Evaluating and Enhancing the Security of Cyber-Physical Systems using Machine Learning Approaches

by

Mridula Sharma
B.Sc., University of Delhi, India, 1988
M.Sc., Punjab Technical University, India, 2003
M.C.A., Punjab Technical University, India, 2006

A Dissertation Submitted in Partial Fullfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering
University of Victoria

Evaluating and Enhancing the Security of Cyber-Physical Systems using Machine Learning Approaches

by

Mridula Sharma
B.Sc., University of Delhi, India, 1988
M.Sc., Punjab Technical University, India, 2003
M.C.A., Punjab Technical University, India, 2006

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Haytham Elmiligi, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. M. Watheq El-Kharashi, Departmental Member
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Yvonne Coady, Outside Member
(Department of Computer Science, University of Victoria)

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Haytham Elmiligi, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. M. Watheq El-Kharashi, Departmental Member
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Yvonne Coady, Outside Member
(Department of Computer Science, University of Victoria)

ABSTRACT

The main aim of this dissertation is to address the security issues of the physical layer of Cyber Physical Systems. The network security is first assessed using a 5-level Network Security Evaluation Scheme (NSES).

The network security is then enhanced using a novel Intrusion Detection System that is designed using Supervised Machine Learning. Defined as a complete architecture, this framework includes a complete packet analysis of radio traffic of Routing Protocol for Low-Power and Lossy Networks (RPL). A dataset of 300 different simulations of RPL network is defined for normal traffic, hello flood attack, DIS attack, increased version attack and decreased rank attack. The IDS is a multi-model detection model that provides an efficient detection against the known as well as new attacks.

The model analysis is done with the cross-validation method as well as using the new data from a similar network. To detect the known attacks, the model performed at 99% accuracy rate and for the new attack, 85% accuracy is achieved.

Contents

List of Tables

List of Figures

Nomenclature

6LowPAN	Low Power IPv6 Personal Area Network
ACK	Acknowledgement
APT	Accessibility, Position and Type classification
BAN	Body Area Network
BR	Border Router
CAP	Composed Assurance Package
CC	Common Criteria
CPeSC3	Cyber Physical enhanced Secured wireless sensor networks integrated Cloud Computing
CPS	Cyber Physical systems
DAG	Directed Acyclic Graph
DAO	Destination Advertisement Option
DIO	DODAG Information Option
DIS	DODAG Information Solicitation
DL	Deep Learning
DODAG	Destination Oriented Directed Acyclic Graph
DoS	Denial of Service
DTC	Decision Tree Classifier
EAL	Evaluation Assurance Levels

ETX	Expected Transmission Count
FE	Features Engineering
FPR	False Positive Rate
GNB	Gaussian Naïve Bayes
IACS	Industrial Automation and Control Systems
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IEC	International Electro-technical Commission
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv6	Internet Protocol version 6
ISA	Industrial Automation and Control Systems Security
k-NN	K-Nearest-Neighbours
KDD	Knowledge Discovery and Data Mining
LLN	Low Power and Lossy Networks
LRC	Linear Regression Classifier
MAC	Medium Access Control
MCPS	Medical Cyber Physical Systems
ML	Machine Learning
MP2P	Multi Point to Point
NSES	Network Security Evaluation Scheme

Of0	Objective Function
P2P	Point to Point
PAN	Personal Area Network
QoS	Quality of Service
RFC	Random Forest Classifier
RL	Reinforcement Learning
ROLL	Routing over Low-power and Lossy networks
RPL	Routing Protocol for Low-Power Lossy Networks
RSSI	Received Signal Strength Indicator
SAR	Security Assurance Components
SCADA	Supervisory Control And Data Acquisition
SDSE	Sensor Data Security Estimator
SIL	Safety Integrity Level
SL	Supervised Learning
SSH	Secure Shell
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UL	Unsupervised Learning
WAM	Wide Area Monitoring
WAMPAC	Wide Area Monitoring, Protection and Control System
WBAN	Wireless Body Area Network
WSN	Wireless Sensor Network

ACKNOWLEDGEMENTS

For this dissertation work, the required support and encouragement comes from several sources in various ways. In particular, I would like to thank Professor Fayez Gebali for accepting me as a Ph.D. student at the University of Victoria under his supervision. Prof. Gebali's ongoing encouragement, support and belief in me allowed me to grow and learn. His understanding, advice and support have been crucial factors in the successful completion of this work. I consider myself very fortunate to have him as my guide, mentor and advisor.

I would also like to express my deep-felt gratitude to my co-supervisor, Dr Haytham Elmiligi, whose support has been a strong pillar for my development and successful completion of this work. In spite of all his personal issues and constraints, he guided me and pushed me to the level, where I could'nt have reached without his support.

In addition, I would like to thank my good friends, Dr Mila Kwiatkowska and Dr Musfiq Rahman from Thompson Rivers University for having faith in me and supporting me in all my good and bad moments.

I am indebted to my family for their love, advice and support throughout my Ph.D. study, especially to my husband for his support and encouragement, without which this journey was almost impossible. My daughter has not only been supportive and understanding but also an advisor and her contributions to this work are quite considerable.

Chapter 1

Introduction

1.1 Cyber Physical Systems

Cyber Physical Systems (CPS) are the backbone of the personal Internet of Things (IoT) or industrial Supervisory Control And Data Acquisitions (SCADA) applications. In Cyber Physical Systems, communication and computational capabilities are integrated and are in close interactions with the physical world. IoT is all about bridging different CPS's so that information transfers can take place between them [1]. Devices in an IoT network are autonomous, they are connected to each other as well as to physical systems such as grids, automobiles and industrial systems [2]. CPS's three tier architecture is an intrinsic combination of physical and cyber subsystems. The tier 1 of the architecture is the physical subsystems, which are made up of sensors, actuators, RFID etc.. Also known as perception layer, this tier generates data, which can be used by the third tier of CPS architecture i.e. cyber subsystem, where computations are performed for making decision. Using these computations and decisions, the physical subsystem can be controlled. Data transmission between the physical and cyber tier happens through the tier 2 i.e. networking subsystems [3–5]. The core of CPS concepts are the integration of these 3C's: Control, Compute and Communicate [6, 7]. The CPS architecture is shown in Fig. 1.1.

The CPS devices at the physical layer have very specific characteristics like limited battery power, limited processing capacity and storage, and short ranges insecure communication channels. CPS's physical layer always involves real-time constraints and physical phenomena. The communication framework that efficiently manages this

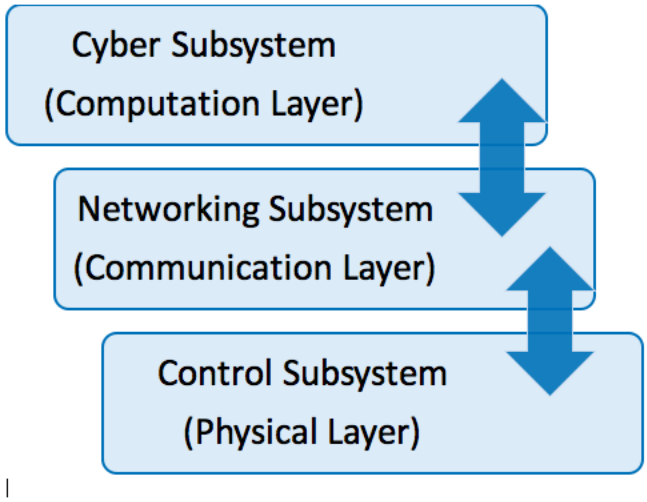

Figure 1.1: CPS's 3 tier architecture integrating Physical/Perception Layer, Communication Layer and Computation Layer

layer is IPv6 over Low-power Wireless Personal Area Network (6LoWPAN). A very special class of CPS is Supervisory Control And Data Acquisition (SCADA) systems, which are used in various industries. In SCADA, system administrators can control the remote sites via a centralized control system [8]. These systems are specifically useful for monitoring and controlling industrial networks such as telecommunications, water and waste control, energy, oil and gas refining, and transportation. The scale of SCADA systems can range from small like monitoring environmental conditions of a small office building, to incredibly complex, such as monitoring the activity of a nuclear power plant.

1.2 Security Issues in CPS

Cyber Physical Systems (CPS) are highly interconnected systems and are providing new functionalities to improve quality of life. It is vital to maintain security of CPS as one of the services they provide is monitoring important personnel and industrial systems. A few critical areas are personalized health care, smart grids, emergency response, surveillance control, traffic flow management, smart manufacturing, highways, and homeland security, energy supply, etc. Security compromise on these areas can lead to several problems, which may be quite simple as service disruptions and economic loss, to highly crucial as compromising natural ecosystems

and human lives [9]. The research on cyber-physical security strives at the intersection of physical security as well as the cyber-security of information, computation, and communication systems. An ongoing need is to address the security and privacy concerns at every level of CPS right from the early stage of design to the final stage of deployment. The physical subsystem of CPS consists of a large number of sensors connected as a Wireless Sensor Network (WSN), which collects data for a variety of CPS. These sensors or motes are of limited power, memory, and processing resources. The firmware of many of these devices are not well maintained, thus they are easily controlled by hackers remotely. The Sensor Layer or Perception Layer is mainly responsible for information collection, therefore, it is pretty obvious that security of IoT should start from here [10].

Massive attacks have been reported on Cyber Physical Systems in the past. A few common ones are listed in Appendix B.

1.3 Motivation

A detailed analysis of the CPS attack incidents leads to several important points:

1. It is a highly strenuous task to physically protect the CPS against attacks; because of its sheer size, numbers of nodes, and power limitations of the physical nodes.

2. Wireless communications channels are insecure.

3. A detailed logging system is required to record all device accesses and commands, especially the ones involving connections to or from remote sites, and must follow them to monitor the networks.

Even though there are many types of research done to protect a sensor network from the attacks, still it is challenging to identify a full-fledged solution as new attacks keep emerging and the provided solutions apply to specific attack types. Therefore, instead of identifying only one attack and providing a solution for that, we should be able to protect the network from several attack types, if possible.

There were two primary motivations for this research:

a. Since, there is no standard measure to assess the vulnerability or degree of security

of different SCADA or IoT systems, we wish to develop a security scheme, and
b. Our next motivation is to develop a systematic approach to enhance the security of the system by deploying the required countermeasures. One of the commonly used countermeasures is an Intrusion Detection Systems (IDS), the second line of defence, which may be deployed in the network based on its security needs. The role of the IDS is to observe the network traffic for the purpose of identifying any anomalies or unauthorized access to the network behaviour [11].

1.4 Research Questions

In this dissertation, we aim to answer the following two research questions:

1. How to define and quantify the vulnerability and security level of physical layer of a CPS?
2. How to build an Intrusion Detection System that can detect several known attacks? How can we extend this IDS to be able to detect new attacks in the same network?

1.5 Contributions

The contributions of this dissertation are summarized as follows:

1. Developed a novel 5-level security evaluation scheme that can be used to evaluate and assess the security needs of the current CPS.
2. Built a complete packet analysis model of RPL protocol. The model identified the features that can be used to distinguish the traffic patterns under different circumstances in cyber physical systems.
3. Created a new dataset of 300 different simulations of RPL network with four different attacks.
4. Proposed a novel predictive model for intrusion detection for four known attacks using machine learning analysis. An extension of the predictive model is added so that it can detect a new unseen attack or a combination of several attacks.
5. Enhanced the predictive model to an optimized model based on time and prediction accuracy using 2 different feature reduction methods.

1.6 Publications

Book Chapter

1. M. Sharma, H. Elmiligi, F. Gebali, "Network Security and Privacy Evaluation Scheme for Cyber Physical Systems (CPS)" in *"Security of Cyber-Physical System: Vulnerability and Impact"*, Springer (Accepted)

Journal

1. M. Sharma, H. Elmiligi, F. Gebali, "A Novel Intrusion Detection System for detecting RPL attacks in Cyber Physical Systems", IEEE Access(Final submission after minor edits)

Conference

1. M. Sharma, F. Gebali, and H. Elmiligi, "3-dimensional analysis of cyber-physical systems attacks," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), Noida, India, Galgotia University, Dec 2018, pp. 1–5. [12]

2. M. Sharma, F. Gebali, H. Elmiligi, and M. Rahman, "Network security evaluation scheme for WSN in cyber-physical systems," in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEM-CON), Vancouver, Canada, University of British Columbia, Nov 2018, pp. 1145–1151 [13]

3. M. Sharma, H. Elmiligi, F. Gebali, and A. Verma, "Simulating attacks for rpl and generating multi-class dataset for supervised machine learning," in 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct 2019, pp. 0020–0026. [14]

1.7 Dissertation Outline

The structure of this dissertation is as follows:
Chapter 1 gave the motivation for the research. Starting from chapter 2, we present an extensive literature review of the previous work in securing a network. First, a comparative analysis of the security schemes is presented and then an analysis of the proposed IDSs for RPL is done.

Chapter 3 provides an overview of the RPL protocol followed by the various attacks on the RPL protocol. The chapter also highlights a detailed description of machine learning concepts and processes including feature engineering, model building and model evaluation techniques and metrics.

Chapter 4 introduces the overall research approach. The first phase of the research is dedicated to the assessment of network security. In this chapter, first, the attack classification is explained, and following that, the novel Network Security Evaluation Scheme (NSES) is described.

Chapter 5 explain the next phase of the research i.e. enhancing the security using novel IDS framework. The details of the IDS framework, experiments and optimizing the performance of the framework are also introduced here. The first part of the chapter includes the details of the empirical side of the research and the second part explains the experiments.

Chapter 6 draws together a comparative analysis of the results using various machine learning algorithms. The model is also evaluated using n-fold cross-validation and new data prediction to make a decision for the prediction model.

Chapter 7 concludes the work done. It also enumerates avenues of future work for further development of the concept and its applications.

Chapter 2

Background and Literature Review

2.1 Security Assessment of CPS Systems

Cyber Physical Systems are very heterogeneous. These systems may suffer with the compromise of hardware components such as sensors, actuators, and embedded systems, or of software products like protocol, firmware, proprietary and commercial software for control and monitoring etc. Every component's interaction can be a loophole for a CPS attack. The need is to understand the security vulnerabilities, attack types, their difficulty measure and various industrial and academic protection mechanisms for these systems.

CPS Security designers often assumed that the attacker lacks knowledge about the internal structure of the target system. However, this assumption seems to be failing as many sophisticated malware's use buffer overflow, code injection, and rootkits, that frequently target CPS [15, 16]. A very popular example is the Stuxnet attack in which nuclear centrifuges in Iran were targeted. The attack stands out because of its complexity and overall impact [17, 18].

Traditionally, industrial control systems are designed around availability and safety. The original design of most CPS applications did not consider cyber-security issues as
(i) Specialized hardware was used in CPSs with proprietary code and only specialists knew the proper way to use them
(ii) Physical security methods were adequate as CPSs operated in closed environ-

ments without any connectivity with other domains.
(iii) As CPSs operated in closed environments, the need for creating secure and robust CPS protocols was not realized.
Due to these reasons, it was difficult to have common standard security protocols developed for any CPS. Although the commonly used security standards for IT systems were not fully applicable, these were the only ones used for all networks. Because of these reasons, several major attack scenarios occurred around the world that led to loss of resources as well as lives. A few of them are listed in the next section.

2.1.1 Well-Known CPS attacks

With the growing use of CPS and its widespread nature, several extensive CPSs attack cases can also be seen around the world. A few of them are listed below:

Stuxnet - An example of substantial damage to nuclear centrifuges in Iran. The cyber worm dubbed 'Stuxnet' and struck the Iranian nuclear facility at Natanz. Targeted at each of the three layers of a cyber-physical system, this worm infected over 200,000 computers and caused 1,000 machines to physically degrade. The effect spread through all across Iran and many other countries including India, Indonesia, China, Azerbaijan, South Korea, Malaysia, the United States, the United Kingdom, Australia, Finland and Germany.

Ukraine SCADA Attack - December 2015 the Ivano-Frankivsk region of Western Ukraine experienced a massive power outage. The attack affected nearly 230,000 people and was regarded as the first high severity cyber attack that caused power outage. [19].

Mirai Attack - not as popular as its counterparts, this brute-forced IoT device attack occurred in August 2016. This malware was created using ELF binaries that target SSH or Telnet network protocols [20]. It hijacked nearly half a million internet-connected devices and resulted in the inaccessibility of several high-profile websites such as GitHub, Twitter, Reddit, Netflix, AirBnB and many others.

Maroochy Water Service Attack - In March 2000, a SCADA system breach on Queensland's Sunshine Coast in Australia was discovered. The attack caused

800,000 litres of raw sewage to spill out into local parks, rivers and even the grounds of a Hyatt Regency hotel. The marine life died, the creek water turned black and the stench became unbearable for residents [21][1].

2.1.2 Security Countermeasures

There are a few security countermeasures that can be used to protect the systems against tampering and breaches. Security of the physical layer can be enhanced by using some forms of physical protection. A special lock and key arrangement or making the node package tamper-proof can physically protect the hardware from intentional tampering.

Another effective defence mechanism is using a firewall. A firewall acts as a gatekeeper over the communication traffic entering and exiting a network. [22]. This may be done using some physical (extra hardware) security management or using cryptographic keys to ensure that only the authorized nodes can join the network. It is claimed that firewalls are quite impossible for the wireless networks, yet, it is possible, either to selectively control and block radio communication or by using rule definition language [22, 23].

Access Control is the enforced restriction on the network to prevent unauthorized users to be able to access the network. It also imposes restrictions on access rights of authorized users, thereby forming and additional layer of protection. It utilises user authentication, a tool that helps identify and validate the identity of a particular user. Access control ensures that the new node has the correct identity, and helps prove that the new node truly is new and authenticated to be admitted into the sensor network. [24]. This is possible using key establishment, which is a part of access control, that will help the new nodes to establish shared keys with its neighbours to ensure secure communications with them.

Encryption ensures privacy in the network by maintaining confidentiality of the data travelling through the network. Data is encrypted before sending and is decrypted by the receiver to read it back. This may be achieved using key management. When a single key is used for both encryption and decryption, it is called symmetric

[1]More details of these attacks are also discussed in Appendix II.

key and is a preferred method in WSN. It consumes less battery power, memory and has minimum computation overhead. The other method is Asymmetric key cryptography, which uses two separate keys, one for encryption and another for decryption, and the two keys are interconnected with complex mathematical algorithms. This method, even though more reliable and safe, is rarely used in WSN as it has a huge overhead on power, computation and memory [25, 26].

Cryptography and key exchanges directly may not prevent an intrusion, but plays a big role in protecting the network by restricting the entry of an unauthorized user and also help in secure data transmission across different nodes and may protect data from tampering i.e. helps in maintaining privacy in the network.

The basic protocol can be improved(specifically in enhancing security) by defining handshakes or tracking control messages.

In spite of all efforts of prevention, intrusion still may occur. Detection of one or several compromised nodes is extremely critical and difficult. The need is to make the systems capable of detecting an intrusion as early as possible. Intrusion detection systems (IDS) and Intrusion Prevention Systems (IPS) are the popular solutions.

Based on the above discussions, there must be two stages of maintaining security:
1. Make the network secure by using preventive measures.
2. Add the further countermeasure to monitor the network to detect unwanted activity.

So the security maintenance of a network starts with the security assessment of the network that assigns a score for the required security needs. Based on the score, more countermeasures may be added. Not all networks need the same level of security and hence the measures deployed in every network may also differ. Decisions about security needs may be taken based on network design principles, traffic flows, and the types of users using the network. Knowing the security needs of the system helps to decide on the deployment of the systems security infrastructure.

The security infrastructure can be built using well-known security standards or by deploying several available counter-measures in an organized way. The industry,

as well as academia, have extensively worked on this issue. There are several security standards and security scheme available in the industry and proposed by the researchers to secure systems. In the following sections, security standards and the security schemes proposed by several researchers are discussed:

2.2 Security Standards

There are four main global security standards available in the industry.

2.2.1 Common Criteria (CC)

Common Criteria is an international standard for computer security certification [27, 28]. It is a framework that allows computer system users to specify their security needs and the technology vendors matches them to certified products [28, 29]. A few purposes for which CC certifications exists are to improve the availability of well certified security-enhanced IT products and profiles, maintain consistent security standards, avoid duplicate IT product evaluations and provide cost-effectiveness as well as efficiency.

The CC's security scheme has seven Evaluation Assurance Levels (EAL's). Considered for the security of applications in extremely high risk situations, these levels span from Level-1 to level-7 [28, 30]. Level-1 is basic and level 7 is the most advanced. The EAL indicate the levels of testing done on the product. It does not ensure that the product itself is more secure. Each EAL level introduces a set of security assurance components (SARs) that must be included in the evaluation such that the level requirements are met. For the organizations to achieve a particular EAL level, they have to meet very specific assurance requirements, which may lead from design documentation and analysis to various testing, or implementation of extra hardware/software. To gain a higher EAL, the organization may need to have more detailed documentation, analysis, and testing than the lower ones, which costs more money and time. The main benefit of this level number assignment is the indication of the testing level maintained by the organization. These security standards are applicable for the IT products or systems, and have been in effect since 1999. The EAL levels state the level of testing at the time of certifying. The Common Criteria evaluations are done solely on computer security systems and products. The EAL

level itself is only one indicator on the security of a product and does not measure the security of the system itself, and especially not of the WSN.

2.2.2 Federal Information Processing Standard (FIPS)

It is a standard published by U.S. government's National Institute of Standards and Technology (NIST), to enhancing computer security by approving cryptographic modules [31]. This standard is strictly enforced in Canada and includes both hardware and software components. The standard specifically applies to the areas related to the secure design and implementation of a cryptographic module like module specification, module ports and interfaces; their roles, services, and authentication etc. Cryptography has a major contribution in maintaining security, but it alone does not ensure or qualify a network to be fully secure.

A Cybersecurity Framework [32] is developed by NIST to help the ever expanding cyber security threat. This framework helps the organizations in understanding their cybersecurity risks. It provides a common language to every level of the organisation's user using the framework. It also offers customised measures for reduction of risks. The Framework design allows organizations to be able to respond and recover from the security incidents that they face or have faced previously. The solutions are derived after analyzing root causes of the problem and then taking preventive measures to improve them for the future. This framework is already used by 30% of the U.S. organizations, and is expected to reach 50% by 2020.

NIST also worked for Industrial Control Systems Cybersecurity. The Guide to Industrial Control Systems (ICS) Security, has a section for SCADA systems security recommendations as well. This guidance is given to modify common IT security controls to be used by ICS and SCADA systems to enhance their performance, reliability and safety requirements.

2.2.3 Industrial Automation and Control Systems Security (ISA99)

An Industrial Automation and Control Systems [33] Security standard was formed after Several cyber security experts came together to make a standard for the security of the hardware and software systems such as SCADA, networked electronic sensing, and monitoring and diagnostic systems. The main role of this standard is to pro-

vide control, safety, and manufacturing operations functionality to the development processes. The multi-standard IEC 62443 series are available to provide regulatory requirements for different types of systems. They establish standards, recommended practices, technical reports, and related information to define the procedures for implementing security at the manufacturing and control systems electronically. They also work to enhance security practices and also assess electronic security performances. Using over 150 standards, developed using the expertise from over 4,000 industry experts around the world, the organization provides the guidelines for adequate system design and implementation, and also for operation and maintenance. This is done to promote manufacturing units reliability, safety, and security.

Several IACS standards and technical reports are prepared by ISA to support different systems for risk assessment, security, product development, protection rating etc. Although, physical security is an integral component to maintain the integrity of any control system environment, it is not addressed in their documents anywhere.

2.2.4 International Electro-technical Commission (IEC)

International Electro-technical Commission (IEC) [34] is an organization involved in the preparation and publication of International Standards for all electrical, electronic and related technologies. They define the safe failure fraction (SFF) and the Safety Integrity Level (SIL) that are quite useful in defining the degree of safety for making the related system fail-safe. IEC 61508 is known to be the basic safety standard applicable to all kinds of industries. The process industry follows ISA 84 / IEC 61511. Car manufacturers use IEC 61508. It works for risk reduction by calculating safety integrity levels known as SIL. The required SIL is based on a hazard and risk analysis, combined with risk acceptance criteria. IEC works actively to provides a platform to companies, industries and governments for meeting, discussing and developing the International Standards they require.

Table 2.1 provides the summary of various security evaluation standards.

Table 2.1: Security Evaluation Standards

Evaluation Standard	**Description**	**Application Domain**	**Explanation**
EAL	7 level security scheme defined by Common Criteria	IT product and systems [27]	Need to have detailed documentation, analysis, and testing
FIPS	Published by U.S. government computer security to approve cryptographic modules	Design and implement cryptographic modules [31]	4 security levels for applications using cryptographic modules. Also states recommendations for SCADA and ICS systems.
ISA	A vendor-neutral global standards and certifications in the field of automation	To promote plant and operational reliability, safety, and security [33]	A standard for the automation of manufacturing, transportation, utilities, defence and other building automation.
IEC	A global solution for defining the safety of the process	Automotive, Process industries, Machinery and Nuclear plants [34]	Performs hazard and risk analysis, combined with risk acceptance criteria.

2.3 Security Evaluation Schemes

Cardenas et al. [35] proposed a security scheme for SCADA (Supervisory Control and Data Acquisition Systems), which are essentially the the old form of CPS systems. They proposed different countermeasures for different attacks by categorizing the attacks in three categories; (1) physical attacks from outsiders, (2) key compromise attacks and (3) insider attacks from somebody controlling a legitimate node. The threats were ranked to calculate the score of the difficulty of accomplishing the attack. For their security scheme, they have considered extra hardware installation, physical access security, and required technical skills to enforce attacks. They did discuss various issues related to SCADA systems, but failed to provide any security scheme or levels.

A state-based semi-Markov chain model framework [36] is used for modelling the security of CPS for the cyber attacks that can lead to physical damages. It is based on traditional Byzantine model, where the attacker and system behaviour over time are studied. A few quantitative security analysis are presented using several metrics like mean time to security failures, steady state security, and steady state physical availability failures. This model does not consider deployment of any countermeasures in the network.

A game-theoretical approach [37] for cyber-physical security for wide area monitoring, protection and control applications (WAMPAC) is proposed, only timing-based attacks, integrity attacks and replay attacks are considered in this work. The security is dealt as three components: Wide-Area Monitoring (WAM), Wide-Area Control (WAC) and Wide-Area Protection (WAP). The model works on various cyber attack scenarios based on the attack model, and the information sets available to both attacker and the defender.

Sensor data security estimator (SDSE) [38], a new comprehensive security estimation module defined for WSNs. Based on cryptographic algorithm, key management scheme and intrusion detection system this module calculates the security level of the network. It is deployed on the base station. The main goal of this work is to calculate the security level (SL) of sensor data based on the three countermeasures and provide that to the WSN users.

Wu et al. [39] proposed the calculation of a comprehensive value Q to define the security level of the network. Common criteria EALs are used to calculate this value. A higher value of Q ensures that the network is secured. They used CCs CAP (Composed Assurance Package), which is a method to evaluate the composed information security where two or more IT products are used. Since CC is a well established standard, it makes this scheme more trustworthy. But absence of discussion about countermeasures, makes it less applicable.

Han et al. [40] proposed a Three-Dimensional Model for software security evaluation which provides a systematic way to analyze software security in three dimensions i.e. technology, management and engineering. In technological dimension, CCs 7 se-

curity levels based on Evaluation Assurance Levels (EALs) are considered. For the management dimension, the management of software infrastructures, development documents and risks are considered and the engineering dimension is mainly focused on 5 stages of software development life-cycle.

Mike, Emmanouili and Vassilis [41] proposed a special security framework specifically for CPS that covers both cyber as well as physical aspects [41]. With the help of Russian-Ukraine dispute for the price of natural gas case study, this framework has combined the vector attacks and the synchronization issues.

Amer [42] in his article proposed a 3-D scheme to classify hardware attacks based on three criteria i.e. Accessibility (A), Resources/money (R), and Time/effort/experience (T) is proposed, that could be represented in 3D space.

Table 2.2 provides the summary of various security evaluation schemes.

Table 2.2: Security Evaluation Schemes

Evaluation Scheme	**Description**	**Application Domain**	**Explanation**
SMC based model	SMC chain based model to describe attacker and system behaviour over time	For any CPS [36]	Considers only cyber attacks leading to physical damages
Security scheme for SCADA	Taxonomy made up security properties of WSN, threat model, and security design space to protect SCADA systems	Any CPS [35]	Provides a view of the security of WSN based on threat ranking done by calculating the difficulty level of an attack
WAMPAC security	A game-theoretic framework to model cyber-physical security using attacker/defender model	For WAMPAC [37]	Framework looks the attacker strategies based on the defender actions, defender progressively updates strategy
Sensor Data Security Estimator (SDSE)	Estimates the sensor data security level based on security metrics by analyzing both attack prevention and detection mechanisms	For any WSN [38]	Security evaluation module is deployed at the base station monitoring the entire network and compares sent message with returned message
CAP based security scheme	Adopts the Geometric mean method, then determines the security value of the network	For any network [39]	Deals with the analysis and testing of the vulnerabilities of the network
3-D model for security evaluation	A systematic way to analyze software security in 3-D i.e. technology, management and engineering	Applies to Software security only [40]	Security evidence, collected from three points of view, are evaluated under a rule to calculate the value.
Security Framework for CPS	Combines cyber & physical aspects as threat model, then protect it using common security policies	CPS at both Cyber and Physical Layer [41]	Identifies the features need to be protected, then apply the common security policies

2.4 Enhancing the security of CPS

Security may be enhanced by knowing the different attack types and then working on building counter measures to protect the network. Over the last few years, researchers have explicitly studied the numerous security issues associated with these low power devices, namely low power and lossy networks(LLNs) [43]. Many taxonomies of attacks are available in the literature and several IDSs have been proposed to secure the networks. Intrusion detection system (IDS) including many others has been a very common solution for securing the network besides others. Role of an IDS is to observes the network traffic, analyzes it and then identify the possible anomalies in the network behaviour. RPL is a very common protocol used for LLN. Also, a plethora of solutions have been proposed for attack detection and protection on RPL [44]. Several countermeasures and IDS has been proposed for RPL too, that may be for some specific attacks. To build these solutions, machine learning and many other techniques may be used. In the next section, we review the attack detection and protection methods proposed for RPL.

2.4.1 Protection Schemes for RPL

VeRA [45] is a version number and rank authentication security scheme, that is based on one-way hash chains, and is used to secure the IPv6 routing protocol (RPL) . The security scheme mainly deals with an internal attacker impersonating a DODAG root and then intentionally increasing the Version Number. It also looks for an internal attacker that can illegitimately decrease the rank value to introduce rank attack in RPL network.

TRAIL [46] an extension of VeRa identifies any topology attack in RPL. Trust Anchor Interconnection Loop works by validating the upward paths to the root with the help of round-trip messages. In TRAIL, each node can conclude its rank integrity using the recursive algorithm to intact the upward path.

Dodge-Jam [47] is another lightweight anti-jamming technique suitable for LLN environments. It is proposed to address the problem of jamming attacks with small overhead. The proposed solution has three components i.e. ACK channel hopping, multi-ACK channel hopping and multi-block data shift. The main rule considered is

that to address any fake ACK attack, the sender channel would be changed with the recipient channel of data packets by ACK channel hopping.

A single checkpoint-based countermeasure, SCAD [48] is proposed as a monitor-based approach to mitigate the forwarding misbehavior in WSN. In this case, each node monitors the forwarding behaviors of the preferred parent node. This is followed by observing the packet loss rate and then comparing the observation result with the collected packet loss rate. This helps in detecting the forwarding misbehavior of the preferred parent node.

A dynamic threshold mechanism is proposed to mitigate the destination advertisement object (DAO) inconsistency attack in RPL-based LLNs. In DAO inconsistency attack, a malicious node drops the received data packet intentionally [49]. Then it replies the forwarding error packet so that the parent node will discard the valid downward routes in the routing table.

SecTrust-RPL [50] is a method of securing protocol. This is mainly making the RPL protocol against rank and sybil attacks. It uses a trust-based mechanism to detect and isolate attacks and optimizes the network performance at the same time.

SPLIT [51] is also working to increase the security, and availability in data communication process of RPL. SPLIT manages a lightweight remote attestation technique by piggybacking it on RPL's control messages. Therefore, it is able to achieve more usage. Due to this reason, it offers low energy consumption and enjoys scalability.

David et al. [52] worked on securing RPL Routing Protocol from blackhole Attacks Using a Trust-based Mechanism. The protocol is scalable as it is computationally inexpensive and does not impose extra overhead on network traffic.

Glissa et al. [53] also secured RPL using threshold with hash chain authentication against rank and sinkhole attacks. It uses cryptography with hash chain, and hence it is computationally expensive. They have used the concept of rank threshold along with hash chain authentication technique and have dealt with the internal attacks like sinkhole, black hole, selective forwarding attacks etc. Simulation results show that SRPL is robust and resistant to this kind of attacks based on malicious manipulation

of RPL metrics.

Table 2.3: Summary of methods proposed to secure RPL

Name	Attack detected	Description
VeRA	version number and rank authentication	One-way hash chains are used to secure the IPv6 routing protocol
Dodge-Jam	Jamming attacks	Lightweight anti-jamming technique for jamming attack
SCAD	Forwarding misbehavior	Observes the packet loss by comparing with the parent node behavior
TRAIL	Topology attacks	performs DAO inconsistency check
SecTrust-RPL	rank and Sybil attacks	
SPLIT	Ensures software integrity of network nodes	makes data communication process available
TrustbasedRPL	Blackhole attack	Uses trust based mechanism to secure RPL
Hash chain based authentication	Rank and Sinkhole attacks	Uses cryptography with hash chain, so computationally expensive

2.4.2 IDS for RPL

SVELTE [54] is an IPv6 based IDS that detects spoofed or altered information, sinkhole and selective forwarding attack. This IDS identifies all malicious nodes that lead to sinkhole and/or selective forwarding attacks in the network. It is a combination of anomaly-based and specification-based IDS methods, and offers both very little overhead and a high success rate of detection. This IDS has three components: a node based module, a border router based module and a firewall that protects the 6LoWPAN network against global attackers. The IDS module is placed in the centralized BR and all network nodes send the data to BR as shown in Figure 2.1.

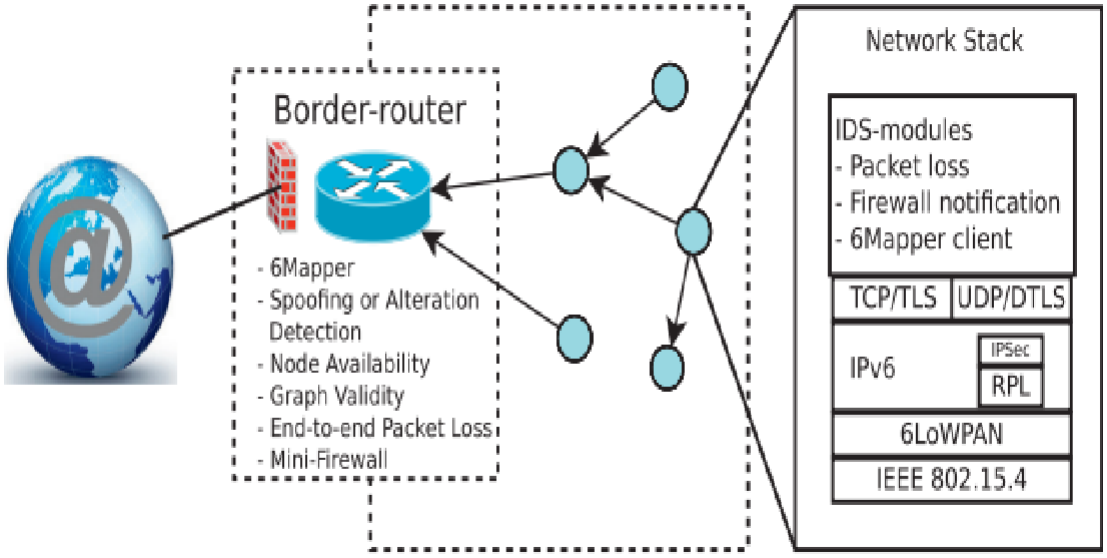

Figure 2.1: SVELTE Framework [54]

An extension to the SVELTE intrusion detection system is proposed where ETX metric are used [55]. The new intrusion detection is based on geographical hints, and is applicable in two situations: first, when both rank-based and ETX-based solutions are not working and second, when a large number of attacks go unnoticed.

CHA-IDS [56] is another IDS developed using machine learning based on the analysis of a compression header. This IDS solution is developed for RPL using Cooja and has used many machine learning algorithms for its implementation and testing. Figure 2.2 illustrates the four layered framework of the IDS. Layer 1 captures compression header data using Cooja traffic analyzer and called as Sensor Agent. This data is analyzed in layer 2 named Aggregator Agent (AGA), where features are extracted. The class labeling is done at layer 3, Analyzer Agent (ANA) layer. The Actuator Agent (ACA) at the layer 4 alerts the users about the malicious activities in the network.

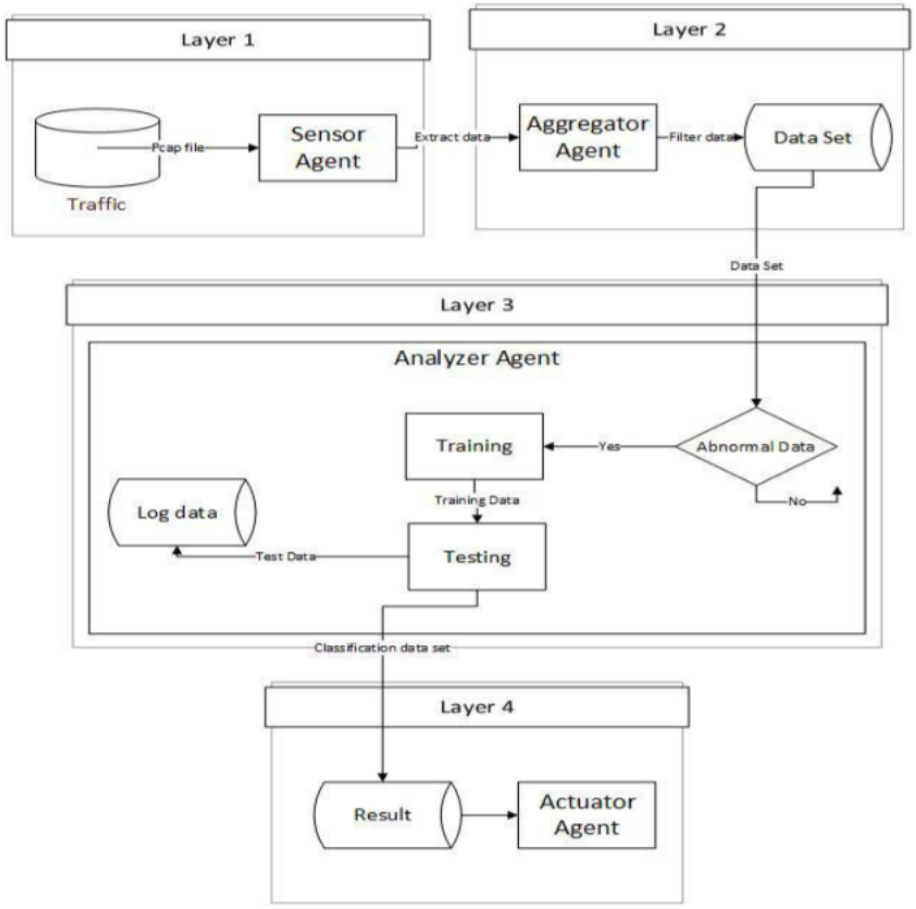

Figure 2.2: CHA-IDS Framework [56]

The main drawback of this IDS is that it can only detect three attacks and there is no scope of any anomaly detection otherwise.

Pongle's IDS [57] is designed to detect wormhole attacks on RPL. It is a hybrid architecture where the main IDS is located at BR, and lightweight modules are located at the nodes. Detection of an attack takes place at the root node. The root node maintains record of all node locations and their transmission ranges. Each node periodically sends information about their neighbors and Received Signal Strength Indicator (RSSI) to the root node. The root node holds both new as well as old information so that it can compare and then detect an attack. The framework is shown in figure 2.3.

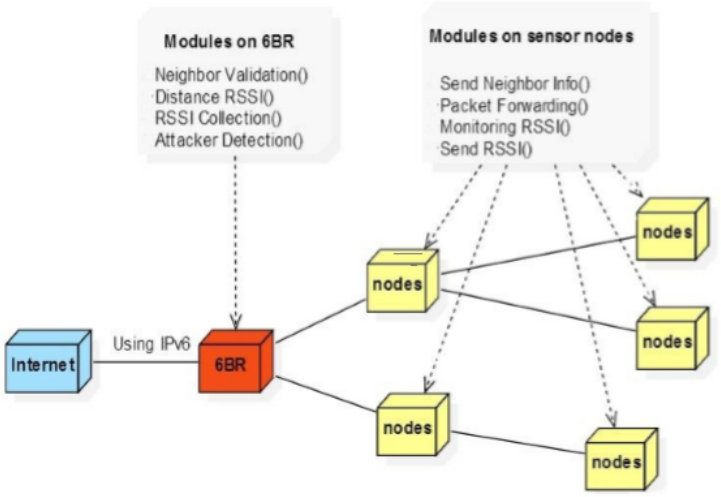

Figure 2.3: Pongle IDS Framework [57]

Farzaneh et al [58] proposed and built an anomaly based lightweight intrusion detection system that was tested on Cooja. This is capable of detecting neighbor and DIS attacks. The IDS placement is fully distributed, and hence, each node in the network collects information and performs the intrusion detection as shown in Figure 2.4.

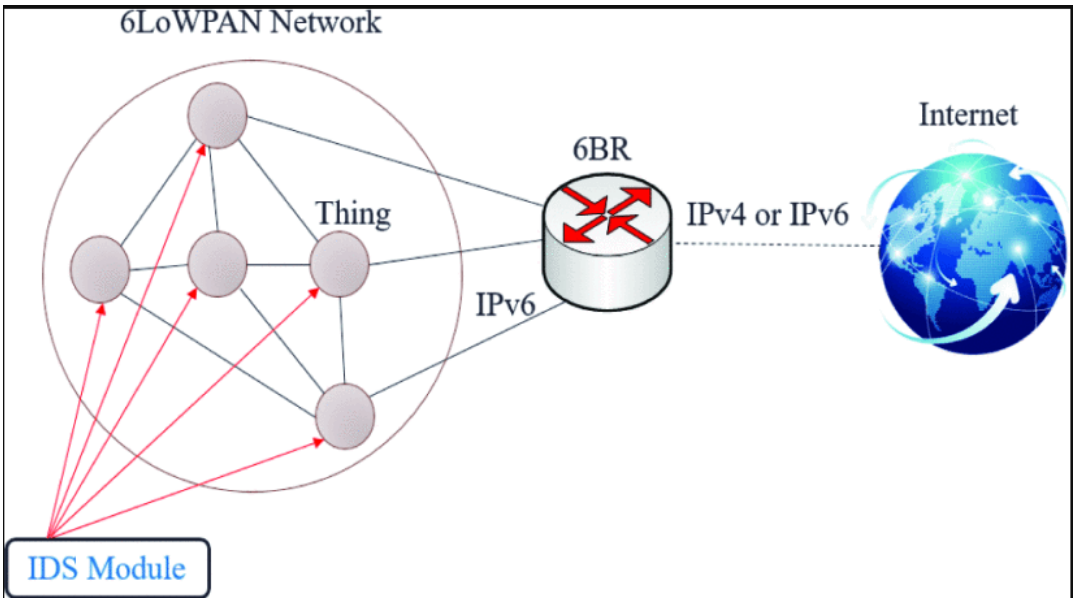

Figure 2.4: Anomaly based IDS Framework [58]

This IDS detects the intrusion based on threshold values on RPL protocol. The results are also analyzed on Cooja and show elevated True Positive Rate (TPR) up to 100% in some cases. It also claims a small False Positive Rate (FPR), is fully effective in attack detection and can be applied to large-scale networks.

INTI [59] is another RPL based IDS that establishes dynamic clustering in order to support data transmission in IoT. By observing the behavior of router nodes in the forwarding task, suspicious nodes are detected by reputation and trust mechanisms. This proposed tool detects sinkhole attacks by testing and analyzing the network traffic on Cooja.

InDRes [60] is proposed as an enhancement of INTI, which is also a mathematical-based, anomaly-detection, hybrid intrusion detection system. It works by dividing the network into separate clusters, and each cluster has a leader node. The leader node collects rank from all its nodes, which is used later to detect and isolate the attacker. As soon as an attacker is detected, the cluster leader notifies the root node, and then DODAG is reconstructed after excluding the attacker node. The System Architecture of InDReS is depicted in Figure 2.5.

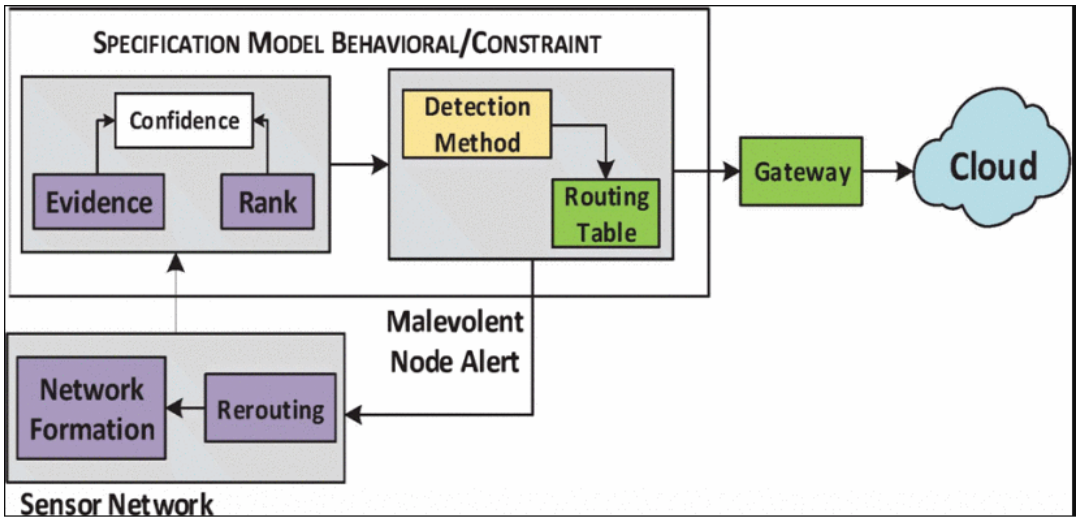

Figure 2.5: InDres IDS Framework [60]

Mayzaud et al. [61] proposed an IDS for version number attacks. A specification-based, signature-detection, hybrid-placement IDS, mainly detects and mitigates ver-

sion number attacks by deploying several monitoring nodes as shown in Figure 2.6. The experimental results of this IDS shows very good detection rates and to minimize the false positives, nodes need to be monitored.

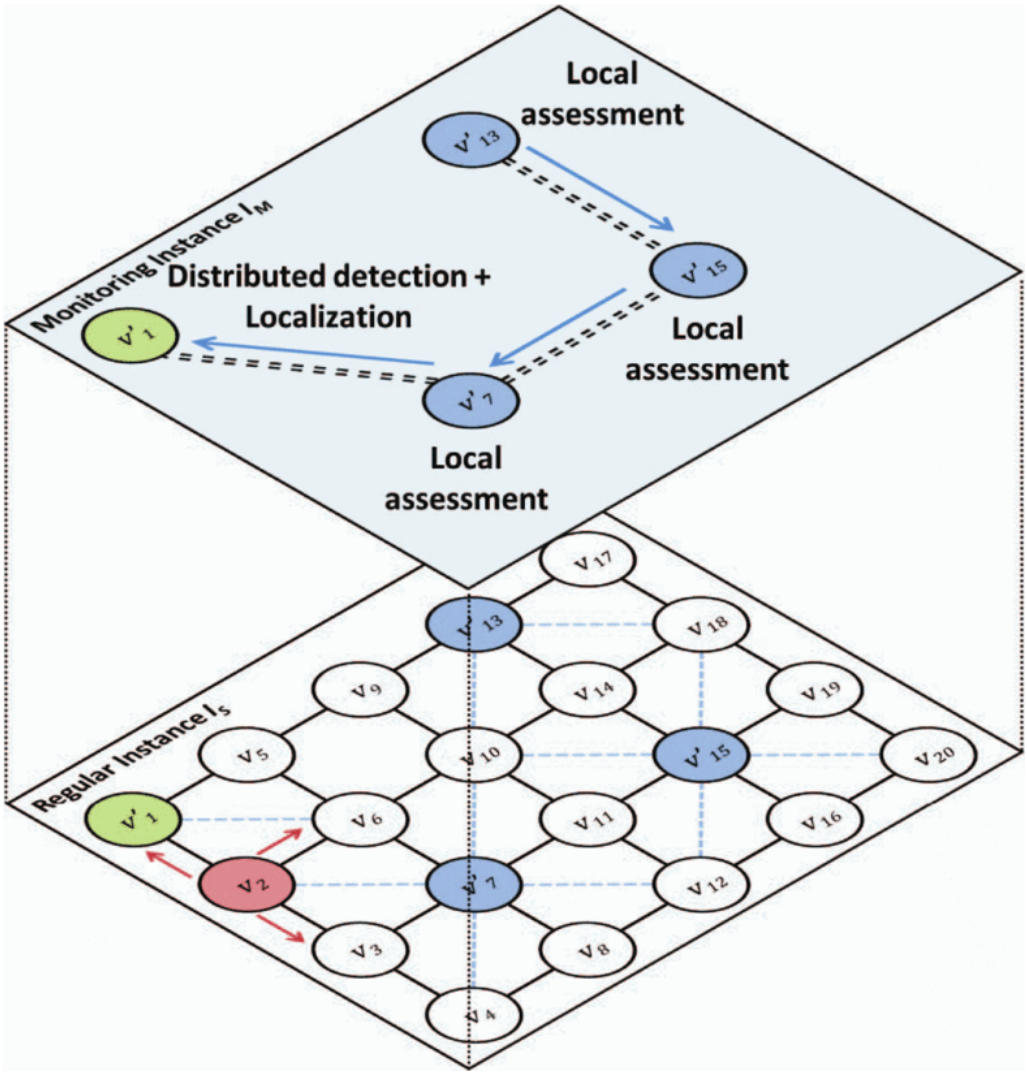

Figure 2.6: Version number detection strategy [61]

A Real-Time Intrusion Detection System is proposed [62] to detect wormhole attack in Cooja, using signal strength indicator (RSSI) to identify the attack and attacker node. In wormhole attack, a pair of attacker nodes form a tunnel and misguide other traffic. The proposed IDS uses a hybrid approach, where the IDS's distributed module are placed at sensor nodes and the centralized module is placed at Border

Router. The proposed system is shown in Figure 2.7.

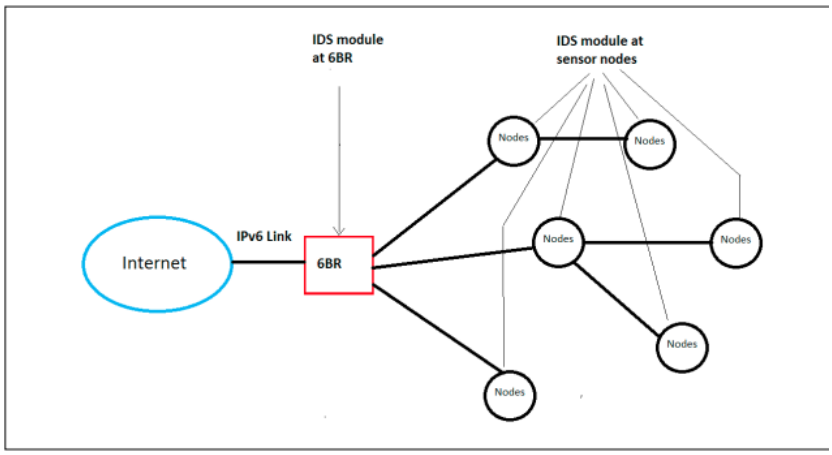

Figure 2.7: Wormhole detection strategy [62]

A specification based IDS is defined to detect topology attacks such as Sinkhole, Rank, Local Repair, Neighbor, and DIS attacks [63]. They use the network traces to extract the states, transitions, and their statistics to identify specifications. The IDS has a high accuracy rate in detecting topology attacks but has a significant overhead; because of which scalability is not achievable.

A K-nearest neighbor based technique is used for detecting the rank attack in RPL protocol [64]. Detection is done based on the distance calculations between the nodes with respect to the sink node or border router.

Another IDS is based on self-organizing map (SOM) neural network to cluster the WSN routing attacks using unsupervised learning [65]. SOM is very effective in converting high dimensional spaces to low dimensional spaces and use it for visualizations. The High-level System Architecture of this IDS is shown in Figure 2.8. This system is capable of detecting multiple types of RPL attacks scenarios i.e. HELLO Flood Attack, Sinkhole Attack, Version Attack and the network with no attack.

ELNIDS [66] is the latest IDS proposed for RPL that uses an ensemble-based machine learning model for creating a network intrusion detection system. This IDS is capable of detecting Sink Hole, Black Hole, Sybil, Clone ID, Selective Forwarding,

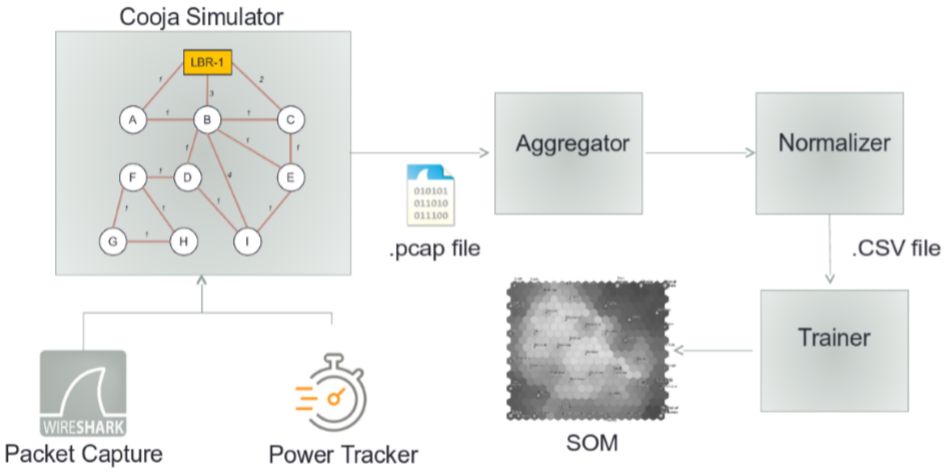

Figure 2.8: SOM Framework [65]

Hello Flooding and Local Repair attacks. They have used ensemble-based machine learning models to build this IDS as shown in Figure 2.9.

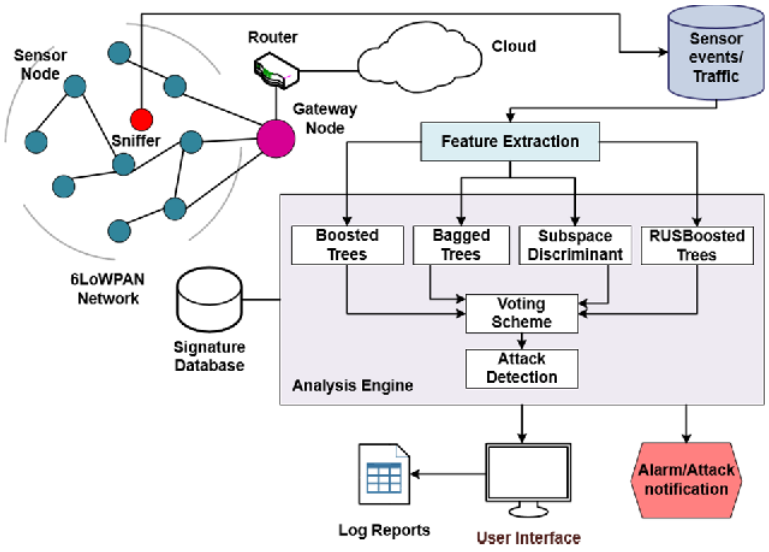

Figure 2.9: ELNIDS Framework [66]

Table 2.4: Summary of IDS for RPL

Name	IDS type	Attack detected	Description
SVELTE	Hybrid	Sinkhole and selective forwarding	Made up of three components, has high success rate
SVELTE-e	Distributed	Rank based attacks	ETX matrix are used to analyze the data
INIT	Distributed	Sinkhole	It follows a specification rule based approach to detect the attack
InDRes	Hybrid	Sinkhole, Rank, Version number	Cluster head compares the measures and inform root node
Pongle's IDS	Hybrid	Wormhole	The main IDS is at BR and lightweight modules at nodes
Mayzaud	Centralized	Version Number	A hybrid placement IDS that detects version number attacks and needs node monitoring
CHA-IDS	Hybrid	Sinkhole and selective forwarding	It uses the analysis of compression header to detect three attacks
Anomaly-based IDS	Distributed	Neighbor and DIS attack	Model is adaptable and is applicable to large scale networks
Real-time IDS	Hybrid	Wormhole attack	Uses signal strength indicator (RSSI) to identify the attack and attacker node
Real-Time IDS	Specification based	Topology attacks	Uses the states, transitions, and their statistics for detection
Rank attack IDS	Centralized	Rank attack	Uses a K-nearest neighbor based technique to calculate distance between the nodes
ELNIDS	Network	Several attacks	An ensemble-based ML model is used for an IDS

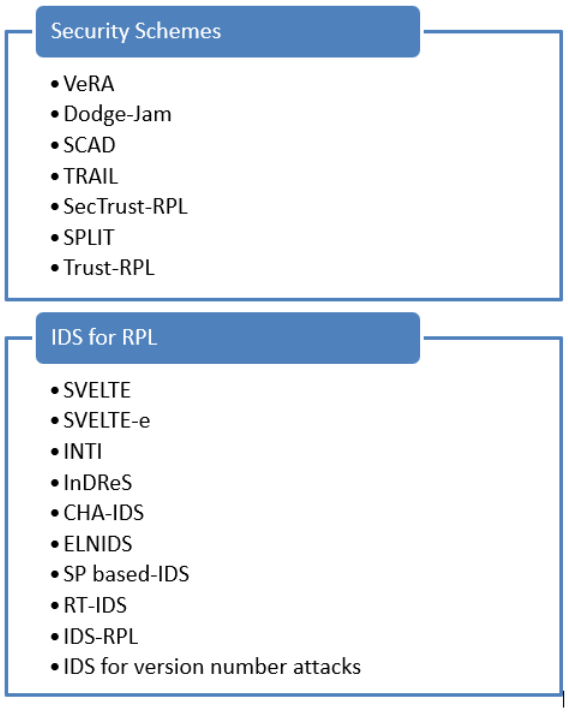

Figure 2.10: RPL protection mechanisms

2.5 Review of Available Datasets

Extensive work has been accomplished on building Intrusion Detection Systems using Machine learning techniques. As known, machine learning algorithms use data for learning and prediction. There are a couple of datasets publicly available for research. Even though an extensive list of public datasets is given in [67], a few commonly used for IDS development are discussed below:

DARPA is the first and most popular data sets used for intrusion detection. This dataset was created in an emulated network environmentat the MIT Lincoln Lab. The two versions of dataset are DARPA 1998 and DARPA 1999 contain seven and five weeks of network traffic in packet-based format. These dataset contains the data about normal traffic as well as of attacks like DoS, buffer overflow, port scans, or rootkits.

KDD dataset - The KDD data set was used at The Third International Knowledge Discovery and Data Mining Tools Competition, with the purpose of creating an

intrusion detector, a predictive model that can analyze the traffic as bad or good. As summarized in the latest paper [68], most of the IDS's are build and tested using KDD and NSL-KDD datasets [69–72]. This dataset is a standard set of data including a wide variety of intrusions simulated in a military network environment.

NSL-KDD [73] is an improved version of the same dataset (KDD) and is also quite popular in latest researches. The new dataset has more refined subsets as the creators have removed duplicates from the KDD CUP 99 data set.

RPL-NIDDS17 is a synthetic dataset used in the literature for studies. It is created using NetSim tool, that simulates various networking environments i.e. IoT, MANET, FANET, VANET etc. This dataset for IoT network scenario compromises of 20 features and 2 additional labelling attributes. This dataset contains traces of attacks including Sinkhole, Black hole, Sybil, Clone ID, Selective Forwarding, Hello Flooding and Local Repair attacks. Features of the dataset have been classified into three categories namely flow, basic and time [74].

AWID is another publicly available data set focused on 802.11 networks [67]. Built for a small network of 11 clients, the dataset is labelled and split into training and testing subsets. The WLAN traffic was captured in packet-based format for an hour. Total 37 million packets were captured, from which 156 possible distinct features are extracted from each packet for 17 classes on a 802.11 network. The 17 class represents 16 attack scenarios and one normal network scenario without any attack.

CICIDS2017 dataset contains pcaps of the benign and most up-to-date common attacks [75]. The network traffic has been analyzed using CICFlowMeter and the data is labeled having features as time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). The dataset is also supported with definitions of the features used in data extraction.

CIC DoS Canadian Institute for Cyber-security dataset was defined to create an intrusion detection dataset with application layer DoS attacks [67]. It has data about eight different DoS attacks on the application layer along with normal user behavior. This data set is available in packet-based format and contains 24 hours of network traffic.

LBNL [75] another common dataset used for intrusion detection. This dataset is developed by analyzing characteristics of network traffic within enterprise networks.

NGIDS-DS [75] data set contains network traffic two formats i.e. packet-based format and host-based log files. Generated in an emulated environment, this dataset has the data about normal user behavior and of seven different attacks like DoS or worm. This dataset is generated using IXIA Perfect Storm tool.

Table 2.5: A few commonly used Datasets for Intrusion Detection

Dataset	Description /Attacks
DARPA	DoS, privilege escalation (remote-to-local and user-to-root), probing
KDD	DoS, privilege escalation (remote-to-local and user-to-root), probing
NSL-KDD	DoS, privilege escalation (remote-to-local and user-to-root), probing
RPL-NIDDS17	Normal traffic and seven other routing attacks
AWID	Popular attacks on 802.11 like authentication request, ARP flooding, injection, probe request etc.
CICIDS2017	Botnet, cross-site-scripting, DoS, DDoS, heartbleed, infiltration, SSH brute force, SQL injection etc.
CIC DoS	Application layer DoS attacks
LBNL	Port scams
NGIDS-DS	backdoors, DoS, exploits, generic, reconnaissance, shellcode, worms

Most of the above dataset have been built in regular wireless networks. A dataset containing data packets from the wireless sensor network data packets seems to be missing. This was the motivation for us to build our own data set for the novel IDS we proposed in this thesis.

2.6 Research findings about security enhancement of CPS

As seen above, intrusion detection system is always seen as a key aspect of the security management tool. For the development of an IDS and its research, several machine learning techniques have been used, however, there are still shortcomings and further research is needed. After reviewing the aforementioned papers and researches in the literature, a few findings are:

1. Almost all of these IDS are only effective for specific attack types and cannot detect multiple or combination of attacks. They are also unable to detect a brand new attack.

2. Almost in all the works, IDS only undergo n-fold cross validations testing on the data set. None of the model are tested using new data in the training phase.

3. The most commonly used dataset for IDS research is KDD. A few other openly available datasets are also used in similar research. Since these datasets are not of the wireless sensor networks, their applicability is quiet questionable.

That leads to the two main challenges, which are:

a. The security standards discussed show that extensive research has been done for defining security evaluations, but most of the schemes are for IT networks and systems. Since the vulnerabilities, attacks, and security mechanisms of CPS are much different from those of traditional networks, a standard scheme that can certify the security level of physical layer of CPS seems to be missing.

b. Most of the IDS's listed in Table 2.4 are only effective for a specific attack and cannot detect multiple or combination of attacks. These are also unable to detect a brand new attack. Another drawback is the testing, which is only done on the dataset used for building the model and not using new data. The testing should be done with the new data from the similar network, but is not seen by the model in the past.

These outcomes provide us with sufficient motivation to do our research and build a security scheme to evaluate the security of the network and develop and test the Intrusion Detection System to enhance it further.

Chapter 3

Review of Concepts

Routing Protocol for Low-Power and Lossy Networks (RPL) is an IPv6 based protocol by IETF ROLL working group. It is commonly used for low power and lossy networks. RPL is a promising, proactive, lightweight, Distance Vector protocol with several advantages for tiny resource constraint devices used at the physical layer of CPS [64, 76–79].

3.1 RPL - Protocol

RPL is distance-vector and a source routing protocol. The setup of the multi-hop pathway is done using a Directed Acyclic Graph. The user initially sets the border route or UDP-server as the root node. Several UDP-clients are established to generate and route data to the UDP-Server from where it goes to the cyber subsystem. A Destination-Oriented Directed Acyclic Graph (DODAG) is built, which contains the paths from the leaves to the root i.e. UDP-Server. There are 4 types of control messages that are working in RPL [80]: 1) DODAG Information Solicitation (DIS) – It is used to look for a DIO from the RPL node. 2) DODAG Information Object (DIO) – The carrier of information regarding the RPL instance and its configurations. 3) Destination Advertisement Object (DAO) – One that propagates the information regarding destination to the upward nodes. 4) Destination Advertisement Object Acknowledgement (DAO-ACK) – The unicast communication by the receiver in response to a unicast message by the sender.

Table 3.1 lists the control messages in RPL

Table 3.1: DODAG Control Messages

Control Message	Description/Purpose
DODAG Information Object (DIO) message	Initiated by root node, this message is broadcasted to all nodes within reach of root. This message is adopted by node to join DODAG as it carries the configuration information
DODAG Information Solicitation (DIS)	Unicasted towards the neighboring nodes, this is critical for a node to join DODAG
Destination Advertisement Object (DAO)	A multicast message sent from one point to multipoint, so that the nodes may transfer information in upward direction towards root
Destination Advertisement Object Acknowledgement (DAO-ACK)	This is a unicast message transmitted by a node which receives DAO message

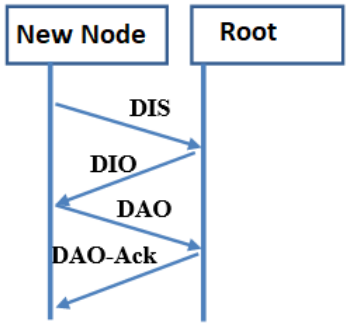

Figure 3.1: Control Messages Flow

To setup the DODAG for the packet transmission, RPL sets up the route information. To do so, root starting with an ID number of 1 starts sending out the (DIO) message that contains parameters to the neighbours. After receiving the message, neighbours calculate their rank and forward the messages to the node with the lower rank, as that is the preferred parent. The process is finished when all the available nodes are joined into a DAG.

By default, RPL has an inbuilt security mechanism that can mitigate the external attacks, but mitigating the internal attacks is still an issue [64]. Several attacks such as rank attack, version attacks and many more are possible in RPL network due

to the control frames being unauthenticated or un-encrypted. Devices may also be compromised or unauthenticated, hence external security measures are required [79].

3.1.1 Attacks on RPL

Two main problems on the security of the physical layer can be: failed sensing and disrupted/failed Communication. The failed sensing occurs because of physical removal of the nodes or a hardware attack that makes hardware non-functional [81]. The disrupted communication can happen because of several reasons like Spoofing/Altering/Replay Routing attack, Denial of Service (DoS) attack, Sybil attack, and node capture attack etc. [82,83]. Pavan and Chavan [84] also presented a survey of the RPL attacks. Some of the attacks studied by them are selective forwarding attack, sinkhole attack, sybil attack, hello flood attack, wormhole attack, black hole attack, DoS attack, clone ID etc. They discussed Some of the spoofing attacks such rank attack, version attack, local repair attack, neighbor attacks and DIS attack. The attacks can be categorized broadly in three main categories, attacks on the resources,attacks on network topology and attacks on traffic. Figure 3.2 shows the classification with attack types [85].

Common RPL attacks are explained below [86]:

1. Flooding attack - It generates large amount of traffic in the network making nodes exhaust faster. Then it make both the nodes and links unavailable.

2. Routing attacks - The routing information is forged or modified to advertise invalid routes to other nodes

3. Increased Rank Attacks - this attack occurs due to increase in the rank value of a RPL node. This leads to the generation of loops in the network [87].

4. Version Number Attack - An important field of each DIO message is version number of a DODAG. This can only be incremented by root. A change in version number indicates a new DODAG and leads to confusion and possibility of loops in the route [87,88].

5. Sinkhole Attack - This attack occurs in two steps. First, the malicious node manages to attract a lot of traffic in any case and then, it modifies the data or drops it [89].

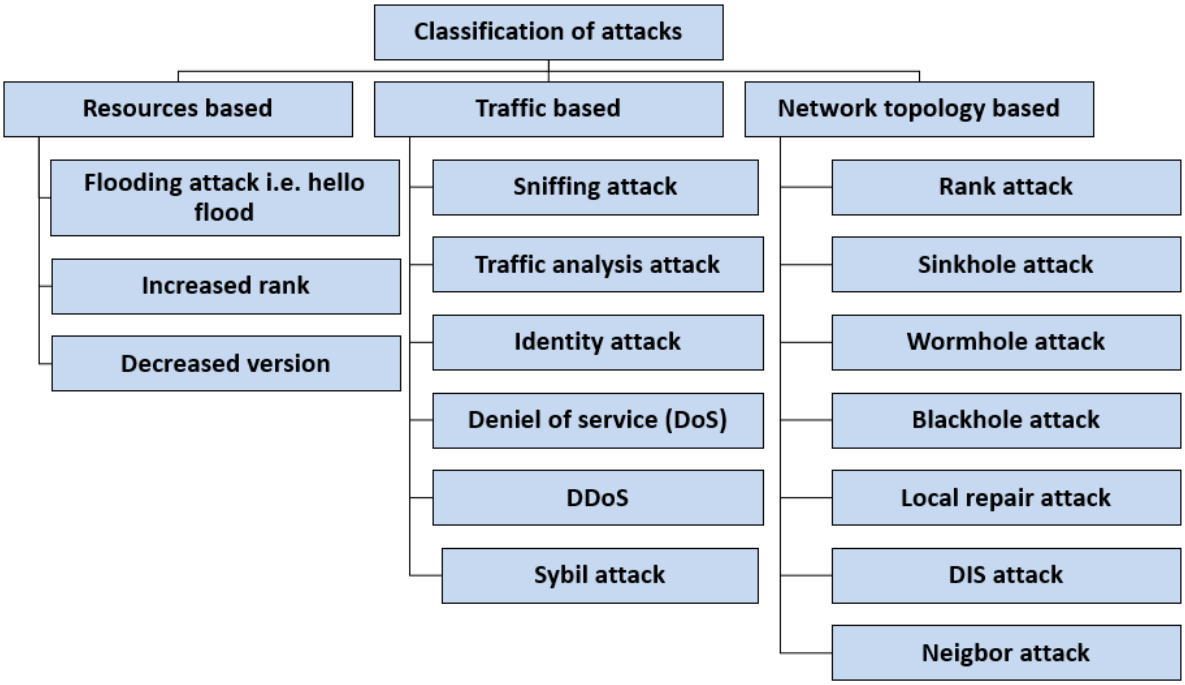

Figure 3.2: Attack classification

6. Wormhole Attack - This attack uses of a pair of RPL attacker nodes, A and B, which are linked via a private network connection. The packets received by A are forwarded to B and replayed later, distorting the routing path [90].

7. Black hole Attack - The malicious node drops all packets that are supposed to be forwarded [90].

8. Sniffing Attack - A passive attack where intruder is only listening the packets transmitted over the network [87].

A new type of Denial-of-Service attack is also identified and investigated, called "hatchetman" attack in LLNs. A malicious node manipulates the source route header of the received packet, and then generates and sends the invalid packets with error route to legitimate nodes [91].

Another new and severe DoS attack, called spam DIS attack, against RPL routing protocol in LLNs is investigated [92]. In this attack, malicious node starts sending a large number of DIS messages each with different fictitious identity. This causes the legitimate nodes to restart the routing by sending excessive number of DIO messages. This leads to denial of service of the malicious node.

3.2 Intrusion Detection System

An intrusion is an unwanted activity performed by an intruder or an unauthorized user to disrupt the normal functioning of a network. Intrusion detection system detects any malicious activity in the network. An IDS works using a monitoring component that can analyse the traffic flow to detect any unauthorized activity in the network and then raise an alarm [58, 93]. It can identify attacks at run-time, but it is not defined to provide a response to the intrusion and hence cannot prevent any further disruptions of service [94, 95]. As the second line of defence, an IDS may perform the following actions within the framework as shown in Figure 3.3:

- Monitors and analyzes the system and user activities
- Audits the vulnerabilities and system configuration
- May also assess the integrity of data files and critical systems
- Commonly analyze abnormal activities

It can be a software or hardware, or a combination system to automate the intrusion detection process [96].

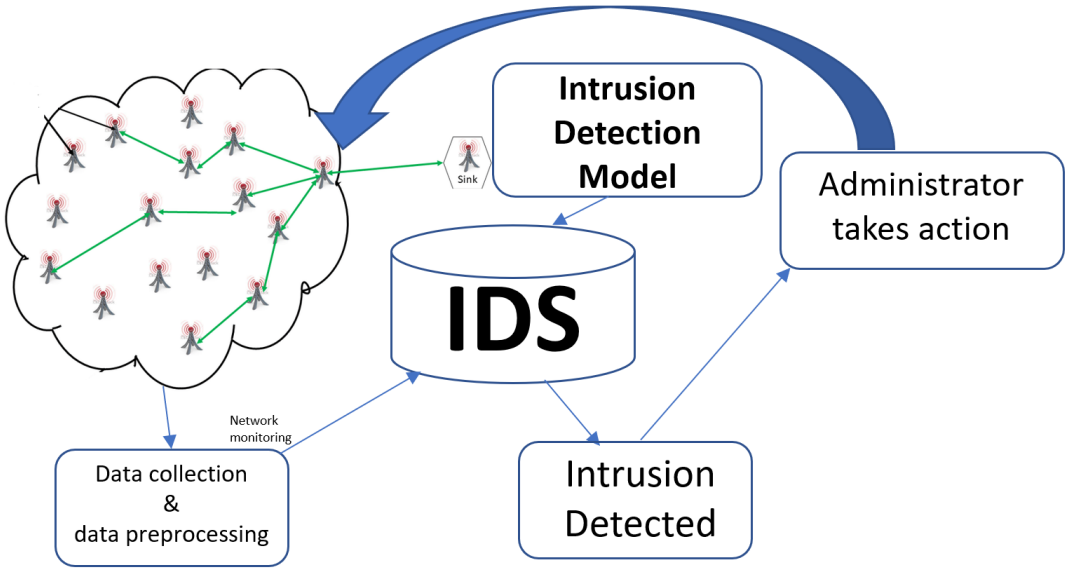

Figure 3.3: A typical IDS Framework

IDS's can be classified into three major categories [97–99]:

3.2.1 Signature based intrusion detection

In this category, IDS detects an intrusion based on the specific signature of an attack. Also known as Rule-based IDS, these IDS's detects intrusions with the help of built-in signatures. They are able to detect well-known attacks with great accuracy as signatures are recognizable and available compared to new attacks with unavailable signatures. [100].

3.2.2 Anomaly based intrusion detection

These IDS's monitors the abnormal behavior of a network. The anomaly-based IDSs can detect intrusion by matching the traffic patterns or resource utilization's of the known networks. They are capable of detecting both known and new attacks. The main problem with this category IDS's is the high rate false alarms i.e. a detection alarm in the absence of an intrusion is there or vice versa [101, 102].

3.2.3 Hybrid intrusion detection

An IDS that works as a combination of two mentioned above. This IDS maintains a signature database as well as monitors the traffic for any changes that may occur in the network behaviour to detect an intrusion. Composed of two detection modules; one for well-known attacks using signatures, and another, detecting the malicious patterns over the normal profiles; these IDS's are more accurate in terms of attack detection with a smaller number of false positives. The main drawback of the hybrid IDSs is the higher consumption of resources. Therefore, they are generally not recommended for WSN's [103].

One principle characteristic or role of the IDS is to detect an intrusion, and hence they are passive in nature. They only detect an intrusion, but cannot take any preventive action. Once they detect an intrusion, they may raise an alarm.

Functionally, an IDS mainly consists of four components, according to the Common Intrusion Detection Framework (CIDF), which are [104]:

1. Event generators
2. Event analysers
3. Event databases

4. Response units

These components are related in a very special way. Event generators are the intrusions in the network, but to build event analysers and event databases, several techniques may be used. Machine learning is a very common methodology used for building an IDS, as reviewed in the literature review chapter. For our work, we have also used the same methodology as machine learning techniques. The technique is about the capability of the machine to learn from data and to categorise the new data is normal or malicious. In the next section, a few terms and concepts related to machine learning are discussed.

3.3 Machine Learning Techniques

Machine learning is the process of making the machine learn from the past data without being explicitly programmed [105]. This technique utilizes several mathematical algorithms to train the model from the data stored in the form of datasets [68]. Main types of Machine Learning are Supervised learning, semi-supervised learning, unsupervised learning, reinforcement learning and deep learning [54]. Each form is supported by different machine learning algorithms [106–108].

3.3.1 Supervised Machine Learning

Supervised machine learning employs a training method using well defined labelled data to create predictions for the specific label. This is one of the benefits of supervised machine learning and the process of detection or decision making is comparatively easier because of the availability of examples or labels. Several algorithms are available that can be used to train the model using labelled data. Another benefit is that supervised learning is more accurate and less computationally intensive. This technique is used for classification problems like predicting color of a car, detecting a specific disease or a specific attack on the network etc. [106,107]. Another application is regression analysis which works on continuous values of data like recommendation and time series prediction etc.. Classification groups data into separate classes or labels with the distinct boundaries, whereas regression models works on the data using a mathematical equation model [109].

3.3.2 Unsupervised Learning

The method of learning, where machine may finds different kinds of unknown patterns in data. Used mainly for clustering and categorization, it is an easier method to work. For supervised ML, the data needs to be labeled, it always needs user intervention, but for unsupervised learning, it is easier to get some unlabeled data as the labels are not defined. Although it is a computationally complex to train the model, it is less accurate and trustworthy. Some of the examples are group of shoppers based on their purchasing needs, mails as spam etc. [107].

3.3.3 Semi-supervised Learning

Sitting between the above two, this method may contain both labelled and unlabelled data. Many real-world problems can fall into this category like organizing a photo archive etc.

3.3.4 Reinforcement Learning

A sub-field of machine learning, where machine is trained using set of actions, parameters and end values i.e. learning by trying the actions within the given parameters and then getting results. The learning continues accordingly. This is very commonly used by the gaming software‘s, where the game learns from the players move and reacts accordingly [108].

3.3.5 Deep Learning

This ML method is an autonomous, self-teaching system in which existing data is used to train the algorithms that may find patterns. These patterns are then used to make predictions about the new data that is given to the predictive model.

There are several algorithms used in the supervised machine learning for training the model. Using these algorithms like Decision Tree, Naive Bayes, SVM etc., a predictive model is built for future intrusion detection.

3.4 Machine Learning Process

Since in supervised machine learning, a model is built that learns from the set of training instances. Each instance is comprised of a vector of feature values and a class label given to that instance. ML analysis system goes through five phases to generate the model, set of feature vector and perform prediction using the model [110]. These five phases are:

1. Data Collection and preparation, which includes identifying the sources of the data, collection of data and pre-process it using different techniques of cleaning, selecting and transforming the data in the independent data format that can be used by any tool or language

2. Feature engineering is the extensive step that classifies and categorises the data. The phase is divided into feature extraction and feature selection that results in an optimal feature set.

3. Building a Predictive Model, includes selecting an optimum ML algorithm and training it using the data

4. Performance Analysis, is achieved by evaluating the results of the model using different measures such as classification accuracy, time and false positive etc.

5. Prediction on new unseen data is done by analyzing the new test data using the predictive model defined above to predict the class value of the new sample.

In many cases, developers have to go through an iterative process until the predictive model is optimized. The overall process of the model development is shown in Figure 3.4

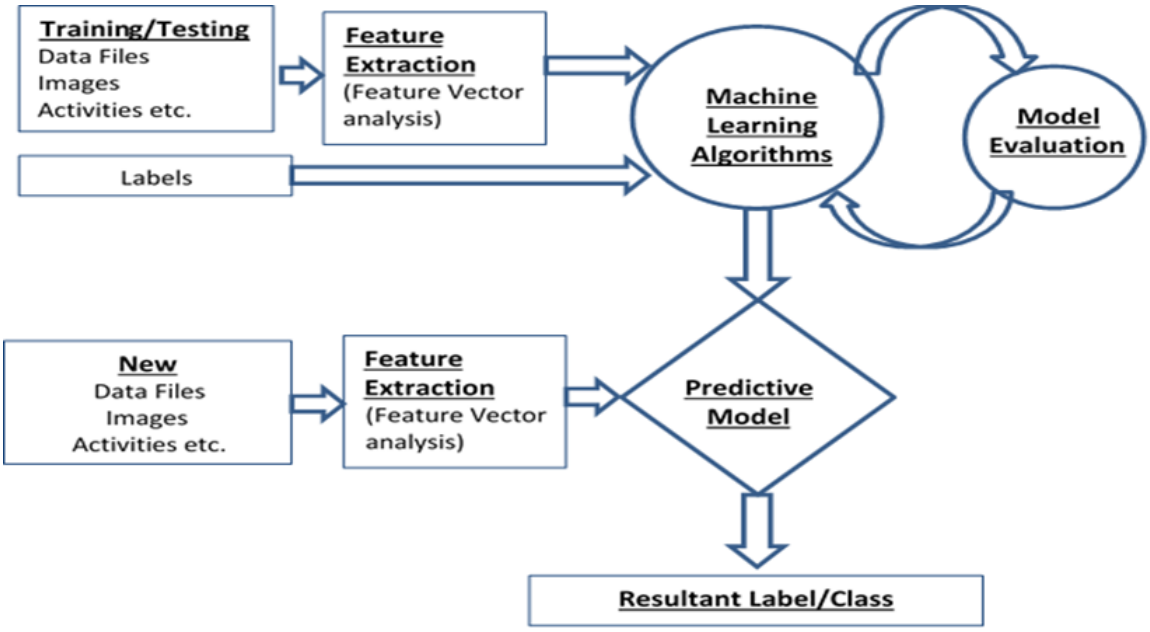

Figure 3.4: Steps in machine learning model development

In the following subsections, each step is described in detail.

3.4.1 Data Collection and Preparation

Data can be collected from various sources in several different forms. We need to collect the data and use it to train the models, so that the model can do predictions in future. But data collected from different sources is in different formats. Some data may be missing and needs to be fixed. Once data is collected, it needs to be completed, cleaned and then transformed into a machine readable form that can be used by Machine Learning algorithms. Generally, the machine learning classifiers may only deal with specific data types such as numeric or categorical, so the data needs to be prepared by filling empty data and then converting it into the usable form to be analyzed further. Some of the methods used for the data preparations may be:

1. Filling the missing data using data imputation
2. Encoding the data for the categorical values
3. Data balancing by feature scaling or data standardization
4. Transformation of the data to the form that can be used by the models or algorithms

3.4.2 Feature Engineering Analysis

Identifying features from the data is a difficult, time-consuming process and requires expert knowledge.*(Andrew Ng, Stanford University)*

Several machine learning experts, including Andrew Ng, consider feature engineering as the core of the machine learning work [111]. It is the process of getting an optimal feature set that can identify different scenarios. Executed in two steps i.e. Feature generation and feature selection, it is the most important step of the machine learning process.

1. Feature Generation i.e. feature selection and feature extraction Not all the features required are directly available. So, we need to perform two key tasks:
   a. Feature Selection - Selecting the all possible features,
   b. Feature Extraction - Extracting new features from the already existing features.

   Feature extraction commonly creates new attributes as combinations of others. For example, in our log files we have the timestamp for each activity. To generate new features, we may need to manipulate the different timestamps. For example to get a feature as time taken to transfer the data packet, we need to manipulate two different time-stamp values. Also known as feature projection, it helps in reducing the features by transforming data from higher features to less features using various mathematical algorithms like Principal component analysis (PCA), Kernel PCA or graph based PCA, semantic analysis, data compression or pattern recognition etc. This is the first step in getting a much smaller and richer set of attributes.

2. Feature Reduction or Dimensionality reduction: Dimensionality reduction is the method of removing features that are not relevant or are redundant [111]. This is done to find an optimal subset of the original features that may be used to build the model as well as for prediction later. There are three main methods of feature reduction [72].

   - Filter method - Also known as attribute evaluator & ranker method, features are extracted from data without any learning involved. This method is faster as features are selected using a pre-processing step, but it does not look for the effect of chosen features on the performance of machine learning classifier. It may look for the dependency between features using corre-

lation coefficient. The commonly used metrics for filter method are Pearson correlation, Mutual Information, Kendall Correlation, Chi squared, Fisher Score, Information Gain etc. [111].

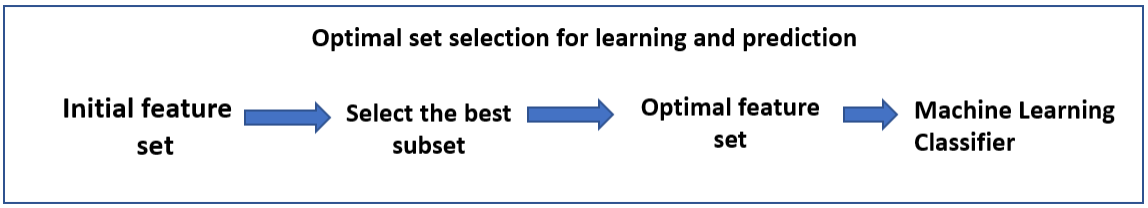

Figure 3.5: Filter method of feature selection

- Wrappers method or Subset evaluation is used to evaluate usefulness of the features by utilising learning techniques. In this approach, features are selected based on their impact on performance of the model. The ML algorithm is taken as a black box i.e. features do not see the algorithm, but accuracy of the classifier is taken as a measure to estimate the prediction accuracy using estimation techniques. Therefore, a wrapper method requires at least one machine learning algorithm that it will use as an evaluation criteria. Although, this method is highly intensive in terms of computation as it needs re-training for each subset yet it provides the best performing feature set for the used classifier. Some of the search algorithms for the features are Forward Selection, Backward Elimination, Recursive Feature Elimination, Genetic Algorithms etc. [112].

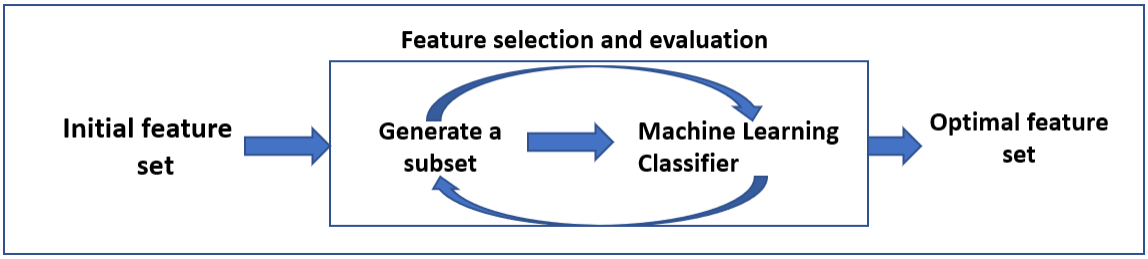

Figure 3.6: Wrapper method of feature selection

- The embedded or hybrid method techniques combines feature selection and classifier construction together. Generally implemented by algorithms with their own built-in feature selection methods.The benefits of embedded hybrid method techniques include high accuracy generalization and

interpretation ability. The most commonly used embedded technique is using decision tree algorithm.

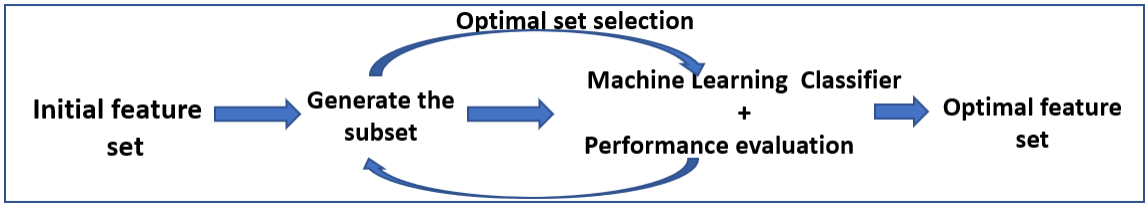

Figure 3.7: Embedded method of feature selection

3.4.3 Model Building

Once, data is ready in the defined format with the optimal feature subset, several machine learning algorithms (classifiers) may be used to train the predictive model, which is used for prediction in future. The role of a classifier is to map the input data to a specific category. There are several commonly used machine learning classifiers used by the ML community and they all follow completely different approaches to train the model.

Some of the commonly used classifiers are:

1. Linear and logistic classifiers such as Logistic regression, Naive bayes, Nearest neighbor, and SVM are used mainly for classification problems. They are easy to implement and show good performance [113, 114].

2. Tree based classifiers namely Decision tree, Boosted tree, and Random forest build the classification model in the form of a tree breaking down the data into miniature subsets. The final result is the tree having both decision the leaf nodes. The best feature or predictor acts as the root node of the tree. They are not essentially as accurate as other techniques [107, 114]. Random Forest grows many classification trees.

3. Layered classifier i.e. Neural network derives inspiration from the human brain. These classifiers consist of neurons and their connections. Each neural network has input and output, and the neural network function connects the inputs to outputs. The neural network can learn to do nonlinear mapping of input to output [115].

3.4.4 Model Evaluation

Once a model is built, we need to test the accuracy of the model. Taking an example of an Intrusion Detection system, the common evaluation metrics are explained below.

Table 3.2 defines the terms TP, TN, FP and FN.

Table 3.2: Metrics Terminology

Terminology	Derivation
True Positive (TP)	An attack is predicted when it is actually present i.e. attack is predicted correctly
True Negative (TN)	No attack is predicted in the network and there is no attack in fact too
False Negative (FN)	Predicts no attack even though network suffers from an attack
False Positive (FP)	An attack is predicted even though there is no attack in the network

The above measures are placed in the confusion matrix or contingency table, which is a technique for summarizing the performance of the classification algorithm. Another variation of these metrics are positive predictive value (PPV) and negative predictive value (NPV) which are the probability of attack predicted when there is an attack and probability of predicting attack when there is no attack.

A few more measures can also be used to evaluate the performance of a classifier as listed in Table 3.3. The goal is to maximise these measures, i.e. to 1 where minimum is 0. A higher values definitely indicates a better classification performance.

Table 3.3: Evaluation Metrics used for the classifiers

Terminology	Formula	Derivation
Sensitivity	$\frac{TP}{TP+FN}$	Measures the proportion of actual attacks correctly identified i.e. how good a model is in predicting the presence of any attack
Specificity	$\frac{TN}{TN+FP}$	Measures the proportion of normal networks correctly identified i.e. i.e. how good a model is in predicting the absence of any attack
Precision (PPV)	$\frac{TP}{TP+FP}$	Measures the ratio of correctly predicted attacks to the total predicted positive results of no attacks and of an attack.
Recall (TPR)	$\frac{TP}{TP+FN}$	Measures ratio of correctly predicted attacks in the network to the total predictions of a class predicted and not predicted
Accuracy	$\frac{TN+TP}{TP+FN+FP+TN}$	Measures the ratio of correctly predicted class to the total results of the specific class.
False Alarm Rate (FAR))	$\frac{FP}{FP+TN}$	The rate at which no attack is classified as an attack

The model evaluation can be done using several techniques:

1. Holdout method

Dataset is broken into training and testing set. As the term states, training set is used to train and testing set is used to test the model. A well used and proved method, but will have a high variance based on the division of training and testing set. The accuracy rate of the evaluation varies widely.

2. Leave-one-out cross validation

Dataset is again broken into training and testing set, but in this technique, testing set contains only one row of data and rest of the dataset is used as training set for training the model. Once the model is trained, it is tested using the testing data. This process is repeated n times, where n is the number of data's in dataset. This method is computationally expensive as several nodes are constructed. i.e. equal

in number to the size of the training set. But it provides accurate results due to extensive testing of the model.

3. n-Fold cross validation

The complete dataset is divided into n subsets. In each iteration, one of the n subsets is used as the testing set, whereas, the other n-1 subsets combined together make the training set. The testing set is used for prediction by the model and the error and success rate is recorded. This process is repeated n times. At the end, the average error across all n iterations is calculated. Since, in this method, every row of the complete dataset gets in the testing set exactly once, and is there in the training set n-1 times, the possibility of test accuracy is higher. The only disadvantage with this method is the long computation time requires to make an estimation as the training algorithm is iterated n times.[1].

4. Validation using new data

The entire dataset is used as the training set instead of breaking it down into testing and training sets. The new data is collected from similar environment for the known label and is used as testing data. The prediction results are analyzed to evaluate the model. [2]

3.4.5 Model Fitting

Once the model is built, it is imperative to calculate the goodness of the model. This can be done by testing the model on the similar data to what it has been trained on.

The well-fitted model will produce quite accurate outcomes. There may be variations in the fitting:

a. Over-fitting - Model matching the data very closely. This may happen when the model learns from very accurate details, even including noise (i.e. non relevant data). The over-fitting generally negatively impacts the performance of the model, while testing with new data.

[1]The original dataset is used both for training and testing in the methods 1, 2 and 3.
[2]This method uses new data from the network

b. Under-fitting - Model is not close enough to understand all variables. If the model is not able to generalize the training data with reference to new data, under-fitting may happen. It is obvious that the under-fit machine learning model will not be a suitable model, and will show poor performance on the new data.

3.4.6 Predictor

The main role of a Machine Learning model is to estimate, or 'predict' the outcomes or results in future. This outcome can be raising an alarm for an intrusion in the system. The predictive model works to provide predictions on the new scenario and re-calibrates the models in real-time automatically, once it has been designed. The role of the predictor is to collect the network traffic for the defined period, map the traffic to the deadset structure and pass it to the model for prediction.

Chapter 4

Assessing the security of a CPS network using NSES

4.1 Overall Research Plan

In order to answer the two research questions, the entire research was conducted in two phases. In first phase, security assessment of the network is done and in second phase, network security is enhanced by building an Intrusion Detection system. The overall work classification is shown in Figure 4.1

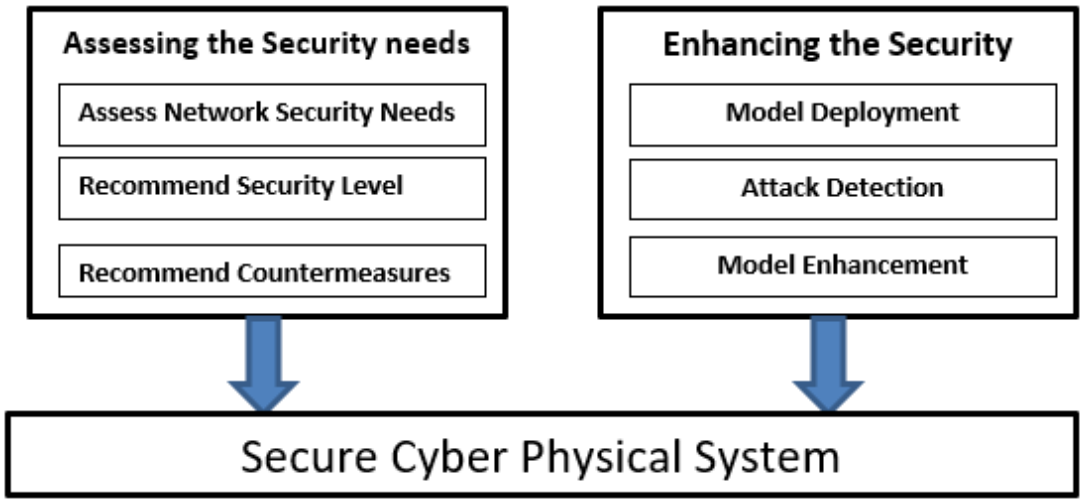

Figure 4.1: Two phases of the overall research

- Phase I - Assessing the security using a Network Security Evaluation Scheme

- Phase II - Enhancing the security using a novel Intrusion Detection system

From the next section, discussion of the phase-I of the research begins.

4.2 3-D Classification Model

Security requirements may vary from one CPS to another. If we can analyze the intensity of an attack, that may happen in a CPS deployment, based on the importance of the CPS application and the information that is being obtained and/or exchanged, its very easy to define the required security need. So we first worked on calculating the severity of an attack, and then using a novel scheme, we evaluated the security needs of the network.

We proposed a 3-D classification model that considers WSN attacks based on three different perspectives; Accessibility (A), Position (P), and attack Type (T) [12]. These perspectives could be represented in 3D space by three axes, where each axis represents one of the dimensions as shown in Fig. 4.2. Each axis could be used to represent one of three levels in one dimension. The following paragraphs explain the way this new classification could be used.

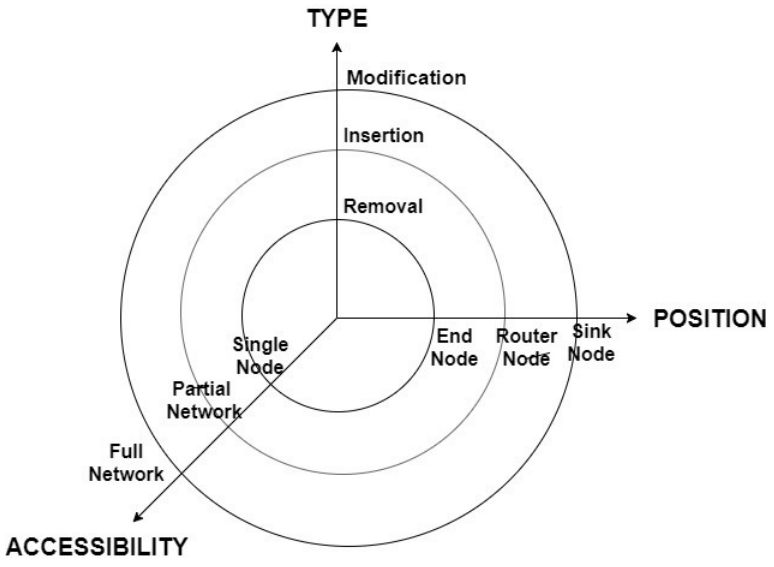

Figure 4.2: 3-Dimensions used for the APT classification

- **Classification based on Accessibility (A)**: This dimension classifies an attack based the number of nodes under attack. The intruder may only get control of one node, one area ranging from 2-3 nodes, or the entire network. The number of nodes under attack is a significant factor in identifying the severity of the attack and possible consequences.

- **Classification based on Position (P)**: This dimension classifies an attack based on the position of the node(s)-under-attack in the network. Nodes-under-attack can be end-nodes, router-nodes or sink-nodes. Since end-nodes do not re-route/forward packets to destinations, removing such nodes does not have a major impact on the network. However, node modification might allow the intruder to introduce malicious data in the network. Router-node hacking can help intruders launch several different attacks with major impact on the network. Similarly, sink-node hacking can shut down the entire network if hackers got access to the required resources.

- **Classification based on Type (T)**: This dimension classifies an attack based on the attack type. This includes node deletion, node insertion or node modification. The node removal is usually the easiest attack and the only impact it will have is the absence of data from a specific location, whereas node insertion allows the insertion of multiple similar nodes so that the data flooding may occur from a specific area. The most difficult and dangerous attack is node modification. It is a form of an active attack that allows the intruder complete control on the node that includes making changes in the node configuration. Alteration in the configuration can lead to different types of attacks such as sink hole, black hole etc. This is only possible when the attacker is experienced, and is capable of breaking all access controls and/or, may also has the ability of modifying the network protocol. Node modification is one of the most hazardous attacks and is the most difficult to detect. [116].

4.2.1 APT Quantification Description

A new quantification model is proposed to evaluate the severity of WSN attacks. Each axis in Fig. 4.2 is divided into three levels for simplicity. Based on this model, an attack may be quantified as a single point in this 3D space. The quantification levels of A, P and T are listed in Table 4.1.

Table 4.1: Quantification levels of A, P and T

Axis	Level	Value
Accessibility (A)	Single node only/Limited Access	1
	A few nodes/Partial Access(2 to 5)	2
	Entire network/Full Access (greater than 5)	3
Position (P)	End Node	1
	Router Node	2
	Border Router or Sink Node	3
Type (T)	Node Removal	1
	Node Insertion	2
	Node Modification	3

Using the quantification level on the 3-D axis, it is possible to calculate *totalValue*, which is an indication of the attack severity level. The calculation of the *totalValue* can done as follows.

$$totalValue = A + P + T \tag{4.1}$$

Where $A \in \{1, 2, 3\}$, $P \in \{1, 2, 3\}$, *and* $T \in \{1, 2, 3\}$ as shown in Table 4.1.

Table 4.2 shows all possible combinations of the proposed quantification scheme in Table 4.1 based on the 3D classification in Fig. 4.2. As shown in Table 4.2, there are 27 scenarios that represent all possible WSN attacks.

4.2.2 Using APT Quantification

The APT quantification scheme can be used to evaluate the severity of a WSN attack. As shown in Table 4.2, attacks may be quantified by different *totalValue* ranging from 3 to 9. The following is a few examples to explain by what means this quantification is applied to different attacks.

Table 4.2: Calculating the attack severity

No.	Number of nodes	Position	Type	*totalValue*
1	Limited	End node	Removal	3
2	(Single node)		Insertion	4
3			Modification	5
4		Router Node	Removal	4
5			Insertion	5
6			Modification	6
7		Sink Node	Removal	5
8			Insertion	6
9			Modification	7
10	Partial	End node	Removal	4
11	(2-5 nodes)		Insertion	5
12			Modification	6
13		Router Node	Removal	5
14			Insertion	6
15			Modification	7
16		Sink Node	Removal	6
17			Insertion	7
18			Modification	8
19	Almost Full	End node	Removal	5
20	(greater than nodes)		Insertion	6
21			Modification	7
22		Router Node	Removal	6
23			Insertion	7
24			modification	8
25		Sink Node	Removal	7
26			Insertion	8
27			modification	9

Table 4.3: Examples to explain the way our proposed quantification scheme is applied to different attacks.

Example Attack	A	P	T	*totalValue*
physical removal of a router node	1	2	1	4
insertion of extra end nodes	1	2	2	5
modification of one or more router nodes	2	3	2	7

4.3 Network Security Evaluation Scheme (NSES)

Using the *totalValue*, it is now possible to define the security needs. Higher values of *totalValue* will surely claim a better security for the system. To define these security claims, a new security evaluation scheme is described that can help standardize the security evaluation of IoT systems across different domains.

In the area of Cyber Physical Systems, this novel scheme can certify the security level of physical layer of the network and can recommend different security levels based on security needs of the specific domain of CPS.

Objective - Main objective of NSES is to maintain the required security level of physical layer of CPSs. Based on the basic security mechanism of prevention, detection and survivability [35], we have defined various levels. The minimum level of security is defined that may prevent any attack in the network as it will prevent a new insertion or removal of a node. But even then- an attack may occur. The next level will ensure an early detection of the attack. IPS on the next level will prevent it from expanding and breaking the entire network. Countermeasures deployed in the network may ensure a much secured and survived network.

Challenges - The main challenge for us was to design a security mechanism that can be used to prevent, detect and mitigate the attacks. Once the attacks at physical layer are classified, it becomes easier for system engineers to take different measures to protect the network from them. The proposed security evaluation scheme basis itself on countermeasures such as access control, key agreement, data encryption, secure routing protocol and trust management [3]. These countermeasures in a network ensure that any CPS network can achieve the level of security as required.

Advantages - This scheme has multiple levels of security based on the difficulty level of an intruder to get into the network. The applicability of the scheme is based on the knowledge about the processes controlled by CPSs and the required level of security maintenance in the system. The main advantage of this scheme is to define a standard evaluation method to secure physical layer of CPS. For ease and standardization, every level has been given a specific color to depict the security level of a network. This will help developers to better streamline their security expectations for various CPS applications.

Benefit of our security evaluation scheme is both for the network engineers and the clients. The network engineers can always claim to have the security properly implemented and functioning in their network deployment based on the clients specifications. In future, verification of such claims and a stamp of approval by several clients will strengthen the network vendors reputation of setting up the networks as per the clients specifications. The clients gets the surety of the security maintained in their network as required as well as they have the recommendations from the network engineers to enhance their security specifications.

NSES is applicable to almost every IoT and CPS‘s physical Layer.

4.3.1 Scheme Details

The five levels of the security range from A to F. This scheme considers five countermeasures i.e. Physical Security, Key Management, Cryptography & Access Control, IDS, IPS, and Secure Protocol. For gaining high level of security in any network, network engineers may keep adding more countermeasures as recommended.

- **Security Level A: Fully Secured Network** - Defined as the highest level of security, 'A' certified networks can detect any kind of attack and take preventive actions automatically. To ensure the highest level of security, this network has all the five major components i.e. Physical Security + Cryptography + IDS + IPS + Secure protocol in place. The Secure routing protocol ensures a correct and efficient route establishment between a pair of nodes. Any kind of attack can be traced and alarmed through this security level.
- **Security Level B: Highly Secured Network** - The network with 'B' level certification can detect attacks and perform basic level of prevention based on the tool used. This is done by raising alarm or inactivating the affected node to stop the invasion further. This network has all the four components except the secure protocol in place i.e. Physical Security + Cryptography + IDS + IPS. Since an IDS can detect attacks but cannot prevent or respond. To ensure high level security, an immediate action must take place once the attack is detected. The IDS must raise an alarm to inform the controller that may take an action to stop the attack impact further. The Intrusion Prevention System will prevent the invalid node to invade further.

- **Security Level C: Moderately Secure Network** - The network with 'C' level security can only detect the intrusion but cannot take any further action. All the attacks more than *totalValue* of 6 may be caught here. This level has the network with Physical Security, Cryptography and IDS in place. Generally, the security-related solutions like authentication and key exchange can provide some security however they cannot eliminate most of the security attacks [117]. The implementation of IDS can enhance the security of network as it will ensure the authenticity of the data transmission over the network. Known as the second line of defence, this security measure can detect an intrusion into the network at an early stage. Most challenging attacks occur due to node fabrication. An early detection of any attack is the objective of this security level.

- **Security Level D: Above Basic Secure Network** - This level is only defined to ensure the Confidentiality, Integrity and Authentication (CIA) control in the network. The data exchange across the nodes is also protected using encryption/decryption to maintain privacy in the network. D level networks also secures the messages transmitting across the network to maintain privacy. This can be done by adding Cryptography and Key Exchange in addition to physical security. This will ensure the Confidentiality, Integrity and Authentication (CIA) control in the network.

- **Security Level E: Basic Secured Network** - Defined as a minimally secured networks, 'E' level network has low level of security i.e. the nodes are physically secured and join the network using authenticated keys. For the minimum level of security, they only have physical security of the nodes ensured by adding software firewall. This level ensures that the nodes are somewhat physically protected and can join the network by only using the authenticated key. Based on the attack classification in our previous work, it takes care of *totalValue* of 3, 4 or 5 attacks.

4.3.2 NSES Color Codes

Color coding is a way to convey information quickly and effectively. We wish to add this advantage to our security scheme as well. To clearly indicate the level of security required form the basic (minimum security) to highest (maximum) security, we used the color range from red to green. Red color indicates minimum security whereas

bright (olive) green indicates high level of security.

The range of colors used at different levels of scheme are shown the Table 4.4.

Table 4.4: Color Scheme of NSES

Level	Description
Security Level A	A fully secured network
Security Level B	Highly Secured network
Security Level C	Moderately Secure Network that can detect the intrusion at an early stage
Security Level D	Above Basic Secure Network where message transfer is secured in the network
Security Level E	Basic Secured Network that only has physical protection and uses passkey to join the network

4.4 NSES - Case Studies

Use of this scheme can be illustrated by utilizing real life CPS. Since a CPS may range from smaller deployments like Body Area Networks (BAN) to larger one's like Environment Monitoring Systems, we used five different case studies and explained the applicability of this security scheme in their deployment.

4.4.1 Environment Monitoring System

There are several applications of the monitoring system such as agricultural, habitat, greenhouse, climate, forest monitoring etc. [118]. In such systems, reliability of the network is important in order to prevent packet loss. The main threats to this type of networks are physical tampering or unauthorized access by the intruder i.e. preventing a node removal or insertion of an unauthorized node. Not only that, if there is an intrusion, the network should be able to prevent the spreading of the intrusion by either blocking the infected node or by re-configuring the network to re-route packets.

The security specifications for environment monitoring system should be that no unauthorized user is able to join the Network. Also, the data transmission should be secure so that the message broadcast is also secure. Since the network is in large geographical area, we also need to monitor for any unauthorized activity happening

in the network like node removal or an unauthorized node insertion or a node replacement with a faulty node. To meet this security need, the sensors must be physically protected. Also, there must be key exchange mechanism in place that restrict the new unauthorized nodes to join the network. In addition, deployment of IDS/IPS will detect any intrusion at an early stage. So for that purpose, the network is highly secured at **Security Level B** as represented with green color in Figure 4.3. Just to note, Security Level B already covers the measures of Level C, D and E. This will ensure that the intrusion is detected and network is protected for spreading the attack.

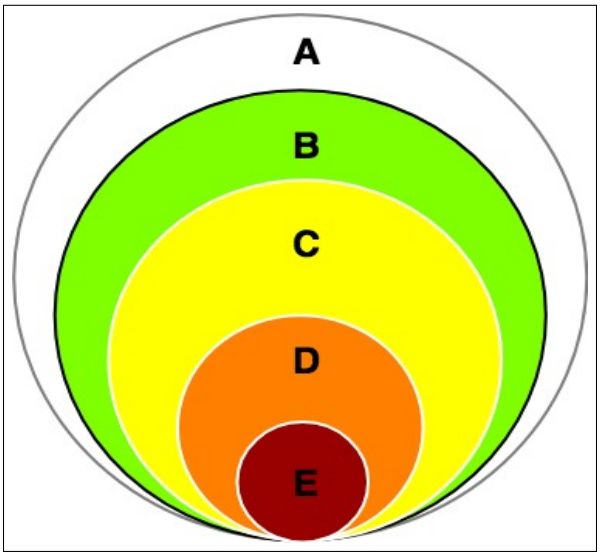

Figure 4.3: Security Level of Environment Monitoring System

4.4.2 Body Area Network

CPeSC3 (Cyber Physical enhanced Secured wireless sensor networks integrated Cloud Computing for u-life Care) is the use of WSN for health care [119]. In this system, the sensed data is either human health data or data to be used for detection of human activities for health care services. Here, the sensors are either attached to the body of the patient or onto the walls in the home environment, and these sensors can track the patients movement. This may collect data about the patients personal health or body movements. The recommendation is to have an Image-based authentication and activity-based access control mechanism to enhance security and flexibility of users access [119]. The Key Management Techniques are most relevant for data protection

in Mobile Cyber Physical Systems (MCPS) [120]. Another similar example is of Wireless Body Area Networks(WBAN), a communication network between the humans and computers through wearable devices [121]. They have also mainly stressed upon Cryptography, Key Management and Trust Management. Secure routing in their work is also to ensure the end-to-end communication verification purposes mainly. The recommendations in this work are also of security and privacy in transferring data like human body signals, requires authentication, integrity, access control, non-repudiation and encryption features.

Physical tampering of the nodes or unauthorized access to modify the readings can be considered as a major threat to this CPS. The need is that the patient‘s vital information must be stored and used with confidence. Moreover, for the patients with a socially unaccepted disease, confidentiality is critical. Any failure or leakage of this type of patient‘s health information could lead to humiliation, wrong treatments, relationship issues, or even job loss. In case of negative perception of the health information can also invariably hinder an individuals ability to get good treatment or coverage. Due to these reasons, it is critically important to ensure the security and privacy of medical data of different patients [122].

In this network, if we may just restrict unauthorized insertion as well secure message transmission, the network will work fine and will be maintaining the privacy of the data transmission. To match this need, the system administrators need to ensure that the network should have an above minimum-security level, where the environment will be controlled physically and has cryptography deployed. The physical protection will protect the sensors so that they are not intentionally tampered externally and Cryptography and key exchange barrier, will secure the message transmission. This will protect the patients data at the physical level as well only allow authenticated sensors and actuators. So, a moderately secured or above basic level of security represented with orange color is appropriately suitable for this network as shown in figure 4.4. For critical patients, who are may be disabled or so, **Security Level D** may be more appropriately suitable.

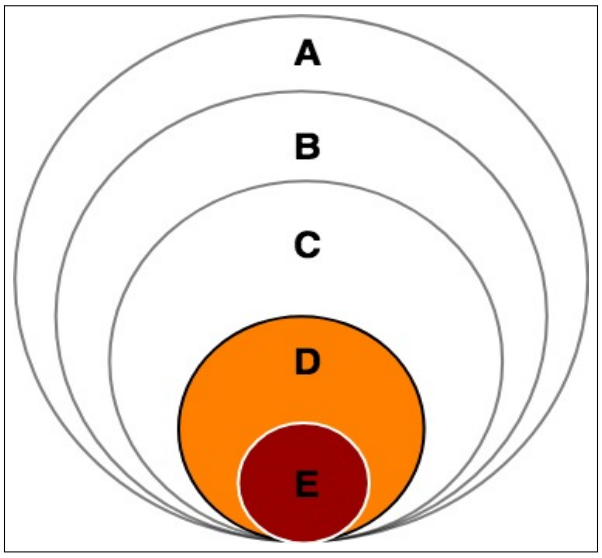

Figure 4.4: Security Level of Body Area Network(BAN)

4.4.3 Surveillance Control

The forest wildfire monitoring application is useful in remote areas. A Sybil attack is known to be a common attack on this network. A two-tier detection scheme is proposed by the authors [123]. Sybil attack is by the attacked nodes that transmit high false-negative alerts to an end user so that they may divert the attention to the less vulnerable geographical regions.

Common threats here are node modifications as well as the new node insertion. We need a fool-proof security and safety for this network. So, the network must be fully secure and protected at **Security Level A** and is coded using an olive-green color as shown in figure 4.5, so that both intruder and ringing of protection alarm takes place. Moreover, through the secure protocol, sybil nodes can immediately be identified and will be stopped from making an adverse effect on the network.

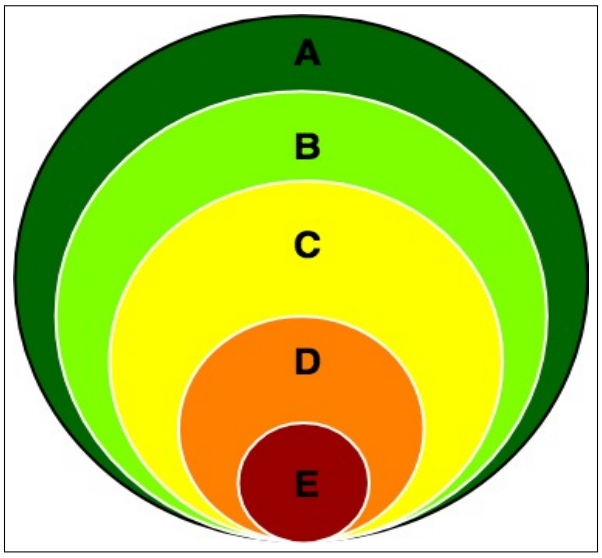

Figure 4.5: Security Level of Surveillance Control and Smart Home Systems

4.4.4 Smart-home System

The current boom is the Internet connected household devices, such as light-bulbs, cameras, smoke-alarms, and door-locks, leading to a s̈mart-home¨ [124]. This helps in family safety, property protection, lighting/energy management, as well as pet monitoring. Garden irrigation system is also an added component.

The various devices in the smart home are from different vendors. This vast heterogeneity in devices makes overall attack vectors very large and highly challenging for a security professional to cover out the entire threat space.

As per our scheme, the sensors need to have some physical protection. Since the coverage in these networks in heterogeneous space and devices, it is important to define security through secure protocol in addition to key exchanges and IDS/IPS so that any intrusion may be detected at an early stage. So for that purpose, the smart home network also needs to be fully secure at **Security Level A** and coded with olive green color as shown in figure 4.5 for the surveillance control as well.

4.4.5 Smart Cars

Smart cars, or popularly known as intelligent cars, are the vehicles that are regarded as environment-friendly, fuel-efficient, and safe by automatic monitoring of road threats. They also have enhanced entertainment and convenience features. The cars have multiple computers networked together, known as Electronic Control Units (ECUs). These ECUs mainly monitor and control various car functions. There are sensors that keep sensing the road threats and accordingly control the car activities like speeding or stopping [125].

The threat to a smart car is a hacker, who can attack a car's ECUs (target) by exploiting weakness in the wireless interfaces (vector) so that it can cause a collision or loss of control (consequence). If we may stop an external node to join the local network that is collecting data and checking with the outside threats, we can protect the car. So the plan may be to protect the car sensor network physically and disable the unauthorized node to join the car network; then, it will ensure security of the smart car. As per our scheme, the sensors need to have some physical protection. Since the coverage in these networks in heterogeneous space and devices, it is important to define security through key exchanges so that no new node can join the network . So for that purpose, the network should follow **Security Level D** as per our scheme and color coded with orange color as shown in figure 4.6.

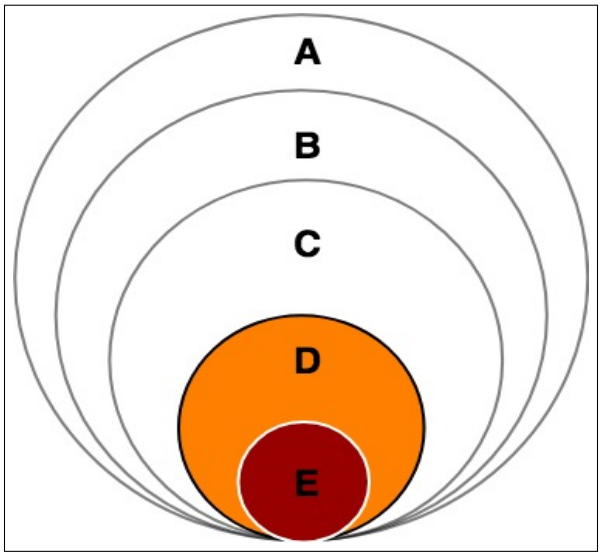

Figure 4.6: Security Level of smart-cars

Chapter 5

Enhancing the security using IDS

The second phase of our research answers our second research question, "How to build an Intrusion Detection System that can detect several known attacks?"

This chapter explains the methodology used for development of the proposed IDS. It is developed using Supervised Machine Learning. Machine learning is the methodology to make a machine learn automatically without much human intervention using the past data. The proposed IDS is trained to detect 5 different scenarios but the plan is to make it adaptable to detect more attacks in future, which is possible using machine learning techniques only.

Unlike previous researchers who used KDD datasets to train and test their models, our IDS will be trained using a novel dataset built from WSN network. The dataset is used to train as well as test the model and then the model is used to make predictions on the same network. This novel IDS is trained to detect four known attacks post-training. It will also be able to detect any new attacks in the network. This IDS can be deployed in the network to make predictions automatically.

5.1 Proposed IDS Framework

The proposed detection system is a complete framework comprised of three layers. The three layers of the proposed framework are:

- Layer 1 - Data sensor to create dataset

- Layer 2 - Model building & evaluation
- Layer 3 - Predictor where model is deployed to monitor the network and make predictions

The framework is shown in Figure 5.1

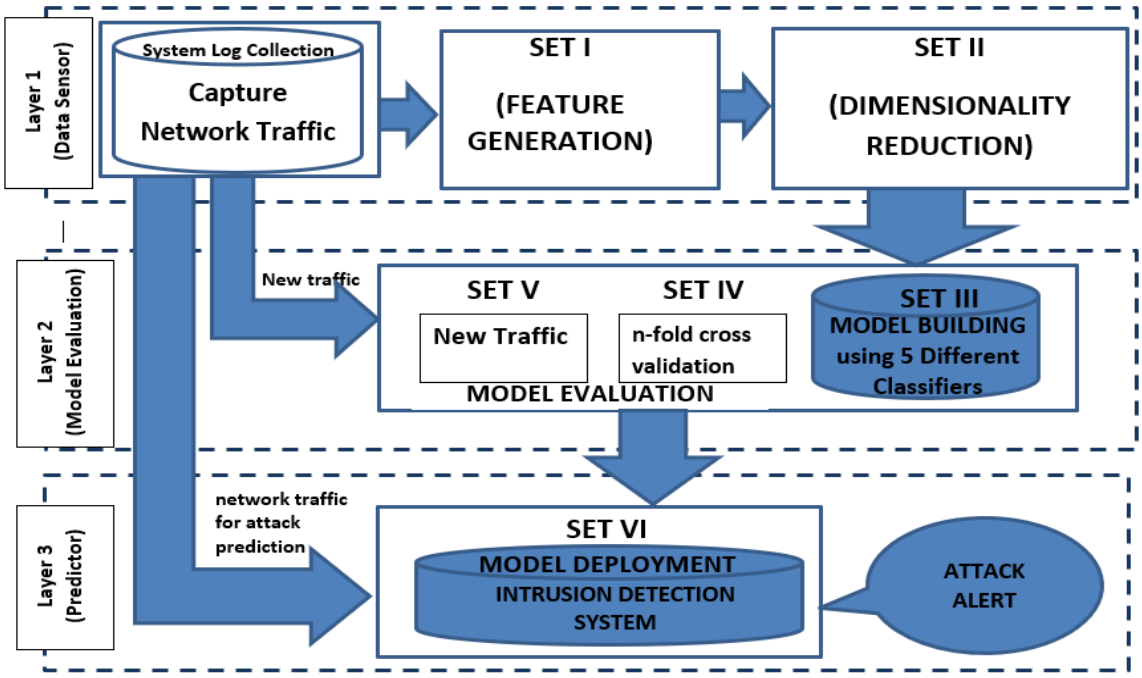

Figure 5.1: Proposed IDS Framework

Several algorithms are used for feature extraction, feature reduction, model building, model evaluation and model applications. As seen in the Figure 5.1, at every layer some algorithms are used. Different algorithms used together to complete a step is termed as a set. In total we defined six algorithm sets. Table 5.1 lists the summary of algorithm sets used at each layer of the model.

In the following subsections these layers are further explained.

5.2 Layer 1: Dataset Building

A brand new dataset is built from a simulated environment using 5 different scenarios. Several python scripts are written for different purposes to achieve the goal. These

Table 5.1: Summary of algorithms used at three layers of the Predictive model

Layer	Set #	Algorithm #	Description
Layer 1	Set I	Algorithm 1, 2, 3 & 4	Feature creation
Layer 1	Set II	Algorithm 5 & 6	Feature reduction
Layer 2	Set III	Algorithm 7	Model building
Layer 2	Set IV	Algorithm 8	n-Fold model validation
Layer 2	Set V	Algorithm 9	Model validation using new data
Layer 3	Set VI	Algorithm 10	Attack prediction

scripts may be used to analyze the .pcap file from the network and generate a new data to be added to the original dataset.

For the purpose of our work, initially four different attacks, belonging to different categories of attacks for RPL are chosen. The model is trained for normal networks; direct attack like DIS flooding and hello flood attacks; and topology attacks, like increased version and decreased rank attack. We have done the full radio packet analysis to build the dataset, which is further used to train the model. The selective forwarding attack is used to test the model. In order to train the model for the four type of known attacks in addition to its normal behavior, the need is to collect and analyze the network packets, for five different scenarios, which are:

1. Normal network with no intrusion
2. Network under hello flood attack
3. Network with DIS attack
4. Network with increased version attack
5. Network with decreased rank attack
6. Network with selective forwarding attack

For the purpose of analysis and prediction, a class value is given to each scenario. These class values are 0 for normal network, 1 for hello flood attack, 2 for DIS attack, 3 for increased version and 4 for decreased rank attack. A new class 5 is for selective forwarding attacks, which is not used in the dataset building, but is used for testing the model to detect a new attack, for which model is not trained.

5.2.1 Attack Vector

The simulation environment needs to be re-configured for introducing the attacks by modifying the source code. The network scenario is shown in the figure 5.2

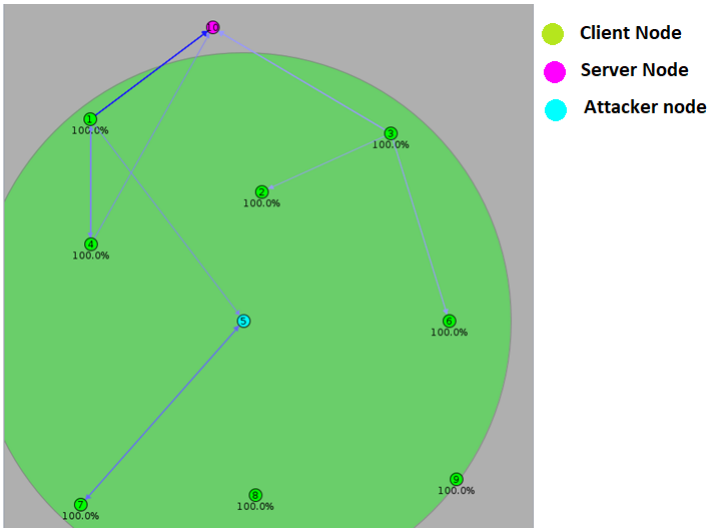

Figure 5.2: The network with 10 nodes

The way these attacks were introduced is explained below:

1. Hello flood attack : In order to make a node act as hello flood intruder, the original time gap between consecutive packet generation is reduced so that the attacker node starts generating data packets at the faster rate than the normal defined rate in the network.

Attack Scenario 1: Hello Flood Attack

Input:get node-id
if node-id *matches* attacker-node-id
increase the data packet generation interval
else
use default value of data packet generation
endif

2. DIS attack in the network: This attack occurs when the DODAG message transfer rate is increased by the attacker node [126]. The attacker node broadcasts

DIS meddages that are recieved by reciever nodes. The receiver node in turn reset the DIO timer, assuing there is an error with the topology around the timer. Extra DIS messages leads to an increase in end to end delay resulting in more control overhead, eventually leading to energy exhausting.

Attack Scenario 2: DIS Attack

Input:get node-id
if node-id *matches* attacker-node-id
  redefine the DIS message transfer interval
else
  DIS message transfer interval is default
endif

3. Increased version attack in the network: The attack results in DAG re-computation as the victim node increases DAG version upon receiving a DIO message and it sends out a new (poisoned) message forcing the re-computation [127, 128].

Attack Scenario 3: Increased Version Attack

Input:get node-id
if node-id *matches* attacker-node-id
  increase the DAG version
else
  regular DAG version
endif

4. Decreased rank attack in the network: The main aim of this attack is to attract traffic by advertising a low rank to disrupt routing paths [127].

Attack Scenario 4: Decreased Rank Attack
Input:get node-id
if node-id *matches* attacker-node-id
decrease the rank,
do not drop parents with a larger rank
else
keep regular rank calculations
endif

5. Selective forwarding attack: The main role of this attack is to forward only the packets passing through a specific filter, and dropping all other packets [127]. **This attack is only used for testing the model to classify a new attack.**

Attack Scenario 5: Selective Forwarding Attack
Input:get node-id
if node-id *matches* attacker-node-id
decrease the rank,
do not drop parents with a larger rank
else
keep regular rank calculations
endif

The radio traffic log is generated as a .pcap file. The purpose of this file is to analyze the changing patterns in absence and presence of an intrusion in the network. The .pcap file is analyzed using a network analyzer and converting it into 2 separate .xml files. These .xml files are used to gather required information about the specific scenario.

The .pcap file as seen in Wireshark is shown in Fig. 5.3.

The dataset building is accomplished in two steps:
A. Feature generation - Using the Algorithms Set-I, the pcap file is analyzed and first converted to an XML file using tshark network analyzer. The XML file is parsed using python scripts to make an entry into the datset fie in the .csv format. This dataset file has the information about one session in total 58 different values, called features.

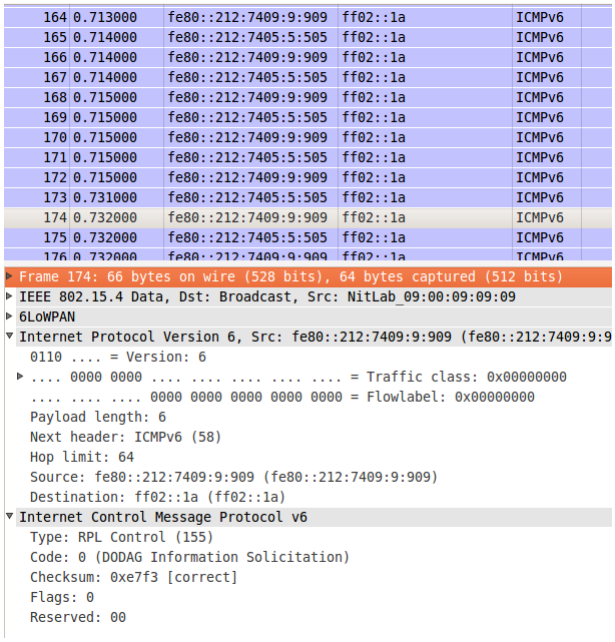

Figure 5.3: Packet file of a session as seen in Wireshark protocol analyzer

B. Feature reduction - Not all the 58 features are useful. We need to extract the most useful features and this is done by using feature reduction. Feature reduction is done using two methods: Filter method and Embedded method [72]. Both the methods gave different results. The time and accuracy scores are compared to make the optimized model. Next section explains these steps in more details.

5.2.2 Building Dataset

A supervised machine learning model is trained using a training set. The training set contains instances of both normal network traffic as well as instances of the network under attacks. Data is collected from Cooja simulator log files and is converted into a form that is easier to interpret by the scripts that train the model. Following that, the features engineering is done by performing two steps: feature generation and feature reduction to get an optimal feature subset.

Algorithm 1 Filtering Packet Data

Require: pcap file

1: get pcap from traffic
2: loop
3: if protocol=udp
4: transfer in udpXML
5: else transfer into allXML
6: end if
7: end Loop

A. Feature Generation Using the .pcap capture of each run, UDP data is filtered from the rest of the data in two separate XML files using command line parser, which is shown in Algorithm 1. A sample of XML frame from data packet is shown in Fig. 5.4.

Figure 5.4: XML data format

Post filtration, each XML file is parsed using a custom script to generate selected data in .csv file. This process is explained in Algorithm 2 for UDP data and Algorithm 3 for all data. Finally, all generated .csv files are processed and analyzed to generate a final list of features, as shown in Algorithm 4.

A test environment is created in Cooja simulator to generate the data used to build the training set. The details of the environment setup is discussed in details in section 3.3. Once each packet of one sample run is parsed, a summary of complete simulation is generated as one row of data in the final dataset.

Algorithm 2 Parsing UDP packets from XML format

Require: initialize counters
1: for every packet in the network do
2: check for every protocol in each packet
3: for each field in protocol do
4: check for frame_number
5: check for frame_time
6: get the 6lowpan source and destination
7: Ipv6 source and destination
8: write information on .csv file

Algorithm 3 Parsing all packets

Require: initialize counters
1: for every packet in the network do
2: check for every protocol in each packet
3: for each field in protocol do
4: check for frame_number
5: check for frame_time
6: get the 6LoWPAN source and destination
7: Ipv6 source and destination
8: Write PacketNo, TimeDelta, FrameNo, Frame length, frameProto, wSrc, wDest

Based on the simulation, a class label is assigned to every row. This class label represents the type of network traffic in each run. If the traffic flow is normal, then the class label is set as normal i.e. 0. If the traffic flow is captured while the network is under attack, the class label is set as the type of attack. The result of this labelling is creation of a dataset containing 58 features.

A snapshot of the dataset showing a few features is shown in Fig. 5.5.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	totSimTin	totalPacke	out1	out2	out3	out4	out5	out6	out7	out8	out9	out10	in1	in2	in3	in4	in5	in6
2	103.663	3418	261	300	223	351	329	362	316	431	349	224	379	236	0	50	0	
3	85.594	3046	231	265	220	302	287	316	294	385	299	224	290	218	0	46	0	
4	94.797	3316	261	299	221	343	324	357	298	414	332	224	306	236	0	50	0	
5	84.64	3035	231	265	220	302	287	316	286	385	299	224	282	218	0	46	0	
6	125.342	3818	265	382	227	375	333	426	337	456	378	306	498	282	11	89	0	1
7	109.576	3464	262	302	223	351	333	366	316	431	349	252	389	264	0	50	0	
8	110.049	3466	262	302	223	351	333	366	316	431	349	253	389	265	0	50	0	
9	108.667	3434	262	301	223	351	333	366	316	431	349	224	389	236	0	50	0	
10	103.489	3418	261	300	223	351	329	362	316	431	349	224	379	236	0	50	0	
11	136.061	3941	267	416	228	384	351	431	346	465	386	306	553	290	11	89	32	1
12	113.258	3578	264	333	223	367	333	366	329	431	349	291	432	273	0	85	0	
13	111.495	3466	262	302	223	351	333	366	316	431	349	253	389	265	0	50	0	
14	108.856	3434	262	301	223	351	333	366	316	431	349	224	389	236	0	50	0	
15	109.727	3466	262	302	223	351	333	366	316	431	349	253	389	265	0	50	0	
16	112.863	3590	264	337	223	367	333	371	329	431	349	291	437	273	0	89	0	

Figure 5.5: A snapshot of dataset

Table 5.2 lists all the features generated at this step with their descriptions. The generated dataset represents the traffic patterns of four attack vectors in addition to the normal network traffic. The attacks represented by the dataset are hello flood attack, DIS attack, decreased rank and increased version attacks with the corresponding class values of 0, 1, 2, 3 and 4.

Algorithm 4 Summarizing data

Require: initialize counters

1: import totalData.csv in dataframe
2: count total number of packets
3: calculate total transmission time
4: calculate messages going out of each node, and going in each node
5: import udpData.csv in another dataframe
6: analyze data frame to get number of hops for each message
7: get time delta for a message travel to final destination
8: calculate number of messages going in and out
9: calculate number of different data travelling in the simulation
10: total time for data transfer
11: average time for each message transfer
12: calculate ratio of data packets to total packets
13: calculate total number of messages originating from each node

To make an efficient and successful IDS, dimensionality reduction is required to reduce the overall time and processing complexity of implementation. Moreover, the features are not equally important and do not contribute the same weight to the classification accuracy of the IDS system. Therefore, feature reduction plays a vital role in improving the overall performance of the system.

Table 5.2: The dataSet with the collected features

Feature	No.	Description
totSimTime	1	Total Simulation Time
totalPackets	1	Total number of packets flowing in the simulation
out1..out10	10	total number of messages going out of the nodes 1 to 10
in1..in10	10	total number of messages going in the nodes 1 to 10
totalUDP	1	total number of UDP data packets travelling in the network in simulation time
totalTimeMess	1	time of the UDP messages passing in the network
totalMess	1	total number of UDP messages in the network in simulation time
avgTimeMess	1	time of the UDP messages passing in the network
uout1..uout10	10	udp messages going out of the nodes 1 to 10
uin1..uin10	10	udp messages coming in the nodes 1 to 10
avgHops	1	average number of hops for data to travel to destination in the simulation
ratio	1	ratio of UDP packets to total data packets
dropped	1	number of packets dropped
totalFromNodes	9	Number of packets originating from each node
class	1	the class identifier identifying the each row in dataset
Total	58	Total number of features

B. Feature Selection/Reduction The generated dataset includes a list of 58 features, as shown in Table 5.2. These features are saved in comma separated value (CSV) format file. However, some of the information is not needed during the training phase as it creates patterns that overlap between classes and may lead to incorrect classification in some cases. Moreover, having many features requires a lot of processing cycle on the target sensor nodes.

Feature reduction is required to select only the most significant features that can

be used to distinguish different attack patterns. This step also helps the machine learning algorithm to get trained faster and reduces complexity of the generated model.

The two main reasons for the feature reduction are :
(1) to reduce the mathematical complexity of the feature space so that the modeling algorithm can work more efficiently, and
(2) to reduce the noise for the target signal by discarding the features that do not contribute to the identification of the attack patterns [129].

We used two different methods for the purpose of feature reduction:

1. Calculate the importance of each feature using Random forest classifier (Embedded method, where a ML classifier is used)
Embedded methods use a subset of features and train a model using them. Based on the results from the model, features are added or removed. We used random forests classifier to train and validate. We have only used Random forest classifier as the first feature selection procedure so that it may provide an indication of the changing classifiers accuracy when features are changed. Random forests are very commonly used for feature reduction, as it uses tree-based strategies that naturally ranks the features. Nodes with the greatest importance happen to be placed at the start of the trees, while the less important features are at the end of trees. So, the pruning of the trees below a particular value can generate a subset of the most important features. Final feature set is defined with the features having importance > 0.04, as explained in Algorithm 5.

Algorithm 5 Generating optimal feature set(Embedded Method)

Require: Object of Random Forest Classifier
1: fit the model with original dataset
2: calculate feature importance
3: for each feature in list do
4: check for threshold value
5: value > 0.04, keep the feature
6: else
7: discard feature
8: store feature set as optimal set

2. Correlation based feature selection (Filter method, no classifier is needed)
This method selects features that are highly correlated with the class, but uncorrelated with each other [130]. The results are again checked after reducing the features.

Final feature set is defined with the features having correlation > 0.30, as explained in Algorithm 6.

Algorithm 6 Generating optimal feature set(Filter Method)

Require: original data set
1: calculate feature correlation
2: for each feature in list do
3: check for correlation value
4: value > 0.30, keep the features
5: else
6: discard feature
7: store feature set as optimal set

5.3 Layer 2: Model Building and Evaluation

A very popular theorem, "no free lunch" [131], explains the importance of several machine learning classifiers especially for classification problems. The theorem states that for any algorithm, any elevated performance over one class of problems is offset by performance over another class [131]. So, the recommendations are to always test different classifiers for testing on specific domain problems. For our problem of intrusion detection, we also used ensemble method, where we selected five different classifiers for building a model. This approach will help in improving the overall results by combining several models, allowing the model to produce better predictive performance compared to a single model.

Different machine learning classifiers takes different approaches for using features to make predictions. We chose five different classifiers using different approaches. The used classifiers are [132]:

1. State Vector Machine - A commonly used algorithm used for classification problems. The algorithm maps each feature into an n-dimensional feature space. These features are then defined into hyper-plane separating the data items by maximum margins, defining the classifications.

2. Decision Tree Classifier - Used for classification problems, it is a tree based algorithm, where input variables are used to make branches. Travelling through a tree provides the information to reach to a defined class.

3. Random forest classifier - An extension of Decision tree classifier, where a collection of many trees is used for classification. Very deep tree leads to over-fitting of the training data. Several trees analyzes the input data and then using the classification of "votes" produces the final result. In Random forest, since the outcomes are derived from many different Decision Trees, the variance is reduced as compared to getting the result from a single DT for the same dataset.

4. Naive Bayes - This classification algorithm is based on the probability of an event based on the prior knowledge of conditions related to that event. The main assumption of this algorithm is that a specific feature of a class is not directly related to any other feature, even though other features may be interdependent.

5. Logistic Regression Model - This classification algorithm is quite powerful as it classifies only as true or false. It can further be modelled to be used for categorical predictions for several class problems.

5.3.1 Model Building

The model is built using the features and then evaluated. Algorithm 7 shows the procedure used for model building.

Algorithm 7 Predictive Model Building

Require: Object of Random Forest Classifier
Require: Object of State Vector Machine
Require: Object of Decision Tree Classifier
Require: Object of Naive Bayes Classifier
Require: Object of Logistic Regression Classifier
1: fit the models with training dataset
2: for each classifier in list do
3: train the model
4: fit the model
5: store model in package file

Since the model is being used for detecting known attacks as well as for new

attacks, we have selected decision tree algorithm as one of the choices. Bouzide has experimentally proved that decision trees are more interestingly capable in detecting new attack [133]. Mukherjee and Panda [134, 135] have used naive bayes successfully for intrusion detection. SVM has already been approved to be an excellent choice for intrusion detection [66, 136]. Thus, the above mentioned five algorithms were chosen for the research. For the purpose of detection, prediction results from all the classifiers are taken and the majority decision is taken as final.

First, the model is built and evaluated using all 58 features. Then model performance is again compared with the new optimal dataset generated using two methods as listed above. The results are compared in next chapter.

5.3.2 Model Evaluation

The model is evaluated using two methods:

1. n-fold cross validation evaluation
2. new data from the similar network (Not seen by model ever)

n-fold cross validation

Algorithm 8 shows the procedure used for n-fold cross validation

Algorithm 8 Model Validation (n-fold Cross Validation)

Require: get the optimal feature set
1: split the dataset into test and train set
2: train the model using train set
3: test the model with test set
4: perform this n times (based on n folds)
5: train using training set
6: test using test set
7: store the results

It works by splitting the dataset into k-parts (e.g. k=5 or k=10). Each split of the data is called a fold. The algorithm is trained on k-1 folds with one held back and tested on the held back fold. This is repeated so that each fold of the dataset is given a chance to be the held back test set.

Validation with new data

Algorithm 9 shows the procedure used for validating the model using new data

Algorithm 9 Model Validation (Using new data)

Require: Model using 5 classifiers
1: get the feature set from the data
2: pass the features through the 5 classifiers
3: perform polling
4: if majority decision found
5: declare the class
6: else
7: repeat the predictions 20 times
8: perform polling 20 times
9: store the majority decision if more than 10 times
10: declare the decision

As mentioned in the algorithm, the class prediction is accomplished using polling. Since there are five different classifiers used for prediction, it is possible they may give different results. So, the decision is made based on ***Prediction method - Polling*** as explained in section 5.4.4.

5.4 Layer 3: Predictor

The proposed model is trained with a training set that represents 1 normal scenario and 4 attacks. When the model is deployed in the network, we may come across three different scenarios, which are:

1. A normal network traffic with no attacks.
2. A network with one of the four known attacks in active mode.
3. A network with a new attack is active or any other combinations, but not the normal network traffic.

The proposed predictive model is capable of identifying any of the above mentioned scenarios as explained in the algorithm below:

Algorithm 10 Predicting attack in new network

Require: 5 TRAINED MODELS

in the first step, prediction results from all the five models are collected
if at least three models declare a same class, then
    alert the predicted class i.e. from 0 to 4
else
if the polling result is < 0 (i.e no clear prediction), then
repeat 20 rounds of predictions
    keep adding the final predictions of every round
if any class prediction > 10 times, then
    predict the class,
else
    alert as a new attack, and store data in new archive for future analysis

5.4.1 Attack prediction

For predictions, the proposed IDS is deployed at the node, which can observe the overall traffic flow of the entire network. The data is collected after every 30 minutes for checking an intrusion. The traffic for 15 minutes is collected as a .pcap file. Using the defined algorithm, the new traffic makes a single line .csv file. This file is passed through the models and predictions are recorded.

For the predictor to work, Set I and Set II of the algorithms (listed in Table 5.1) are applied on the .pcap files and then the newly formed .csv file is passed through the model for prediction.

5.4.2 Attack prediction for known attacks

For the known scenarios 1 and 2 as listed above, the class is clearly predicted as 0 to 4. Since there are 5 classifiers used for prediction, the polling is done and the decision is taken as the final result.

5.4.3 Attack prediction for new attacks

When the network has a new attack for which model is not trained for, the different classifier predict them separately. In that case predictions are repeatedly done and the decision is made as explained in algorithm 10.

5.4.4 Prediction method - Polling

Each new log data from the network is passed through the algorithms 1 to 4 to generate a new .csv file containing latest network signatures. This signature file is passed through the model for prediction. Model predicts the result of the new data using five classifiers namely, SVM, Random Forest Classifiers, Decision Tree Classifier, Naive Bayes and Logistic Regression Classifier. All the five results are then passed for polling. The majority decision is taken as the final result. If in the first test, no majority decision is found, the testing is done 19 more times to get the final majority out of all runs, and the classification of more than 50% is taken as final result. If there is no majority decision, we lead to the decision of alarming it as an attack but that is a new attack as is not in our class definitions. We may claim that as a new attack and this leads to our future work.

5.5 Experimental Setup

To set up RPL network for this work, a network simulator named Cooja is used which is designed for simulating the sensor networks over the Contiki Operating system. This is a Java based simulator but allows sensor nodes to be written in C. It is a flexible, cross-level simulator, and allows the nodes to be in different levels of both software and hardware. Cooja combines both low-level and high-level simulation, and is extensible to different sensor node platform, operating system software, radio transceiver and radio transmission models [137].

5.5.1 Network setup

For our experiments, 9 sensor nodes (UDP-Client) were deployed randomly with a single sink(UDP-Server). Each sensor node sends the data periodically that must reach the sink or UDP-Server. The sensor data may directly reach the UDP-Server, if it is in the range of the Server, else it travels using the hops. The sink is located within the network. We do assume that all sensor nodes and the sink are time synchronized. At the time of setting-up, the UDP server gains knowledge about the network topology by the nodes that report to their neighbouring nodes(as the network set-ups). The UDP server or border router performs network analysis as it is more powerful thus there is a confirmation that the UDP-Server or border Router

is trustworthy and always free of compromises. Unfortunately, compromised sink discussion is out of the scope of our work.

The inbuilt default OF0 objective function is chosen for the sample simulation. The traffic is captured as radio messages in the .pcap file, which is further analyzed using command line t-shark network analyser.

Here, each node is a TMoteSky4 node. The normal network is shown in Figure 5.6 and the network with attacker is in Figure 5.7. Each UDP-client sends an UDP packet to the server once a minute in normal cases. For the purpose of simulating the attacks, the source code of the simulator written in C, is modified. Hello Flood and DIS are the resource consuming attacks, whereas reduced rank and increased version are routing attacks. In all the attack scenarios, node 5 is the victim or compromised node, nodes 1, 2, 3, 4, 6, 7, 8, and 9 are the normal clients and node 10 is the UDP-server node.

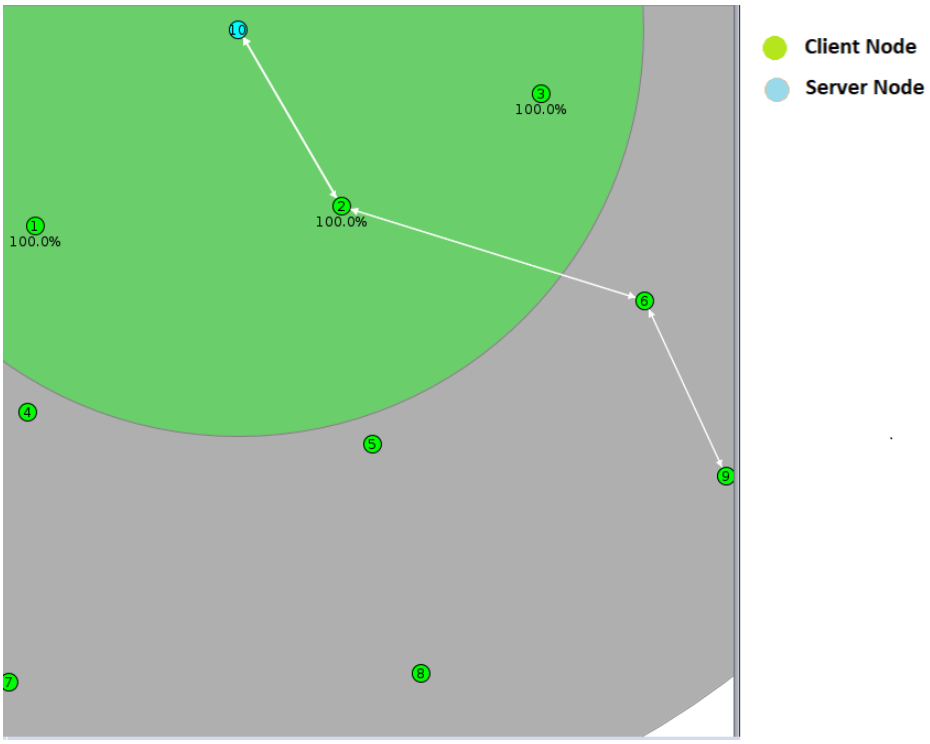

Figure 5.6: Nodes in the normal network

The data flowing in the network is stored in .pcap file which we have used for building the dataset as shown in Figure 5.8. Several simulations were run for each scenario ranging from 10 time units to 15 time units. .pcap capture of each scenario is used to build one row of the dataset. Each row is assigned a specific label based on

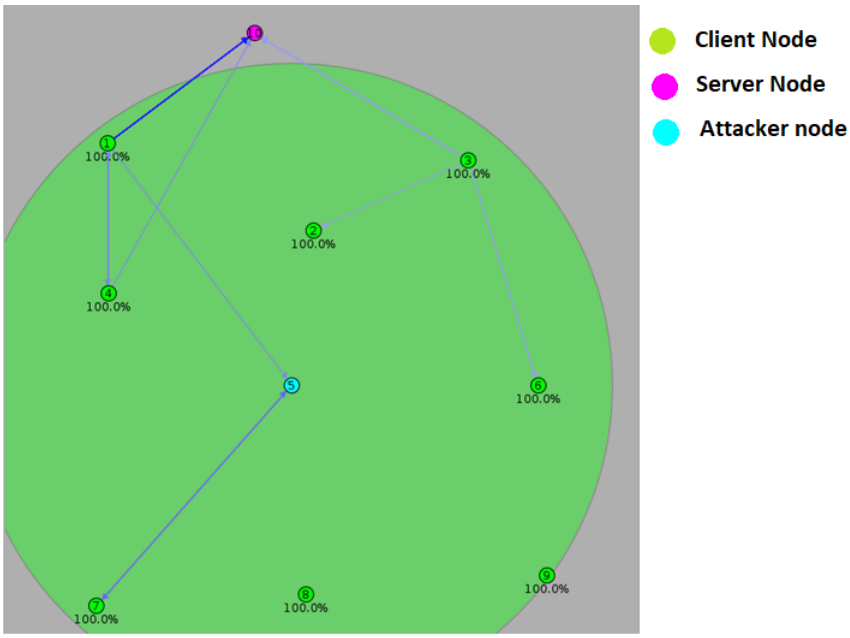

Figure 5.7: Nodes in the compromised network

the scenario it belongs to as 0, 1, 2, 3, or 4. The label is the class value that represents a specific attack vector and will be use to train the network using supervised machine learning.

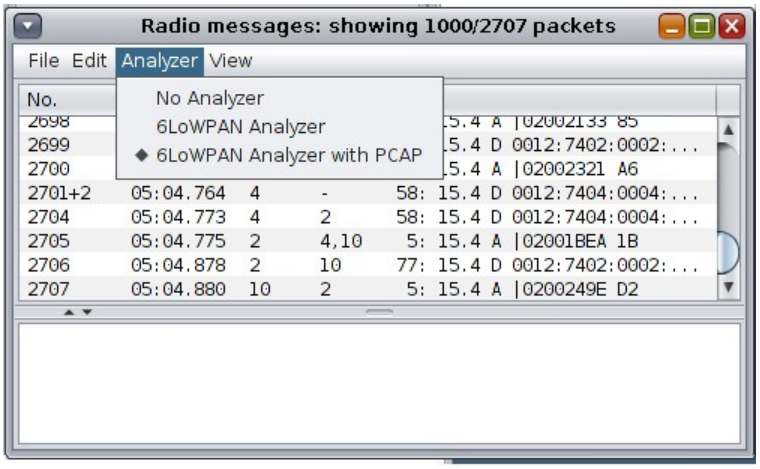

Figure 5.8: Gathering data as a .pcap file

5.5.2 T-shark Network Analyzer

The pcap file from the simulator can be analyzed using the Wireshark network protocol analyzer. But for the purpose of less memory usage, t-shark is used which is

command line version of the network analyzer. This powerful open source network analyzer can capture the network traffic and can inspect closely the happenings in the network. The .pcap file, generated from Contiki was used by t-shark to explore and divide the messages as required. The analysis of this .pcap file may provides a plethora of information about the network, link, node and the packets, which we have used for the purpose of full packet analysis of the network. T-shark analyzer converts the .pcap files to a .XML file having the values in key pair values which can be extracted using the algorithms defined using python. Algorithm 1 does this function for us.

Chapter 6

IDS Results Analysis

6.1 Layer 1: Dataset Building

After the creation of the .pcap file with 58 features and one class value, features need to be reduced further.

Two different feature reduction methods namely the filter method and embedded feature reduction method using Random Forest Classifier are applied in layer 2. In the embedded method, RFC used a classifier to observe the effect of the feature of the final result. For filter-based feature reduction using correlation, the correlation between the features is calculated to extract the features.

6.1.1 Correlation based feature reduction (Filter Method)

The model accuracy was tested with different values of correlation. If the correlation was greater than 0.25, 15 features were extracted, for the values greater than 0.30, 13 features and the correlation greater than 0.35, only 6 features were needed for the model building. The accuracy scores are listed in Table 6.2.

Table 6.1: Accuracy score with original features and reduced features with different values of correlation

Model	Accuracy with original features	Correlation >0.25	Correlation >0.30	correlation >0.35
features#	58	15	13	6
RFC	100%	100%	100%	98.33%
SVM	61.66%	71.66%	73.33%	90%
DTC	100%	100%	100%	100%
GNB	80.00%	80.00%	80.00%	85.00%
LRC	98.33%	93.33%	98.33%	98.33%

6.1.2 Feature reduction using Random Forest Classifier (Embedded method)

The Embedded method utilises a random forest classifier to check the impact of feature(s) on prediction accuracy of the model. For different values of impact ranging from greater than 0.02 (yields 17 values), than to 0.03 (yields 14 values), than to 0.04 (9 features) and then 0.05 (yields 6 features)

Table 6.2: Accuracy score with original features and reduced features with different values of importance

Model	Accuracy with original features	Importance > 0.02	Importance > 0.03	Importance > 0.04	Importance > 0.05
features#	58	17	14	9	6
RFC	100%	100%	96.66%	100%	96.66%
SVM	61.66%	61.66%	66.66%	71.66%	75.00%
DTC	100%	100%	100%	100%	100%
GNB	80.00%	76.666%	78.33%	78.44%	78.33%
LRC	98.33%	93.33%	98.33%	100.00%	93.33%

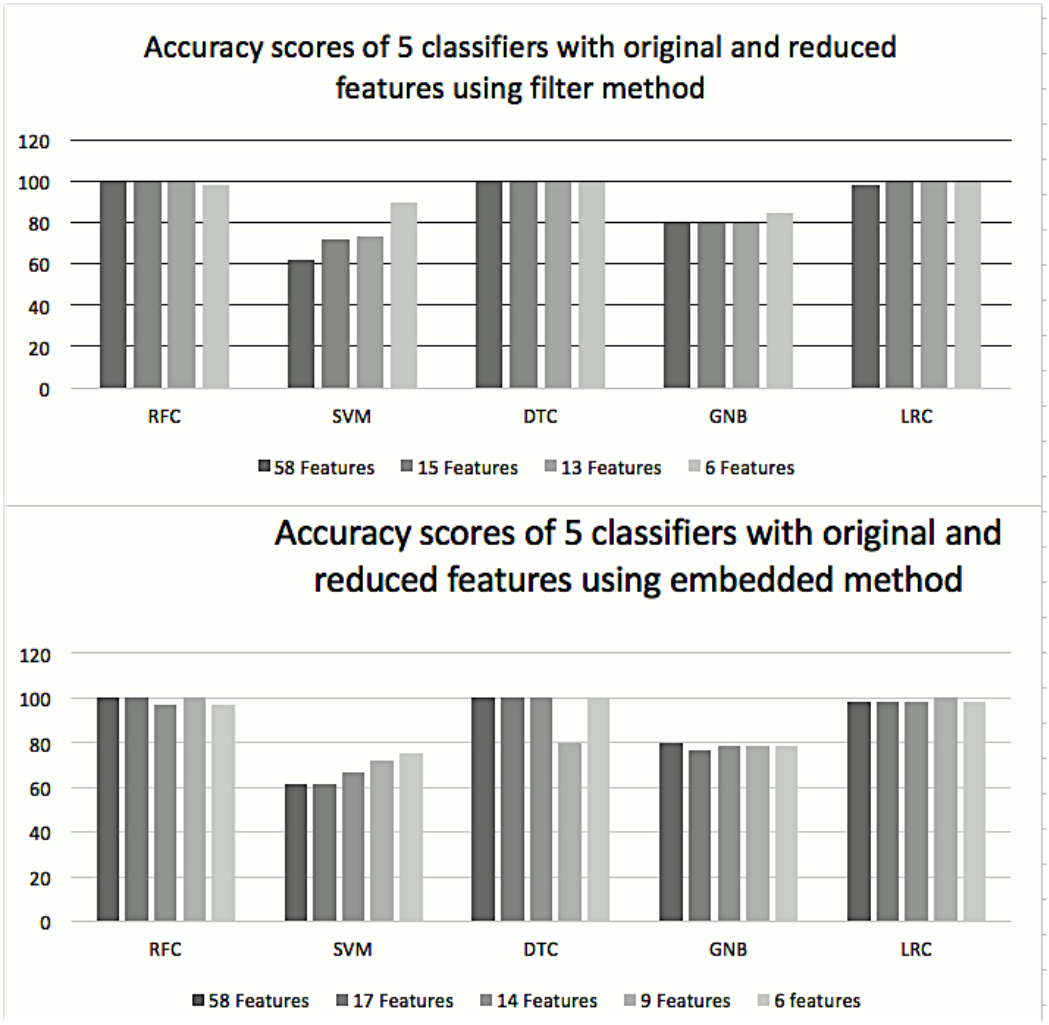

Figure 6.1: Comparison of accuracy scores of five classifiers using embedded and filter methods with different values.

6.1.3 Decision on the optimal set from both the methods:

Filter method:

The accuracy scores of the feature set with a correlation greater than 0.30 was same for four classifiers and increased the score of one classifier so the value of 0.30 was chosen as final. The number of reduced features are 13, which are: out5, in1, totalUdp, uout2, uout5, uout6, uin1, uin2, uin4, uin5, uin6, ratio, o2.

Embedded method:

The accuracy scores with features having importance greater than 0.04 were the same for three classifiers, improved for one and reduced a little bit for the last one i.e. GNB. The number of features selected is 9, which are out5, in2, in4, in5, uout5, uin2, uin4, uin5 and o5.

In the next layer, the optimal set selected here is used for the model building and evaluation.

6.2 Layer 2: Model Building and Evaluation

In this layer, two models were built using the reduced features from the two methods discussed in layer 1. Each model have objects of five different classifiers so that the model can predict the results with high accuracy rate.

Once build, model evaluation was done in two steps:

- Step 1 - n-Fold cross validation - using the original dataset
- Step 2 - Model Evaluation using new data from the similar network

Table 6.3 lists the terminologies used in the evaluation process.

Table 6.3: Metrics Terminology

Terminology	Derivation
True Positive (TP)	Correct prediction of an attack
True Negative (TN)	Correct prediction of no attack
False Negative (FN)	Incorrect prediction of no attack
False Positive (FP)	Incorrect prediction of attack

Table 6.4 explains a few other evaluation metrics that may be used for the analysis.

6.2.1 n-Fold cross validation results

For model testing, 80/20 cross validations were performed. The analysis for class 0, class 1, class 2, class 3 and class 4 are summarized in following subsections.

Terminology	Formula	Derivation
Sensitivity	$\frac{TP}{TP+FN}$	Measures the proportion of actual attacks correctly identified i.e. how good a model is in predicting the presence of any attack
Specificity	$\frac{TN}{TN+FP}$	Measures the proportion of normal networks correctly identified i.e. i.e. how good a model is in predicting the absence of any attack
Precision	$\frac{TP}{TP+FP}$	Measures the ratio of correctly predicted attacks to the total predicted positive results of no attacks and of an attack.
Recall	$\frac{TP}{TP+FN}$	Measures ratio of correctly predicted attacks in the network to the total predictions of a class predicted and not predicted
Accuracy	$\frac{TN+TP}{TP+FN+FP+TN}$	Measures the ratio of correctly predicted class to the total results of the specific class.

Table 6.4: Evaluation Metrics used for the classifiers

Results of model built using embedded method

The performance of the classifiers are evaluated against individual scenarios.

In every case, TP and FP rates are evaluated. For a normal scenario, DTC, GNB and LRC show 100% detection rate. For Class 1 (i.e. Hello flood attack), the accuracy score is 95% for Decision Tree, 93.0% for RFC and GNB, 96.7% for SVM and 83.3% for LRC. For DIS attack, RFC, SVM and DTC has 100%, but 96.7% for GNB and 90% for LRC. Rates for increased version and decreased rank, none of the model has 100%. Table 6.5, Table 6.6 and Table 6.7 lists the TP scores of models, confusion matrices of attack classification for each machine learning algorithm and results like precision, recall, accuracy, specificity and sensitivity for 5 different scenarios using embedded methods respectively.

Table 6.5: TP scores of X-validations using embedded method

Model	Class 0	Class 1	Class 2	Class 3	Class 4
RFC	98.3	93.3	100	83	95
SVM	98.3	96.7	100	96.7	96.7
DTC	100	95	100	95	88.3
GNB	100	93.3	96.7	86.7	95
LRC	100	83.3	90	88.3	95

Table 6.6: confusion matrices of attack classification for RFC, SVM, DTC, GNB, LRC using embedded method

(a) Random Forest Classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	0	0	1	0	Normal
3	56	0	1	0	Hello Flood
0	0	60	0	0	DIS attack
1	0	3	53	3	Increased version
1	0	2	0	57	Decreased rank

(b) SVM classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	0	0	1	0	Normal
1	58	0	1	0	Hello Flood
0	0	60	0	0	DIS attack
0	2	0	58	0	Increased version
0	0	2	0	58	Decreased rank

(c) Decision Tree classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
60	0	0	0	0	Normal
4	50	5	1	0	Hello Flood
1	2	54	2	1	DIS attack
2	1	3	53	1	Increased version
0	0	1	2	57	Decreased rank

(d) Naive Bayes classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
60	0	0	0	0	Normal
2	56	0	1	1	Hello Flood
1	0	58	1	0	DIS attack
3	0	1	52	4	Increased version
0	0	2	1	57	Decreased rank

(e) Logistic Regression classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
60	0	0	0	0	Normal
1	57	0	1	1	Hello Flood
0	0	60	0	0	DIS attack
1	0	1	57	1	Increased version
1	0	2	4	53	Decreased rank

Table 6.7: Precision, Recall, Accuracy, Specificity and Sensitivity for 5 classifiers

(a) Normal Scenario

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	97.5%	97.5%	97.5%	97.5%	97.5%
DTC	85.1%	100.00%	100.00%	100.00%	100.00%
RFC	95.2%	100.00%	100.00%	100.00%	100.00%
GNB	85%	100.00%	100.00%	97.50%	100.00%
LRC	93.00%	100.00%	100.00%	100.00%	100.00%

(b) Hello Flood

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	95.1%	97.5%	97.5%	97.5%	97.5%
DTC	92.10%	87.50%	87.5%	100.00%	87.5%
RFC	97.4%	92.5%	75.00%	97.50%	92.50%
GNB	100.00%	92.50%	92.50%	100.00%	92.50%
LRC	97.50%	97.50%	97.50%	25.00%	97.50%

(c) DIS Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	100.00%	100.00%	100.00%	96.11%	100.00%
DTC	92.10%	87.50%	87.50%	78.50%	87.50%
RFC	97.00%	100.00%	100.00%	97.50%	100.00%
GNB	97.60%	100.00%	100.00%	97.50%	100.00%
LRC	97.60%	100.00%	100.00%	100.00%	100.00%

(d) Increased Version Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	97.40%	95.00%	95%	95.00%	95.00%
DTC	94.60%	87.50%	87.50%	100.00%	87.50%
RFC	100.00%	92.50%	87.50%	97.50%	92.50%
GNB	100.00%	100.00%	87.50%	97.50%	87.50%
LRC	100.00%	92.50%	92.50%	97.50%	92.50%

(e) Decreased Rank Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	100.00%	100.00%	100.00%	100.00%	100.00%
DTC	100.00%	100.00%	100.00%	100.00%	100.00%
RFC	100.00%	92.50%	82.50%	97.50%	92.50%
GNB	100.00%	100.00%	100.00%	97.50%	100.00%
LRC	100.00%	97.50%	97.50%	100.00%	97.500%

Results of model built using Correlation/Filter method

Correlation-based feature reduction removes features that have low correlation with the class variable. The performance of the classifiers is evaluated against individual scenarios.

Experimentally, for every case, TP and FP rates are evaluated. Table 6.8, Table 6.9 and Table 6.10 lists the TP scores of a model, confusion matrices of attack classification for each machine learning algorithm and results like precision, recall, accuracy, specificity and sensitivity for 5 different scenarios using filter method respectively.

Table 6.8: TP scores of X-validations using filter method

Model	Class 0	Class 1	Class 2	Class 3	Class 4
RFC	88.3	86.7	93.3	88.3	96.7
SVM	98.3	95	95	95	95
DTC	98.3	90	88.3	85	95
GNB	98.3	95	95	91.7	96.7
LRC	98.3	95	95	93.3	81.7

Table 6.9: Confusion matrices of attack classification for the 5 classifiers using filter/correlation method

(a) Random Forest Classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
53	0	5	1	1	Normal
2	52	2	2	0	Hello Flood
1	0	56	2	1	DIS attack
1	1	3	53	2	Increased version
1	0	0	1	58	Decreased rank

(b) SVM classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	0	0	1	0	Normal
1	57	2	0	0	Hello Flood
0	0	57	3	0	DIS attack
0	1	1	57	1	Increased version
0	0	0	3	57	Decreased rank

(c) Decision Tree classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	1	0	0	0	Normal
4	54	1	1	0	Hello Flood
1	2	53	4	0	DIS attack
2	1	3	51	3	Increased version
0	0	1	2	57	Decreased rank

(d) Naive Bayes classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	0	1	0	0	Normal
1	57	2	0	0	Hello Flood
1	0	57	2	0	DIS attack
0	1	0	55	4	Increased version
0	0	0	2	58	Decreased rank

(e) Logistic Regression classifier

Normal	Hello Flood	DIS	Inc. Ver.	Dec. Rank	Classified as
59	0	1	0	0	Normal
1	57	2	0	0	Hello Flood
1	0	57	2	0	DIS attack
1	1	1	56	1	Increased version
0	0	4	7	49	Decreased rank

Table 6.10: Precision, Recall, Accuracy, Specificity and Sensitivity for 5 classifiers

(a) Normal Scenario

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	95.00%	95.00%	95.00%	8796.11%	83.47%
DTC	86.80%	82.50%	82.50%	98.75%	82.50%
RFC	87.00%	100.00%	100.00%	92.50%	100.00%
GNB	87.2%	85.00%	85.00%	92.50%	85.00%
LRC	81.80%	90.00%	90.00%	92.50%	90.00%

(b) Hello Flood

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	97.4%	95.00%	95.00%	87.4%	95.00%
DTC	100.00%	75.00%	75.00%	97.50%	75.00%
RFC	97.4%	92.50%	92.50%	97.50%	92.50%
GNB	95.00%	95.00%	95.00%	95.00%	95.00%
LRC	90.50%	95.00%	95.00%	95.00%	95.00%

(c) DIS Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	100.00%	100.00%	100.00%	100.00%	100.00%
DTC	86.70%	97.50%	97.50%	92.5%	97.5%
RFC	94.5%	92.5%	92.50%	97.50%	92.50%
GNB	97.60%	100.00%	100.00%	97.50%	100.00%
LRC	93.00%	100.00%	100.00%	100.00%	100.00%

(d) Increased Version Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	92.30%	90.00%	90.00%	90.00%	90.00%
DTC	68.6%	87.5%	87.5%	87.5%	87.5%
RFC	100.00%	90.00%	90.00%	87.50%	90.00%
GNB	87.50%	87.5%	87.5%	87.50%	87.50%
LRC	89.70%	87.50%	87.50%	85.00%	87.50%

(e) Decreased Rank Attack

Algorithm	Precision	Recall	Accuracy	Specificity	Sensitivity
SVM	95.20%	100.00%	100.00%	100.00%	100.00%
DTC	94.4%	85.00%	85.00%	85.00%	85.00%
RFC	95.10%	97.50%	97.50%	87.50%	97.50%
GNB	97.50%	97.50%	97.50%	85.00%	97.50%
LRC	100.00%	87.50%	87.50%	85.00%	87.50%

6.2.2 Model evaluation using new data

In order to see the results more accurately, the testing is done with new unseen data. Now the .pcap files from the same environment were collected for the network without attack, with hello flood attack, DIS attack, increased version and decreased rank attacks. For the purpose of model testing, 10 simulations were done for each scenarios i.e. from class 0-4. The results found are summarized in following subsections.

Results of model built using embedded method

Table 6.11 lists the results of model built using embedded method.

Table 6.11: Accuracy score of model testing with new data using 5 classifiers

Model	Class 0	Class 1	Class 2	Class 3	Class 4
RFC	100%	90%	100%	100%	100%
SVM	80%	100%	90%	60%	90%
DTC	100%	100%	100%	90%	100%
GNB	60%	100%	100%	60%	90%
LRC	100%	100%	100%	100%	100%

Normal scenario: Logistic Regression, Decision Tree Classifier and Random Forest classifiers always gave the correct results. SVM predicted correctly 8 times and 2 times predicted it as an attack. Naive Bayes was able to detect only 6 times, and was incorrect 4 times. But, as we adopted polling for the final decision, the result predicted was always a normal scenario.

Attack 1 (hello flood attack): RFC was able to predict 9 times as an attack, whereas all other classifiers gave correct prediction all the 10 times. Though polling was done, the correct result was achieved in the very first attempt.

Attack 2 (DIS attack): For this attack SVM was correct 9 times, whereas, all others were able to correctly predict class 2.

Attack 3 (increased version attack): Random Forest, and Logistic Regression classifiers always gave correct results. SVM and Naive Bayes correctly predicted 6

timed and DTC was able to provide 9 correct predictions, but polling always got the correct result.

Attack 4 (decreased rank attack): Random Forest, Decision Tree and Logistic Regression classifiers always gave correct results. SVM and Naive Bayes were able to predict accurately 9 times, but polling always got the correct result.

Table 6.12: Overall results of model built using embedded method

Metric	Value
Overall Success Rate	90%
Overall Time taken	0.0329 ms
Overall Sensitivity	1.0
Overall Specificity	1.0

Results of model built using correlation method

Table 6.13 lists the results of model built using correlation method.

Table 6.13: Accuracy score of model testing with new data

Model	Class 0	Class 1	Class 2	Class 3	Class 4
RFC	100%	80%	100%	100%	100%
SVM	80%	80%	80%	70%	50%
DTC	80%	0%	50%	100%	60%
GNB	100%	80%	50%	60%	100%
LRC	60%	70%	80%	100%	100%

Normal scenario: RFC and Naive Bayes classifiers always gave the correct results. SVM and DTC gave correct prediction 8 times and 2 time it predicted it as attack 2. LRC prediction were least to 6 out of 10. But polling result was always clearly predicted a normal scenario.

Attack 1 (hello flood attack): All classifiers except Decision Tree classifier gave correct prediction all the 10 times. Decision Tree was wrong all the 10 times. But polling gave a correct result in the very first attempts.

Attack 2 (DIS attack): This is the only scenario where the first attempt did not lead to the correct prediction. All the classifiers predicted different results, mainly switching between class 2 and class 3. The first attempt in polling led to 20 iterations of prediction for the final result in our IDS. Since Random Forest classifiers always gave correct prediction and Logistic Regression was predicting the correct class most of the time, the net result after 20 iterations gave a correct prediction of 2. (19 times it was 2)

Attack 3 (increased version attack): Random Forest, Decision Tree and Logistic Regression classifiers always gave correct results. SVM was correct 7 times and GNB only 6 times, but polling always got the correct result.

Attack 4 (decreased rank attack): Random Forest, Naive Bayes and Logistic Regression classifiers always gave correct results. SVM was given incorrect prediction 50% times but Decision Tree predicted correctly 6 times, but polling always got the correct result.

Table 6.14: Overall results of model built using correlation method

Metric	Value
Overall Success Rate	90%
Overall Time taken	0.0319 ms
Overall Sensitivity	1.0
Overall Specificity	1.0

6.2.3 Selecting the model for the predictor layer

Three sets of evaluation metrics are considered for the final decision making on selecting the predictor model to be deployed.

1. Detection rate: Ability of the model to accurately classify input data (i.e. average accuracy score of each model)

2. Sensitivity: Calculating Sensitivity of each model

3. Detection time: The amount of time taken by each model to predict the result

Table 6.15: Decision making for the predictor model

Metrics	Filter Method	Embedded Method
Detection Rate	90%	90%
Sensitivity	1.0	1.0
Detection Time	.0319 ms	0.0329 ms

Both methods seem to work equally well based on the above figures. But, since the embedded method uses DTC for feature reduction, we chose to keep the features extracted through the embedded method for our final predictor.

6.3 Layer 3: Predictor

The IDS is deployed at the node that is observing traffic flow of the entire network. Data is collected every 30 mins for checking an intrusion. Traffic for 15 mins is collected as a .pcap file and using the defined algorithm, the new traffic makes a single line csv file. This file is passed through the models and predictions are recorded.

6.3.1 Predicting known attacks

We tested with 10 instances each of 5 different scenarios. The performance for both the normal traffic and the four defined attacks were quite similar to the previous section. The summarized result is shown in Table 6.16.

Table 6.16: Predictive model performance for the known attacks

Model	Normal Traffic	Class 1	Class 2	Class 3	Class 4
RFC	9	8	10	10	10
SVM	8	5	5	8	9
DTC	7	2	6	10	9
GNB	10	10	6	10	10
LRC	7	8	8	10	10

For attack 2 and attack 3, predictions were correct 50% of the time when polling was used.

6.3.2 Predicting new attack

The model was tested for a new attack by introducing it (the new attack) in the network. For 20 different simulations, data was collected as a pcap file and the new dataset for these 20 runs was created. When passed through the model, the prediction was as follows:

1. None of the model predicted class 0
2. Random Forest Classifier predicted 2 or 4 several times
3. SVM predicted class 1 all the time
4. Decision Tree Classifier predicted 3 or 4 a few times
5. Naive Bayes classifier predicted it as 3 all the time
6. Logistic Regression Model classified it as class 2 all the time

As we discussed polling, that when in the first prediction, at least three models predict the same class, we iterate the prediction 20 times. If in the end, at least in 10 iterations, a clear class is predicted, we declare the attack of that class. But if there is no clear class prediction made for at least 10 iterations, we call it a new class of attack. As a result of polling, only 3 times class 3 was predicted and rest of 17 times, it was a new attack leading to a prediction accuracy of 85%.

The results found is summarized in Table 6.17.

Table 6.17: Predictive model performance for the new attack

Model	Class 0	Class 1	Class 2	Class 3	Class 4	Final Decision
RFC	0	1	11	1	5	2
SVM	0	20	0	0	0	1
DTC	0	0	0	7	13	4
GNB	0	0	0	20	0	3
LRC	0	0	20	0	0	2

For future work, we archive this in a separate file. Our future work will be to analyze these archives to see a pattern and maybe define them as a new class or classes.

Chapter 7

Conclusion & Future work

7.1 Conclusion

This chapter concludes the research work presented in the dissertation. The first part of the research work provides first, an APT scheme to quantify attacks based on A (accessibility), P (Position) and T (Time). Followed that, a new assessment method for the security of the Cyber Physical Systems is proposed named as Network Security Evaluation Scheme(NSES). The proposed NSES is divided into 5 different security levels based on the deployed countermeasures. The five levels of the security start from the very basic security level 'E' to the highest security level 'A', which is the fully secured network. Level 'A' covers all the levels from E to A regarding the security countermeasures deployed at every level.

Supported by a five CPS/IoT examples, the use of NSES is explained with a particular focus on the security need of every network and the recommendations of the security level from the NSES. These recommendations can be used by network administrators at early design phases to define the security needs of the network and then match them at the time of deployment.

The second part of the research is focused on enhancing the security of the network. For this purpose, a novel Intrusion Detection System is developed using supervised machine learning. The IDS is trained using the training data from the WSN network, without direct programming. The IDS also has the capability of adapting itself independently through iterations.

This IDS can detect 4 known attacks, as well as several unknown attacks, is proposed. This novel IDS works in three layers.
Layer 1: Data sensor to create dataset
Layer 2: Model building & evaluation
Layer 3: Predictor where model is deployed to monitor the network and make predictions

At layer 1, data is collected for the five different scenarios including a network without any attack and 4 other networks with the 4 different attacks i.e. Hello Flood attack, DIS attack, Increased version and decreased rank attack. Attacks have been introduced using the codes of the simulator. A dataset with 58 features representing 5 different classes on Cooja for RPL is built using radio signal files of these simulations.

At layer 2, the predictive model is built using the dataset from layer 1. The model is built using five different supervised machine learning classifiers i.e. RFC, SVM, DTC, GNB and LRC. The model is evaluated using n-fold cross-validation as well by using the new date from a similar network. For the purpose of evaluation using new data, we have used polling for the final result prediction.

Layer 3 is a prediction layer, where the model is used to analyze the network traffic and predict the result. when tested for 5 different scenarios, accuracy score of the known attacks was above 90% for all scenarios

Model is also extended to detect a new attack. The new selective forwarding attack was introduced in the network. When tested for a new attack, an accuracy score of 85% was achieved.

The IDS developed is planned to be deployed at the Border Router, from where the entire network may be monitored. The BR is suitable as to support overall processing, this node may have more resources in terms of battery backup and processing capabilities.

Table 7.1 presents the comparison of this new IDS with other IDS's reviewed in the literature.

Table 7.1: Updated Summary of IDS for RPL

Name	Attacks Detected	No of simulations	Validation method	New attack detection	Dataset Used
SVELTE	Sinkhole and selective forwarding	unknown	Cooja Network for only current setup	No	No dataset
SVELTE-e	Rank based attacks	4,6,or 9 nodes setup	Cooja Network for only current setup	No	no dataset
INIT	Sinkhole	35 simulations	Cooja Network for only current setup	No	no dataset
InDRes	Sinkhole	NS-2 simulation tool	Comparison with INTI and SVELTE	No	Sim. only
Pongle's IDS	Wormhole	Unknown	Cooja with only k-fold cross validation	No	no dataset
Mayzaud	Version Number	Several	Cooja Network for only current setup	No	KDD
CHA-IDS	Sinkhole and selective forwarding	Unknown	Cooja with only k-fold cross validation	No	no dataset
Anomaly-based IDS	Neighbor and DIS attack	Unknown	Cooja Network for only current setup	No	no dataset
Real-time IDS	Wormhole attack	8, 16 and 24 nodes	Cooja Network for only current setup	No	no dataset
Real-Time IDS	Topology attacks	8-12 nodes setup	Cooja Network for only current setup	No	no dataset
Rank attack IDS	Rank attack	30 nodes setup	Cooja Network for only current setup	No	no dataset
ELNIDS	Several attacks	Unknown	Hold-out validation method	No	RPL-NIDDS
New IDS	4+	10 node setup with 300 simulations	n-Fold and new data testing	YES	new RPL dataset

7.2 Future work

We plan to extend our work into three different directions.

7.2.1 Extending the security assessment

Our proposed model for security assessment works based on the deployed counter measures at every level. Based on the efficiency of the countermeasures installed in the network, the scheme can further be extended to have more categories at every level. Defined as A+, A++ and A+++ etc., it can define the extend of security deployed at every level by considering the standards of the deployed countermeasures.

We have tested the application of the model using specific case studies. We need to extend our testing to include more scenarios using these new categories and working on the proven efficiencies of the available counter measures available in the industry. This will also elaborate the applicability of the security scheme using the available solutions and their abilities in different scenarios.

7.2.2 Implementing it on the actual network

At present the whole work has been tested and approved on the simulated environment. Since it is a controlled environment, the prediction accuracies are quite high. Our immediate need is to deploy it on the actual physical network, so the next step is to build a network on same number of nodes and collect the data to generate real dataset and use that to train the model. Once the model will be deployed in the physical environment, we are very sure that the results may not be similarly accurate as the IDS may have an issue of under-fitting i.e. having some variance in the features that were constant in the controlled environment.

The deployment plan of the IDS is on the border router, from where the entire network may be monitored. It is practically not possible to deploy the IDS as a distributed system on several sensor nodes, as it will lead to node battery exhaustion because of extra processing and memory needs. Once the IDS is deployed at the Border Router, it is even possible to support the BR with additional battery backup which is not possible for all other sensor nodes.

7.2.3 Extending the predictive model

Right now our model predicts no attacks and four known attacks. The model is also capable of detecting a new attack, which is an unclassified attack yet. In real deployment scenarios, every time a new attack is detected, it may be same as before or a new one, which the network has never seen.

Our plan is to keep the dataset of the new attack as an archive and study the patterns in dataset to classify them into different classes, if required. The new class is named and added in the dataset back, so that the IDS model is extended to detect more attacks. Although, this will have some manual processing, but still model will be extended. It is also possible to automate this process using other machine learning technique. Then, we also plan to analyze the profile similarities between the unknown attacks and previously trained attacks. Future work also includes retraining the IDS model using deep learning and reinforcement techniques.

Appendix A

List of publications

1. M. Sharma, F. Gebali, H. Elmiligi, and M. Rahman, "Network security evaluation scheme for wsn in cyber-physical systems," in2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Nov 2018, pp. 1145–1151 [13]

2. M. Sharma, F. Gebali, and H. Elmiligi, "3-dimensional analysis of cyber-physicalsystems attacks," in2018 4th International Conference on Computing Communication and Automation (ICCCA), Dec 2018, pp. 1–5. [12]

3. M. Sharma, F. Gebali, H. Elmiligi, A. Sharma, "Simulating Attacks for RPL and Generating Multi-class Dataset for Supervised Machine Learning", IEMCON 2019

4. M. Sharma, H. Elmiligi, F. Gebali, "Network Security and Privacy Evaluation Scheme for Cyber Physical Systems (CPS)" in *"Security of Cyber-Physical System: Vulnerability and Impact"*, springer (Accepted)

5. M. Sharma, H. Elmiligi, F. Gebali, "A Novel Intrusion Detection System for detecting RPL attacks in Cyber Physical Systems", IEEE Access(Final submission after minor edits)

Appendix B

A few popular CPS attacks

Stuxnet, a cyber work stuck the Iranian nuclear facility at Natanz and caused substantial damage to nuclear centrifuges. The worm targeted each of the three layers of a cyber-physical system. It used the cyber layer to distribute malware and identify its targets, then control layer (in this case, specific models of programmable logic controllers (PLCs) manufactured by Siemens) to control physical processes, and finally the physical layer, causing physical damage. It infected over 200,000 computers and caused 1,000 machines to physically degrade. The effect spread through all across Iran and other countries including India, Indonesia, China, Azerbaijan, South Korea, Malaysia, the United States, the United Kingdom, Australia, Finland and Germany. Stuxnet malware is comprised of three modules, a worm that executes all code related to the main payload of the attack, a link file to propagate the copies of the worm, and a rootkit component that may hide all malicious files and processes, so that the attack cannot be detected.

Ukraine SCADA Attack took place in the Ivano-Frankivsk region of Western Ukraine, in December 2015, where SCADA network controllinf the power grid got hijacked by attackers who gained access to crucial systems. The attackers gained access to systems controlling the breakers. They sabotaged the system by remotely controlling the breakers leading to a massive power outage. This incident affected about 230,000 people and was regarded as the first high severity cyber-attack that caused power outage [19]. As studied in details, the skilled attackers planned their careful strategies over many months. In this period, first they studied the networks and found operator‘s credentials, and then launched synchronized assault in well-structured manner. They used social engineering and phishing in email to gain access.

It is claimed that the lack of two factor authentication and insecure remote log-in of the SCADA network controlling the grid allowed attackers to hijack credentials and gain access to crucial systems.

Mirai Attack was discovered in August 2016 by a white-hat security research group, MalwareMustDie! [75], Mirai is not as popular as its counterparts.The malware is created using ELF binaries and it targets SSH or Telnet network protocols allowing it to exploit default and hardcoded credentials[19]. It used brute-force IoT devices utilizing factory default usernames and passwords, and logged into them to infect these IoT devices with the malware. The malware hijacked nearly half a million internet connected devices, and resulted in inaccessibility of several high-profile websites such as GitHub, Twitter, Reddit, Netflix, Airbnb and many others. The scale of the attack was unprecedented, and the exploitation of IoT devices to launch this DDoS attack may lead to more cyber-attacks in an even larger scale in the future. By the end of November 2016, approximately 900,000 routers were infected and crashed due to failed TR-064 exploitation attempts by a variant of Mirai, which resulted in Internet connectivity problems for the users of these devices. The attack has almost reached 600Gbit/s.

Maroochy Water Service Attack is another SCADA system breach that occured on Maroochy Water Services on Queensland's Sunshine Coast in Australia. It was discovered in March 2000, when Maroochy Shire Council experienced problems with its new wastewater system [138]. The Communications been sent to waste-water pumping stations by radio links were being lost, and as a result, pumps started malfunctioning. This case is also an important case that has been cited around the world as an example of the damage that could occur if SCADA systems are not secured. This attack lead to severe disruptions of the plant, including disruption of proper pump operation, suppression of alarms, and even releasing of untreated sewage into local waterways. The attack caused 800,000 liters of raw sewage to spill out into local parks, rivers and even the grounds of a Hyatt Regency hotel. The marine life died, the creek water turned black and the stench became unbearable for residents [21].

WannaCry is a new attack on Microsoft windows operating systems for the ransom payments in bitcoin cryptocurrency. The attack worked by encrypting the data from the user machine and then demanding huge ransom payments in the Bitcoin

cryptocurrency to release the data. Propagated through EternalBlue, this attack was mainly on older windows systems. Although, Microsoft released patches to close the exploit, the WannaCry's was able to spread, as it was mainly working on older Windows systems, which could not upate systems to install the patch.

VPNFilter in an example of interception of a SCADA system attack. It emerged in 2018 as a very strong IoT threat. This attack has an ability to survive a system reboot (i.e. it stays even after the system setting are rest), hence it is very difficult to remove. This threat had a destructive capability to break or wipe a device as instructed by the attacker using the commands. A well skilled and well-resourced threat on the IoT devices.

Bibliography

[1] R. Atat, L. Liu, J. Ashdown, M. J. Medley, J. D. Matyjas, and Y. Yi, "A physical layer security scheme for mobile health cyber-physical systems," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 295–309, Feb 2018.

[2] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained iot devices," *Mechanical Systems and Signal Processing*, p. 106436, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327019306570

[3] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: Analysis, challenges and solutions," *Computers & Security*, vol. 68, no. Supplement C, pp. 81 – 97, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817300809

[4] M. Barrère, C. Hankin, A. Barboni, G. Zizzo, F. Boem, S. Maffeis, and T. Parisini, "CPS-MT: A real-time cyber-physical system monitoring tool for security research," in *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Aug 2018, pp. 240–241.

[5] Z. Zhang, J. Porter, E. Eyisi, G. Karsai, X. Koutsoukos, and J. Sztipanovits, "Co-simulation framework for design of time-triggered cyber physical systems," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, ser. ICCPS '13. New York, NY, USA: ACM, 2013, pp. 119–128. [Online]. Available: http://doi.acm.org.ezproxy.library.uvic.ca/10.1145/2502524.2502541

[6] W. Di, C. Ling, G. Lili, W. Yimei, and J. Xuesong, "Application research of zigbee protocol in CPS based on multi-agent," in *2017 29th Chinese Control And Decision Conference (CCDC)*, May 2017, pp. 6843–6846.

[7] J. Jamaludin and J. M. Rohani, "Cyber-physical system (cps): State of the art," in *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, Nov 2018, pp. 1–5.

[8] M. R. Asghar, Q. Hu, and S. Zeadally, "Cybersecurity in industrial control systems: Issues, technologies, and challenges," *Computer Networks*, vol. 165, p. 106946, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128619306292

[9] A. Burg, A. Chattopadhyay, and K. Lam, "Wireless communication and security issues for cyber physical systems and the internet of things," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 38–60, Jan 2018.

[10] A. Bhattacharjya, X. Zhong, J. Wang, and X. Li, *Security Challenges and Concerns of Internet of Things (IoT)*. Cham: Springer International Publishing, 2019, pp. 153–185. [Online]. Available: https://doi.org/10.1007/978-3-319-92564-6_7

[11] C. Pu and X. Zhou, "Suppression attack against multicast protocol in low power and lossy networks: Analysis and defenses," *Sensors (Basel, Switzerland)*, vol. 18(10):3236, 2018.

[12] M. Sharma, F. Gebali, and H. Elmiligi, "3-dimensional analysis of cyber-physical systems attacks," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec 2018, pp. 1–5.

[13] M. Sharma, F. Gebali, H. Elmiligi, and M. Rahman, "Network security evaluation scheme for wsn in cyber-physical systems," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov 2018, pp. 1145–1151.

[14] M. Sharma, H. Elmiligi, F. Gebali, and A. Verma, "Simulating attacks for rpl and generating multi-class dataset for supervised machine learning," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct 2019, pp. 0020–0026.

[15] J. Chen and X. Mao, "Bodhi: Detecting buffer overflows with a game," in *2012 IEEE Sixth International Conference on Software Security and Reliability Companion*, June 2012, pp. 168–173.

[16] E. Chien, L. O'Murchu, and N. Falliere, "Duqu. the precursor to the next stuxnet," in *5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2012.

[17] B. Schneier, "To kill a centrifuge," *The Langner Group*, 2013.

[18] M. Asplund and S. Nadjm-Tehrani, "Attitudes and perceptions of IoT security in critical societal services," *IEEE Access*, vol. 4, pp. 2130–2138, 2016.

[19] M. B. Farrell and J. Detsch, "Hard lessons for Energy Dept., power sector after Ukraine hack," Jun. 2016. [Online]. Available: https://link.gale.com/apps/doc/A452362153/CPI?u=uvictoria&sid=CPI&xid=02ff6650

[20] Botnets, "Mirai," *NJ Cybersecurity & Communications Integration Cell(NJCCIC)*, 2016. [Online]. Available: https://www.cyber.nj.gov/threat-profiles/botnet-variants/mirai-botnet

[21] M. D. Abrams, "Malicious control system cyber security attack case study: Maroochy water services, australia," *Technical Papers*, August 2008. [Online]. Available: https://www.mitre.org/publications/technical-papers/

[22] M. S. Hossain and V. Raghunathan, "Aegis: A lightweight firewall for wireless sensor networks," in *Distributed Computing in Sensor Systems*, R. Rajaraman, T. Moscibroda, A. Dunkels, and A. Scaglione, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 258–272.

[23] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Wifire: A firewall for wireless networks," in *Proceedings of the ACM SIGCOMM 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Toronto, ON, Canada, August 15-19, 2011*, 10 2011, pp. 456–457.

[24] Y. Zhou, Y. Zhang, and Y. Fang, "Access control in wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 3 – 13, 2007, security Issues in Sensor and Ad Hoc Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870506000497

[25] S. Kumaran, N. Kailasanathan, and S. Mohan, "Review of asymmetric key cryptography in wireless sensor networks," *International Journal of Engineering and Technology (IJET)*, vol. 8, pp. 859–862, 01 2016.

[26] S. Kumaran and P. Ilango, "Evolution of key management and variations of random pre key distribution in wireless sensor network: Survey," *International Journal of Applied Engineering Research*, vol. 9, pp. 11 681–11 688, 01 2014.

[27] C. C. Portal, "Common criteria for information technology security evaluation," www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R4.pdf.

[28] (2018, Aug.) Purpose of the arrangement @ONLINE. [Online]. Available: www.commoncriteriaportal.org

[29] K. M. Jefcoat, "What is common criteria certification, and why is it important?" December 2017. [Online]. Available: https://www.blancco.com/blog-what-is-common-criteria-certification-why-is-it-important/

[30] K. Caplan and J. L. Sanders, "Building an international security standard," *IT Professional*, vol. 1, no. 2, pp. 29–34, March 1999.

[31] N. I. of Standards and Technology, "Security requirements for cryptographic modules," http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf.

[32] ——, "Cybersecurity framework," https://www.nist.gov/industry-impacts/cybersecurity-framework.

[33] I. S. of Automation, "Isa99, industrial automation and control systems security," urlhttps://www.isa.org/templates/two-column.aspx?pageid=124560.

[34] I. E. Commission, "Functional safety and iec 61508," urlhttps://www.iec.ch/functionalsafety/?ref=extfooter.

[35] A. A. Cardenas, T. Roosta, and S. Sastry, "Rethinking security properties, threat models, and the design space in sensor networks: A case study in SCADA systems," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1434 – 1447, 2009, privacy and Security in Wireless Sensor and Ad Hoc Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870509000468

[36] H. Orojloo and M. A. Azgomi, "A method for modeling and evaluation of the security of cyber-physical systems," in *2014 11th International ISC Conference on Information Security and Cryptology*, Sept 2014, pp. 131–136.

[37] A. Ashok, A. Hahn, and M. Govindarasu, "Cyber-physical security of wide-area monitoring, protection and control in a smart grid environment," *Journal of Advanced Research*, vol. 5, no. 4, pp. 481 – 489, 2014, cyber Security. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2090123213001495

[38] A. Ramos and R. H. Filho, "Sensor data security level estimation scheme for wireless sensor networks," *Sensors*, vol. 15, pp. 2104 – 2137, 2015.

[39] X. Wu, J. Li, and W. Yao, "A network security evaluation model based on common criteria," in *2008 International Conference on Apperceiving Computing and Intelligence Analysis*, Dec 2008, pp. 416–420.

[40] Z. Han, X. Li, R. Feng, J. Hu, G. Xu, and Z. Feng, "A three-dimensional model for software security evaluation," in *2014 Theoretical Aspects of Software Engineering Conference*, Sept 2014, pp. 34–41.

[41] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Modeling security in cyber physical systems," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3, pp. 118 – 126, 2012.

[42] S. Moein, F. Gebali, and I. Traore, "Analysis of covert hardware attacks," *Journal of Convergence*, vol. 5, 10 2014.

[43] B. Li and L. Zhang, "Security analysis of cyber-physical system," *AIP Conference Proceedings*, vol. 1839, no. 1, p. 020178, 2017. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.4982543

[44] A. Raoof, A. Matrawy, and C. Lung, "Routing attacks and mitigation methods for rpl-based internet of things," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1582–1606, Secondquarter 2019.

[45] A. Dvir, T. Holczer, and L. Buttyan, "Vera - version number and rank authentication in RPL," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, Oct 2011, pp. 709–714.

[46] H. Perrey, M. Landsmann, O. Ugus, H. Hamburg, M. Wahlisch, and T. C. Schmidt, "Trail: Topology authentication in rpl," 2015.

[47] J. Heo, J. Kim, S. Bahk, and J. Paek, "Dodge-jam: Anti-jamming technique for low-power and lossy wireless networks," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2017, pp. 1–9.

[48] C. Pu and S. Hajjar, "Mitigating forwarding misbehaviors in rpl-based low power and lossy networks," *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, 2018.

[49] C. Pu, "Mitigating dao inconsistency attack in RPL-based low power and lossy networks," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 570–574.

[50] D. Airehrour, J. A. Gutierrez, and S. K. Ray, "Sectrust-rpl: A secure trust-aware rpl routing protocol for internet of things," *Future Generation Computer Systems*, vol. 93, pp. 860 – 876, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17306581

[51] M. Conti, P. Kaliyar, M. M. Rabbani, and S. Ranise, "Split: A secure and scalable rpl routing protocol for internet of things," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2018, pp. 1–8.

[52] D. Airehrour and S. Ray, "Securing rpl routing protocol from blackhole attacks using a trust-based mechanism," in *IEEE Global Communications Conference (GLOBECOM)*, 12 2016.

[53] A. M. G. Glissa, A. Rachedi, "A secure routing protocol based on rpl for internet of things," in *26th International Telecommunication Networks and Applications Conference (ITNAC)*, 12 2016.

[54] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the internet of things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661 – 2674, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870513001005

[55] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the rpl-connected 6lowpan networks," in *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*, ser. IoTPTS '17. New York, NY, USA: ACM, 2017, pp. 31–38. [Online]. Available: http://doi.acm.org.ezproxy.library.uvic.ca/10.1145/3055245.3055252

[56] M. N. Napiah, M. Y. I. Bin Idris, R. Ramli, and I. Ahmedy, "Compression header analyzer intrusion detection system (CHA - IDS) for 6lowpan communication protocol," *IEEE Access*, vol. 6, pp. 16 623–16 638, 2018.

[57] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in internet of things," *International Journal of Computer Applications*, vol. 121, no. 9, 2015.

[58] B. Farzaneh, M. Montazeri, and S. Jamali, "An anomaly-based IDS for detecting attacks in RPL-based internet of things," in *2019 5th International Conference on Web Research (ICWR)*, 04 2019.

[59] C. Cervantes, D. Poplade, M. Nogueira, and 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)*, May 2015, pp. 606–611.

[60] A. U. M. Surendar, "Indres: An intrusion detection and response system for internet of things with 6lowpan," in *IEEE International Conference Wireless Communication Signal Process. Netw. (WiSPNET)*, 12 2016.

[61] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in RPL-based networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, June 2017.

[62] S. Deshmukh-Bhosale and S. S. Sonavane, "A real-time intrusion detection system for wormhole attack in the RPL based internet of things," *Procedia Manufacturing*, vol. 32, pp. 840 – 847, 2019, 12th International Conference Interdisciplinarity in Engineering, INTER-ENG 2018, 4–5 October 2018, Tirgu Mures, Romania. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978919303282

[63] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," *MDPI*, 2016.

[64] V. Neerugatti and A. R. Mohan Reddy, "Machine learning based technique for detection of rank attack in rpl based internet of things networks," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, July 2019.

[65] E. Kfoury, J. Saab, P. Younes, and R. Achkar, "A self organizing map intrusion detection system for rpl protocol attacks," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 11, 2019.

[66] A. Verma and V. Ranga, "Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, April 2019, pp. 1–6.

[67] M. Ring, S. Wunderlich, D. Scheuring, D. Land es, and A. Hotho, "A Survey of Network-based Intrusion Detection Data Sets," *arXiv e-prints*, p. arXiv:1903.02460, Mar 2019.

[68] Kunal and M. Dua, "Machine learning approach to IDS: A comprehensive review," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, June 2019, pp. 117–121.

[69] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes - class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842 – 851, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915020190

[70] Y. EL Mourabit, A. Toumanari, A. Bouirden, H. Zougagh, and R. Latif, "Intrusion detection system in wireless sensor network based on mobile agent," in *2014 Second World Conference on Complex Systems (WCCS)*, Nov 2014, pp. 248–251.

[71] T. Mehmood and H. B. M. Rais, "Machine learning algorithms in context of intrusion detection," in *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, Aug 2016, pp. 369–373.

[72] K. Anusha and E. Sathiyamoorthy, "Comparative study for feature selection algorithms in intrusion detection system," *Automatic Control and Computer Sciences*, vol. 50, no. 1, pp. 1–9, Jan 2016. [Online]. Available: https://doi.org/10.3103/S0146411616010028

[73] M. Almiani, A. A. Ghazleh, A. Al-Rahayfeh, and A. Razaque, "Intelligent intrusion detection system using clustered self organized map," in *2018 Fifth International Conference on Software Defined Systems (SDS)*, April 2018, pp. 138–144.

[74] A. Verma and V. Ranga, "Evaluation of network intrusion detection systems for rpl based 6lowpan networks in iot," *Wireless Personal Communications*, vol. 108, pp. 1571–1594, 2019.

[75] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on mqtt-iot," *Protocol Complexity*, vol. 2019, 2019.

[76] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the rpl protocol: A focus on the objective function," *Ad Hoc Networks*, vol. 96, p. 102001, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870519300319

[77] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," RFC 6550, Mar. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6550.txt

[78] T. Zhang and X. Li, "Evaluating and analyzing the performance of RPL in contiki," in *Proceedings of the First International Workshop on Mobile Sensing, Computing and Communication*, ser. MSCC '14. New York, NY, USA: ACM, 2014, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2633675.2633678

[79] J. Kaur, "A ultimate approach of mitigating attacks in rpl based low power lossy networks," 2019.

[80] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL)," in *2011 IEEE 7th*

International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct 2011, pp. 365–372.

[81] J. Phukan, K. F. Li, and F. Gebali, "Hardware covert attacks and countermeasures," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, March 2016, pp. 1051–1054.

[82] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of things (IoT): Taxonomy of security attacks," in *2016 3rd International Conference on Electronic Design (ICED)*, Aug 2016, pp. 321–326.

[83] S. Ali, S. B. Qaisar, H. Saeed, M. F. K. M. Naeem, and A. Anpalagan, "Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring," *sensors*, 2015.

[84] P. Pongle and G. Chavan, "A survey : Attacks on rpl and 6lowpan in iot," in *International Conference on Pervasive Computing (ICPC)*, 2015.

[85] B. A. Alabsi, M. Anbar1, and S. anickam, "A comprehensive review on security attacks in dynamic wireless sensor networks based on rpl protocol," *International Journal of Pure and Applied Mathematics*, vol. 118, pp. 653–667, 2018.

[86] A. Kamble, V. S. Malemath, and D. Patil, "Security attacks and secure routing protocols in rpl-based internet of things: Survey," in *2017 International Conference on Emerging Trends Innovation in ICT (ICEI)*, Feb 2017, pp. 33–39.

[87] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in rpl-based internet of things," *International Journal of Network Security, IJNS*, pp. 459–473, 2016.

[88] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3685–3692, Oct 2013.

[89] A. Mathew and J. S. Terence, "A survey on various detection techniques of sinkhole attacks in wsn," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, April 2017, pp. 1115–1119.

[90] S. Ali, M. A. Khan, J. Ahmad, A. W. Malik, and A. ur Rehman, "Detection and prevention of black hole attacks in IoT WSN," in *2018 Third International*

Conference on Fog and Mobile Edge Computing (FMEC), April 2018, pp. 217–226.

[91] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, June 2018, pp. 12–17.

[92] C. Pu, "Spam dis attack against routing protocol in the internet of things," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2019, pp. 73–77.

[93] N. K. Mittal, "A survey on wireless sensor network for community intrusion detection systems," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, March 2016, pp. 107–111.

[94] I. Tomić and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, Dec 2017.

[95] H. Karimipour, S. Geris, A. Dehghantanha, and H. Leung, "Intelligent anomaly detection for large-scale smart grids," *IEEE CCECE, Edmonton*, May 2019.

[96] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16 – 24, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804512001944

[97] S. Duhan and P. Khandnor, "Intrusion detection system in wireless sensor networks: A comprehensive review," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, March 2016, pp. 2707–2713.

[98] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: A review," *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, p. 167575, 2013. [Online]. Available: https://doi.org/10.1155/2013/167575

[99] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, April 2019.

[100] H. Karimipour, A. Dehghantanha, R. Parizi, R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, May 2019.

[101] S. Geris and H. Karimipour, "A feature selection-based approach for joint cyber-attack detection and state estimation," *IEEE Int. Conf. on Smart Energy Grid Engineering (SEGE), Oshawa*, August 2019.

[102] M. R. Begli, F. Derakhshan, and H. Karimipour, "A layered intrusion detection system for critical infrastructure using machine learning," *IEEE Int. Conf. on Smart Energy Grid Engineering (SEGE), Oshawa*, August 2019.

[103] J. Sakhnini, A. Dehghantanha, and H. Karimipour, "Smart grid cyber attacks detection using supervised learning and heuristic feature selection," *IEEE Int. Conf. on Smart Energy Grid Engineering (SEGE)*, August 2019.

[104] C. E. Kahn, P. Porras, S. Staniford-Chen, and B. Tung, "A common intrusion detection framework," in *semanticscholar.org*, 2000.

[105] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, July 1959.

[106] A. Dey, "Machine learning algorithms: A review," *International Journal of Computer Science and Information Technologies*, vol. 7, 2016.

[107] K. Das1 and R. N. Behera, "A survey on machine learning: Concept, algorithms and applications," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, 2017.

[108] B. S. Sasikala, V. G. Biju, and C. M. Prashanth, "Kappa and accuracy evaluations of machine learning classifiers," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, May 2017, pp. 20–23.

[109] A. Hassan, S. Tahir, and A. I. Baig, "Unsupervised machine learning for malicious network activities," in *2019 International Conference on Applied and Engineering Mathematics (ICAEM)*, Aug 2019, pp. 151–156.

[110] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117 – 123, 2016.

[111] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *advances in BioInformatics*, 2015.

[112] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," *Proceedings of 2014 Science and Information Conference, SAI 2014*, pp. 372–378, 10 2014.

[113] C. Burges and C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, January 1998. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-tutorial-on-support-vector-machines-for-pattern-recognition/

[114] R. K. Sharma, H. K. Kalita, and B. Issac, "Are machine learning based intrusion detection system always secure? an insight into tampered learning." *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 3, pp. 3635 – 3651, 2018. [Online]. Available: http://ezproxy.library.uvic.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=132098629&site=ehost-live&scope=site

[115] D. Nikolov, I. Kordev, and S. Stefanova, "Concept for network intrusion detection system based on recurrent neural network classifier," in *2018 IEEE XXVII International Scientific Conference Electronics - ET*, Sep. 2018, pp. 1–4.

[116] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks," in *Security in Pervasive Computing*, J. A. Clark, R. F. Paige, F. A. C. Polack, and P. J. Brooke, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 104–118.

[117] Y. Ping, J. Xinghao, W. Yue, and L. Ning, "Distributed intrusion detection for mobile ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 19, no. 4, pp. 851–859, Aug 2008.

[118] M. F. Othman and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204 – 1210, 2012, international Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705812027026

[119] J. Wang, , H. A. an Shu Lee, and F. Xia, "Secured health care application architecture for cyber-physical systems," *ArXiv e-prints*, Dec. 2012.

[120] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber physical systems," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 3, pp. 401–416, May 2016.

[121] S. Pathania and N. Bilandi, "Security issues in wireless body area network," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 4, 2014.

[122] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, and S. Shamshirband, "Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications," *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 113 – 122, 2017.

[123] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "A sybil attack detection scheme for a forest wildfire monitoring application," *Future Generation Computer Systems*, vol. 80, pp. 613 – 626, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16301522

[124] V. Sivaraman, D. Chan, D. Earl, and R. Boreli, "Smart-phones attacking smart-homes," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '16. New York, NY, USA: ACM, 2016, pp. 195–200.

[125] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security - A survey," *CoRR*, vol. abs/1701.04525, 2017. [Online]. Available: http://arxiv.org/abs/1701.04525

[126] D. Sharma, I. Mishra, and S. Jain, "A detailed classification of routing attacks against RPL in internet of things," *International Journal of Advanced Research, Ideas and Innovation in Technology*, vol. 3, no. 1, pp. 692 – 703, 2017.

[127] A. Verma and V. Ranga, "Analysis of routing attacks on RPL based 6LoWPAN networks," *International Journal of Grid and Distributed Computing*, vol. 11, pp. 43–56, 08 2018.

[128] A. Aris, S. Oktug, and B. Ors, "RPL version number attacks: In-depth study," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 04 2016, pp. 776–779.

[129] R. Nisbet, G. Miner, and K. Yale, "Chapter 18 - a data preparation cookbook," in *Handbook of Statistical Analysis and Data Mining Applications (Second Edition)*, R. Nisbet, G. Miner, and K. Yale, Eds. Boston: Academic Press, 2018, pp. 727 – 740. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124166325000189

[130] S. M. Kasongo and Y. Sun, "A deep learning method with filter based feature engineering for wireless intrusion detection system," *IEEE Access*, vol. 7, pp. 38 597–38 607, 2019.

[131] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[132] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Medical Informatics and Decision Making*, 2019.

[133] Y. Bouzida, "Neural networks vs decision trees for intrusion detection," in *ResearchGate*, 2006.

[134] S. Mukherjee and N. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119 – 128, 2012, 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017312002964

[135] M. Panda and M. Patra, "Network intrusion detection using naive bayes," *IJCSNS International Journal of Computer Science and Network Security*, vol. 7, 12 2007.

[136] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33 789–33 795, 2018.

[137] H. ali, "A performance evaluation of RPL in contiki," Ph.D. dissertation, Blekinge Institute of Technology, Sweden, 2012.

[138] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," *International Federation for Information Processing Digital Library; Critical Infrastructure Protection;*, vol. 253, pp. 73–82, 03 2007.