network. Each watchdog has a different rule-set based on the behaviour of the components in the network. According to the security rule-sets in the centralized IDS, the patterns are identified from the monitored messages. Thus, a hybrid approach is used in this work. The flexibility of using a different set of rules is the main advantage of this system. The rule-set has to be updated very often in order to make the system up-to-date for new attacks. Dynamic attack detection is not possible in this IDS as it has some predefined set of rules..

Thanigaivelan et al. [43] developed a hybrid attack detection system for internal anomalous activities. It was used to monitor and evaluate their neighbors within a one-hop distance and to report them to their parents only when it detected an anomaly. When an intrusion is detected, the monitoring node is isolated, and data packets are discarded in the link layer to avoid unnecessary network overhead. The system also included a fingerprinting function that allowed the border router to detect network changes and locate the source of the threats. The router and other nodes were given different tasks and they were coordinated. This system is capable of detecting and banning flooding attacks, selective forwarding attacks, and clone attacks. This system is quite complex to handle, and it mainly focuses on limited types of attacks only.

4.2. IDS Types Based on its Techniques

There are many algorithms for detecting intrusions and improving the performance of the IDS. These algorithms and techniques can be applied in various stages of intrusion detection. Based on the techniques and methods implemented along with it, the IDSs are grouped into four types: signature-based, anomaly-based, specification-based, and hybrid IDSs.

4.2.1. Signature-Based IDS

This kind of intrusion detection system is also termed as a "Misuse-based IDS". All possible known attack patterns are stored in the IDS database. These IDSs analyse the generated information and find out whether there is any match with the known attack. This type of IDS is very effective against known attacks. It needs a periodic update because the efficiency of this system depends on attack signatures available in the database [44]. Although it gives a higher true-positive rate, it is incapable of detecting new patterns of attacks.

Kumar et al. [45] proposed a unified IDS (UIDS) for detecting DoS attacks, probe attacks, generic attacks, and exploit attacks. The decision tree algorithm is applied to the UNSW-NB15 dataset. Various forms of rule sets are defined in order to develop the system. This signature-based IDS detects the attacks more effectively than the existing research work. It needs further refinement to detect new attacks. The dataset used in this research is not specific to IoT. It is difficult to detect unknown attacks using this approach.

4.2.2. Anomaly-Based IDS

This kind of IDS can classify the behavior of the system as abnormal or anomalous. This categorization is based on rules or heuristics rather than patterns or signatures. First, the IDS should be trained to understand the normal behavior of the system. If there is any activity that violates the normal behavior, then the IDS can identify it as an attack. This type of IDS detects unknown attacks effectively. However, it considers everything an intrusion, which means it is deviating from the normal behavior. Therefore, anomaly-based intrusion detection systems normally have higher false-positive rates than other types of IDSs [46]. In general, to train the normal behavior of the systems, machine learning algorithms can be used. But implementing machine learning for the resource-constrained IoT nodes is a challenging research issue. The lightweight aspects should be considered in such cases.

REVIEW ARTICLE

Ulla and Mahmoud [47] proposed an anomaly detection system for IoT networks using deep learning. The Convolutional Neural Network algorithm was the backbone of this research. The proposed IDS model was evaluated using IoT-related IDS datasets such as BoT-IoT, IoT-DS-2, IoT-23, and MQTT-IoT-IDS2020. This multiclass model detects various attacks like DoS, DDoS, flooding attacks, OS Scan, Port Scan, Mirai, etc. efficiently in terms of accuracy and other metrics. Multiple IDS datasets were combined in this research for the purpose of developing the model. The deep learning approach and the multiple data sources require more training time and computational costs.

4.2.3. Specification-Based IDS

This kind of intrusion detection system is also called "Rule-based IDS". These IDSs contain a rule-set and some thresholds associated with the rule-set. These rules are defined by the experts regarding the normal and abnormal activities of the nodes and protocols in the networks. Like anomaly-based IDS, these IDSs also detect attacks whenever there is a deviation from the specified thresholds and rules. In specification-based IDS, the rules and thresholds are set by the human experts, but in anomaly-based IDS, the system should be trained. This is the difference between these two types of IDSs. Since there is human involvement in these IDSs, they have a lower false-positive rate compared to the anomaly-based IDSs [48]. The specification-based IDSs are not flexible and error-prone due to the manually defined specifications. Periodic upgrading of the rules and thresholds is essential to make the system relevant for current needs.

Astillo et al. [49] recommended a specification-based system to detect the malicious acts of an implanted Artificial Pancreas System (APS) which maintains the blood glucose level of the human body. In this research, the security challenges and associated risks related to patients' health and safety were studied. The behavior-rules of the APS were defined. The UVa/Padova simulator was used to emulate the functionalities of APS. SVM and kNN are the classifiers used in this research to validate the proposed model. The recommended system monitors the components of the APS continuously, and abnormal glucose levels are identified with better accuracy. Since it is related to human life, better refinements should be required. The behavior-rules of the APS have to be updated in order to include new symptoms that lead to abnormalities in blood glucose levels.

4.2.4. Hybrid IDS

Hybrid IDSs are developed by combining one or more of the aforementioned types of IDSs. These IDSs are established to optimize the performance by minimizing the drawbacks and maximizing the advantages of these IDSs. By merging the merits of such IDSs, the detection accuracy and the performance of the hybrid IDS are enhanced.

By using the Map Reduce approach and the unsupervised Optimum-Path Forest (OPF) algorithm, Bostani et al. [50] developed a hybrid IDS with anomaly and specification- IDS. Based on their experimental results, the authors defend that their IDS performed well by reducing false-positives and increasing true-positives. This hybrid system is suitable for detecting sinkhole and selective forwarding attacks in IoT networks. This system has its own limitations in unsupervised learning and the Map-Reduce approach. The raw data packets from the simulated Wireless Sensor Networks (WSN) are used in this research. Hence, the dataset used in this research is not specific to the Internet of Things.

5. ANALYTICAL SURVEY OF IDS FOR IOT

The Table 1 shows the summary of the reviewed literature. Here, IDS research work, the type of IDS it belongs to, techniques used in the IDS, advantages, and the research gaps of these IDSs are briefly given.

Research	IDS Type	Techniques/Tools	Attack Detection	Required Refinements
Fu et al. [14]	Centralized	Automata	jam-attack false-attack replay-attack	State-space problem
Raza et al. [15]	Hybrid	SVELTE	Sink-hole attacks	Additional Control overhead due to 6Mapper module
Shreenivas et al. [16]	Hybrid	Extension to SVELTE using ETX metric, the geographical detection algorithm	ETX and Rank attack	Maximum 8 nodes only used.
Mbarek et al. [17]	Centralized	ENIDS protocol	Clone attacks	Consumes more energy in normal scenario
Ioulianou et al. [18]	Hybrid	Cooja Simulator,	DoS	IDS functionalities are

REVIEW ARTICLE

		Pattern Matching Algorithm		not considered
Qu et al. [23]	Hybrid	Sliding window Protocol, One-Class SVM, Fuzzy C-Means	Anomalous events and routing attacks	Refinements required for diversity of attacks
Moustafa et al. [25]	Centralized	AdaBoost ensemble method	Botnet attacks	Limited to three IoT application layer protocols
Jan et al. [26]	Centralized	SVM classifier	DDoS attacks	Single attribute only used
Eskandari et al. [27]	Centralized	Passban IDS, iForest	Port Scanning, Brute force, flooding attack	Not considered the attacks in the training phase, flooding attack reduces the detection rate
Alkadi et al. [28]	Distributed	Blockchain, Bidirectional Long Short-Term Memory (BiLSTM)	DoS, DDoS, Port Scanning, OS Scan etc.	Need further refinement for real-time implementation
Cheema et al. [29]	Distributed	Blockchain, Spectral Partitioning	Routing attacks and Botnet	Real-world conditions should be addressed
Parra et al.[30]	Distributed	Deep Learning	Phishing, DDoS, Botnet	More training time
Kumar et al. [32]	Distributed	Ensemble	Backdoor, Reconnaissance, DoS	Real-time deployment requires lightweight mechanisms for IoT nodes
Oh et al. [35]	Distributed	auxiliary shifting, early decision	Conventional attacks using signatures	Single device only
Lee et al. [36]	Distributed	Energy consumption models	Routing attacks, DoS	Single device only
Mehmood et al. [37]	Distributed	Naïve Bayes Algorithm, Multi-agent	DDoS Attack	Low capacity systems are not considered
Cervantes et al. [38]	Hierarchical - Distributed	INTI	Sinkhole attacks	Low capacity systems are not considered
Sforzin and Conti [39]	Distributed	Snort tool	Conventional Attacks	Single Node is considered
Midi et al. [40]	Centralized	KALIS	DoS, Routing attacks	Complex functionalities
Wani and Revathi [41]	Centralized	Software-Defined Networking (SDN)	Flooding attacks	Only flooding attack is considered
Amaral et al. [42]	Hybrid	Watchdogs	Routing attacks based on a different set of rules	Requires optimization in enforcing and storing new security rules

REVIEW ARTICLE

Thanigaivelan et al. [43]	Anomaly-based, Hybrid	Network fingerprinting	Clone, Flooding, selective forward	Complex to handle
Kumar et al, [45]	Centralized Specification-based IDS	Decision Tree	Exploit, DoS, Probe, Generic	Requires refinement for detecting new attacks.
Ulla and Mahmoud [47]	Anomaly-based	Convolutional Neural Networks	Dos, DDoS, Mirai, Flooding, Port Scan	Training takes more time
Astillo et al. [49]	Centralized Specification-based	UVa/Padova simulator, SVM, KNN	Abnormal blood glucose level	Human life related. Periodic update required
Bostani et al. [50]	Hybrid	Optimum-Path Forest (OPF), Map Reduce Algorithm	Sinkhole, wormhole, selective forward attack	Simultaneous different types of attacks reduce the performance

Table 1 Intrusion Detection Systems for IoT

According to this review, when machine learning algorithms are deployed, the performance and efficiency of the intrusion detection systems will be better and the hybrid IDS will provide better accuracy, which reduces false positives and improves the true positives.

6. RESEARCH DIRECTIONS BASED ON THE REVIEW

The IoT has evolved from the traditional network architecture. Hence, it also incorporates all the vulnerabilities and threats associated with traditional networks. As IoT is connected to the global network, all the security issues that lie on the Internet also propagate to the IoT environment. The following are the reasons for various security-related issues in the IoT environment:

- The devices in IoT networks are resource-constrained; they have less memory, processing power, and limited energy.
- Voluminous IoT devices from heterogeneous sources are linked to the Internet, which tends to make the IoT more vulnerable.
- IoT devices use different technologies and platforms. Hence, providing interoperability among such devices is a challenging issue.

These issues make the IoT vulnerable and cause serious damage like data breaches and tampering of IoT nodes. If the nodes are compromised, then the security risk will rise to a higher level. Cryptography is one of the technologies used to secure data. Here, secure keys are the core elements. But, when the attacker compromises the internal nodes to get the security keys, preventing the network from attacks is not possible. In such a scenario, IDSs are a boon for providing security to the IoT networks. Therefore, it is essential to have an intrusion detection system to monitor the IoT network and detect the attacker and compromised IoT devices.

IDSs have been used in traditional network and information systems for more than two decades. The usage of IDS and its implementation in IoT compared to traditional networks is still in the initial stage. Moreover, current IDS solutions for the IoT are not sufficient. The research gaps for deploying intrusion detection systems in IoT networks are given below:

- The intrusion detection systems used in traditional networks are heavyweights, which mean they will not be suitable for resource-constrained IoT networks. The lightweight aspects in terms of processing, memory, and battery power consumption should be considered for developing IDS for the IoT.
- In traditional network, once the connection is established, there will be an end-to-end data transmission. But in the IoT network, the data packets traverse multi-hops from the sender to the receiver. Hence it is more vulnerable. The connectivity and link stability issues of the IoT network should be kept in mind when designing IDS for IoT.
- The IoT uses advanced protocols and technology that have their own vulnerabilities in the networks. So, the IDS developed for traditional networks are not applicable in the IoT environment.
- The sensors generate voluminous data. The security aspects of such data and managing such voluminous data also lead to research challenges.

The above facts summarize the issues and challenges of implementing IDS while deploying them in IoT networks.

7. CONCLUSION

One of the most important security tools deployed in traditional networks is the IDS. While implementing the IDS in an IoT environment, the characteristics of the IoT should be considered. The deployment of IDS in the IoT has a lot of

REVIEW ARTICLE

emerging scope and challenges for research. In this paper, the security issues in the IoT, the need for IDS in the IoT, and the different types of IDS for the IoT are reviewed. An analytical survey based on the review is also given. The analysis clearly shows that they did not reach a consensus, implying that additional research and development for IDS in IoT networks is still required. The intrusion detection systems also necessitate periodic refinement to keep the systems suitable for current needs. Hence, it provides a wider scope for IoT security researchers.

REFERENCES

[1] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", Cyber Security 4(18), 2021, DOI: 10.1186/s42400-021-00077-7

[2] A. Colakovi and M. Hadziali, "Internet of Things (IoT): A Review of Enabling Technologies, Challenges, and Open Research Issues", Computer Networks, 2018, DOI: 10.1016/j.comnet.2018.07.017.

[3] E. C. Ugwuabonyi and E.Z. Orji, "Issues and Challenges in Security and Privacy of Internet of Things (IoT)", International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), 7(12), 2018, ISSN 2278-2540.

[4] B. B. Zarpaelo, R.S. Miani, C.T. Kawakani and S. C. Alverenga, "A Survey of Intrusion Detection in Internet of Things", Journal of Network and Computer Applications, 2017, DOI: 10.1016/j.jnca.2017.02.009.

[5] A. Mayzaud, R. Badonnel and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things", International Journal of Network Security", ACEEE a Division of Engineers Network, 18 (3), pp.459-473, 2016, DOI:10.6633/IJNS.201605.18(3), hal-01207859.

[6] T. A. Tchakoucht and M. Ezziyyani, "Building A Fast Intrusion Detection System For High-Speed Networks: Probe and DoS Attacks Detection", Procedia Computer Science, 127, pp. 521–530, 2018.

[7] K.K. Patel and S.M. Patel, "Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application and Future Challenges", International Journal of Engineering Science and Computing, 6(5), ISSN 2321- 3361, 2016, DOI: 10.4010/2016.1482.

[8] V. Adat and B. B. Gupta, "Security in Internet of Things: issues, challenges, taxonomy, and architecture", Telecommunication System, 2017, DOI: 10.1007/s11235-017-0345-9.

[9] A. Tewari and B.B. Gupta, "Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework", Future Generation Computer Systems, 108, ISSN: 0167-739X, pp: 909-920, 2020, DOI: 10.1016/j.future.2018.04.027

[10] R. Sahay, G. Geethakumari and K. Modugu, "Attack Graph based Vulnerability Assessment of Rank property in RPL-6LowPAN in IoT", IEEE Explore, 2018, DOI: 10.1109/WF-IoT.2018.8355171

[11] J. Deogirikar and A. Vidhate, "Security Attacks in IoT: A Survey. International Conference on IoT in Social, Mobile, Analytical and Cloud", I-SMAC- 2017, IEEE, 2017.

[12] A. R. Sfar, E. Natalizio, Y. Challal and Z. Chtourou, "A Roadmap for Security Challenges in the Internet of Things", Digital Communications and Networks, 4, pp.118-137, 2018.

[13] E E. Hemdan and D.H. Manjaiah, "Cybercrimes Investigation and Intrusion Detection in Internet of Things based on Data Science Methods", Cognitive Computing for Big Data Systems over IoT, 2018, DOI: 10.1007/978-3-319-70688-7_2.

[14] Y. Fu, C. Yan, J. Cao, O. Kore and X. Cao, "An Automata based Intrusion Detection method for Internet of Things", Mobile Information Systems, Hindawi Publications, 2017(1750637), 2017, DOI: 10.1155/2017/1750637.

[15] S. Raza, L. Wallgren and T. Voigt, "SVELTE: real-time intrusion detection in the Internet of Things", Ad Hoc Network, 11(8), ISSN: 2661-2674, 2013, DOI:10.1016/j.adhoc.2013.04.014.

[16] D. Shreenivas, S. Raza and T. Voigt, "Intrusion Detection in the RPL connected 6LoWPAN Networks", Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, IOTPTS'17, Abu Dhabi, United Arab Emirates, 2017.

[17] B. Mbarek, M. Ge and T. Pitner, "Enhanced Network Intrusion Detection System Protocol for Internet of Things", Proceedings of ACM SAC Conference (SAC'20), ACM, New York, Article 4, 2020, DOI: 10.1145/3341105.3373867.

[18] P. P. Ioulianou, V. G. Vassilakis, I.D. Moscholios and M. D. Logothetis, "A Signature-based Intrusion Detection System for the Internet of Things", International Conference on Information and Communication Technology Forum (ICTF-2018) ,Graz, Austria, 2018, https://www.researchgate.net/publication/ 326376629.

[19] P. Wanda and H. J. Jie, "A survey of Intrusion Detection System", International Journal of Informatics and Computation (IJICOM) 1(1), ISSN: 2685-8711, 2019.

[20] H. Abdul-Ghani, D. Konstantas and M. Mahyoub, "A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Model", International Journal of Advanced Computer Science and Applications, Springer, 9(3), 2018.

[21] Y. Lu and L.D. Xu, "Internet of Things (IoT) Cyber Security Research: A Review of Current Research Topics", IEEE Internet of Things Journal, 2018, DOI: 10.1109/JIOT.2018.2869847.

[22] C. Ramakrishna, G.K. Kumar, A.M. Reddy and P. Ravi, "A Survey on various IoT Attacks and its Countermeasures", International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), 5(4), ISSN: 2394-2320, 2018.

[23] H. Qu, L. Lei, X. Tang and W. Ping, "A Lightweight Intrusion Detection Method Based on Fuzzy Clustering Algorithm for Wireless Sensor Networks", Advances in Fuzzy Systems, Article ID: 4071851, 2018, DOI: 10.1155/2018/407185.

[24] S. K. Biswas, "Intrusion Detection Using Machine Learning: A Comparison Study", International Journal of pure and Applied Mathematics, 118 (19), pp.101-114, ISSN: 1311-8080 (print); ISSN: 1314-3395 (online), 2018.

[25] N. Moustafa, B. Turnbull and K. R. Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things", IEEE Internet of Things Journal, 2018, DOI:10.1109/JIOT.2018.2871719.

[26] S. U. Jan, S. Ahmed, V. Shakov and I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things", IEEE Access, 2019, DOI: 10.1109/ACCESS.2019.2907965.

[27] M. Eskandari, Z. H. Janjua, M. Vecchio and F. Antonell, "Passban IDS: An Intelligent Anomaly Based Intrusion Detection System for IoT Edge Devices", IEEE Internet of Things Journal, pp. (99):1-1, 2020, DOI: 10.1109/JIOT.2020.2970501.

[28] O. Alkadi, N. Moustafa, B. Turnbull and K. R. Choo, "A Deep Blockchain Framework-enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks", IEEE Internet of Things Journal, 2020, DOI:10.1109/JIOT.2020.2996590.

[29] M. A. Cheema, H. K. Qureshi, C. Chrysostomou and M. Lestas, "Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things", 16th International Conference on Distributed Computing in Sensor Systems (DCOSS-2020), IEEE Xplore, 2020, DOI: 10.1109/DCOSS49796.2020.00074.

[30] G. D. L. T. Parra, P. Rad, K. R. Choo and N. Beebe, "Detecting Internet of Things Attacks using Distributed Deep Learning", Journal of Network and Computer Applications, 163(102662), ScienceDirect, 2020, DOI: 10.1016/j.jnca.2020.102662.

[31] M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed and M. Nasser, "Anomaly-based Intrusion Detection Systems in IoT using Deep Learning", Applied Sciences, 11(18), 8383, 2021,DOI:10.3390/app11188383.

REVIEW ARTICLE

[32] P. Kumar, G. P Gupta and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks", Journal of Ambient Intelligence and Humanized Computing, 12, pp. 9555–9572, 2020, DOI:10.1007/s12652-020-02696-3

[33] L. Santos, R. Gonçalves, C. Rabadao and J. Martins, "A flow-based intrusion detection framework for internet of things networks", Cluster Computing, Springer, 2021, DOI: 10.1007/s10586-021-03238-y

[34] E. Benkhelifa, T. Welsh and W. Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Towards Universal and Resilient Systems", IEEE, 2018, DOI:10.1109/COMST.2018.2844742.

[35] D. Oh, D. Kim and W. W. Ro, "A Malicious Pattern Detection Engine for Embedded Security Systems in the Internet of Things", Sensors, 14 (12), ISSN: 24188–24211, 2014, DOI: 10.3390/s141224188.

[36] T. H. Lee, T. H. Wen, L. H. Chang, H. S. Chiang and M.C. Hsieh, "A lightweight Intrusion Detection Scheme based on Energy Consumption Analysis in 6LowPAN", Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Lecture Notes in Electrical Engineering 260, Springer Netherlands, pp. 1205–1213, 2014.

[37] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song and M. M. Malik, "NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks", Journal of Supercomputers, Springer Science+Business Media, LLC, Springer Nature, 2018, DOI:10.1007/s11227-018-2413-7

[38] C. Cervantes, D. Poplade, M. Nogueira and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things", IFIP/IEEE International Symposium on Integrated Network Management (IM), pp.606–611, 2015.

[39] A. Sforzin and M. Conti, "RpiDS: Raspberry Pi IDS-A fruitful Intrusion Detection System for IoT", International IEEE Conference on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress, 2016, DOI:10.1109/UIC-ATC-Scalcom-CBDCom-IOP-SmartWorld.2016.114.

[40] D. Midi, A. Rullo, A. Mudgerikar and E. Bertino, "KALIS: A system for knowledge-driven adaptable intrusion detection for the Internet of Things", Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17), 2017.

[41] A. Wani and S. Revathi, "Analyzing Threats of IoT Networks Using SDN Based Intrusion Detection System (SDIoT-IDS)", Smart and Innovative Trends in Next Generation Computing Technologies (NGCT-2017), Springer, CCIS 828, pp. 536–542, 2018.

[42] J. Amaral, L. Oliveira, J. Rodrigues, G. Han and L. Shu, "Policy and Network-based Intrusion Detection System for IPv6-enabled Wireless Sensor Networks", IEEE International Conference on Communications (ICC-2014), pp. 1796–1801, 2014.

[43] N. K. Thanigaivelan, E. Nigussie, S. Virtanen and J. Isoaho, "Hybrid Internal Anomaly Detection System for IoT: Reactive Nodes with Cross-Layer Operation", Security and Communication Networks, Article ID: 3672698, 2018, DOI: 10.1155/2018/3672698.

[44] O. A. Okpe, O. A. John and S. Emmanuel, "Intrusion Detection in Internet of Things", International Journal of Advanced Research in Computer Science, 9(1), ISSN: 0976-5697, 2018, DOI:10.26483/ijarcs.v9i1.5429.

[45] V. Kumar, A. K. Das and D. Sinha, "UIDS: A Unified Intrusion Detection System for IoT Environment", Evolutionary Intelligence, 14, pp. 47–59, 2021, DOI: 10.1007/s12065-019-00291-w

[46] L. Santos, C. Rabadão and R. Gonçalves, "Intrusion Detection Systems in Internet of Things: A Literature Review", ResearchGate, 2018, DOI: 10.23919/CISTI.2018.8399291.

[47] I. Ulla and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks", IEEE Access, 9, e-ISSN: 2169-3536, pp. 103906–103926, 2021, DOI: 1109/ACCESS.2021.309402.

[48] R. Mitchell and I. Chen, "A Survey of Intrusion Detection Techniques for Cyber-physical Systems", ACM Computing Surveys (CSUR), 46 (4), 55, 2014.

[49] P. V. Astillo, J. Jeong, W. C. Chien, B. Kim, J. S. Jang, I. You, "SMDAps: A Specification-based Misbehavior Detection System for Implantable Devices in Artificial Pancreas System", Journal of Internet Technology, 22(1), e-ISSN:2079-4029, 2021, DOI: 10.3966/160792642021012201001

[50] H. Bostani and M. Sheikhan, "Hybrid of Anomaly-Based and Specification-Based IDS for Internet of Things Using Unsupervised OPF Based on MapReduce Approach", Computer Communications, 98(15), pp. 52-71, 2017, DOI:10.1016/j.comcom.2016.12.001.

Authors

A. Arul Anitha is pursuing her Doctoral Degree at St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India, affiliated to the Bharathidasan University, Tiruchirappalli. She received her Master's degree in Computer Applications (MCA) from Manonmaniam Sundaranar University, Tirunelveli, India and her B.Sc in Computer Science from Madurai Kamaraj University, Madurai, India. Her research interests are in computer networking and security, intrusion detection systems, the Internet of Things (IoT), and machine learning. She has published six research articles in reputed journals. She has cleared the National Eligibility Test (NET) for Assistant Professors.

Dr. L. Arockiam is working as an Associate Professor in the Department of Computer Science at St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has published four books and 359 research articles in reputed journals. He has guided more than 34 M.Phil Research Scholars and 30 Ph.D. Research Scholars, and at present he is guiding eight Ph.D. Research Scholars. He received various awards for his academic excellence. His research interests are in the Internet of Things, Cloud Computing, Big Data, Data Mining, Software Engineering, Web Services, and Mobile Networks.

How to cite this article:

A. Arul Anitha, L. Arockiam, "A Review on Intrusion Detection Systems to Secure IoT Networks", International Journal of Computer Networks and Applications (IJCNA), 9(1), PP: 38-50, 2022, DOI: 10.22247/ijcna/2022/211599.

A. Arul Anitha, MCA, NET, (Ph.D).

Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli

Internet of Things
Network Security
Intrusion Detection Systems
Machine Learning Algorithms
Data Mining

	All	Since 2017
Citations	21	20
h-index	2	2
i10-index	1	1

TITLE	CITED BY	YEAR
Annids: artificial neural network based intrusion detection system for internet of things AA Anitha, L Arockiam Int. J. Innov. Technol. Explor. Eng. Regul, 8	12	2019
Network Security using Linux Intrusion Detection System A Anitha International Journal of Research in Computer Science 2 (1), 33	6	2011
VeNADet: version number attack detection for RPL based Internet of Things AA Anitha, L Arockiam Solid State Technology 64 (2), 2225-2237	1	2021
Promoting a Clean and Hygienic Environment Using IoT AA Anitha, L Arockiam International Journal of Recent Technology and Engineering (IJRTE) 8 (5 …	1	2020
A Hybrid Method for Smart Irrigation System AA Anitha, A Stephen, DL Arockaim International Journal of Recent Technology and Engineering (IJRTE), ISSN …	1	
A Review on Intrusion Detection Systems to Secure IoT Networks AA Anitha, L Arockiam International Journal of Computer Networks and Applications 9 (1), 38-50		2022
Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things AA Anitha, L Arockiam International Journal of Advanced Computer Science and Applications 12 (11 …		2021

A Arul Anitha Edit
MCA NET (Ph.D) · Researcher at St. Joseph's College of Tiruchchirappalli
Tiruchirappalli, India | Website
Ph.D Research Scholar

Research Inter... 81.3
Citations 23
h-index 3
Citations over time

View your latest weekly report >

This page is private — only you can see it.

Overall publications stats

9,667	23	17
Reads	Citations	Recommendations
+148 last week	--	—

Reads of your research

Total reads	9,667 (+148)
Publication reads	9,029 (+145)
• Full-text reads	3,862 (+73)
• Other reads	5,167 (+72)
Project reads	14
Question reads	539 (+3)
Answer reads	85

People who read your publications in the last 8 weeks

Read your publication's full-text
Arnaud Rosay
System Architect
Institution and department
STMicroelectronics · Research & Development
Skills

This author profile is generated by Scopus Learn more

Anitha, A. Arul

St. Joseph's College, Tiruchirappalli, Tiruchirappalli, India

https://orcid.org/0000-0002-1256-5418

Metrics overview

4
Documents by author

7
Citations by 7 documents

1
h-index: View h-graph

Document & citation trends

Most contributed Topics 2016–2020

Intrusion Detection System; Network Security; Denial-Of-Service Attack
1 document

Irrigation (Agriculture); Automatic Irrigation Systems; Internet Of Things
1 document

View all Topics

Note:

Scopus Preview users can only view an author's last 10 documents, while most other features are disabled. Do you have access through your institution? Check your institution's access to view all documents and features.

Review • *Open access*

A Review on Intrusion Detection Systems to Secure IoT Networks

Arul Anitha, A., Arockiam, L.

International Journal of Computer Networks and Applications, 2022, 9(1), pp. 38–50

Show abstract Related documents

0 Citations

Article • *Open access*

Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things

Anitha, A.A., Arockiam, D.L.

International Journal of Advanced Computer Science and Applications, 2021, 12(11), pp. 499–506

Show abstract Related documents

0 Citations

Article • *Open access*

A hybrid method for smart irrigation system

Arul Anitha, A., Stephen, A., Arockiam, L.

International Journal of Recent Technology and Engineering, 2019, 8(3), pp. 2995–2998

Show abstract Related documents

0 Citations

Article • *Open access*

ANNIDS: Artificial neural network based intrusion detection system for internet of things

Arul Anitha, A., Arockiam, L.

International Journal of Innovative Technology and Exploring Engineering, 2019, 8(11), pp. 2583–2588

Show abstract Related documents

7 Citations

Back to top

Web of Science Master Journal Li | Web of Science - Researcher ID | My publications | Publons | Clarivate

publons.com/dashboard/records/publication/

Big news! Your Publons™ profile is moving to the Web of Science™. Click here to read more.

publons BROWSE COMMUNITY FAQS

WEB OF SCIENCE

Private Dashboard ▸ Publication Records

Private dashboard - My publications

For Arul Anitha

Web of Science ResearcherID
AAB-6395-2022

- Account
- Profile
- Affiliations
- Email
- Permissions

DATE ▾	TITLE	JOURNAL	CITATI...	ALTMETRIC	
2021-11	Ada-IDS: AdaBoost Intrusion Detection System for IC...	International Journal of A...	0	-	
2021-01-01	VeNADet: Version Number Attack Detection for RPL b...	Solid State Technology		-	
2020-01-30	Promoting a Clean and Hygienic Environment using IoT	International Journal of R...		-	
2019-09-30	A Hybrid Method for Smart Irrigation System	International Journal of R...		-	
2019-09-10	ANNIDS: Artificial Neural Network based Intrusion Det...	International Journal of In...		-	
2011-12-30	Network Security using Linux Intrusion Detection Syst...	International Journal of R...		-	

No more results.

32°C Partly sunny

ENG IN

10:37 AM 25-04-2022