A robust self-healing and intrusion detection model in software-defined wireless sensor networks

RK Thupae

orcid.org/0000-0001-5708-9745

Dissertation accepted in fulfilment of the requirements for the degree Master of Science in Computer Science at the North-West University

Supervisor: Prof N Gasela

Co-Supervisor: Prof B Isong

Co-Supervisor: Prof A Abu-Mahfouz (CSIR)

Graduation: July 2023

Student number: 20115563

Declaration

I, **Ratanang Kenneth Thupae** declare that this project "**A robust self-healing and intrusion detection model in software-defined wireless sensor networks**" is my work carried out at the North-West University, Mahikeng Campus and has not been submitted in any form for the award of a degree to any other university or institution of tertiary education or published earlier. All the materials used as sources of information have been duly acknowledged in the text and the references.

Signature:

Supervisor: **Prof. Naison Gasela**

Department of Computer Science
Faculty of Natural and Agricultural Science North-West University
Mafikeng Campus

Co-Supervisor: **Prof. Bassey Isong**

Department of Computer Science
Faculty of Natural and Agricultural Science North-West University
Mafikeng Campus

Co-Supervisor: **Prof. Adnan Abu-Mahfouz**

Council for Scientific and Industrial Research
Pretoria, South Africa

Dedication

I dedicate this research to my beloved Thupae and Kgaudi family, my father Mr T.J. Thupae, my mother Mrs M.S. Kgaudi-Makgale and other family members, for the support and sacrifices they have made for me to finish my studies. To my daughter Refemele Gladness Thupae, I hope this level of work motivates and guides you regarding your career choice.

Acknowledgements

Let me pour out all my thanks to my God, with His shadow of protection I adore the knowledge, wisdom, and blessings He has granted me. With heartful appreciation, I am grateful to my supervisor, Prof. N. Gasela, and co-supervisor Prof. Adnan Abu-Mahfouz (CSIR) for endless guidance through the journey of the research study. Finally, the passion and drive they had in Computer Science played a huge role. My sincere appreciation also goes to my co-supervisor Prof B.E. Isong who believed that with patience and support, this will be possible.

Abstract

Software-defined wireless sensor network (SDWSN) is a networking model that brings software-defined networking (SDN) benefits such as effortlessness, innovation, and flexible network management and configuration to the wireless sensors network (WSN) world. However, the network model is still faced with several challenges in terms of security and reliability. The centralized controller, which is the "brain" of the network, is always the primary target of attacks and poses a single-point failure. A security compromise on the controller can result in access to vital users' data, and network resources and may bring about the total failure of the SDWSN due to the absence of a robust self-healing ability. Though multi-controllers architecture is the rescuer, they are only cost-effective for large-scale SDN. Moreover, several solutions such as intrusion detection systems (IDS) and fault-tolerance (FT) mechanisms have been proposed and developed. However, research has shown that these solutions are disjointed in terms of implementation. This study considered the existing solutions as not cost-effective and therefore, seek for a viable solution that is both self-healing and attack-aware in the SDWSN.

A comprehensive literature review of the FT mechanisms and IDSs has been conducted to bring together the state-of-the-art SDN, WSN, SDWSN, and machine learning algorithms, to gain insight into their challenges, strengths, and weaknesses for improvements. The literature review provided insight into the performances of both the replication scheme in the aspect of FT and the flow-based anomaly detection approach in terms of IDS. This study, therefore, proposed an integrated FT and ID model known as the Fault Tolerance-Intrusion Detection Model to detect faults and intrusions in the SDWSN together. FT and IDS mechanisms utilized the controller - OpenFlow network statistics collection technique to achieve their functions: *opf_flow_stats_Request* and *opf_flow_stats_Reply*. The system architecture for each model is designed and their components or functionalities are presented and discussed. In addition, the flow-based anomaly detector is machine learning based and to identify the best algorithm for a resilient controller, empirical analysis using four Machine learning models: support vector machine (SVM), logistic regression (LR), naïve Bayes (NB) and random forest (RF) is

performed to determine classification accuracies and time efficiencies. The NSL-KDD dataset is used to train and test the model. Results of the model showed that the RF model outperformed all other models considered with an accuracy of 99% and 0.1 and 0.6 secs for training and testing time respectively, and performed well in terms of classification accuracy. The designed FaToID model was implemented in the SDWSN environment and its performance was evaluated using network latency and throughput with three controllers for FT while a DDoS dataset was used to evaluate the accuracy of the IDS.

The simulation results showed a good and improved network delay and throughput for the FT mechanism in POX and default controllers compared to floodlight controllers. Moreover, the ID model showed about 98.7 % detection accuracy, 99.9 % specificity and sensitivity, 97 % precision and recall, and 96.8 % F-measure by the RF-based IDS model. Therefore, for SDWSN to be resilient, a model that incorporates both faults and attack detection must be in place to protect the network from all malicious attacks and unexpected faults that can result in access to network-sensitive resources and even failure. Integrating the proposed FaToID Model into the SDWSN model can significantly increase the dependability and resiliency of the SDWSN.

Keywords: *WSN, SDWSN, Fault tolerance, Controllers, Self-healing, IDS, Machine learning, Security*

Table of Contents

List of Figures

List of Tables

List of Acronyms

WSN	Wireless Sensor Network
SDN	Software-defined network
SDWSN	Software defined wireless sensor network
APIs	Application programming interfaces
IoT	Internet of things
DoS	Denial of service
SLR	Systematic literature review
FT	Fault tolerance
NOS	Network operating system
IHR	Informer homed routing
DHR	Dual homed routing
IDS	Intrusion detection system
ID	Intrusion detection
DR	Detection rate
GPS	Global positioning system
RM	Research methodology
CRM	Constructive research methodology
QA	Qualitative analysis
FaToIDM	Fault tolerance intrusion detection model
FaToM	Fault tolerance model
IDM	Intrusion detection model
DMCs	Distributed multiple controllers
S2C	Sensor to controller
C2C	Controller – to-Controller
SPC	Shortest path computation

SOF	Sensor OpenFlow
DM	Detection module
FTypeDB	Fault type database
RM	Recovery module
ML	Machine learning
DDoS	Distributed denial of service
DoS	Denial of service
KM	Kilometre
M	Metre
TP	True positive
FP	False positive
TN	True negative
FN	False negative
TPR	True positive rate
FPR	False positive rate
SVM	Support vector machines
RF	Random Forest
LR	Logistic regression
PR	Precision
ROC	Receiver operating characteristic
NB	Naive Bayes
DSRM	Design science research methodology
DFCA	Distributed fault-tolerant clustering algorithm
CHs	Cluster Heads
PCA	Principal component analysis

Definition of Concepts

SDWSN: SDWSN is defined as a network computing paradigm extracted from the SDN concept to WSNs strategies to improve technological applications [1, 2].

WSN: WSN is defined as a network of two or more nodes working cooperatively to sense and control the environment surrounding all of them [3, 4].

Fault Tolerance: In WSN, FT refers to the enablement of a system which continues to be operational in terms of failure or fault occurring within system components such as sensor nodes and controllers [5, 6].

Self-Healing: Self-healing refers to the feature which allows the network to automatically self-heal after a sensor node or controller attack. It refers to the ratio of links which are compromised and decreases with time even if the sensor nodes or controllers are corrupted [7, 8].

Latency: Latency refers to the time delay between the cause and effect of some change in controllers or system being observed and input to simulation often occurs because of the network [9].

Throughput: Throughput is defined as the rate of messages delivered over the communication channel between controllers it can pass through network nodes, and it is measured in bits per second [10].

Intrusion Detection System: It refers to a system that monitors a network for malicious activity and is divided into network-based intrusion detection systems and host-based intrusion detection systems within the network [11].

Machine Learning: Machine learning (ML) refers to artificial intelligence, a technology that studies algorithms to imitate human learning. Finally, ML improves accuracy and has caught more attention among Internet of Things strategies [12].

Security: Security in WSN refers to protection from vulnerable networks or resilience mechanisms to guard against harmful attacks from intruders aiming to damage network systems [13].

Attack-aware: in WSN Is defined as a malicious attempt to disrupt normal network traffic of a targeted system or network. This occurs when a target is overwhelmed with a flood of internet traffic which can lead to serious damage to the network [14, 15].

Chapter 1

Introduction and Background

1.1 Introduction

This section presents an introduction to this research which involved self-healing and intrusion detection in the software-defined wireless sensor networks (SDWSN).

A wireless sensor network (WSN) contains several small, low-powered sensor nodes which are highly cost-effective and check environmental effects including temperature, sound, pressure, and humidity [1]. These sensor nodes send information to a host wirelessly on the network, so that it can be in a readable format [16]. However, WSN has its weakness due to poor communication infrastructure [1][14]. Moreover, it is categorized into four groups such as bi-directional networks, star networks, mesh networks and one-way networks [17]. Due to the sensitivity of information carried by nodes, they pose a problem for software engineers to design secure models for these devices. However, the nodes perform computation, sensing, actuation, and wireless communication functions [14][15]. Today, WSN has been significantly considered the future of the Internet of Things (IoT) [18]. Since WSNs play an important role in IoT, sensor nodes are regarded as important aspects of this concept [19]. Furthermore, it has been estimated that almost 70 billion devices will have sensors envisioned to be connected to the network by 2025 [20]. Thus, with the network challenges faced, the SDN paradigm evolved to advance WSN. In particular, SDN is capable to neutralize network issues and bringing innovations by offering new functions to the entire network topology, yielding a new network model known as SDWSN.

However, disjointing the control plane and data plane in the network breaks vertical integration [21]. It is also a technology that allows efficient provisioning of future networks by lowering operating costs through simplified management, hardware, and software. Moreover, regarding security, wireless network has advantages and the capability of extracting traffic flows based on network state. WSNs have several nodes that collect data, put an update on the internet and possibly different aggregations that depend on the sensing devices at the first layer. Despite advances in SDN-SDWSN deployment, based on security, research is still in its early stages and thus, needs more attention for better improvement. The identified problems include FT which is faced with a self-healing ability issue due to identified single point of failure when the centralized controller is affected. However, from the perspective of security in SDWSN, several attacks and threats have been identified as primary targets of attacks on the network. In essence, such malicious attacks are seen in the form of intrusions. In terms of FT and intrusion detection (ID), aspect types of faults can be identified in the control plane's domain and the

issue of intrusion nodes that appear to be legitimate need a defensive intrusion detection system (IDS). The issue of intrusion nodes appears to be legitimate in the network and therefore, an IDS need to be provided for a network to be defensive against all faults and intrusions. Currently, some techniques that deal with intrusion have been proposed or developed while some are yet to be implemented.

To address the identified problem and avoid existing single points of failure based on SDWSN controllers, this research seeks to implement a fault tolerance mechanism by designing single but physically distributed controllers. From the perspective of SDWSN, the aim is to protect SDWSN from catastrophic faults and intrusions. According to SDN, WSN paradigm, the application of IDS is important when protecting SDWSN against different attacks that bug the network. Therefore, this research is performed to design and implement a self-healing and ID model-based SDWSN. SDWSN is a paradigm that still needs more attention and improvements based on the latest technologies [22]. The technology is faced with several challenges inherited from SDN and WSN such as FT, network intrusions (NIs), etc, which serve as existential threats to the adoption and development of SDWSN.

1.2 Problem Statement

The OpenFlow-based SDN architecture has one centralized controller which is the powerhouse of the network and if it is compromised, the whole network may fail [23, 24]. The SDWSN is faced with a self-healing ability issue, which could lead to single-point failure when the centralized controller is affected. It poses a situation that can lead the network operating system (NOS) to a state of failure in the SDWSN [2]. From the security perspective, several attacks and threats threaten the entire network, and the controller has been identified as a primary target. Some of these attacks are in the form of NIs and if successful, can maliciously compromise the controller [25]. According to Abdulaliyev et al. [26], the security concerns originate from the fact that many attacks can be launched on the controller by compromised sensor nodes in WSN. Moreover, the issue of intrusion nodes appears to be legitimate in the network. Therefore, an IDS is needed to provide a defensive network.

Therefore, to address this challenge, some mechanisms have been proposed and developed to tackle faults and intrusions in the network model, but robust and efficient solutions are yet to be achieved. For intrusions, several studies have used machine learning (ML) techniques in detecting and identifying intrusions. The solutions that provide FT and intrusion detection (ID) are separated and disjointed, which means FT cannot detect faults emanating from security attacks and threats. Likewise, IDS cannot detect faults emanating from system components and having these solutions separately is considered not cost-effective. This research designed and implemented an integrated model that incorporates FT and ID to ensure resiliency and reliability in the network. The model combines FT and ID aspects by incorporating security into FT to eliminate faults and intrusions in the SDWSN. In terms of usage, FT will increase controller performance and the IDS will detect and stop intrusions early before they pose catastrophes to the entire network.

1.3 Research Questions

The following research questions (RQs) are extended from the objectives, and they seek to be answered so that the aim of this research can be achieved.

RQ1: How can we prevent the centralized controller from being a single point of failure due to faults and intrusions in the SDWSNs?

This Research Question was answered by comprehensively conducting a literature review based on existing or related work in SDN, WSN and SDWSN outlined in Chapter 2. Security issues and other mentioned countermeasures experienced in SDNs, WSNs and SDWSNs were outlined in the literature. Fault tolerance and intrusion detection approaches and mechanisms in specified networks were studied to help in designing an integrated FatoID model or framework for SDWSN.

RQ2: How can we design and develop an efficient model that incorporates both fault tolerance and intrusion detection or identification mechanisms for a reliable and secured SDWSN?

RQ2 has been answered in Chapters 4 and 5 where the researcher evaluates FaToM using network throughput, and latency through Wireshark network analysis. Thus, it is for self-healing capability in proposed multiple controllers (Pox, default, and floodlight). Based on the four selected ML models (NB, LR, RF and SVM), RF was chosen to design the integrated model (FaToIDM) for SDWSN. FaToIDM is designed to identify faults and detect intrusions or attacks in the SDWSN. The goal of this model is to bring the capability of self-healing and intrusion detection (ID) to avoid single points of failure in controllers and secure SDWSN.

RQ3: How can we evaluate the developed model for effective reliability and security performance?

RQ 3 was answered by performing a simulation of the proposed FatoIDM respectively in Chapter 4. To assess effectiveness, reliability and security using Wireshark in FaToM is based on analysing network packets in Mininet topology and to identify intrusions or security attacks such as DDoS, the RF model is used in IDM. Therefore, in satisfaction with all the stated research objectives, and to answer research questions, the research methodology and design in Chapter 3 are followed.

1.4 Research Aim and Objectives

This subsection focuses on research questions, research aim and objectives.

1.4.1 Research aim

This research aims to develop a robust self-healing and intrusion detection model to handle faults and intrusion attacks in SDWSN.

1.4.2 Research objectives

To achieve the research aim, the following research objectives(ROs) shall be achieved:

RO 1: Determine mechanisms or techniques which can lead to the deployment of logically centralized but physically distributed controllers to avoid single points of failure and to identify intrusion or attacks regarding traffic flows.

RO 2: Designing and developing a model with a self-healing capability and intrusion detection or identification in the SDWSN.

RO 3: Simulate and evaluate the model for effectiveness and performance.

1.5 Research Motivation

SDWSNs are innovative in the realm of the network in the sense that they can dynamically manage, control, and modify the network. This can be done in a programmable manner without the network administrator's intervention. However, given the challenges it faced, there are already proposed and developed solutions, and several works have been done and there is a high volume of them. This makes it difficult to have a starting point on what needs to be done especially on FT and ID. Therefore, this research aims to design an integrated and disjointed model including FaToM and IDM known as FaToIDM for SDWSN where security will be incorporated into FT to eliminate faults and intrusions or attacks.

1.6 Method of Investigation

This research adopts design science research methodology (DSRM) and is aimed at producing a knowledgeable solution to the identified problem. The methodology is a mixed method which incorporates qualitative and quantitative methods. Several research methods were selected for answering the research questions. Details about the methodology and the methods as well as the research design are discussed in Chapter 3.

1.7 Research Organization

This research is categorized into the following chapters:

Chapter 1 – Introduction: This chapter explains all definitions of concepts, and acronyms while highlighting the research aim, research objectives and questions. It also presents the introduction and the problem statement.

Chapter 2- Literature review: In this chapter, literature is explored, and related works from other researchers are considered, studied, and analysed thoroughly. The chapter also highlights previous work that was done in the computer science field. Finally, it provides a detailed study of the work currently done, as well as related works from other researchers as a basis for the current work.

Chapter 3 - Research methodology and design: This chapter focuses on DSRM to achieve the aim of this research and the study methods are explained in detail.

Chapter 4 - Integrated model design: This chapter presents the proposed robust self-healing and intrusion detection model, a detailed report of the tools used, and all the steps involved from the design to the evaluation. This chapter generally provides solutions to RQ 1 and RQ 2 by following RO 1 and RO 2.

Chapter 5 - Result analysis and discussion: This chapter thoroughly provides the experimental setup to run simulations, briefly analyses the results obtained and discusses and evaluates the model to answer RQ 3 by following up on RO 3.

Chapter 6 - Summary, recommendation, conclusion, and future work: This chapter outlines a summary regarding the results obtained and potential future work is included. Thereafter, the references used in this research follow.

1.8 Research Scope and Limitations

This research focuses only on designing and simulating a robust model that incorporates FT and ID. The significance is to ensure resilience and security in SDWSN. However, due to time and resource constraints, partial implementation of the model is outlined in Chapter 5.

1.9 Research Output

This research has generated the following outputs:

1. The first review paper is published and detailed by R Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Software-defined wireless sensor networks management and security challenges: A review," in IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp 4736-4741.
2. The second review paper is published and detailed by R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Machine learning techniques for traffic identification and classification in SDWSN: A survey," in IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp 4645-4650.
3. A journal article “A robust self-healing and intrusion detection model in software-defined wireless sensor networks” is currently being drafted to be submitted soon.

1.10 Chapter Summary

The chapter presented an introduction and background information on SDN, WSN and SDWSN, the problem statement to be addressed, as well as the aim and objectives relating to the research questions. It also presents the organization of this research.

Chapter 2

Literature Review

2.1 Chapter Outline

This chapter presents the literature reviewed when carrying out this research looking at the related important aspects such as FT, ID, SDN WSN and SDWSN. The goal is to bring together what researchers have done and point out gaps in their research. In this chapter, the literature review also outlines SDWSNs which is viewed as the main point that differentiates WSNs from traditional SDNs, while paving the way to creating a solution to the research problem. As outlined in Chapter 1, a single point of failure in centralized controllers is concerning and this chapter highlights the identified causes and their solutions. This is to close the identified gap observed in our problem statement. Furthermore, the literature review helps in evaluating and interpreting relevant research which leads to answering our research questions. In essence, security is the core concern of why sensor networks need to be improved for better performance and functionality.

2.2 Introduction

Da Costa et al. [27] describe WSN as a network constituted of small mobile devices with the functionality of sensors. The IDS-based WSN vary from traditional networks, mainly in three aspects such as no fixed network infrastructure; how it communicates and available resources such as CPU, memory, etc. A WSN normally comprises sensor nodes deployed in a specific area of the network and organised in a manner where it can be vulnerable to faults identified. Since SDWSN is still in its infancy or early developmental stage, sensor nodes often suffer from various attacks. In addition, these sensor nodes are usually deployed in severe environments which are some factors that can cause failure. If data is monitored, a reduction in accuracy can be experienced. Thus, fault detection in controllers using sensor nodes is very important to ensure the accuracy of monitoring results. From the perspective of traditional networks, SDN became the backbone paradigm for networks and the implementation of a robust FT mechanism for SDWSN. In essence, with the existing paradigm, SDN data plane and FT techniques were discovered to be reactive and proactive aspects which may or may not rely on the controller. Furthermore, ID is discovered to be a way of monitoring what is happening within a network to detect abnormal traffic and what appears to be malicious attacks that can breach security policies.

Therefore, we highlighted important facts including FT and ID aspects whereby more helpful mechanisms or techniques can be designed to avoid unauthorized access through information, and data manipulation.

2.3. Overview of WSN

A wireless sensor network (WSN) is regarded as a special wireless network that consists of several densely deployed sensor nodes. According to Narayanaraju et al. [28], sensor networks are referred to as heterogeneous systems with a combination of tiny sensors, actuators and their important computing elements. WSN is divided into three important categories namely: passive (one-directional sensor), passive (narrow-beam sensor) and lastly active sensor. In particular, each node may consist of numerous sensors which incorporate wireless transceivers to allow communication and networking. However, in the WSN realm, FT is one of the aspects that need the researcher's attention. Additionally, based on the structural view of WSN shown in Figure 2.1, from a logical point of view, sensor nodes can be contacted via services of middleware layers.

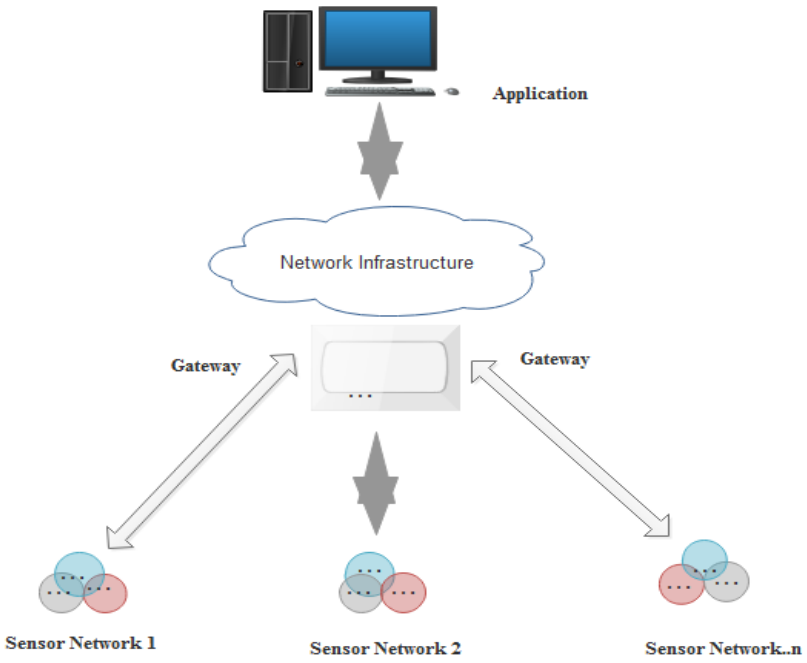

Figure 2.1. Schematic of WSN [29]

According to Modieginyane et al. [30], WSNs are technologies that are deployed on factors such as security applications, and material sensing. However, these sensor networks work separately on group networking and compute sensors. The authors further reported that depending on the design principle, WSN consists of a WSN server, sensor nodes, switches, routers etc. They also advised on security since it is one of the most significant aspects when planning to optimize network functionalities. Furthermore, in WSN, collected data was processed and routed to the nearest gateway node and it comprises a huge number of densely deployed sensor nodes. Figure 2.2 shows the overall WSN architecture where each node

comprises one or more sensors, low-power radio with a power supply, and possibly a global positioning system (GPS).

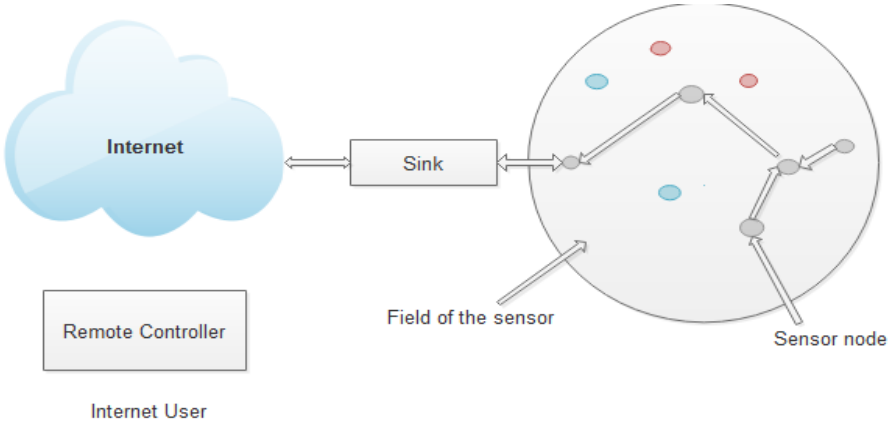

Figure 2.2. Schematic of WSN architecture [31]

This subsection extensively presents WSN security issues from the literature and the subsequently proposed solutions.

Table 2.1: WSN security issues

Security issue	**Description**
DoS	General attack applies to the data link, network, and transport layer where an attacker can put fake broadcast packets. Kocher et al. [32], reported that at the on-link layer, an intruder may internationally violate the communication by transmitting messages or attempting to generate collisions of ZigBee or IEEE 802.11b protocol. Secondly, at the routing layer, a node may attempt and take advantage of a multi-hop network, and this may occur by simply refusing to transport messages. Finally, the transport layer has vulnerability to such attacks due to flooding. However, flooding is when an attacker sends many connection requests to a malicious node.
Traffic analysis attacks	These attacks occur when an intruder negatively affects the network where they can easily disable the base station. However, Zhu et al. [33], demonstrated two attacks that can classify the base station in a network through the contents of packets.
Physical attacks	Anwar et al. [29], stated that WSN consists of layers which protect the sensor from different attacks. These types of attacks may destroy the sensor node permanently and packets can be lost. In contrast, attackers can easily have access to cryptographic secrets and tamper with spoofing in sensor nodes. However, attackers may be able to replace them with malicious nodes.

Sybil attack	Chelli et al. [34], defined these attacks as malicious nodes that illegitimately take on multiple identifiers. Furthermore, it defeats the redundancy techniques of distributed data storage systems.
Node replication attacks	Kocher et al. [32], stated that based on node replication, an intruder can duplicate a sensor node in a sensor network and this can be done by copying an existing node ID. Moreover, replicated nodes can disrupt the performance of a sensor network. Therefore, in this situation packets can either be misrouted or corrupted.
Secure localization	Anwar et al. [29], also reported that secure localization is regarded as a significant factor during the implementation of security in the network and this can prevent the attacker from searching the header of packets and data. However, few attacks relate to sensor location and WSN uses location-based information to identify the position of these nodes.

Security issues: Mohan et al. [35], stated that security is an established field for computing where its mechanisms address authentication, intrusion detection and other computing services to provide a secure transaction. Though different challenges in WSN have been discovered, more focus has been put on various security issues. Anwar et al. [29], described security as one of the main characteristics of traditional WSNs experiencing different types of attacks. However, these attacks are still present and WSNs are very prone to them. Furthermore, Giruka et al. [36], posited that the attacks can be handled by different security architectures and services such as confidentiality, integrity and authentication. Some of the security issues are presented in Table 2.1.

Proposed solutions: Kocher et al. [32], stated some of the countermeasures for the attacks discussed in Table 2.1 as the following: key management, link-layer encryption, adaptive antennas and spread spectrum. However, mitigation or solutions to DoS or DDoS attacks would be effective with an analysis of flow statistics found in the switches as well as the flow behaviour [32].

2.4 SDN in WSN

SDN is a network paradigm or model decoupling control and data plane making the control plane programmable by utilizing different application programming interfaces (APIs) shown in Figure 2.3. The decoupling of control and data planes can be done by a defined programming interface between switches and controllers [21, 37]. SDN is regarded as a persistent approach and a promising solution whose key idea is to allow flexible, efficient network operation and management through software programs. Furthermore, network efficiency is improved by SDN through the utilization of high-level novel abstractions. In particular, the application of SDN to WSN has put some effort to allow versatility and flexibility in the SDWSN realm [38]. SDNs are however faced with challenges such as single point of failure, communication between switches and controllers, and most importantly security. In the SDN paradigm, most of the FT and ID functions become significant if removed from physical nodes to a logically centralized

controller. SDNs have the capability of handling network traffic flows in terms of packets so that switches and routers can receive flow policies. These devices store these flow policies in a flow table [39].

SDN has potential benefits namely: configuration enhancement, improved performance, and encouraging innovation in the network model. However, despite its benefits, a control plane is implemented on a controller including devices in the network. Hassan et al. [16], stated that throughout the network, traffic flow can be manipulated by the controller. In this situation, if the sensor node is about to fail, a signal will be sent to the controller for changes to the routing table in no time. Furthermore, after taking over the control plane functionalities such as traffic management; quality of service (QoS) can be achieved [40]. In WSNs, the monitoring capacity of SDN has the capability of providing a better view of network status. This is due to an entity that is in control of detecting intrusions or abnormal behaviour on the network [41]. Moreover, vendor independence, heterogeneous network management, reliability and security are introduced by SDN which were still impossible in the traditional network, and it is beneficial [21]. The extension of SDN to WSN has become beneficial to SDWSN due to cost-effective infrastructure upgrades and, the delivery of new services and improvements of user experience to existing infrastructure.

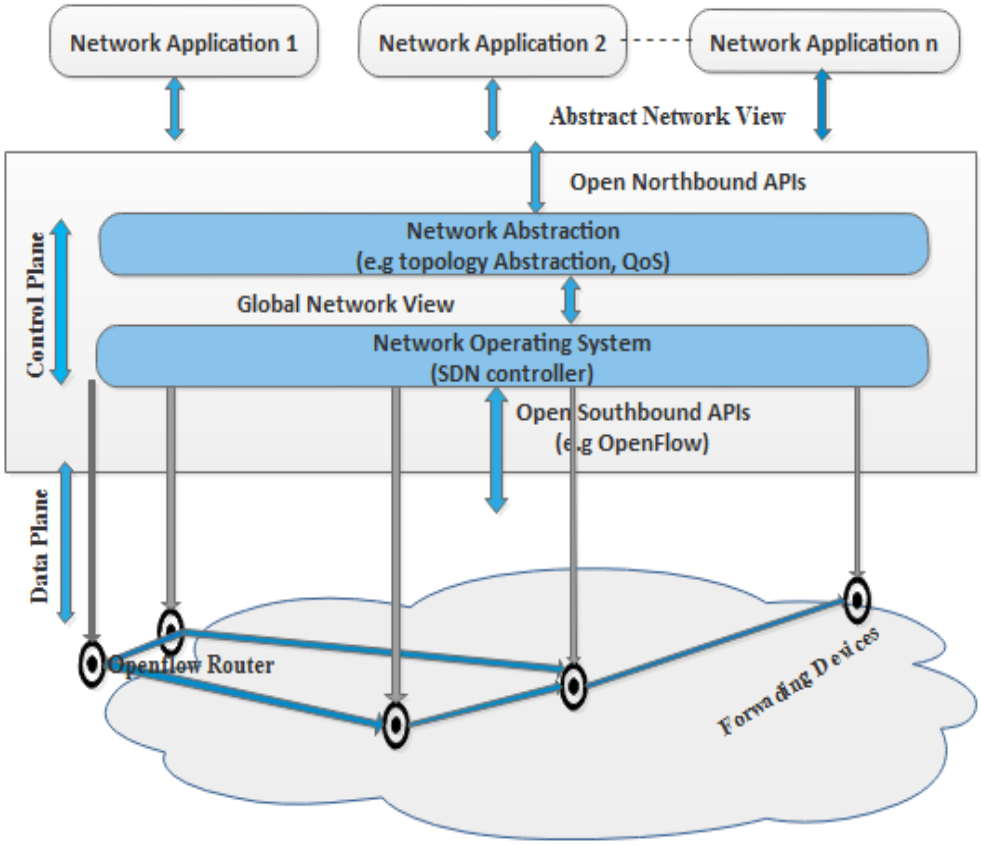

Figure 2.3. Schematic of a simplified view of the SDN architecture [21]

2.4.1 SDN security issues

This subsection extensively presents SDN security issues, and the solutions proposed.

Table 2.2: SDN-WSN security issues

Security issue	Description
SDN Controller	Another security threat within the SDN is the controller. The controller is prone to many vulnerabilities as it is a single point, and its failure would mean total failure to the entire SDN network. Threats that are targeting the controller have the ultimate win over the entire network. Furthermore, security bridges may result because the controller itself is more programmable than traditional networks and thus opens room for security threats. In essence, Raza et al. [42] and Akhunzada et al. [43], reported calls for SDN designers to build them in such a way that security issues are addressed or anticipated. Application-based threats are caused by applications that may be implemented at the top of the control plane. Scott-Hayward et al. [7], stated that the challenge emanates from the security measures where the controller may not have the capability to authorize and authenticate applications and resources used by applications with the right auditing and tracking. However, Zhang et al. [44], also stated that due to centralized architecture, the SDN controller is vulnerable to security threats and the security of the targeted network will importantly depreciate if the controller is compromised by intruders. El Moussaid et al. [45], further reported that due to SDN central architecture, the controller is the one responsible for network configuration and decisions. Therefore, exploitation and vulnerability can easily allow attackers to get access, and this can cause damage to the whole network.
Hardware-based	Since the controller is regarded as the leader of the network, it might be the main choice for many attackers. However, in SDN, the controller is a logically centralized device and is prone to malicious attacks. For example, a DDoS attack can target the controller to destroy it. Furthermore, Zhang et al. [44], reported that this can happen when attackers produce several fake packets and send them to switches. In contrast, the computational resources of the controller can be drained at any time and affect the whole network. Communication channels are the first to be attacked and links between controllers and switches can be cut down. Therefore, this can dramatically affect network performance.
Protocol-based	Zhang et al. [44], stated that a controller commonly consists of modules namely network management and monitoring which are also known as third-party applications. However, if these modules are compromised in the controller, detrimental problems can occur and induce vulnerability to the system. Furthermore, for these security challenges to be avoided better network mechanisms need to be established. Therefore, this can be done by applying newly developed SDN technologies to detect them and respond to malicious attacks in advance.

Threat-based physical, node or base station attack, privacy-based threat, routing mechanism, transmission channel experiencing threats and link level experiencing attacks.	Thupae et al. [46], stated that there is a need for the usage of side-channel analysis for the physical destruction of sensor nodes in SDWSN, in capturing and compromising sensor nodes. Furthermore, privacy is experiencing a threat that leads to accessing different types of personal information utilizing WSN's vulnerability. Therefore, security-based threats also prevent message forgery and information alteration from eavesdropping on a wireless channel.

Security issues: SDN has two types of rules that make the establishment of networking transformation and security problems. First, is the capability of controlling the network by using software and the other is the centralization of intelligence within the controllers. The main security challenges or attacks in this interface are threats from applications and threats due to scalability. Moreover, regarding SDN structure, security issues can be concluded from hardware and protocol-based challenges. Zhang et al. [44] and Scott-Hayward et al. [22], stated that SDN generally experiences several security challenges such as SDN controller, hardware-based, and protocol-based as shown in Table 2.2.

Proposed solutions: Scott-Hayward et al. [7], presented many network programming languages such as Procera, NetCore and Frantic that can be utilized. In particular, the FRESCO scripting language allows researchers to execute new security applications using the OpenFlow controller. Moreover, different security models are designed to check whether SDN applications abide by security policies. In contrast, the solution to address security issues towards SDN is to enhance security measures during real-time network management.

Therefore, maximizing the memory of controllers is also a possible solution. It can check flow rule complications that are occurring in real-time, and they have authorization from OpenFlow applications before altering flow rules.

2.5 Overview of SDWSN

SDWSN is an essential system that comprises divisions such as generation, delivery mechanism and a reconfigurable WSN [47]. It is a paradigm that still emerges for low-rate wireless personal area networks (LR-WPAN) [2]. However, SDWSN is still in its early stages and improvements in the research community have been noticed. Moreover, the main existing SDWSN architectures are categorized into communication mediums and constrained resources [48]. However, when designing a secure SDWSN system user mobility, multiple operators, overhead and compatibility need to be given attention. SDN still experience important issues such as standardisation and adoption which have not yet been resolved. However, although this paradigm has promised on reducing energy, the area of this assertion needs evaluation and quantification. In WSN, traffic management, sensor node mobility localization accuracy and network management still pose concerning issues [49]. The WSN realm also consists of inherent issues such as communication, routing, security, and configuration which have ignited research interest in previous years. Modieginyane et al. [30], reported that WSNs are

information technologies categorized under modern networking and computing platforms. These sensor networks are widely used in today's network computing applications. Moreover, WSNs are still faced with the demand for network functionalities that can reach customer satisfaction. This subsection extensively presents security issues in SDWSN and the proposed solution.

Security issues: According to [13, 43, 50], SDWSNs face several security challenges namely: middle-boxes and security using the transport layer. However, security is needed in the control plane, and this is to safeguard SDWSNs against all threats. Therefore, data sensing challenges on the data plane will be solved. Other security issues in SDWSN that need close attention are error-prone, performance tuning and network management.

Table 2.3: SDWSN security requirements [51]

Requirements	**Description**
Responsive alerts	Security events should be in real-time and fast.
Adaptation	User mobility and dynamic network conditions should be taken care of.
Consistency	Is defined as flow rules that are characterized by different applications without any measure of deviation.
Confidentiality	Third parties are not allowed to access unauthorized information.
Integrity	Information should be kept safe and not be modified by attackers.
Authorization	Only legitimate users should access resources.
Authenticity	All entities should be fully secured.

Proposed solutions: Based on the above-stated issues, the security model should be designed and implemented with network security and other protocols used for communication. In such a manner, the advancement of the IoT paradigm should address security issues for SDWSN to be well-secured. Therefore, for accurate future SDWSN networks, a good technique should ensure the reliable utilization of resources. Some of the security requirements are presented in Table 2.3 [47].

2.6 Fault Tolerance

This section fully discusses FT in general, and it also outlines other researchers' work, limitations and finally mechanisms with their design principles. FT is described as a paradigm feature in networks that supports the functionality of important systems such as defence systems, power grids, financial trades, and transportation systems. Botelho et al. [52, 53], defined FT as an important part of any network system control and is typically a built-in design. They also stated that SDN FT accommodates different fault domains namely: the data plane,

control plane and the controller itself. Controller architecture allows controller instances to coordinate their actions through a dependable data store. They further reported that related literature on FT-based SDNs is relatively scarce. Jerlin et al. [54], stated that FT-based SDN maintains sensor network functionalities without interruption due to sensor node failure. Reasonably, each node in the sensor network has unlimited energy, and hence the failure of a single node does not affect the entire task of the sensor network.

Duran et al. [55], proposed select and fast-fail mechanisms in Openflow groups to improve FT and the results showed that the mechanism provides proactive redundancy mechanisms. The mechanisms can recover themselves in terms of link failures without the controller intervening. In essence, the approach is efficient because it reduces packet loss due to link congestion. Based on this research study, the authors emphasized that they would extend this work by dynamically assigning and creating a group's structure; the main reason is to optimize memory usage in core switches. Francesca et al. [56, 57], reported that a single point of failure in SDN can compromise network functionality. Furthermore, they elaborated that the replication approach was used in their work due to its backup servers that are kept consistent with the state of the primary server. Pfeiffenberger et al. [58], stated that achieving a robust controller hybrid approach to FT is the best method to follow. They further explained that with this mechanism, network fault tolerance but without limitation to restoring bandwidth efficiency and FT after a fault occurred is possible.

Qiu et al. [59], presented their work by designing an FT mechanism in WSN. Based on their principles, they designed and developed an energy-aware mechanism called informer homed routing (IHR), which is an FT mechanism and an advanced algorithm to an existing one called dual informer routing (DIR). This mechanism reduces WSN lifetime by transmitting more significant data during a situation when a node is faulty. In contrast, Botelho et al. [52], proposed fault detection and election algorithms alternatively to implement the database in the coordination service.

2.7 Intrusion Detection

This section presents an overview of types of attacks, densification systems, existing mechanisms or approaches and design principles. It fully discusses ID in general and outlines other researchers' work and their limitations.

An IDS is useful in network security and gives little information to other information-supportive systems where attacks occur [60] and such information is useful in mitigating the cause of intrusions. Furthermore, authors in [27] stated that since intrusion is related to a security perspective, WSN is generally faced with new security challenges namely: passive attacks, active attacks, internal attacks, and external attacks. Attacks are categorized into three sections namely: physical, data link and network layer. Zwane et al.[61], proposed a flow-based mechanism in IDS to detect intrusions using an ensemble learning technique. However, for the development of this method, the authors used ML algorithms. Moreover, they evaluated DT, NB, and SVM and based on the results, obtained that DT performed better compared to the

other algorithms. According to the authors, the limitation of this approach is that the flow-based IDS proof of concept prototype uses the DT novel ensemble technique. Also, their proposed IDS was implemented in SDN based wireless network and evaluated according to classification accuracy.

Furthermore, Baraneetharan et al. [60] discussed the important WSN-based approaches as follows:

a) *Anomaly-based detection approach:* is categorized into node, network, and data anomalies. However, node anomalies deal with software or hardware challenges occurring in sensors and this is due to power limitations. In essence, network anomalies deal with connection challenges while data anomaly occurs due to datasets disorder and irregularities caused by sensor problems [60].
b) *Misuse-based detection approach:* known as signature-based IDS used to detect well-known intrusions, and its weakness is that any new intrusions cannot be identified. Therefore, using this approach in WSN provides difficult tasks and is less effective [60].
c) *Hybrid-based detection approach:* is the combination of anomaly and misuse detection approaches, used for clustered WSNs to achieve accurate IDS. It also uses distributed learning algorithm (DLA) to train a well-known support vector machine (SVM) [60].

The limitations of these researchers' work clearly state that ML techniques have been applied and the following applications are needed for future research studies:

a) Detecting data spatial and temporal correlations using hierarchical clustering
b) Compressive sensing sparse coding
c) Distributed and adaptive ML approach for WSN
d) Resource management mechanism using ML

Table 2.4 presents some of the network intrusions and attacks [62] while Table 2.5 highlight the IDS densification system.

Table 2.4: Types of Intrusions or Attacks

Attack type	**Description**
Probe	This attack alters spoofed routing information, it also replies to routing data. However, the training patterns dataset in probe attacks is inside the sniffer.
Unknown	It is regarded as an attack that is classified as zero days.
Buffer overflow	These are attacks known as triggering vulnerabilities.
R2L	This attack is based on information spoofing and affects the efficient use of resources. However, the training dataset in the R2L attack includes IMAP, multihop and guessing passwords.

DoS	This attack is based on the back, land Neptune, teardrop, pod etc.
U2R	This attack is based on load module, buffer-overflow xterm and Perl.

Table 2.5: Densification system

Proposed system	**Detection mechanism**	**Type of detected attack**	**Data Collection**	**Tool utilised**	**Summary**
Global hybrid-based IDS [63]	Anomaly detection	DoS (Selective forwarding)	Traffic based	Network Simulator 2 (NS2)	This model utilises SVM based clustering approach to decrease energy consumption in sensor nodes.

Some of the ID approaches proposed or developed are discussed: Sedjelmac et al. [64], introduced a technique to investigate the issue of intrusion detection (ID) in sensor networks. They proposed a technique called lightweight that can be applied to such networks; their scheme was designed to demonstrate its effectiveness in detecting the attacks before it is late. Ha et al. [65], proposed an approach for traffic sampling rate decisions for efficiently exploiting limited resource-based IDS in detecting malicious traffic.

2.8 Machine Learning Techniques

This section presents selected ML algorithms for an IDM design in Chapter 4 and it includes SVM, RF, NB and LR [66]. Authors in [67] highlighted that in the 1950s, ML was introduced as a technique for AI. The authors further elaborated that the focus of ML evolved and shifted more to computationally robust algorithms. Authors in [68] stated that ML is an inspiration for many practical solutions. It maximizes the usage of resources and prolongs the lifespan of the network. Over time, WSNs must dynamically monitor environments that change rapidly because sensor networks often adopt ML mechanisms. Therefore, the main reason for the adaptation of ML techniques is to eliminate unnecessary redesign. In essence, ML is divided into supervised and unsupervised learning. Authors in [69] stated that compared to unsupervised learning algorithms, supervised learning algorithms are conveniently and extensively used to solve different security challenges and intrusions in WSNs. Machine learning techniques are used to avoid re-programming and this technique is useful for deploying the sensor nodes in a hostile environment [70]. The main goal of using ML is to extract data from different levels of abstraction in WSN [71, 72].

i. Support vector machine (SVM): is mainly a classification algorithm but can also be used for regression classification which is considered by finding the hyper-plane that

separates two classes and this relies on N-dimensional feature space [11, 25, 66, 73]. The SVM training samples are divided into subsets called support vectors specified by the decision function.

ii. Random forest (RF): is mainly a collaborative technique that works based on proximity search and is also regarded as a decision tree-based classifier. For better performance, it uses conquer approach and standard divide, and its main principle applies to the disjunctive hypothesis [66, 74, 75].

iii. Naïve Bayes (NB): is classified as a supervised algorithm that assumes that the probability of each feature or attribute belonging to a given class is not dependent on all other attributes. In essence, conditional probabilities are identified in the value of the attribute known while distance probability can be found by multiplying together all attributes with conditional probability [11, 66]. Therefore, Equation 2.1 is used to find the prediction.

$$P(M|N) = P(N|M) * P(M)/P(N) \quad \text{(Equation 2.1)}$$

iv. Logistic Regression (LR): is used to solve classification challenges and prediction is done by fitting data to the logistic function. Values selected by the logistic function range between 0 and 1 but if the value is 0.5 and above then it is labelled as otherwise 0 [66, 76] as shown by Equation 2.2.

$$\log(p/1-p) \quad \text{(Equation 2.2)}$$

Table 2.6: Summary of ML model

Model	**Description**	**Motivation**
Support vector machine (SVM) [11, 25]	SVM is described as an ML model utilized for identification purposes. Data is represented in n-dimensional space where each feature has a particular coordinate.	The classification process is performed by identifying the hyper-plane and then classifying the classes. Using SVM with eliminated number of features resulting from a feature selection stage avoids over-fitting. Therefore, SVM is adopted in this research due to its speed and high scalability.
Random forest (RF) [74, 75]	RF model is used to improve classification accuracy and consists of many decision trees (DTs).	RF classifier provides the solution to the over-fitting of features in a dataset.
Naïve Bayes (NB) [11]	Is categorized as a probabilistic model with a strong independence assumption. Specificity=TN/TN+TPs between features dataset in the Weka tool.	This classifier provides practical learning where prior knowledge of ML observed data can be combined.

Logistic regression (LR) [76]	A classifier that observes a discrete set of classes in data.	This ML model utilizes logistic sigmoid functionality to take a probability. Thus, its prediction analysis applies the probability concept.

Table 2.7: Model evaluation parameters

Parameter	**Description**
True Positive (TP)	Is defined as a parameter that represents the number of times the intrusion occurred through traffic classification and correct patterns were found [11].
True Negative (TN)	Is defined as a parameter that represents the number of times the normal traffic is classified correctly [11, 77].
False Positive (FP)	Is defined as a metric that represents the number of times normal traffic is identified as an intrusion [11, 77].
False Negative (FN)	Is defined as a parameter that represents the number of times traffic is classified as normal [11, 77].
Classification Accuracy (CA)	is described as the ratio of samples that are correctly classified to the total number of classified samples as per the equation shown below. Furthermore, it classifies the number of correct features predicted against the number of input features in a dataset [11, 77, 78]: $$\textbf{Accuracy} = \frac{(\textbf{True Positive}+\textbf{True Negative})}{(\textbf{True Positive} + \textbf{False Positive}+\textbf{True Negative}+\textbf{False Negative})}$$
Receiver operating characteristic (ROC)	The receiver operating characteristic graph is determined after evaluating ML models with different classification thresholds [72].
Precision	is defined as the number of correct positive features divided by the number of positive features predicted by the classifier: $$\textbf{Precision} = \frac{\textbf{True Positive}}{\textbf{True Positive}+\textbf{False Positive}}$$ [11, 77]
Recall	is defined as the number of correct positive features divided by the total number of relevant features: $$\textbf{Recall} = \frac{\textbf{True Positive}}{\textbf{True Positive}+\textbf{False Negative}}$$ [11, 77]
The area under the curve (AUC)	This parameter measures how well predictions are determined to range from value 0 to 1 and when the score of a model is 100 % wrong that means it has an AUC of 0.0. However, predictions that are 100 % correct have an AUC of 1.0 [72].

F-measure	F-measure defines a harmonic mean between precision and recall and its score ranges between [0, 1]. This is how precise the model regarding correct classified features is [77]. Thus, it is shown by the equation as follows: $$\boldsymbol{F-measure} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive} + \frac{1}{2}(\boldsymbol{False\ Positive} + \boldsymbol{False\ Negative})}$$
Specificity	is defined as the number of correct negative features divided by the total number of negative features plus positive features [78] as shown by the equation below: $$\boldsymbol{Specificity} = \frac{\boldsymbol{True\ Negative}}{\boldsymbol{False\ Positive} + \boldsymbol{True\ Negative}}$$
Sensitivity	Is defined as a metric that evaluates the ability of the model to predict the TPs of each category available [78]: $$\boldsymbol{Sensitivity} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive} + \boldsymbol{False\ Negative}}$$

2.9 Anomaly Flow detection in SDN-SDWSN

This section presents a study of anomaly detection and comprehensively states its overview. Anomaly detection has gained success due to different applications such as fault detection, and cyber-intrusion detection among others. SDWSN-based anomaly flow detection has shortcomings and to solve them, this research discusses methods to apply concerning the reference.

Anomaly detection is a novel technique of finding effective processes to discover faults or intrusions in a network flow. Ali et al. [79], surveyed related works on anomaly detection utilizing ML techniques, and based on their review, challenges such as high false alarm rate and real-time anomaly detection were identified. Pachauri et al. [80], found that discovering faults in medical WSNs is important and they explained that to solve this challenge different fault or anomaly detection techniques are developed. These authors used ML techniques to show that experimentation on real medical datasets experienced good results compared to other anomaly detection techniques. The authors concluded that their research was important in detecting sensor faults or anomalies accurately with low false alarm rates. Dey et al. [81], presented a deep learning mechanism which produced better results compared to ML techniques. The authors used a gated recurrent unit-long short-term memory (GRU-LSTM) model for anomaly detection using a flow-based approach. The approach used achieved higher accuracy and speed in SDN when detecting intrusion. The authors also stated that anomaly detection systems using the flow-based technique in an OpenFlow controller can protect the SDN environment. However, their research focussed on investigating two flow-based IDS approaches in Openflow controller. Thus, the first approach used the ML algorithm, where the NSL-KDD dataset with feature selection obtained an accuracy of 82 % using the random forest (RF) technique. The other approach used a deep neural network (DNN) based on IDS based on

GRU-LSTM. Based on experimental results with comparative analysis, it showed that deep learning appeared as a better approach for ID in OpenFlow controllers. Mousavi et al [82], proposed an entropy-based mechanism for detecting malicious traffic targeting the pox controller, while David et al [83], further extended this mechanism in a flow-based network. However, the drawback of the work stated in [82] is that when the number of hosts increases, more false positive rates (FPR) are reported. Authors in [84] presented their work on anomaly detection using deep learning techniques to classify network traffic using two deep learning algorithms namely: convolutional neural network (CNN) and recurrent neural network (RNN). However, this method was introduced on a graphical processing unit (GPU) with TensorFlow and the results demonstrated better improvement regarding detection accuracy and strong potential usefulness in SDN security.

In [85], authors introduced a novel detection and recovery technique using a multivariate statistical analysis approach in the WSN environment. They considered WSN situations based on temperature sensors for their simulations. They considered three routing algorithms showing strategy, the negative effect of data loss and data recovery capability as well as a new data arrangement mechanism to exploit spatial correlation among sensors. It was concluded that the multivariate mechanism performed best and improved the robustness of WSN due to data loss. According to authors in [86], few statistical approaches for traffic analysis are used. Authors in [87] presented their work using statistical and ML techniques for detecting DDoS attacks. Since network flows statistics and communications are recorded by switches, also referred to as sensors, include total bytes sent, total packets sent and flow time.

Therefore, based on the studies above, this research work used the ML approach for intrusion detection-based anomaly detection method as shown in Chapter 4.

2.10 Overview of Sensor-Flow

This section gives a comprehensive study of Openflow based on the SDN-SDWSN paradigm. Sensor-flow is defined as a component in OpenFlow sensors that measures the network traffic packets. That is, authors in [88], stated that OpenFlow has a unique duty of providing a way to manage traffic in sensors and exchanging data between controller and sensors. An OpenFlow sensor has two logical components following the SDN paradigm which decouple the control plane and data plane. However, an OpenFlow sensor comprises flow tables and an abstraction layer that communicates with the controller through the OpenFlow protocol. In [2], Kobo et al. stated that the OpenFlow protocol is whereby a sensor maintains a flow table consisting of flow entries that are determined to handle traffic packets. The authors clarified that OpenFlow is divided into three communication classes namely: controller to sensor (C2S), Symmetric and Asynchronous classes.

Authors in [89] proposed an IDS which is statistical-based for an OpenFlow-based SDN for detecting encrypted insider attacks. It operates by utilizing the network statistical information from the OpenFlow-enabled sensors. The controller requests this information and the OpenFlow switch sends logs of the statistics to the controller. The controller then mines the

features using a new flow, stores them and then hands them over to the IDS which performs deep analysis to detect anomalies or malicious traffic flow. Detection is by flow classification using features such as average bytes per flow, growth of single flows, average packets per flow, growth of different ports, average flow duration and percent of airflow. To match traffic flow, source and destination IP address, and port number address are used [89]. This approach was not implemented and evaluated but is future work.

In [90], authors proposed an ID module for OpenDayLight (ODL), an SDN controller to detect and prevent unnecessary traffic flow from entering the network. IDM is integrated into the controller and constantly monitors the packet count per flow. Once the network flow statistic is available to the controller from the OpenFlow-enabled switch or sensor, the IDM monitors packet count and computes the packet rate at a different interval. With a given threshold value, if the computed rate exceeds the value, a higher priority flow having the same match criteria is added with action as a punt to the controller. The flow is given an idle timestamp of 5 secs to minimize message overhead and the other few packets follow the same punting to the controller. Deep inspection is carried out by the controller, for known packet senders appropriate rate limiting is performed immediately and for unknown packet sources, a new flow is added alongside relevant match information. Furthermore, drop action which enforces the dropping of all packets from unknown sources is introduced to protect the entire network. The approach implemented using the OpenFlow plugin module in ODL has not yet been evaluated. Figure 2.4 illustrates the important process applied by the OpenFlow protocol which is further extended in Chapter 4.

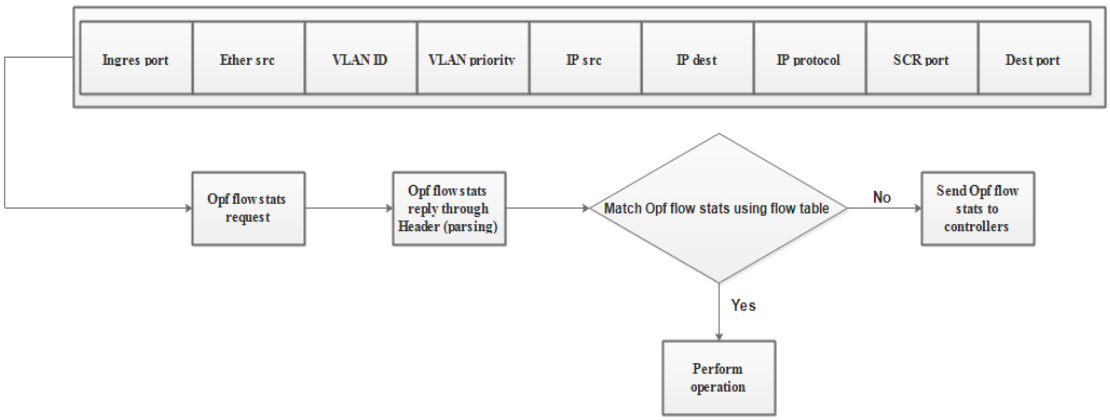

Figure 2.4 OpenFlow architecture [2, 91]

2.11 Related Works

This section presents some of the related works found in the literature. According to Azzabi et al. [31], SDN-WSN with its different paradigms offers benefits to address network problems. However, the existence of a centralized controller easily brings a lot of challenges such as fault tolerance (FT) and intrusion detection. We study several related works and discuss proposed

mechanisms, tools and approaches used. Though researchers have proposed and developed multiple controllers as solutions, the issue of consistency and global view of the network is yet to be solved, and the design is still in its infancy stage. Furthermore, Thupae et al. [46, 92], stated that several issues cannot be given attention by researchers if there is a possibility that the SDWSN system can suffer from reliability and FT challenges. Since security is one of the important challenges, in the perspective of FT and IDS after architectural design SDWSN is still faced with reliability and resilience issues. Several research findings have shown that ML applications in the realm of the network are beneficial when the best-performing or appropriate algorithm is employed.

Therefore, this section presents existing work from peer-reviewed sources to construct a relevant literature review.

2.11.1 Fault tolerance in SDN-SDWSN

Kim et al. [93], stated that the issues of fault tolerance are experienced because of the vulnerability of WSN dynamics such as packets dropping and so on. Thus, to protect against such, the management system should act carefully to the network dynamics through the configuration of the network. Ortega et al. [94], also added that failures ranging from controller and sensor nodes, to communication can occur at any time without warning which poses a critical challenge to the entire network. Hence, if the failure occurs, WSN still needs to be able to recover to normal functioning without human intervention. There are types of fault domains such as application, data and control plane which are covered by [52, 95]. However, efficient SDN controllers should have the capability of withstanding control and data plane faults or attacks [96].

Froceca et al. [56, 57], reported that controller failure in an SDN environment can damage the functional network. They further elaborated that the replication approach was used in their work due to the backup servers that are kept consistent with the state of the primary server. Qiu et al. [59], presented their work by designing an FT mechanism in a WSN. Based on their principles, they designed and developed an energy-aware system called informer homed routing (IHR) which is an FT mechanism and an advanced algorithm compared to an existing one called dual informer routing (DIR). This mechanism extends the lifetime of WSN and transmits more significant information in a situation where node faults occur.

Li et al. [97], presented their work by designing FT which is based on a restoration approach. In their study, the authors used failure detection in a controller and determined new routes to maintain better communication. Furthermore, the authors designed a restoration approach by using a local optical method whose aim was to reduce the path calculation time. Song et al. [98], introduced the control path reliability idea for out-of-band controllers to prevent a network problem due to data plane failures. Cascone et al. [99], utilized a state machine in the data plane to detect and recover failure in both the data and control planes. AI Aghbari et al. [100], investigated FT capability-based WSN and also evaluated the robustness of the particular network. The authors proposed a new algorithm for FT with coverage preservation

in face-based WSN by selecting substitute nodes to replace the failing ones. They also presented a distributed algorithm for failure recovery, which is in conjunction with repairing and restoring the network structure. In comparison to the following authors' [101, 102] existing techniques, the performance of the FCAFT scheme was evaluated through simulations and showed effective results of FCAFT in handling failures by restoring the connectivity of the face-structured WSN.

Yamansavascilar et al. [103], stated that existing SDN-based data plane FT mechanisms can be categorized into reactive and passive which may or may not depend on the controller. However, these mechanisms are still at an early stage because they have partial solutions. In contrast, authors [103], proposed dynamic protection with quality of alternative paths (DPQoAP) that does not only cover existing network faults but also the quality of alternative paths (QoAP). Hu et al. [104], highlighted that link fault tolerance (FTlink) has been a long-serving paradigm in SDN, in contrast to reactive, proactive techniques proposed to sustain network robustness. These authors proposed FTlink which is a flexible and efficient scheme in SDN. The scheme follows a two-step heuristic algorithm: firstly, an algorithm that transmits elephant flows with a greedy tracing method, and secondly an algorithm that transmits mice flows with a bidirectional searching method. Therefore, once a link is monitored and a fault is detected, FTlink checks the matching table entries and enables backup links.

2.11.2 Intrusion detection in SDN-SDWSN

Bysani et al. [105], introduced an approach to investigate the issue of intrusion detection (ID) in sensor networks. They proposed a technique called lightweight that could be applied to such networks. Their scheme was designed to demonstrate its effectiveness in detecting the attacks before it is late. Ha et al. [65], proposed an approach to rate traffic sampling, and, decisions for exploiting limited IDS resources in detecting malicious traffic were also considered. Dwivedi et al. [70], defined an ML approach for anomaly detection and the authors noted the high accuracy of the Bayesian classification algorithm which can detect malicious sensor nodes. Husain et al. [71], showed a comparison between neural network (NN) and SVM and stated that the SVM ML approach performed better than NN regarding accuracy.

Panda et al. [106], utilised supervised and unsupervised techniques for data filtration and selected the NSL-KDD dataset. They proposed using a DT algorithm and the technique worked only for binary classification. Mourabit et al. [107], proposed a feature selection technique and used RF, NBs, K-means and SVM to identify types of anomalies or attacks. They stated that RF outperformed other used ML algorithms and concluded that the hierarchical clustering technique can be used to improve performance and effectiveness. Furthermore, anomaly IDS which collects information on normal traffic patterns was used to detect anomaly traffic patterns. Gao et al. [108], proposed the NSL-KDD dataset to develop an efficient IDS. The authors applied adaptive ensemble learning techniques involving DT, RF, K-nearest, and deep neural networks. According to the results acquired, the DT as a stand-alone algorithm showed tremendous accuracy compared to the adaptive algorithms.

Taher et al. [73], also proposed a supervised ML system to classify network traffic and used the NSL-KDD dataset for training and testing to detect whether traffic was normal or an anomaly. When using SVM and artificial neural network (ANN) algorithms and feature selection methods, the authors discovered that ANN combined with feature selection methods performed better than SVM. Al-Issa et al. [109], used specific datasets to detect attacks in the network. However, with the implementation of ML algorithms, results showed that DT has a lower false positive rate (FTR) and true positive rate (TPR) than SVM. In terms of FNR, SVM is higher than DT. Using SVM, Mourabit et al. [107], proposed three mobile agents (MA) namely: collector, misuse detection and anomaly detection which were used to detect intrusions or attacks in the network.

Authors in [100] proposed a machine learning mechanism for IDS to timely detect and respond to known and unknown network intrusions or attacks in the SDN environment. The approach is called Eunoia and uses both decision tree (DT) and random forest (RF) to construct a classification model using the KDD'99 dataset for model training. The phases of the system include feature selection, anomaly detection, and decision-making for intrusion response [100]. Once an anomaly is detected by the system, a flow rule takes action to drop it before entering the network, and for massive undetected traffic flows, reactive routing is performed. The approach, according to the authors, can effectively protect the network in real-time and minimize uncertainty in classification. Implementation and performance evaluation show a high detection rate and reduced uncertainty in decision-making with small-sized features.
Authors in [110] also implemented an ML-based IDS based on flow information in the SDN to detect and prevent malicious activities in the network. The approach utilized neural network (NN) model pattern recognition constructed on the existing signature-based architecture. An NLS KDD dataset is used to train the model using a backpropagation algorithm with four main categories: Denial of Service (DoS), Remote to a User (U2R), Remote to Local (R2L) and Probes. Anomaly-based attacks are detected when a deviation from normal network behaviour is based on the controller-OpenFlow flow statistics request and reply approach via deep inspection using the NN model in real-time for traffic classification. However, for performance, evaluation was done using open daylight (ODL) controller, running on the Ubuntu 16.04 operating system and Open Virtual Switches which is based on the protocol of OpenFlow. The experiment was simulated using Mininet and 7 features selected from the NLS KDD dataset. Results obtained show promising improvement in the detection rate, achieving an accuracy of over 97 %.

In [111], authors stated that SDN provides the basis for autonomous response and mitigation against attacks on networked computer infrastructures. A novel distributed intrusion prevention system that is collaborative to identify and thwart coordinated intrusion or spurious behaviour was proposed. The technique is an artificial neural network, ANN-based and is used as a virtual network over the substrate of networks. It disperses the computational capability of neurons to part or all the programmable switches in the network where each switch requires little resources in terms of communication and neuron computation. Moreover, the approach is considered scalable, effective at detecting distributed attacks on a global view as well as robust. To

determine the effectiveness, a prototype was realized in OpenFlow-based SDN to validate it. Also, simulations were performed, and the results obtained show that the collaborative intrusion prevention system is scalable and performed better in detecting flooding DDoS than other existing approaches. It was effective at detecting attacks such as Slammer, Witty and Conficker worms.

Carvalho et al. [112], presented an SDN-based ecosystem that can monitor network traffic flow and proactively detects anomalies that can interrupt the functioning of the network. When an anomaly is detected, the technique performs analysis to find loopholes at the network traffic level. The implementation of a good approach to guaranteeing availability is also done, and constant monitoring in the SDN environment relies on OpenFlow protocol and traffic patterns which are detected based on available normal attack profiles. In essence, mitigation policy was invoked due to the detected anomalies. Moreover, simulations were performed using a testbed to detect DDoS attacks. The results obtained revealed a high detection rate compared to other techniques that exist. Thus, it can be used due to its resilience on the network.

Jeong et al. [113], proposed a scalable IDS architecture on an SDN environment using a Kernal-based virtual machine (VM) which employed virtualization. In this approach, the OpenFlow-based switches connect the IDS executing in virtual machines, and the controller and the network attach software in the virtual domain. The focus is on distributed traffic sampling at the switches to inspect and detect malicious traffic packets. A web GUI was developed to visualize the network topology as well as the configuration of the IDS. A performance evaluation was conducted on a virtual OpenFlow-based SDN using 5 attackers' VM and 2 victims and compared with an NB algorithm. The results showed that the proposed approach performed better at detecting malicious packets. This paper [25] conducted a study on OpenFlow-based SDN concerning IDS and DDoS. It also analysed several ML techniques or algorithms which are important for modelling IDS and handling DDoS attacks in SDN. The ML algorithms include Neural Networks, Bayesian Networks, Decision Trees, SVM, Genetic Algorithms and Fuzzy Logic [25]. Based on the analysis, the study recommends the application of ML algorithms for the mitigation of attacks in the SDN. The techniques hold a great research prospect both in academia and the industry.

Zanna et al. [114], proposed a novel distributed IDS which is integrated into the SDN controller to provide scalable threat management solutions in real time. The approach employed the core mechanisms of Bro IDS, an open-source commercial-oriented application for security monitoring and analysis. The authors implemented the IDS core mechanisms that analyse and detect malicious traffic that is forwarded to the controller. The results show the proposed approach's effectiveness in the detection of malicious traffic in the SDN. This paper [88] proposed an efficient and manageable IDS for an OpenFlow-based SDN to detect and prevent malicious attacks from the SDN landscape. The approach detects anomalies by building a classification model using the Bat Algorithm (BBA) for efficient feature selection. It operates by capturing traffic packets and detecting anomalies by selecting essential features of the packets using the BBA. It then classifies them as either normal or abnormal packets and generates the appropriate rules. The system was evaluated using the NSL–KDD dataset

containing normal and 4 types of different attack categories such as a Probe, DoS, R2L and U2R [88]. The results show an accuracy of 98.85 % for U2R with less FP rate than other attack categories. Sayeed et al.,[115] proposed an anomaly-based IDS mechanism for an OpenFlow-based SDN to detect and prevent malicious attacks or network anomalies. The mechanism employs a packet filtering firewall over the floodlight SDN controller and uses the Apriori algorithm's association rules to search for patterns among the data transmitted through the firewall. The patterns are then classified as either normal or anomaly packets where any deviation from normal behaviour is considered potential threat which triggers alarm or alert. Implementation was performed on Mininet and has not been evaluated. According to [116], if implemented in real time, it could produce significant results and give insights into new threats and vulnerabilities identification.

In the study by Ha et al. [65], a sampling method-based IDS was designed to detect anomalies in the network. In this case, suspicious traffic is constantly monitored using a suitable sampling rate (SSR) in each optimized switch, thereby quantifying the network throughput, and deeply inspecting malicious traffic and flow path information using the SDN functionalities. Once malicious behaviour is detected, a notification is sent for proper mitigative action to foil the attack. Simulations were conducted to evaluate the performance and the results showed improvement in malicious traffic detection in large-scale networks. Therefore, this indicated that the sampling scheme performs better compared to the naïve scheme.

Tang et al. [62], proposed a flow-based IDS in an SDN environment to detect flow-based attacks using a deep learning approach based on the controller-OpenFlow switch network statistics request and reply after a given time interval. The authors used a deep neural network (DNN) approach on the IDS module to analyse statistics and to check if there is any suspected malicious behaviour [62]. If any malicious behaviour is identified, the OpenFlow protocol is used for encountered faults and security policies will be sent to the switches to defend the network. Therefore, obtained results showed that the optimal hyper-parameter for DNN works accurately with 75.5 % due to detection and false positive rate (FPR). The author in [117], proposed a lightweight flow-based IDS for the SDN. It operates network statistical information requested periodically by the controller from OpenFlow switches. It then analyses the traffic flow information to mine features such as duration, packet count, byte count, source IP, destination IP, protocol, source port and destination port using the Bagged Tree to classify the packet flow as normal or anomalous. Finally, the Ryu controller and open switch (OVS) were used to experiment, and results showed a highly accurate 0.98 detection rate, with low FPR.

2.12 Critical Literature Analysis

This section briefly discusses critical literature analysis based on two aspects namely FT and ID outlined in WSN-SDWSN.

A. Fault Tolerance

Mohapatra et al. [118], state that fault occurrence is caused by hardware and software failure or malfunction including node fault, communication link etc. However, to close the gap of the problem area necessary enhancement of robustness and self-healing process needs to be adopted. However, their work still needs improvement. Thus, the simulation results have shown satisfactory performance based on the proposed algorithms. Authors in [119, 120] made highlighted the issue of FT where reliance on the controller might not be feasible. However, this resulted since the controller is a single point of failure in the traditional network (SDN) or WSN. In contrast, tissue of FT was proposed using slavery architecture with local mechanisms of virtual controller redundancy and synchronization between controllers. Thus, simulation results showed that concerning Cbench performance tests the proposed light synchronization mechanisms imposed important performance penalties in new network traffic scenarios like those generated by Cbench.

B. Intrusion Detection

Gite et al. [121], state that intrusions are vulnerable to the sensor networks, so authors proposed a detection technique or mechanism that analysed traffic patterns, identifying intruder nodes challenge in WSN. Furthermore, the technique or mechanism addressed four common attacks. In particular, the strength of the mechanism was discovered when a lightweight and reliable model was constituted by identifying attacking nodes or intruders in WSN without increasing the sensor nodes' workload. Thus, the performance of this technique showed promising results which need improvement. Gandhimathi et al. [122], state that packet-based signature IDS match every packet with a malicious signature or activity and it is a challenge to the research community. However, it is a time-consuming process because analysis of every packet needs high computation but it produces few false alarms. According to these authors, Flow-based IDS is a hopeful security technique in sensor networks because it cannot analyse the entire packet load. Thus, its strength is that it offers the best solution for sensor networks because it takes less processing time. In [123] authors discovered that SDWSN is faced with different security problems including SDNs and WSNs. However, novel architecture with an unsupervised detection algorithm using a hierarchical approach was proposed. This was to improve based on the security of integrated SDWSNs. In conclusion, to check or examine the effectiveness of the proposed architecture and algorithms, sensors were simulated on Cooja and NSL-KDD standardized datasets. According to results output, proposed method or output was able detect 97% of abnormal traffic.

Thus, a comparison of findings to the literature is highlighted in Chapter 5 under sub-section 5.5 and this is to show how obtained results differ from the current literature results discussed above.

2.13 Chapter Summary

This chapter presented a literature review of FT and ID in SDN, WSN and SDWSN and related work done by other researchers. In brief, it discussed security challenges faced by the controller and their vulnerabilities to attacks and faults. It also presented some expressed solutions in literature to some of the problems and challenges in WSNs. It presented some practical approaches or mechanisms tried out by a few researchers.

Chapter 3

Research Methodology and Design

3.1 Chapter Outline

This chapter presents the introduction, the discussion of design science research methodology and the mixed methods approach on how to solve the identified research problem. After the discussion, the research design follows as well as the mechanisms on how to achieve the research goal. Data collection, analysis and evaluation follow as part of the discussion. In addition, as part of following the proper procedure in this research study, administrative procedure and ethical considerations are also discussed.

3.2 Introduction

Research is defined as a scientific search or art of scientific investigation as well as an academic activity used in a technical sense [124]. According to [125], research defines and redefines challenges, outlines hypotheses, collects, and evaluates data, reaches conclusions and tests them to check if they fit according to an outlined hypothesis. In this context, this research is designed to answer the research questions outlined in Chapter 1. In the computer science aspect, the research aims to find a hidden solution that has not yet been discovered or a solution to improve on existing problems. Authors in [126] stated that the importance of research inculcates scientific thinking and promotes logical development.

Research in computer science or information systems comprises four paradigms which are:
Interpretivism: applies the qualitative method to discover new research ways and it approaches knowledge by making emphasis on the significance of other researchers' points of view to understand social reality [126].
Positivism: provides a systematic approach to conducting research and uses quantitative methods by focusing on reliable and valid tools [126].
Design science or pragmatic approach: is regarded as a paradigm that takes place as an equal companion to natural science research in CS or IF field and uses mixed methods to solve research problems [126, 127].
Socio-Critical applies an ideological review method to uncover some research inequalities [126].

Based on the nature of this research, the suitable paradigm adopted is the pragmatic approach to answer all the research objectives mentioned in Chapter 1. Consequently, in Chapter 4, this research details the design of the proposed FT and ID mechanisms for SDWSN to prevent the centralized controller from being the single point of failure due to faults and intrusions. As argued by other researchers, designing controller mechanisms for FT and ID performing platform detection is a process which is repetitive for SDWSN because they lack identification and self-healing capability. However, to construct a solution to the identified issue, a controller-

efficient mechanism has been designed by developing a cost-effective framework that incorporates FT and ID.

The significance is to ensure resilience and security in programmable networks. FT was used to increase the controller performance proposed and the best-chosen ML algorithm was utilized to detect and stop intrusions as proposed by [26]. The discussion of research methodology follows in section 3.3. Design science research (DSR) [128] follows the knowledge flows, process steps and outputs to illustrate the functionality of a construct. Figure 3.1 shows the flow of the proposed research methodology and detailed DSR discussed in depth. The DSR is aimed at producing new knowledge by providing new solutions to existing challenges. Figure 3.1 shows the research workflow.

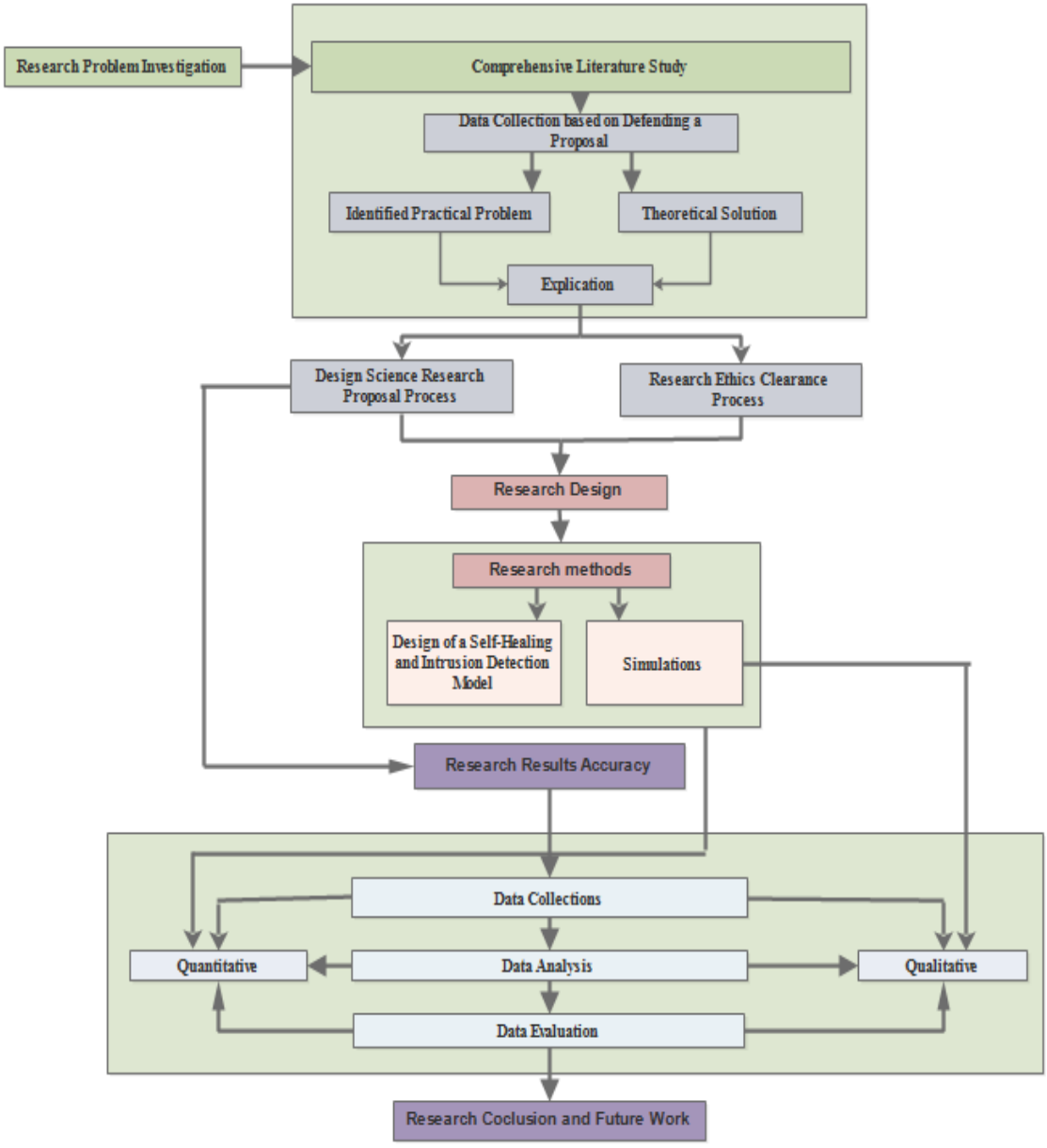

Figure 3.1: Research design process

3.3 Research Methodology

A research methodology (RM) is a technique to systematically solve identified research problems and has many dimensions and methods [129], and its scope is much wider than that of methods. RM does not only have an impact on research methods but also the logic behind the method or technique utilized. It is also discovered as a scientific, systematic, and academic activity for obtaining important information on a research topic. Furthermore, it comprises defining and redefining problems while its purpose is to formulate solutions or answers to questions through the application of scientific procedures. Therefore, for this entire research, we adopted a constructive research methodology whose aim is to solve a practical problem in SDWDNs utilizing simulation and it also aims to produce an academically appreciated theoretical contribution. This, research study involves a practical problem statement, a comprehensive literature survey, the designing of a model or framework, a demonstration of the feasibility of a solution to the problem and an evaluation of results. Moreover, a case study is also presented to demonstrate how the chosen methodology and related methods are applied to our research problem.

3.3.1 Design Science Research Process

As discussed in section 3.2, the design science approach aims to outline new information by providing novel answers to existing problems. The design science follows the construction approach to illustrate the functionality of a model. Therefore, these are the steps that were followed for the entire research.

a) **Formulation of practical or research problem:** To solve the nature of the existing problem identified, an insightful survey on the FT and ID in a controller's challenges on SDWSNs was conducted. We discovered that FT and intrusion were the dominant or major issues in the SDN and WSN realms.

b) **Understanding the research topic:** After studying existing SDN and WSN literature in Chapter 2, we discovered that FT and ID were done separately. However, as an ongoing process, we needed a clear understanding of our research area. That is, the research title is discovered by conducting a comprehensive review or analysis of the existing problems in SDN-WSNs. Moreover, this need had an impact on the way to contribute to a novel SDWSN paradigm. Therefore, to design a secured SDWSNs model, the two aspects were combined.

c) Construct design: Utilizing a design science research methodology (DSRM) [130], the method used to design the proposed technique was based on determining efficient mechanisms following the insightful comprehensive literature review of the existing non-fault and intrusion tolerant controller problems of SDN-WSN and the impact on SDWSNs paradigm**.** The design of these mechanisms was facilitated by collecting information based on the use and implementation of the chosen ML model in SDWSNs.

d) **Evaluation and testing the construct:** This step allowed us to effectively and analytically evaluate our model to see how fault tolerance increases the performance of the controller and how machine learning detects malicious intrusion in SDWSNs. However, illustrating the existing challenge can be solved with the proposed construct which will illustrate or demonstrate full functional and operational simulations. To have a concrete validation of the proposed mechanism, after the design of the model, simulations were run to test the performance and to validate the reliability of the collected simulation data. After completion of this dissertation, it will be published in a university database and also in peer-reviewed conferences and journals platforms or other digital libraries

e) Research on theoretical connection and contribution: Theoretical aspects of the SDN and WSN approach extracted from the literature study must be aligned with the model designed for this research. Therefore, this had a huge impact or contribution to our work.

f) Applicable scope evaluation: Both the scope evaluation and potential limitations of findings based on this research were discussed to clarify the extent of the design of our proposed model.

3.4 Research Design and Technique

This section presents the research design and methods utilized in this research.

3.4.1 Research Design

The research design strategies are based on chosen methods used to approach the outlined research questions in the first chapter. A research design can be referred to as the way forward, outlining how the identified challenge will be circumvented and how the proposed solution will be proffered to achieve the desired research goal [58]. Research methodologies and methods are the key elements in selecting relevant research design techniques. This research follows mixed research design strategies and more specifically model design and simulations. Figure 3.2 outlines research questions, literature study (constructive theoretical model, instructional methods, and design strategies), and implementation and simulation of SDWSN-based FT and ID model (quantitative validation). Data were collected, analysed, and evaluated (qualitative validation), while accurate and efficient results lead to reliable research connection and contribution. On the foregoing, there is a need for research methods and designs to correlate and be used jointly. That is, to allow further extensions and changes to the research. If research design is done, then possibly the methods might lead to the selection of a relevant research method that correlates to a novel research design.

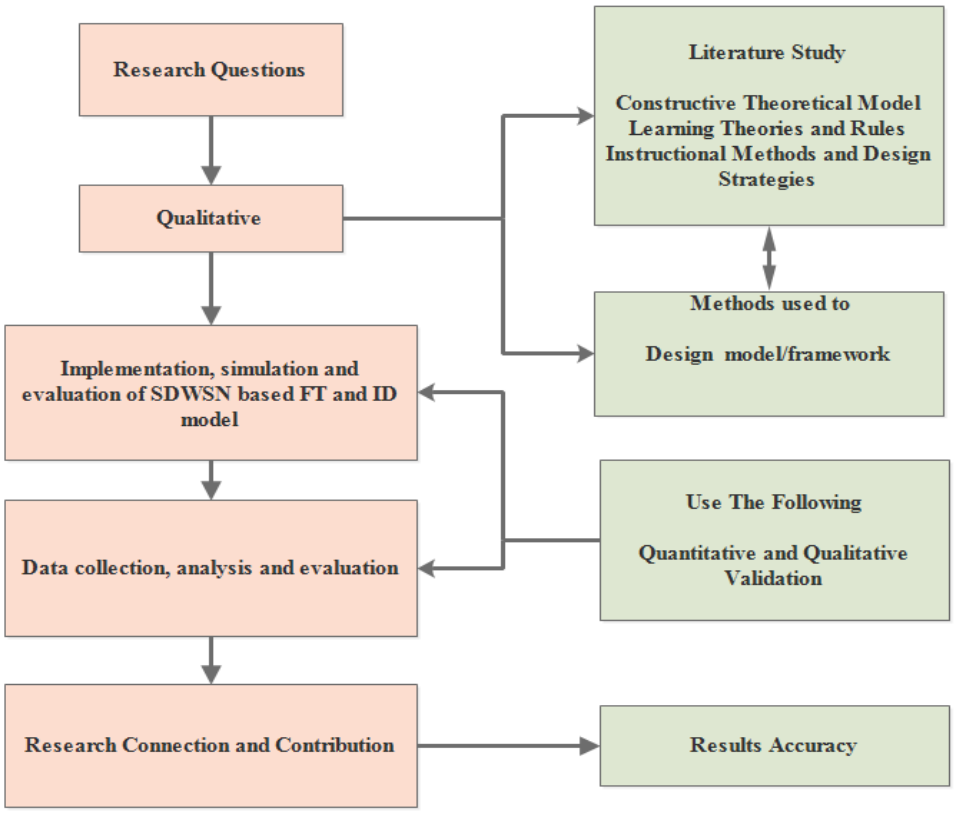

Figure 3.2: Research design mechanism

3.4.2 Research Methods

The research methods play an important role in any research, however, it initiates the researcher to be clear on which methods to apply. This is to accordingly do a research study on the identified research area. The research methods provide a defined guideline model where different research questions will be answered to provide a solution to the identified research problem [57]. In section 3.4, mixed analyses (qualitative and quantitative) are used in this research. Moreover, research methods are concurrent with a specific aim based on the identified research area.

In the context of this research, methods suitable for answering the RQs are discussed as follows:

A. ***Comprehensive literature review (CLR):*** CLR helped in understanding the identified research problem. However, the research topic was outlined by doing an in-depth review and analysis of existing work on FT and ID challenges for WSN leading to the SDWSN paradigm. In essence, this applies to the use of replication-based light synchronization and anomaly detection-based ML approach in SDWSN and was achieved by conducting CLR in Chapter 2 of this research.

B. ***Framework or model design****:* Regarding the survey of the literature we performed in this research, we designed a robust self-healing model where security was integrated into FT to detect both faults and attacks in the network. The ML algorithms were used

to simulate the model while evaluation and implementation were done using a Wireshark analyser and rapid miner simulators.

C. ***Simulations:*** This method allowed the researcher to evaluate the system. Simulation is a method used in this case for studying newly developed network protocols. The reason for its application is to use more cost-effective tools to build huge networks. The chosen tool for simulation supports the SDWSN properties namely FT and ID to secure the whole network. Therefore, the simulation should support the security model where the SDWSN is vulnerable to attacks or intrusions deployed in controllers and sensor nodes. The proposed research study designed an approach to SDWSN FT and ID mechanism illustrated in Figure 3.2.

3.5 Data Collection and Analysis

In this research, data analysis was not externally collected, however, it was gathered from the simulations. Firstly, the proposed FT and ID mechanism for SDWSN was designed using the chosen design platform and ML algorithms. Then, the design followed the simulations to generate the data that was used for data analysis, validation, research connections and contributions.

3.5.1 Data collection

Data collection is based on qualitative and quantitative data collection as follows: Qualitative data collection was based on the literature review conducted for an in-depth understanding of identified research problem and collection of relevant information to propose a solution to circumvent the identified issue. And, to challenge the assumptions made towards the proposed research.
While the quantitative data collection was based on conducting several simulations for accuracy and consistency, collecting more data for analysis, and considering significant changes in data.

3.5.2 Data analysis

The data used in this research was generated by simulating the proposed FT and ID mechanism in SDWSN. Data analysis is an important stage of any research where the requirement of interpretation of collected data is based on logical and analytical reasoning to relationships for easy understanding. Data analysis was done both qualitatively and quantitatively as explained.

Quantitative analysis: The quantitative analysis is based on numerical analysis, it follows simulation and statistical methods [131]. The use of this analysis approach to conducting relevant analysis on the identified problem and the research topic enabled us to use quantitative analysis in the design of the proposed FT and ID mechanism/construct using experimental approaches like simulation to have a fully functional construct without errors and also collect,

analyse, and evaluate the data after simulations for its performance and effectiveness. Therefore, this is based on:

- Literature study to outline the in-depth understanding of simulated FT and ID mechanisms to measure effectiveness and performance.
- Tackling any applicable assumptions based on the proposed research.
- Conduct simulation to check accuracy and consistency
- Data analysis through simulation
- Collecting data to perform analysis and present them in visualized graphs
- Consider important any important applicable changes in data

Therefore, DSRM which is based on the FT and intrusion ID process was used. In this case, analysis involving a mixed research approach of qualitative and quantitative provided a firm guideline for conducting this research to achieve the desired objectives and answer outlined research questions [131].

3.6 Research Evaluation

Following data collection and analysis as stated in section 3.5, we highlight the research evaluation stage where the proposed research problem and results were evaluated in terms of effectiveness and performance. However, research evaluation concerns quantitative and qualitative analysis and FT/ID mechanisms are important concerning evaluation, effectiveness, and performance. Therefore, the following is a discussion of the metrics used for evaluating the proposed solution.

3.6.1. *Metrics or parameters used for FT and ID*

In this research, the following metrics or parameters were used in the model evaluation under the FT aspect and ML algorithms considered under ID SDWSN as shown in Tables 3.1 and 3.2, and then further discussed. From the perspective of security, the proposed SDWSN-based FT and ID model needs to be reliable, available, and resilient due to single points of failures in controllers and sensor nodes. However, availability is defined while reliability and these aspects lead to the following metrics or parameters for the model to withstand faults in the network. Therefore, the proposed model depends on the following metrics.

1. Throughput: depends on variation in packets or messages delivered at sensor nodes, if the communication path shows a faulty sensor node, then the delivered packets or messages rate drops. This scenario can affect throughput which is defined as:

$$\boldsymbol{Throughput} = \frac{\boldsymbol{delivered\ packets}}{\boldsymbol{propagation\ time}} \quad \text{.....(Equation 3.1)}$$

1. Latency: depends on the source node, where the presence of a faulty sensor node delays the transmission of delivered packets or messages. However, this results in packets or messages taking longer to reach the destination and latency is defined as follows:

$$\boldsymbol{Latency} = \frac{\boldsymbol{end\ to\ end\ delay}}{\boldsymbol{number\ of\ nodes\ in\ path}} \quad \text{.....(Equation 3.2)}$$

Table 3.1: FT evaluation metrics

Metric	Description
Latency and Delay	Is described as a delay from data input into a system to the desired data output.
Throughput	Is described as a measure of the total amount of data transmitted per unit of time.

As for IDS, a confusion matrix is defined as a tabular tool used to outline the performance of each ML algorithm in terms of classification. According to this research, the confusion matrix is explained and applied to help in the performance of each algorithm so that the best model can be chosen for the design and implementation of our SDWSN-based FT and ID model. In this essence, the confusion matrix with some evaluation metrics was used due to ID and was derived from TP (define normal as normal), FP (define normal as an anomaly), TN (define anomaly as an anomaly), and FN (define anomaly as normal) attributes. Furthermore, FAR, accuracy, specificity, sensitivity, detection rate, precision, and F-measure are the common evaluation parameters used by researchers in the ID aspect. The following parameters were used in this research work.

2. Detection rate: determine the ratio between the correctly classified number of anomalies and the total number of anomalies present in the dataset and computed as follows:

$$\boldsymbol{Detection\ rate} = \frac{\boldsymbol{TP}}{\boldsymbol{FP+TP}} \quad \text{.....(Equation 3.3)}$$

3. Accuracy: define the percentage of correctly classified test features by training the algorithm in the dataset and is computed as follows:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad \text{.....(Equation 3.4)}$$

The ID metrics or parameters are utilized to check and good performing chosen ML algorithm in this research whereby they also allowed designed IDS to check attacks or intrusions classified.

Table 3.2: ID evaluation metrics

Metric	Description
Classification accuracy	Is defined as the percentage of correctly classified test features by training the algorithm in the dataset.

Detection rate	Is defined as the ratio between the correctly classified number of anomalies and a total number of anomalies present in the dataset.

Thus, research evaluation is performed to:

- Measure the depth of collected data through simulation which is based on the proposed research solution.
- Provide informative data to address identified existing research problems to answer the why, what and how questions. This was raised by quantitative data analysis.

The evaluation and reliability of this research solution were done by submitting scientific academic articles including peer-reviewed papers, accredited conferences or journal papers. Doing the evaluations helps to:

1) Provide contextual data collection which addressed identified challenges based on the existing research problem and also answered the questions such as:
    - Why did it happen?
    - How did it happen?
    - What caused it?
2) Measure collected data extracted before and after the simulations. This involved data generalization or collection, data analysis and precision and consistency evaluation.

3.7 Simulation set-up

This section presents the system properties and the parameters used for the simulations of the proposed integrated FaToID model for the SWDSN. For FaToM, Mininet is created in Python, provides (API) for user customization, and does not rely on C programming utilities. Firstly, *sudo su* was used to login and installation of controllers was done using this command $ Mininet /util/install.sh. Moreover, the command $ *sudo mn* was used to create a tree network topology with 12 hosts, 9 nodes, and 3 controllers. Generally, the virtual network topology was designed to be used in a proposed simulation and to achieve RO1 as stated in Chapter 1. The command $ *sudo* Wireshark was used to run the emulator with root privileges which have a minimal security risk. This sub-section outlines the system properties used to perform the simulation as shown in Table 3.3.

Table 3.3: System properties

Parameter	**Value**
Hardware	• Device name: r2pae-Veriton-Z4620G • Hard Disk:491.2 GB • Random Access Memory: 3.7 GB • Processor: Intel® Core™ i3-2130 CPU @ 3.40GHz x 4

Software	• Linux-Ubuntu 18.04.5 LTS, 64-bit operating system • Ubuntu server (for pox, default & floodlight) • Mininet 3.1.2 • Wireshark 3.4.2 • Orange 3.3.1 • Rapid miner

In this research, several metrics were employed to assess the performance and effectiveness of the proposed system in terms of FaToM and IDM. This includes the throughput and latency or delays under the FaToM phase whereas, for IDM, we used specificity, sensitivity and accuracy.

A. Simulation parameters

To assess the proposed integrated model based on the data generated through simulation, effectiveness, and performance for FaToM are measured or dependent on latency and throughput as shown in Tables 3.3 and 3.4. However, as for IDM, classification accuracy, F-measure, Precision, Specificity and Sensitivity are also used. In this case, an assessment using specified metrics was based on the chosen dataset and ML techniques.

Table 3.4: Simulation parameters

Parameter	**Value**
Emulator	Mininet
Simulator	Wireshark, Orange and rapid miner
Simulation area	200*200 meters
Data transmission time interval	5.0 seconds
Transport control protocol	TCP
Number of controllers	3
Number of sensor nodes/switch	24
Number of Hosts	12

B. Simulation procedure

This subsection outlines the simulation process which involves FaToM and IDM known as FaToIDM, and Table 3.5 shows a ping test between 12 hosts, 3 controllers (Default, Floodlight and POX) and 24 sensor nodes respectively. Accordingly, the data collected through simulation was assessed, analysed and presented in Chapter 5. This was done by executing Mininet, and the Wireshark concurrently on Linux-Ubuntu 18.04.5 LTS, a 64-bit operating system, and also Orange and rapid Miner, a data mining tool used to choose the best ML algorithm to detect intrusions. Firstly, for the FaToM, Mininet was used to design the tree virtual network topology using tree topology because it can accommodate multiple controllers. We applied a scenario

where the *sudo mn* command is used for the execution. Table 3.5 presents the ping test across hosts to check the reachability between nodes/switches and controllers.

Table 3.5 Ping test between hosts and nodes

Controller(s)	**Host(s)**	**Number of Node(s)**	**Packets transmitted and received**	**Ping test (s)**
Floodlight, Default and Pox	H_1-H_2	24	67	67,543
	H_1-H_3		40	39,909
	H_1-H_4		30	296,54
	H_1-H_5		50	50,134
	H_2-H_6		93	94,167
	H_2-H_7		75	75,710
	H_2-H_8		78	78,634
	H_2-H_9		193	196,551
	H_3-H_{10}		161	163,793
	H_3-H_{11}		160	162,754
	H_3-H_{12}		159	161,747

However, this research study used Mininet, Wireshark, Rapid Miner and Orange, collect data was collected using CIC-DDoS2019 in comparison with NSL-KDD datasets to determine results accuracy.

3.8 Administrative Procedures

This section outlines the administrative procedure required by the Computer Science Department under the Faculty of Natural and Agricultural Sciences (FNAS) at North-West University Mahikeng Campus to authorize conducting the proposed study as well as the study title registration. To do the proposed study, three major activities are required:

- Research needs to be defended
- Conducting corrections outlined by the Department during proposal presentations
- And the application for research ethics clearance.

The proposed research proposal was successfully done on the 30th of August 2018 followed by amending the final proposal which was submitted on the 09th of August 2021 and the proposal corrections in table format submitted to the department head of research on the 10th of August 2021, and lastly was the application of the research ethics which was submitted to the department on the 3rd of September 2018 and after expiry, it was renewed on 30th of August 2022.

3.9 Stages of the Research Study

The aim of this research study is based on to improve the existing energy-aware mechanism by designing the proposed efficient self-healing and intrusion identification SDWSN mechanism. Figure 3.3 illustrates the proposed research study that was done following several research steps, starting from the literature until the final phase evaluating the performance of collected research results. The model illustrates certain steps followed to provide the solution to the identified practical problem and done by achieving desired objectives supporting outlined research questions.

Based on Figure 3.3 of the conceptual model for the research study, the comprehensive literature review was done from Chapter 1 until Chapter 3. The literature study outlined data collection involving depth understanding of the research area. Furthermore, it also identified problems affecting the proposed research area. That is, a theoretical solution was proposed to circumvent the identified practical problem. However, both identified research problems and proposed theoretical solutions were explicated. Meaning identified research problem was thoroughly analysed to reveal its cause in the research area. Furthermore, also how the proposed solution will circumvent the identified research problem and to what extent the proposed solution will improve identified research area. Following the explication stage, the proposed theoretical solution was carried out as the design of a robust self-healing and intrusion detection mechanism for SDWSN to circumvent the practical problem. Detailed qualitative and quantitative data collection, analysis and evaluation have been discussed in sections 3.6 to section 3.8. In Chapter 4, the actual robust self-healing and intrusion detection model in SDWSN to circumvent fault and intrusion tolerant inefficiency is proposed and designed; simulations were conducted to collect the data that was used for performance analysis and evaluation outlined in Chapter 5. Simulations of the proposed solution were carried out several times for data accuracy and consistency; to determine whether the outlined objectives were met and whether the research questions were answered accordingly.

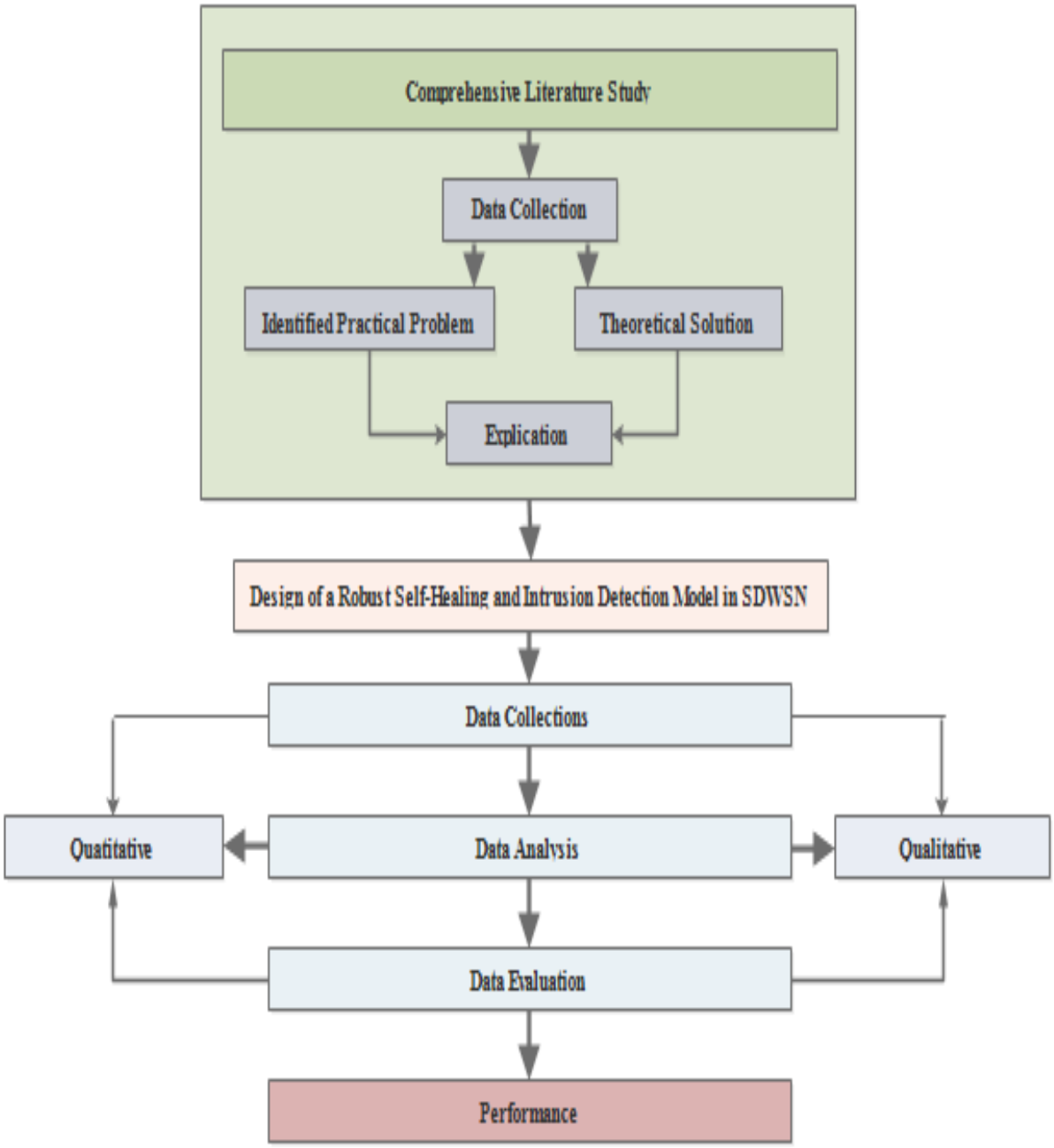

Figure 3.3: Process of the research study

3.10 Chapter Summary

This chapter presented a comprehensive and detailed research methodology and design used in this research study or project. To achieve the desired research goal or aim, the methods, and design of this research constructive research methodology were discussed. Furthermore, administrative, and ethical procedures were discussed, and relevant techniques or mechanisms were outlined.

Chapter 4

Integrated FaToID Model

4.1 Chapter Outline

This chapter presents an integrated framework for SDWSN that involved both fault tolerance (FT) and intrusion detection (ID) mechanisms. It begins with an introduction and progresses to the fault tolerance and intrusion detection model (FaToIDM) system architecture comprised of fault tolerance model (FaToM) and intrusion detection model (IDM) also known as IDS design. The chapter also presents the model of both proposed phases and provides a detailed explanation of each of the components involved.

4.2 Introduction

Software-defined wireless sensor network (SDWSN) is faced with several challenges which are dominated by security threats and attacks of different degrees. Moreover, having a single controller in the control plane of the network, makes it vulnerable to a single point of failure. Though several solutions have been proposed in the literature, existing security solutions are yet to detect the presence of faults in the network nor do FT solutions detect security threats and attacks in the system. Therefore, to ensure that SDWSN is robust and can resist both security attacks and faults in the network, the fault tolerance model (FaToM) and intrusion detection model (IDM) are integrated into one model known as fault tolerance and intrusion detection model (FaToIDM). The purpose is to efficiently guard against network failures or compromises from both faults and security attacks.

Thus, Chapter 4 presents an integrated system framework for the SDWSN architecture. Combining both FT and intrusion will allow the system to have the capability to defend against all forms of attacks and faults or failures while still ensuring consistency of network states and views. To achieve this:

1) We first provided the design of the FaToM which involves the use of multiple controllers. The aim was to answer RQ 2 by ensuring consistency and resiliency to avoid a single point of failure in the network posed by the centralized controller.

2) Secondly, we provided the design of the IDM which is combined with the FT to answer RQ 2. The goal here was to incorporate security into the SDWSN model alongside the FT model to detect intrusions or attacks at an early stage.

3) Thirdly, we presented the overall system architecture which is the FaToIDM in the proposed SDWSN model. To effectively achieve the objective of the IDM, we employed the best ML model in terms of classification accuracy to detect a multitude of security attacks and threats such as DoS and DDoS. The overall objective was to design a resilient SDWSN system with fault-tolerant and secured multiple controllers

which can detect intrusions that bring about performance, reliability, security, and capability of eliminating single points of failure in the controllers.

Certainly, the aim was to propose the design of a robust self-healing FaToIDM for the SDWSN. FaToM was designed using Mininet in terms of its network topology with multiple controllers. Wireshark was used to capture packet flows, identify, analyse, and check if there are lost packets which will determine if there is still an alive controller. Finally, IDM was based on flow anomalies trained and tested using NSL-KDD in the orange tool.

4.3 FaToIDM architecture

This section presents the system architecture as shown in Figure 4.1, its components, and the interrelationships between them. The basic components of the proposed FaToIDM architecture are the FaToM and the IDM. The design was to enable the SDWSN model to be fortified with functionalities that can keep it in continuous operation even in the event of fault/failures or security attacks. To achieve these functionalities, important quality requirements are to ensure the security, reliability, availability, and consistency of controllers. The essence is that the controller, as the intelligence of the SDWSN network, is always the primary target of attacks and is seen as a single point of failure for the centralized SDWSN architecture. Therefore, FaToIDM architecture in the context of this study is designed as a robust model to defend against such catastrophic attacks and faults. To achieve the architectural goal, multiple controllers were used for mitigating the single point of failure which in turn increases the reliability, and availability of the network.

1) *Distributed Multiple Controllers:* In the context of this research, the centralized but distributed multiple controllers (DMCs) placed in different network domains are proposed where each serves as a centralized controller to ensure the operation or function of the network despite faults or failure. These proposed controllers are logically and physically distributed in the FaToIDM to manage the network in each domain and its neighbours. The main function of the DMCs is to guard against unexpected failures in the network.

2) *FaTo Module:* This is designed in the SDWSN to implement reliability or guard against faults or failures in the network. As shown in Figure 4.1, the data plane and control plane are the two most important elements which call for FaToM and its main function is to detect failures and recover from them using the technique of active replication. The active replication ensures all the controllers are active with one controller managing the operations of the network while sensors are connected or assigned to multiple controllers. The objective is to ensure that when a controller fails, other controllers can still control the sensors without switching. However, for this to be effective, a novel controller placement was considered and designed.

3) *ID Module:* IDM is designed in the SDWSN to detect a wide range of security attacks and network intrusions such as DoS, DDoS, black holes etc, as discussed in Chapter 2, to prevent them from being harmful to the network. Its responsibility is to monitor the network

and detect any forms of attacks or intrusions using the ML technique. Though several IDS have been proposed for SDN or SDWSN, the IDS in this integrated system will operate alongside the FaToM to increase the reliability and resiliency of the network.

Figure 4.1 shows FaToIDM architecture.

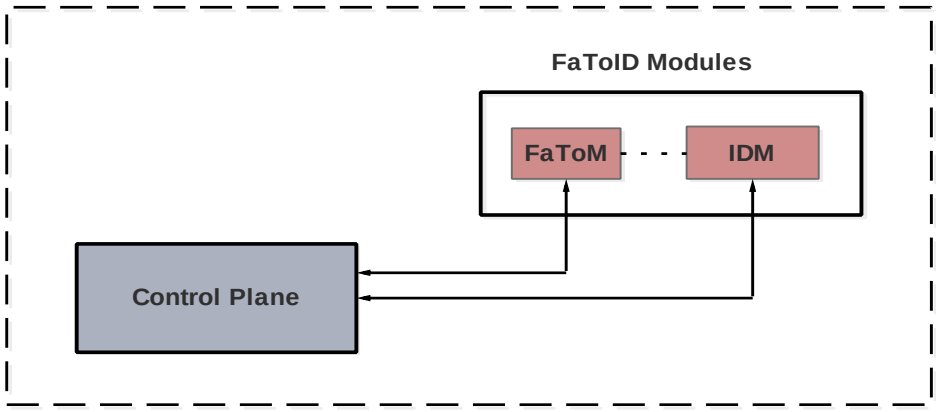

Figure 4.1: FaToIDM architecture

Therefore, the detailed design of each component is discussed in the sections that follow.

4.4 Distributed Multiple Controllers Design

This section provides the main contribution of this research. In this research, multiple controllers are employed, and the network is partitioned into different manageable domains where each controller has its FaToIDM. The design is to achieve a replication mechanism to ensure reliability. This is shown in Figure 4.2, in the DMCs design, controller-to-sensor (C2S) and controller-to-controller (C2C) communication routes are considered. The communication routes can allow a freeway for faults or attacks in a network to be efficiently discovered. Also, this is to allow for effective synchronization of the controller's current states with each other, to ensure network consistency, and to facilitate the takeover of the network control when one controller malfunctions in a certain network domain.

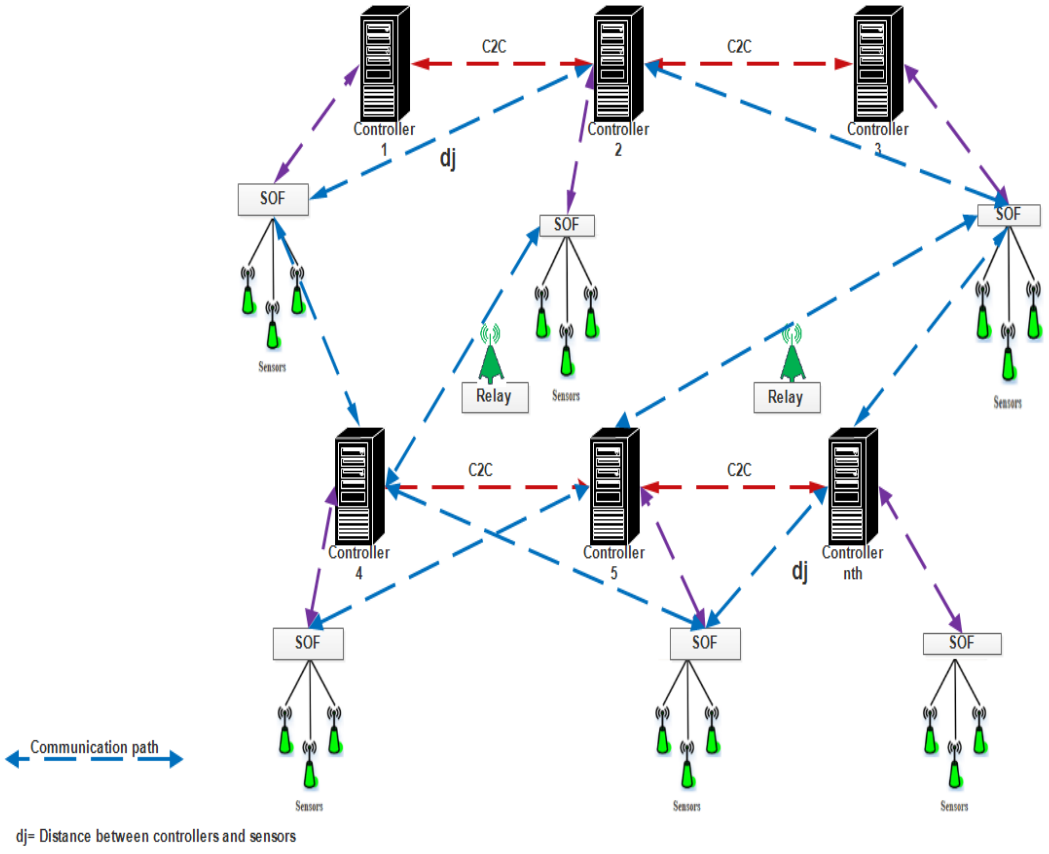

Figure 4.2 SDWSN Network Topology

Figure 4.2 shows the SDWSN network topology which comprises n^{th} controllers and each of them manages the network domain. The controller in each domain communicates with sensor nodes and relay nodes manage to transmit data. The controller performs the task of managing the network and enforcing flow rules, while the sensors perform the tasks of forwarding network traffic. Each sensor node in each domain is connected to multiple controllers of other domains. Moreover, the multiple controllers deployed with FaToM work autonomously for each network domain, and the design illustrates physically distributed but logically centralized controllers which will prevent a single point of failure.

4.4.1 Controller placement

Since this research proposed the use of distributed multiple controllers, effective controller placement must be adopted. The controller placement is specifically to determine the best way to deploy the number of sensor nodes and controllers to maximize reliability and resiliency in the SDWSN model using FaToM while sustaining low latency in the network in terms of communication. In the FaToIDM, DMCs are placed by considering C2C and S2C minimum latency.

To achieve a better C2C or S2C latency in FaToM, the frequency of the inter-controller synchronization needs to be considered. DMC is proposed to have depicted placement in

partitioned SDWSN to allow messages not to relatively travel a long distance across domains. DMCs placement should allow delays to be added to S2C from C2C when evaluating latency perceived at sensors. C2C or S2C communication latency entails controller computation of propagation placement delay between sensor nodes and associating it as latency of the corresponding edge. Thus, based on the assumption, communication between all controllers across domains is routed using the shortest path computation (SPC).

4.4.2 Controller placement algorithm

To effectively place the DMCs and assign the appropriate number of sensors to each, this research proposes the use of SPC. In this novel approach, the network is first modelled as an undirected graph, $\boldsymbol{G = <C, S, E>}$ where C is the set of vertices which are the controllers and S is the set of the sensors while E is the edges which are the connection between the sensors and the controllers measured as the distance with a relation given by $\boldsymbol{S, C}$ is the shortest path latencies between each pair of nodes. For instance, Figure 4.2 can be modelled as a graph shown in Figure 4.3:

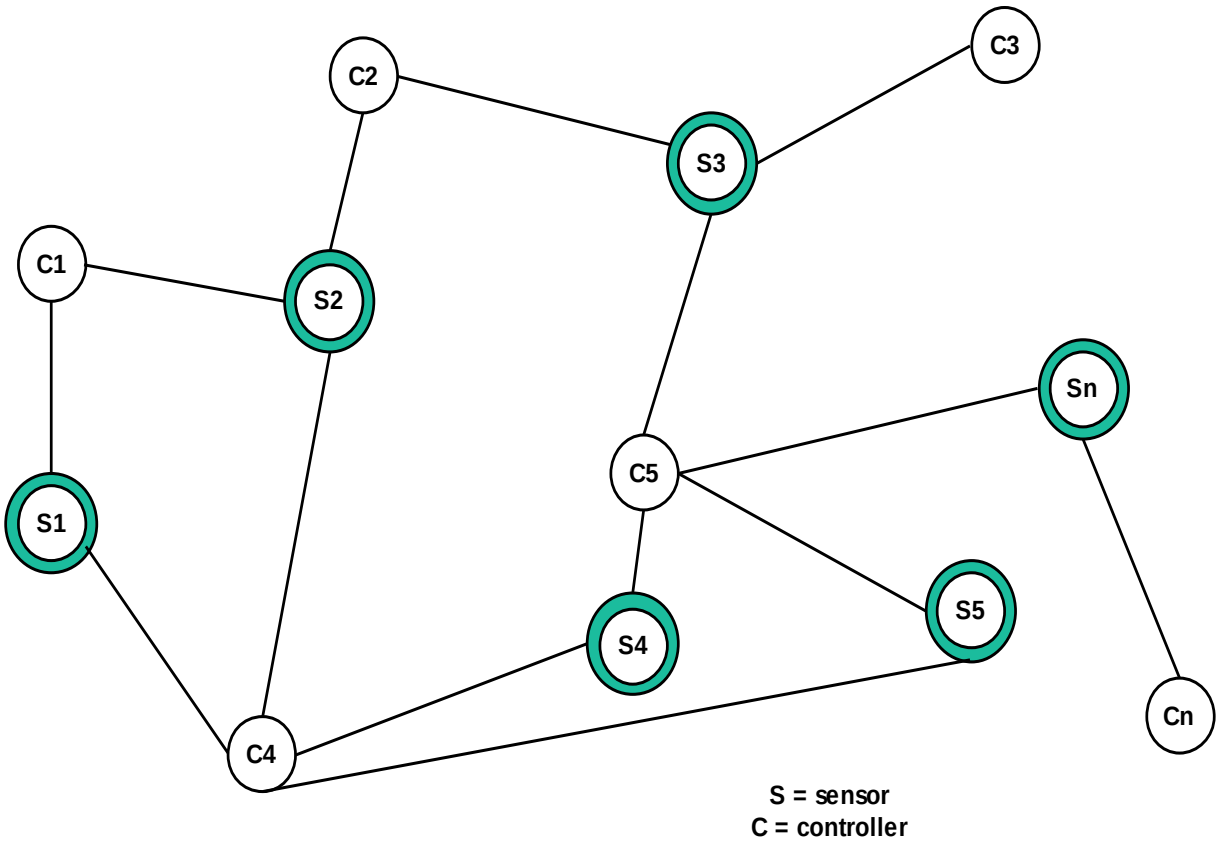

Figure 4.3. Controller-Sensor graph

Where C_i are the controllers and S_i the sensors for i=1, 2…n. However, as shown in Figure 4.3, each controller is connected to sensors while sensors are actively connected to multiple controllers to ensure the reliability of the network. With this connection, when a controller fails, its sensors will automatically be assigned to the nearest controller which is determined by d_{ij} = min (xKm), where x is the value and Km or m refers to kilometre or metre. Also, when the sensors fail, which nearest sensors a controller should be connected to?

In this research, the goal of the controller placement realm is to optimally place controllers with suitable sensors' assignment such that in the event of failure, communication in the network between the control and data plane can continue. Hence, the SPC is adopted using a novel approach that returns the shortest path from the source node to the destination node. With this approach, the shortest path table will be maintained for each S2C placement and will be used for decision-making once a controller is faulty or has failed.

4.5 Fault Tolerance Design

This section outlines the proposed FaToM design and its role in guarding against controller faults in the SDWSN. This ensures that the network continues to be functional and operational in the event of faults. FaToM covers both aspects of faults in both the control and the data plane with a communication link known as the SensorOpenFlow (SOF) or OpenFlow (OPF). Our FT design takes advantage of the network statistics 'REQUEST and REPLY' to detect network faults/failures on the SDWSN model. The FT requirements and design principles followed in designing the novel integrated framework in the SDWSN are discussed as follows:

*A. **Design Principles***

The design of multiple controllers with self-healing capability in distributed control plane architecture is to enhance performance and it assumes a replication mechanism. Thus, design principles are discussed as follows:

1) *Reliability:* Reliability refers to the function of time with conditional probability of how the system will correctly operate. DMCs are deployed to keep consistent communication between controllers where the global state of the network will be achieved. Since this is performed throughout a complete time interval, reliability is achieved through FT. This is a critical principle that will allow DMCs to withstand malicious attacks which can result in severe failures such that the global state SDWSN system can be restored.

2) *Consistency**:*** This is the fundamental principle which acts as physically distributed but logically centralized controllers because the proposed design of DMCs in FaToM will keep the global state of the network. The principle is applied to multiple controllers to generally consider the version update consistency. In this case, this research designs FT to a perspective that consequences based on the performance of the SDWSN system are kept low. Thus, having this principle in FaToM is to enhance synchronization.
3) *Availability:* This is the probability that the network is correctly operating at a specified or given time and is closely related to reliability. In this research, security is the main factor to be considered and this leads to determining availability in the SDWSN system.

4.5.1 Fault types

SDWSN with centralized controllers lacks FT mechanisms against the controller, link failures and even security attacks. Therefore, this section presents the nature of faults and attacks that can affect the SDWSN system:

i. **Controller failures:** Controller failures are caused by faults that affect SDWSN controllers. Consequently, expected faults are normally experienced by centralized controllers. In [3], if a failure occurs, sensor nodes cannot function properly. In the context of this research, a controller fault or failure is considered as the inability of the controller to send *opf_flow_stats_Request* to the SensorOpenFlow or OpenFlow within a specified timestamp.

ii. **Link and node failures:** Link or node failures are discovered by controllers which communicate with sensors, and they affect data flow or communication in the network. In essence, affected data flow interrupts communication between controllers, and sensors can be also interrupted. This research considered link or node failures as the inability of the SensorFlow/OpenFlow to respond with *opf_flow_stats_Reply* to the controller's request for network statistics within a specified timestamp.

iii. **Security attacks:** Security attacks mostly affect both the control and data plane, and this can result to access to sensitive information or data privacy. Generally, the security concept will encompass the protection of SDWSNs and their components. The integration of the FT and IDS model into the SDWSN will have an impact by eliminating unauthorized entries to the network. This research detected security attacks or intrusions based on the analysis of the *opf_flow_stats_Reply* to the controller's request for network statistics.

With the above-discussed types of failures, controllers have been proposed and designed. Once the link or node fault or failure is affected among the DMCs, the latency of the network will be affected. Thus, with the proposed FaToM, robust DMCs will maintain connectivity in each domain.

4.5.2 FaToM

The proposed FaToM's goal is to provide SDWSN with the capability of eliminating the single point of failure with multiple controllers. This is based on active replication which is used to ensure that SDWSN components are continuous in their operation during component failures or faults. The phases of the FaToM are shown in Figure 4.4 and the component design for each phase is discussed in the subsections. In the FaToM, the detection module (DM) aims to detect faults in controllers while the recovery module (RM) is responsible for recovering failures that occurred in controllers, links, or the nodes (sensors). To avoid a single point of failure in controllers of the network if faults are detected, FaToM should activate the RM.

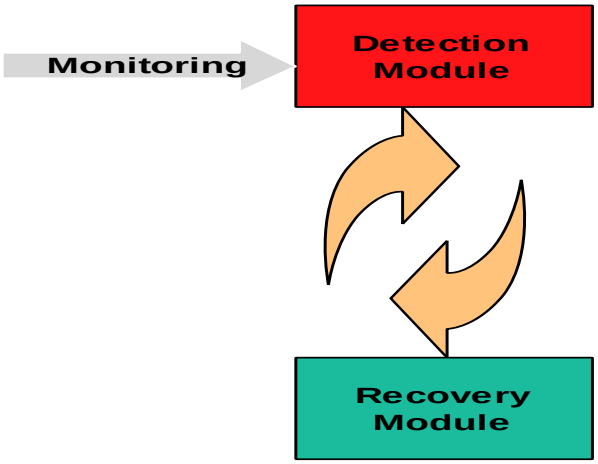

Figure 4.4: Fault tolerance model

A. Detection Module

The detection module (DM) has a component called the DETECTOR that executes the detection strategies in the SDWSN. The components of the DM are the Messenger, Secretary, and the Fault type database (FtypeDB) mechanism as shown in Figure 4.5a and the process or activities involved are shown in Figure 4.5b.

Messenger: This mechanism monitors the internal components and the communication links between controllers and their sensors in each domain, and it also monitors neighbours of external controllers and the IDS. This is to collect information about the DMCs and their sensors across each network domain. For the external controllers and sensors, it collects information based on the placement of the neighbouring nodes. Monitoring involves message exchanges and logging. For an internal controller, external controllers, and their communication link or SOF, will take advantage of the network statistics' collection to detect faults or failures.

The SDWSN controllers collect network statistics by continuously monitoring all the OpenFlow sensors and request all network statistics when needed to provide a global view of the network at a specific time interval. The controller will send *opf_flow_stats_Request* to the OpenFlow sensors after a fixed time window and request the current set of statistics for flows, ports, etc. SOF responds to the request of the controller through the opf*_flow_stats_Reply* by sending the network statistics to the controller.

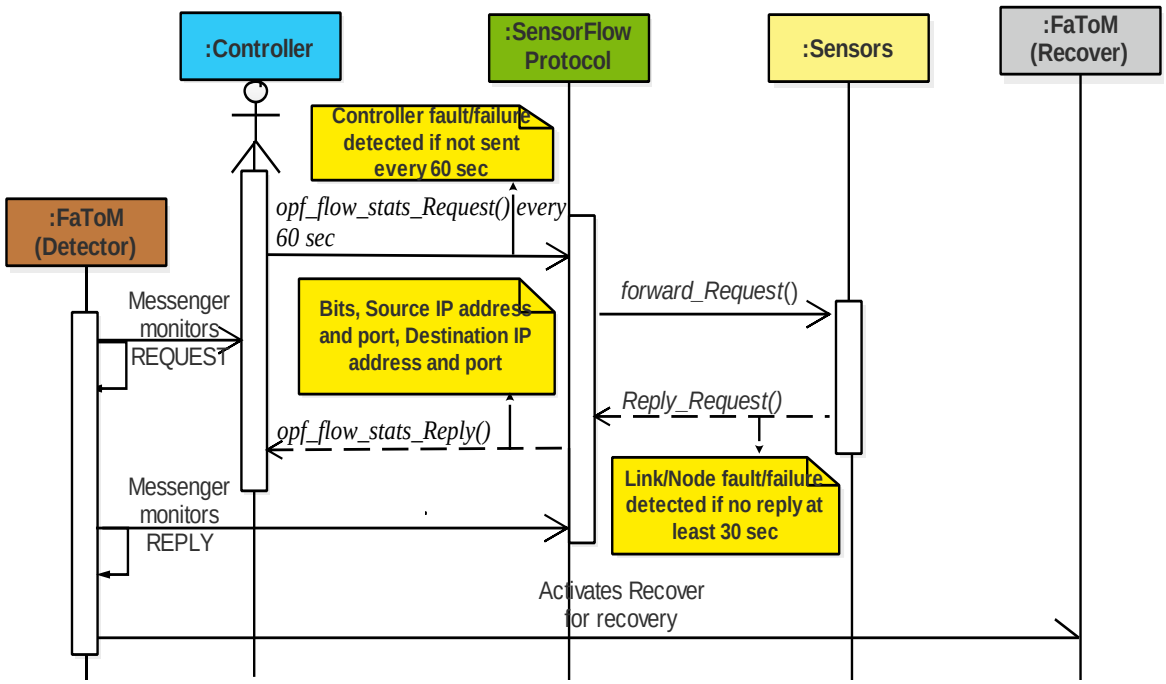

Figure 4.5a: Fault detection process

Given the timestamp allocated, the MESSENGER will then monitor both the *opf_flow_stats_Request* and *opf_flow_stats_Reply* to detect faults after a given time interval.

a) For instance, if the *opf_flow_stats_Request* is scheduled for every 60 seconds, the MESSENGER will declare a fault or failure of the controller if, after 70 seconds, there is no request being sent.
b) For *opf_flow_stats_Reply*, if a response is scheduled for 30 seconds from the OpenFlow for a controller request, a fault or failure of the communication link or sensor nodes is assumed if after 40 seconds no reply or packet is forwarded to the controller.

As the REQUEST or REPLY are executed at the specified interval, the MESSENGER always sends a SUCCESS message to the SECRETARY to log, otherwise, FAILURE is logged. The process involved is shown using a sequence diagram in Figure 4.5a.

Secretary: This component maintains a log of all reports sent by MESSENGER which is either SUCCESS or FAIL for both the controller and the sensors. Table 4.1 shows the design of Figure 4.5a maintained by the SECRETARY. In general, the goal is to facilitate decisions on whether to activate the recovery process or not. In this case, if the report is FAIL, immediately the SECRETARY accesses the FtypeDB to identify the priority of such fault to decide the action to take.

Table 4.1: Secretary log

Time interval (Sec)	Controller/Sensor Node ID	Report	Action
30-60	Controller 1 (S1)-Sensor 2 (S2)	SUCCESS	-
40-70	Controller 1 (C1)-Sensor 2(S2)	FAILURE (C1, S2)	RECOVER
30-60	Controller 1(C1)-Sensor 4 (S4)	SUCCESS	-
40-70	Controller 1 (C1)-Sensor 4(S4)	FAILURE (C1, S4)	RECOVER
30-60	Controller 2 (C2)-Sensor 3(S3)	SUCCESS	-
40-70	Controller 2 (S2)-Sensor 3(S3)	FAILURE (C2, S3)	RECOVER
30-60	Controller 3 (C3)-Sensor 5(S5)	SUCCESS	-
40-70	Controller 3 (C3)-Sensor 5(S5)	FAILURE (C3, S5)	RECOVER
30-60	Controller 4 (C4)-Sensor 5(S5)	SUCCESS	-
40-70	Controller 4 (C4)-Sensor 5(S5)	FAILURE (C4, S5)	RECOVER
30-60	Controller 5 (C5)-Sensor n(Sn)	SUCCESS	-
40-70	Controller 5 (C5)-Sensor n(Sn)	FAILURE (C5, Sn)	RECOVER

1) *Fault type database (FtypeDB):* This is a stable storage that stores all the types of faults or failures for controllers and nodes/links. In this regard, the faults or attacks discovered are based on priority as maintained in stable storage. The priority determines the appropriate actions that can be taken. For instance, as shown in Table 4.2, if the priority for controller fault is LOW, the controller will continue to operate. But if MODERATE, an alert will be given while HIGH will take immediately to avoid failure in the network.

Table 4.2: Priority between controllers and OpenFlow sensors

Failure		Priority	
Controller	H	M	L
Link/Sensor	H	M	L

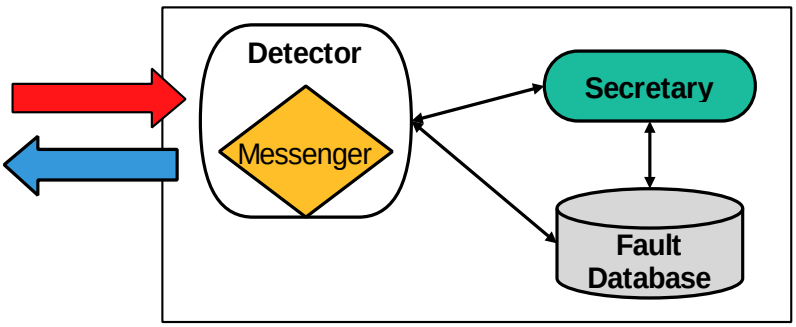

Figure 4.5b: A fault detection module

Algorithm 4.1. Fault detection algorithm

Algorithm 1: Detection

Input: Network statistics requests
Output: SUCCESS or FAILURE
Start
set opf_flow_stats request to every 60 sec
set opf_flow_stats reply to 30 sec
Messenger monitors the *opf_flow_stats_Request/Reply*
if at every 60-sec SDWSN controller sends *opf_flow_stats_Request*
then No Controller_fault/failure → SUCCESS
messenger log report to the secretary
else Controller_fault/failure → FAILURE
messenger log report to the secretary
Detector activate Recover
If SUCCESS
then if opf_flow_stats replies send at least 30 sec
then No Link/Node_fault/failure → SUCCESS
messenger log report to the secretary
else Link/Node_fault/failure → FAILURE
messenger log report to the secretary
Detector activate Recover
End

Figure 4.5b presents the fault detection module (FDM) which represents the architecture of the specified components, and they are explained by sub-section A. Algorithm 4.1 specifies the detection strategy or algorithm in Figure 4.5a and b.

B. Recovery Module

The recovery module has an agent known as the RECOVER, which has the responsibility of executing the recovery strategies of the FaToM in the SDWSN model. The architecture is shown in Figure 4.6 which is the recovery module and the SPC which maintains constant communication with the MESSENGER and SECRETARY. The operations of the components are as follows:

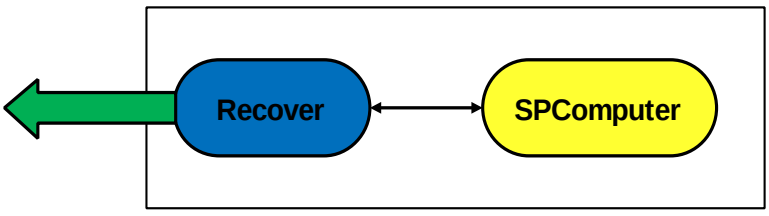

Figure 4.6. Fault recovery module

Shortest paths computer: This module computes the shortest paths of each connection in the network based on the SP algorithm. In this case, it maintains a table, preferably, the adjacency matrix of the G, where the controllers are the rows and the sensors are the columns as shown by Algorithm 4.1, and the shortest path of the controller–sensors connection. The path is

measured in metres (m) or kilometres (km) depending on the proximity. In this case, the network graph modelled in Figure 4.3 is computed in Table 4.3 – C2S distance table.

Table 4.3 is defined as follows: From the related mechanism of Figure 4.6, SPComputer is applied to determine the distance (d_{ij}) between controllers and sensors defined as follows, i represent the x-axis while j is the y-axis. If (d_{ij}) is greater than 1, then sensors are assigned from other controllers, when d_{ij} equals 1 main sensor connected to controllers remains the same. Thus, if d_{ij} is equal ∞, then it shows that sensors are not assigned to controllers.

Table 4.3: C2S Connection distance table

C-S Distance (xm)	**S_1**	**S_2**	**S_3**	**S_4**	**S_5**	**S_n**
C_1	1	6	∞	∞	∞	∞
C_2	∞	1	5	10	∞	∞
C_3	∞	∞	1	∞	∞	∞
C_4	3	8	∞	1	7	∞
C_5	∞	∞	4	2	1	8
C_n	∞	∞	∞	∞	∞	1

Figure 4.7 shows the relationship between Table 4.3, and Figure 4.6 in which we illustrated communication between the SPComputer module, SDWSN controller and SOF. Thus, it simply elaborates how MESSENGER and SECRETARY communicate to outline detection and recovery module mechanisms.

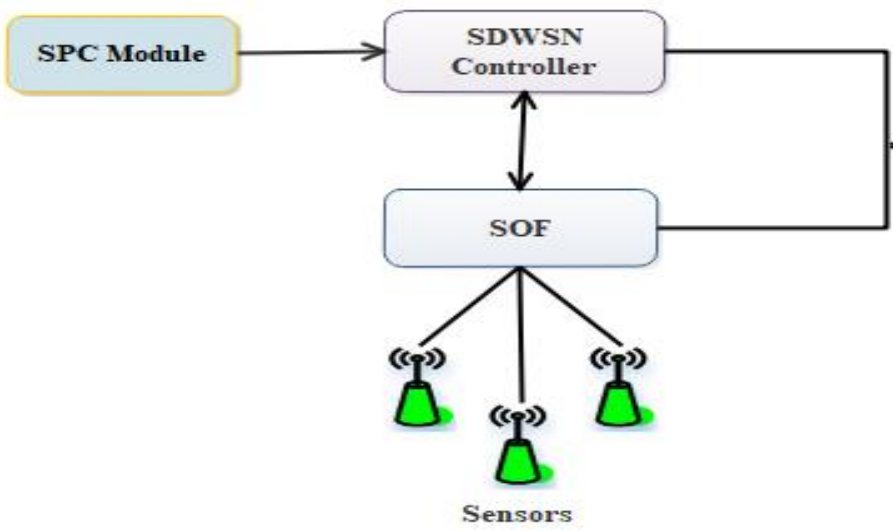

Figure 4.7: Relationship of SPC between SDWSN Controllers and Sensors

The recovery module also maintains constant communication with the MESSENGER and the SECRETARY. With this communication, once “FAILURE” is sent by MESSENGER or

logged in by the SECRETARY, this module immediately signals the SPComputer to compute the current distance of all controllers and sensors in the network. However, before signalling the SPComputer, it first identifies the fault type and the associated priority in the FtypeDB to take accurate and proportional action. Thus, the actions to be taken are as follows:

1. If the **fault type** = controller and priority are HIGH, the OpenFlow sensors are automatically assigned to the controller with which it has the shortest path or adjacent to them.
   For instance, if C_1 is faulty, then the sensors, S_1 will be assigned to:
   S_1_assign = min (dij(C_2,.., C_n))
2. If the **fault type** = sensors or communication link and priority are HIGH, the controller is automatically assigned to the OpenFlow sensors with which it has the shortest path.
   For instance, if C_1 is faulty, then the sensors, S_2 will automatically be assigned to:
   C_1_assign = min (dij (S_2,.., S_n))

The overall algorithm to perform this recovery task is shown by Algorithm 4.2.

Algorithm 4.2: Recovery algorithm

Algorithm 2: Recovery	
Input:	MESSENGER Report, Secretary log and FTypeDB
Output:	Controller or Sensor assignment
Start	
For Messenger report = FAILURE	
Recover check the fault type and priority	
If fault type = controller & priority = HIGH	
then **Recover** identifies the controller, C_k and activates the SPComputer for the distance table	
assign OpenFlow sensors C_{k-1} to controllers adjacent to C_k	
else If fault type = link/sensor node & priority = HIGH	
then Recover identifies the Sensors, S_m and activates the SPComputer for the distance table	
assign OpenFlow sensors S_{m-1} to controllers adjacent → S_k	
The network continues operations as usual	
End	

Once the recovery process is done, control is then passed to the detection module and normal network activities resume. However, this entire process is done in milliseconds to the notice of anyone.

4.6 Intrusion detection design

This section presents the IDM design which is regarded as the last line of defence against intrusions identified. Intrusions or attacks on the controller can either be normal or abnormal traffic packets. Therefore, this design adopts the flow analysis approach, and the intrusion detection model is designed to be a defensive or preventative measure from intrusions or attacks. Thus, IDM will play a role in intensifying security within controllers in SDWSN. Moreover, several architectural requirements were employed to design the model that can

address security challenges in the network. IDM aims to monitor the network and detect any intrusions or attacks; thus, the following requirements are achieved.
Communication: This requirement ensures secure communication between sensors and controllers.
Monitoring: After analysing the network traffic flows, this model continues to monitor packet flows between DMCs through the OpenFlow sensor nodes protocol.

4.6.1 Intrusion Detection Model

The proposed IDM in this research is also referred to as the IDS which applies the same proposed idea used for the FT. Its objective is to provide SDWSN with the capability of identifying intrusions or attacks within DMCs. To achieve this, the main idea is based on flow-based anomaly detection using an ML technique. The ML technique is utilized to classify intrusions or attacks in terms of packet flows within controllers. The design of IDM includes extracted features from the packet header as shown in Table 4.4 and are traffic or packet-flow-based, obtained in the SDWSN environment.

Table 4.4: Extracted features [1-3]

Feature Name	**Feature description**
Duration	Length of connection
Protocol type	Protocol type relies on UDP, TCP etc
Src_bytes	Data bytes from source to destination path
Dst_bytes	Data bytes from the destination to the source path
count	Represents the number of connections to the same host as the current connection in seconds
Srv_count	Represent the number of connections to the same service as the current connection in seconds

In this model, the IDM has been implemented in the SDWSN-based controller. For effectiveness, the IDM has an agent called MONITOR that will allow SDWSN-based controllers to monitor SOF and request network *opf_flow_stats*. This is for IDM to take advantage of the global network overview for identifying intrusions. Therefore, below are the proposed IDM architecture and design principles utilized.

4.6.1.1 IDM architecture and design principles

As per the discussion in the FT part of Chapter 4, this section presents another important part that contributes to this research work. The section also discusses the architecture and design principles associated with the proposed IDM in SDWSN which utilises SDN functionalities to detect and identify attacks and intrusions based on anomaly network flow. However, our design

creates an IDS module integrated into the control plane but outside the controller. In this manner, the reasons are as follows:

1. The aim is to alleviate the SDN-SDWSN's controller from the challenge of monitoring the entire network considering the amount of network traffic transmitted through sensors to detect flow anomalies [2, 112, 132]. However, it is a challenging task for the controller but its intelligence with several important functions needs to be considered. Therefore, SDWSN needs to have a robust IDS that can effectively assist with the task of flow-based anomaly detection.
2. Existing anomaly detection techniques are only based on a centralized SDN-SDWSN controller and are not effective in real-time operation. Our design supports a large-scale network with distributed controllers. In this manner, it can learn the behaviour of attacks and intrusions using the ML algorithm. Therefore, this will facilitate the detection of an anomaly in real-time and identify the types of attacks with high precision and accuracy.

A. Design Principle

In this research, the design of the proposed IDM known as the IDS in the SDWSN environment is based on the network flows-based anomaly detection, and response or mitigation strategy which also takes advantage of the network statistics collection. Therefore, important design principles are discussed as follows:

1. *Automation:* With this, when using the SDN functionalities, it is possible to automate the monitoring and detection of anomalous traffic flow in the entire network of the SDWSN.
2. *Modularity:* To effectively perform the task of flow-based anomaly detection, each component such as traffic features collection, anomaly detector, mitigation and reporting is comprehensively designed with their functions that contribute to the overall IDM success.
3. *Efficient usage of SDWSN functionality:* As discussed above, in this research, we utilize the SDN functionalities to effectively detect and identify flow anomalies. Here, the controller and the OpenFlow or SensorFlow are the core functionalities used. However, the controller manages flow control to devices. It further installs and enforces flow rules, etc while OpenFlow/SensorFlow deals with traffic forwarding, etc.
3. *Monitoring:* Based on the last principle, the network traffic flows are constantly monitored for normal or abnormal behaviour. That is, if abnormal behaviour is observed, mitigation action is taken immediately to identify either the attacker or the attack's target. Therefore, reports are created, and alerts are sent to neighbouring controllers.

B. IDS Architecture for Detecting Anomaly Flows

This subsection presents the proposed IDS that detects network attacks and intrusions based on the anomaly flow mechanism. The architecture is presented in Figure 4.8. Also, as shown in Figure 4.8, the important components are the controller, sensors, Sensor-Flow protocol,

anomaly detection using the RF-based ML algorithm, and Mitigation and Reporting. All these components work together to achieve anomalous flow detection in the SDWSN ecosystem.

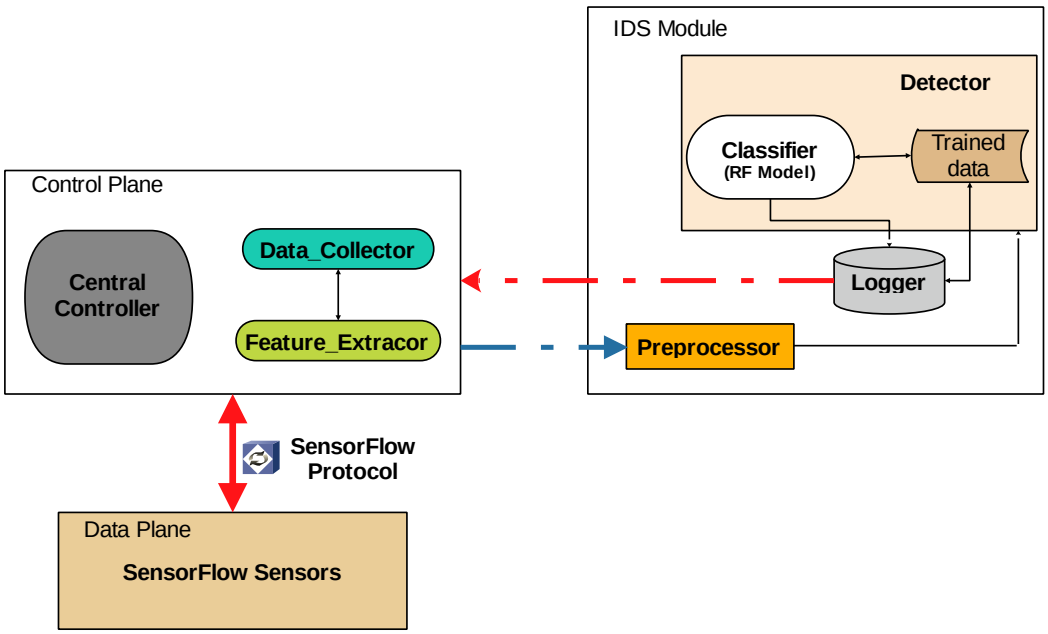

Figure 4.8: Proposed IDS architecture

The different components and their functions are discussed as follows:

A. Sensor-Flow Protocol and Controllers

In these modules, with the usage of the SDN paradigm, the SDWSN sensors are responsible for collecting and forwarding data in the network. According to the SensorFlow protocol, sensors send messages to the controllers regularly. This is to provide network statistics which indicate the present flow status as well as to update the flow table information based on the controller's reports from the IDS module after the detection operation. Below is the discussion of the following operations.

a. *Statistic collector:* In this IDM phase, the operation involves a continuation of the operations discussed above in the FaToM phase of this research. In this case, the controllers are periodically monitored through the statistic collector and send flow statistics messages *(opf_flow_stats_Request)* to the SensorFlow and the sensors [1-3]. Though different studies have proposed different time intervals, in the context of this research, we proposed 20 second time interval. In the FaToM phase, the expiration of the time interval without a response from SensorFlow could signify the presence of faults or failures. However, there is no timestamp response for our IDS. Once the SensorFlow received the REQUEST_Message known as *opf_flow_stats_Request* from the controller, it will immediately respond with a REPLY_Message labelled as *opf_flow_stats_Reply* by sending the network statistics to the controller.

b. *Feature extractor:* As the REPLY_Message is received, the feature extractor then extracts the traffic flow features corresponding to the flow table entries such as bits, packets, source

IP address, destination IP address, source port, destination port, protocols, etc. This collected information is then pre-processed such as classifying, creating a flow feature vector, and forwarding them to the IDS for processing detection of anomalies. Finally, it will provide a timely response to the controller as the results of the detection operation for mitigation action to be taken.

The process involved in the anomaly flows processing is represented by the sequence diagram presented in Figure 4.9.

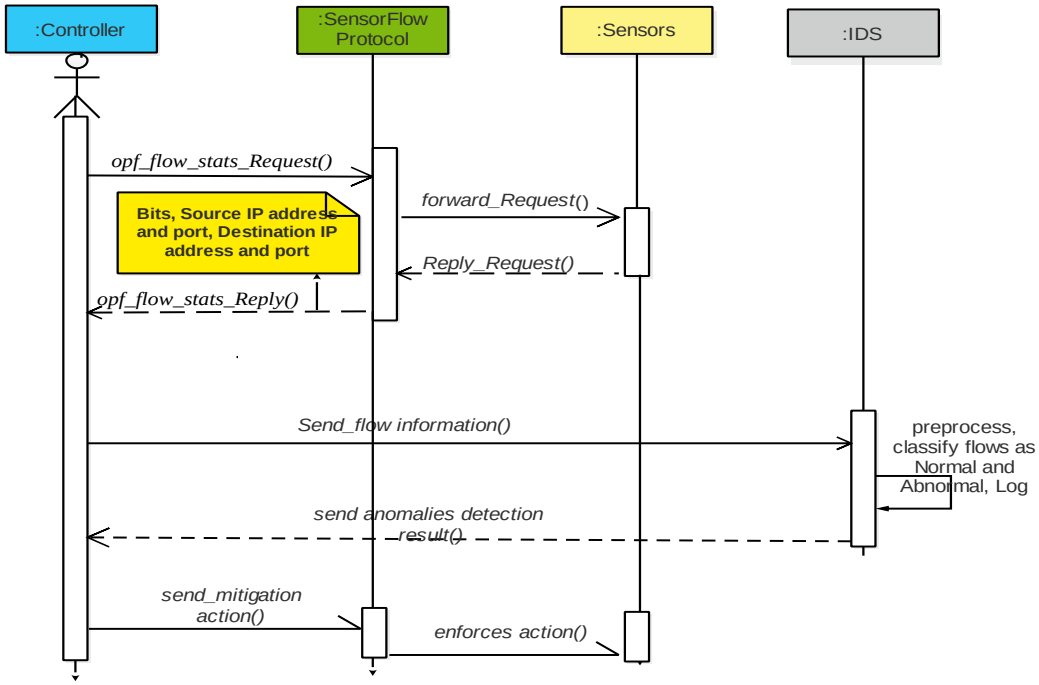

Figure 4.9: Anomaly flows process

B. Anomaly Detection

In this module, the aim is to determine whether the traffic flows collected are not what is expected by analysing the data flow information from the feature extractor and detecting anomalies if present. This module operates in two-fold mechanisms namely: pre-processing and detection.

Pre-processor: This mechanism is used to ensure that the flow features collected from the controller are standard and normalized with each created vector having its features. Also, the previously collected flow features are used to train the detector to learn attacks such as TCP, UDP flooding, etc.

Detector: On completion of the pre-processing operation, the pre-processed flow features' information is forwarded to the detector mechanism for anomaly detection. Here, the detector utilizes the RF algorithm to classify the SDWSN traffic flows. It predicts classified features and when the final identification result is determined by voting of single decision tree output is made. According to [75, 133, 134], RF has good tolerance regarding anomaly values and

good scalability based on the K decision tree (DT) where each selects which class a given independent variable X belongs to. Thus, K is the number of decision trees contained in RF, while θ_k represents independent and identically distributed random vector variables. Only one selection is given to the most correct class and K DT as computed in Equation 4.1. The core of RF is based on the following: The mechanism of random repeat sampling of features is applied to extract K samples randomly from original training data and then K classification regression trees are extracted. Secondly, it assumes that the original training data have n and m features that are randomly selected at each sensor node of each tree ($m \leq n$); while m is used for the amount of information with the most classification ability in each feature. Thirdly, each tree will grow to a maximum level without cutting. Generally, the similarity and correlation of DT are significant features of RF to reflect generalization performance. However, generalization error reflects generalization of the system and applies in Equation 4.2.

$$h(X, \theta_k, k) = 1, 2 \ldots\ldots, K \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(Equation 4.1)}$$

$$PE^* = P_{X,\gamma}(mr(_{X,\gamma}) < 0) \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(Equation 4.2)}$$

Where in Equation 4.2, PE^* represents the generalization error, values $_{X,\gamma}$ defines probability space and $mr(_{X,\gamma})$ is the margin function. Once a result by the RF algorithm is produced, the detector module produces anomaly reports accordingly which are sent to the controller as feedback. In the report, if malicious flows are detected by the RF algorithm, the SDWSN controller(s) is/are warned immediately with an "ANOMALY" report about the attack, which needs the controller to take appropriate mitigation action. The attack type along with the IP addresses and ports involved are included in the report and are used to enforce mitigation actions at the data plane which include dropping, forwarding, and modifying packet fields. The detection algorithm is shown by Algorithm 4.3.

Logger: This sub-module keeps logs of all the attacks detected by the IDS module which are stored as historical data. This data is important for training the ML algorithm in terms of new attacks profile and for effective learnability as accurate classification is critical or dependent on big data or the size of the dataset.

C. Mitigation Action

This stage is purely providing a timely response to counter the malicious event since anomalies in the network can harm the network and have adverse effects on the QoS. This is the resiliency aspect of this research where the network must be effectively managed in the face of attacks or intrusions and reconfiguring the mechanisms for detecting and restoring the network infrastructure the same way as FaToM. However, as the network is automatically and constantly monitored and traffic flow is classified to detect and categorize attacks, a mechanism for repairing such threats in the network is highly important.

In the context of this research, we proposed an approach that should be taken to ensure that the network components and flows caused or impacted by the anomalies enforce new policies. Policies are the set of rules and actions that guide what decision should be taken whenever attacks or intrusions occur in the SDWSN. Moreover, the rules are conditions set, matching the

attacks or intrusions that happened while the action taken is critical for identifying the flows and components involved or targeted. In the proposed IDM, the policy to be enforced is to mitigate flow anomalies in the SDWSN network once detected. Since the Sensor-Flow is flow based, each sensor maintains a flow table which is dynamically changed by the controller and consists of the flow entries that establish how the packets are handled [1,2]. In this case, flow entries which contain the ports and IP address reported by the IDS are injected into the flow tables of the sensors together with actions that should be enforced on the incoming packets matching these flows. The actions are three-fold namely: *Forward*, *Drop* or *Modify* the packet fields [1-3].

Therefore, when anomalies are detected on the network, actions are taken to enforce the following policies just like [1,2]. If more flow entries matches are found, prioritization based on the highest degree of match of other active entries in the flow table will be applied [1] as shown below:

i. ***block_src_IP_Addrs*** will block the traffic of the IP address where the anomalies were detected or simply the malicious host.
ii. ***block_tff_flow*** stops the communication with the identified host to some services as indicated in the IP address of the destinations.
iii. ***L_balance*** is the load balancing action taken and this involves a distributed SDWSN controller. In this case, traffic loads are distributed to the neighbouring controllers to ensure good quality of service (QoS) and performance during this type of event. This is important especially when attacks such as DDoS are involved and ensures the high availability of the controllers the same as FaToM.
iv. ***broadcast_event*** activates the sending of a message or notification to the neighbouring controllers with the identified traffic flows and the mitigation action taken. This will invoke the FaToM's shortest path table to be used as shown in Figure 4.11. This is important to enable them to update their flow tables and avoid harming the entire network.

As shown in Figure 4.10, once anomalies are detected in the SDWSN, based on the shortest path computation, alerts or notification is sent to all the neighbouring controllers or node. Here, we assumed that the controller with ID C4 detected anomalies and immediately sends alert information to neighbouring controllers such as C1, C2 and C5 to update their flow rule and thwart any such anomalies under their network domain. This is shown in Figure 4.10 with the red-dotted lines. This is also applicable to load balancing in the network. As shown in Figure 4.10, the blue-dotted line shows controller C4 load being distributed to C1, C2 and C5 during the attack detection period to ensure good QoS and network performance to the end users.

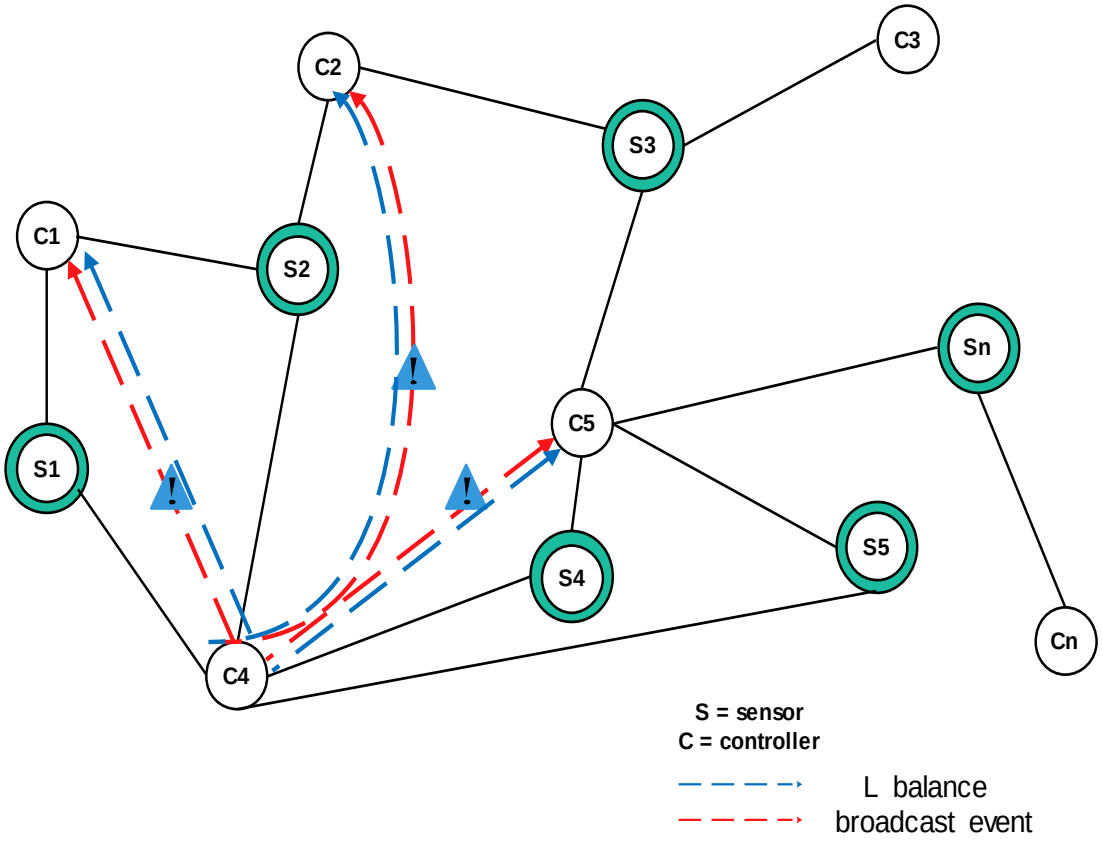

Figure 4.10: SDWSN Controller-Sensor graph with intrusion alert

With these policies in place, this module forces them to the sensors using the Sensor-Flow through the controllers which oversee forwarding the logic of the network devices.

D. Reporting

This module reports the event to the network administrator to assist establish appropriate solutions to the problems as well as to create a log of all detected anomalies or events as well as attacks profile in the SDWSN. The report shows the network resources and components affected by the detected anomalies. To this end, if anomalies such as attacks and intrusions are detected and mitigated, they will be included in the abnormal traffic profile. This will also offer important information that will help network administrators design new network policies for ensuring QoS to the network users.

Algorithm 4.3. Intrusion detection algorithm

Algorithm 3: Intrusion Detection
Input: Anomaly detection using network statistics **Output:** NORMAL/ANOMALY Start Controller sends *opf_flow_stats_request* at a given time interval to OpenFlow protocol OpenFlow protocol reply with *opf_flow_stats_reply* with network statistics. Controllers receive collected network statistics and extract the important features. Forward extracted features to Detection Module for pre-processing and anomaly detection. RF ML algorithm to classify pre-processed features into Normal and Anomaly If the classification report is Anomaly Then the controller is alerted and mitigation action is activated (***block_src_IP_Addrs, block_tff_flow, L_balance, broadcast_event***) Else do nothing End

4.6.2. Justification for Random Forest ML algorithm

This subsection comprehensively presents the ML algorithm chosen for this research and applies it to intrusion detection framework design. To select the appropriate ML algorithm, we carried out an experimental analysis of four ML algorithms: SVM, LR, NB and RF models. The methodology employed is two-fold: the training and testing of the models considered. During the training of the models, the NSL-KDD dataset [11, 135, 136] is fitted to the models to determine their performances on accuracy and other important metrics used. The testing of the models was applied to choose the best ML model that performed well in terms of the metrics considered. This was important to select the appropriate model used in building efficient and robust IDS for the SDWSN environment.

The stages involved are shown in Figure 4.11 and are discussed as follows:

a. Pre-processing data: we focused on processing data, and this is done by eliminating unnecessary features. In essence, this is to reduce the size of data so that it can fit and enhance the ML model's performance.
b. Data classification involves training data for selected ML algorithms with normal or anomaly and processed data have 125973 instances: 103429 for training and 25195 for testing. In general, the training dataset contains 75583 data points with 6 features as shown in Table 4.6.
c. Predictive evaluation and analysis: In this last stage, an intrusion was predicted using the four selected ML models done through classification and testing. For the analysis leading to the effectiveness of the four models, only one best-performing model was chosen to construct the IDM.

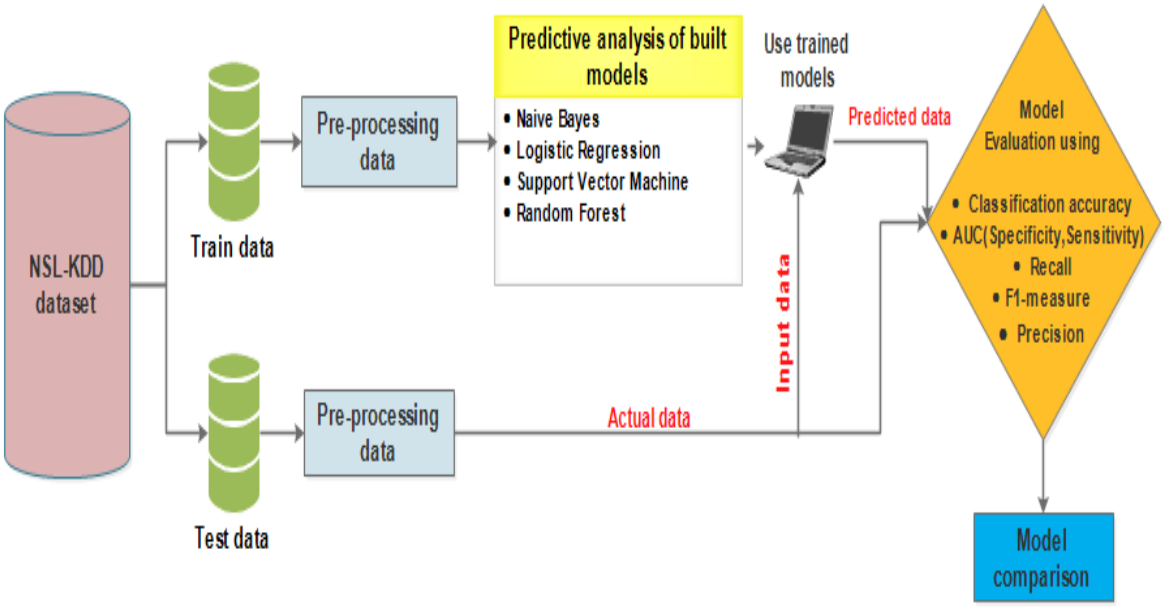

Figure 4.11: Classification method

A. Dataset

To achieve our research objective, the NSL-KDD dataset [11, 135, 136] was utilized for training and testing the models. Based on the literature review done in Chapter 2 of this research, it is proven that the data used is reliable and efficient to develop the ID model. Moreover, several research works have utilised this data set for detection purposes in IDS. In [76], the authors tested their IDS using the NSL-KDD dataset on the decision tree (DT), NB, LR and neural networks and promising results were obtained. Additionally, in this research, the NSL-KDD dataset is chosen to classify six features with 42 attributes as either normal traffic or an anomaly.

Table 4.5: Train and test time

Model	**10-Fold**	
	Train time(s)	**Test time(s)**
SVM	0.12	0.8
LR	0.04	0.3
RF	0.01	0.6
NB	0.06	0.5

B. Time Efficiency and accuracy

This section presents train and test times based on the selected models as shown in Table 4.5. Furthermore, the efficiency of the model is based on classification accuracy. Based on Table 4.5, SVM has the lowest training time of 0.12 secs and testing time of 0.8 secs. RF has a higher training time of 0.01 secs and test time of 0.6 secs, NB has 0.06 training time and 0.5 testing time, and LR has a training time of 0.04 secs and testing time of 0.3 secs. In essence, this indicates that RF is the fastest model in terms of scoring or testing time and is seconded by LR.

Therefore, RF is timeously efficient compared to the other three models and is regarded as the best-performing model used for the experiment.

Table 4.6: Model classification accuracy rate

Model	Correct instance (%)	Incorrect instance (%)
SVM	**76**	**24**
RF	**99**	**1.0**
NB	**83**	**17**
LR	**92**	**8.0**

The results shown in Table 4.6 and Figure 4.11 indicates that RF was the best-performing model compared to other models. That is, RF has an output of 99.0 % accuracy compared to LR, SVM and NB. Thus, the indication is that the RF model is the best-performing model and is considered a good predictor for predicting network anomalies as compared to the other three ML models.

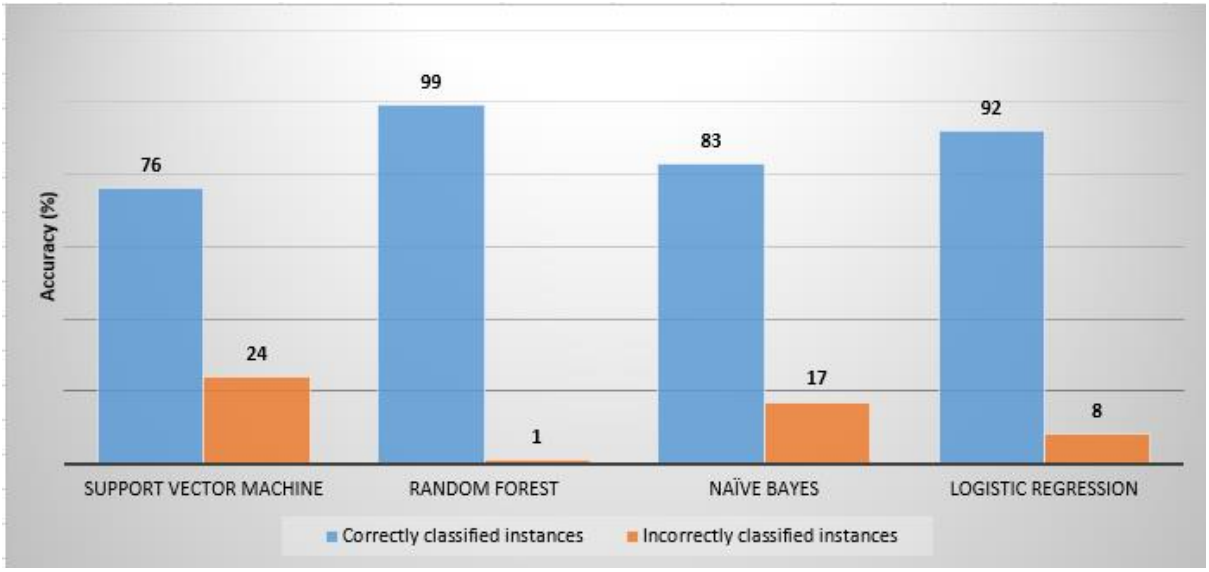

Figure 4.12: ML models accuracy rate

C. Receiver operating characteristic

This section presents the quality of classified ML models in terms of FPR, and TPR with ROC curve graphs for each model. The use of this analysis tool is based on performance evaluation and cost-sensitive learning. In this research, the analysis methodology is used for evaluation and comparison to select the best-performing ML model. Therefore, based on their predictive performance analysed, its graph is shown and explained in Figure 4.13 as follows:

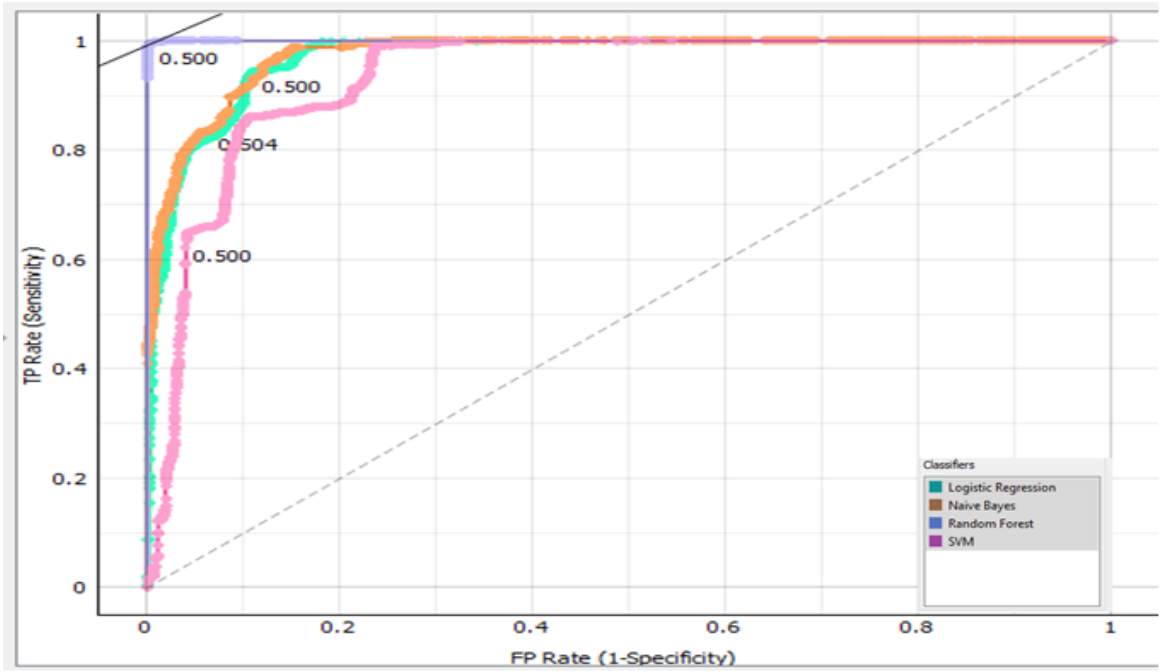

Figure 4.13 ROC graph for ML models

The ROC curve is defined as a plot of the TPR (Sensitivity) against FPR (Specificity) for different possible cut points based on the diagnostic test. However, this graph or ROC curve demonstrates a trade-off between specificity and sensitivity. According to Figure 4.13, the closer the curve follows the left-hand border and then the top border of ROC space as shown by selected models, the more accurate the classification or test. Also, the slope of the tangent at the cut-point determines the ratio for the value of classification or test. The ROC graph shown in Figure 4.13 is determined after training and testing NB (brownish line), LR (greenish line), SVM (purple line), and RF (blue line) ML models with different classification thresholds while the area under the curve measured accuracy. In contrast, AUC provides information based on used ML models to determine the efficiency of the ROC graph. According to Figure 4.13, classification thresholds of LR with 0.500 TPR and SVM with 0.504 TPR are higher compared to RF with 0.500 TPR and NB with 0.500 TPR. Therefore, this shows that the RF model is more efficient in terms of performance compared to other models and is the best in terms of detecting intrusions.

4.7 Proposed SDWSN Architecture

The SDWSN work follows three important layers namely control, application and physical layer and they are similar to a conventional SDN work. However, the physical layer is comprised of communication, sensing and computation resources and the control plane is mainly responsible for data transmission [137]. The core contribution of this research is developing a novel model that presents an integrated system design. This model has the capability of guarding against controller faults and intrusions. In essence, it will trigger proper defence actions to countermeasure the identified challenges in the SDWSN. This section presents the proposed design of a fault and intrusion-tolerant system in an SDWSN environment. Finally, this model is designed to meet the stated research goal in Chapter 1.

Therefore, it will ensure SDWSN security, data confidentiality, and integrity. Figure 4.14 shows the SDWSN conceptual model.

In our model, Wireshark (FaToM) and ML technique (IDM) are incorporated alongside used modules and this is to secure SDWSN and DMCs. The aim of designing FaToIDM is to critically reduce security attacks such as faults and intrusions from the control, data, and application plane. The designed DMCs in this model will monitor SDWSN, while FaToIDM aims to communicate and alert the system if there are faults or attacks, this is also to prevent them. In essence, the proposed FaToIDM is a hybrid approach that deals with FT and ID challenges. In a nutshell, FT deals with faults targeting the controllers while ID deals with intrusions from both data and control plane. Thus, using SPC will be an advantage for fault recovery in controllers leading to a self-healing process and by using an RF-based ML algorithm *opf_flow_stats* will be assessed to identify or detect intrusion. However, the anomaly detection technique is chosen as a support system for both normal and anomaly attack behaviour. Therefore, the proposed self-healing and ID model referred to as FatoIDM has the following important components: FatoM and IDM-IDS.

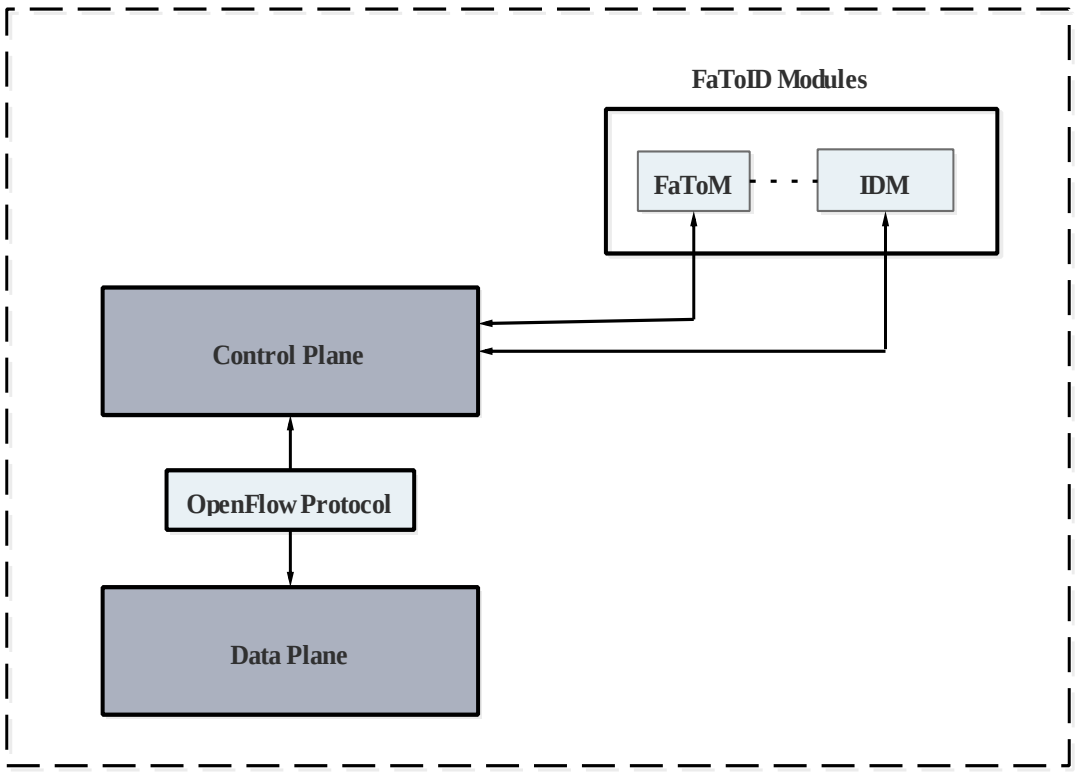

Figure 4.14: SDWSN conceptual model

The SDWSN architecture follows three important layers namely control, application and physical layer and they are similar to a conventional SDN work. However, the physical layer is comprised of communication, sensing and computation resources and it is the control plane mainly responsible for data transmission.

- What process was followed?: The design process followed constructive design with reference from literature studies done in Chapter 2.

- How were the key components derived and formulated?: However, key results informed the design including FaToM comprised of a detection module using a fault detection module as shown by Figure 4.5 b and a recovery module using a fault recover module as shown by Figure 4.6 while IDS M consists of a detector, logger and pre-processor.

- What key results informed the design?: Based on this research work two novel components namely the tolerance module (FaToM) and intrusion detection module (IDM) were derived and formulated from Fault Tolerance and intrusion aspect and disjointed to a new SDWSN conceptual model which illustrates the whole architectural design.

Therefore, based on findings, solutions that provide FT and ID are separated and disjointed, which means FT cannot detect faults emanating from security attacks and threats. Likewise, IDS cannot detect faults emanating from system components and having these solutions separately is considered not cost-effective. Therefore, this research designed and implemented an integrated model that incorporates FT and ID to ensure resiliency and reliability in the network. The model combines FT and ID aspects by incorporating security into FT to eliminate faults and intrusions in the SDWSN.

4.7.1 SDWSN system operations

This section outlines operations as well as the imposed constraints on the execution. These specific operations and constraints are important to satisfy all stated objectives to answer outlined RQs. Therefore, this system operation has requirements given to the integrated model (FatoID). System requirements are important when designing a system, this is to ensure that the whole system is not compromised. In essence, system requirements are considered as an overall or detailed collection providing insight into how the proposed SDWSN-based self-healing and intrusion detection model should operate. This is concerning solving the stated problem as discussed in Chapter 1 of this research. For the proposed system to be fully operational, the system design as shown in Figure 4.15 deployed FaToM and IDM. However, under FaToM, the system initiates a detection module using a fault detection algorithm to identify faults. Furthermore, it initiates a recovery module to recover them using a recovery algorithm.

a. If the detector agent in the detection module executes strategies, the MESSENGER mechanism will monitor *opf_flow_stats* requests and replies.
b. If there is a failure identified in the controller, the self-healing capability will be applied to FT.

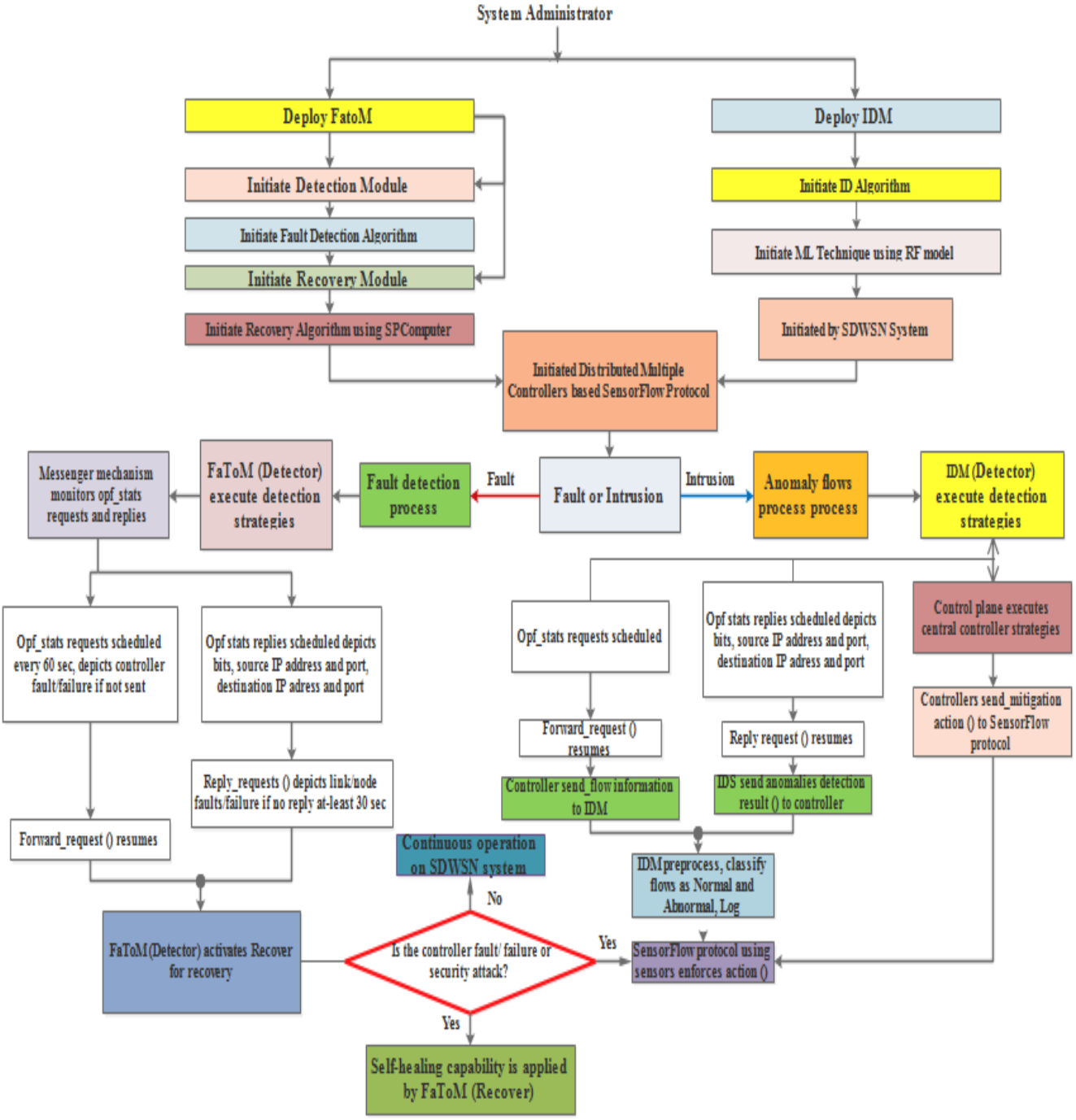

Figure 4.15: SDWSN operations

Moreover, under the IDM, the system initiates IDS using the RF ML algorithm to detect or identify intrusions or attacks.

a. If an anomaly is reported to controllers that will impact a security attack in the SDWSN, then IDS need to take security measures.
b. If controllers experience normality of packets or traffic, then the SDWSN network will continue to be operational.

Therefore, system operations are carried out to have a robust controller which will maintain a global view of the SDWSN.

4.8 Chapter Summary

In this chapter, we presented the design of the proposed integrated FaToID model, however under these two disjointed phases. The first phase of the FaToM aimed to identify controller faults or failures; while the second phase under the IDM, the aim is to detect intrusions or attacks in SDWSN. Moreover, among FaToM, a Wireshark packet or traffic analyser was utilized to capture packets or traffic. In essence, with captured traffic, under IDM we used RF, LR, SVM and NB ML algorithms to check the best-performing model in terms of detecting intrusions in SDWSN. Therefore, for a secure system, the RF model outperformed the others and was selected as best in terms of performance and accuracy.

Chapter 5

Evaluation and Results

5.1 Chapter Outline

This Chapter presents the implementation of the proposed system, the results obtained, and the theoretical evaluation of the proposed self-healing and intrusion detection model discussed in Chapter 4 with existing models. It starts with an introduction, followed by an experimental setup, and progresses to the results and analysis of the simulated data and finally, theoretical comparisons.

5.2 Introduction

This section presents the results of the experiment performed to determine the mechanisms to avoid a single point of failure in the SDWSN controllers and to detect faults and intrusions identified. The experiment was performed based on the methodology outlined in Chapter 3 and the framework designed in Chapter 4 of this research. The researcher started by designing a virtual network topology with multiple controllers and employed Wireshark to analyse the traffic flows. As discussed in Chapter 4, the proposed FaToM's function is to check if there are any faults identified. For the proposed IDM, the goal was to detect spurious intrusions or attacks on the network. In this case, we utilized the NSL-KDD dataset [69] to train and test for ML algorithms: SVM, NB, RF and LR, and to identify the best-performing model in terms of classification accuracy, efficiency, etc. However, due to time constraints, the designed FaToIDM was not fully implemented as an integrated model. We only evaluated the performance of each component: FT and IDS.

For FT, the goal was to evaluate FaToM's impact on the network. Thus, the performance and effectiveness of FaToM were measured via network throughput and latency or delay. The IDM's performance in terms of the selected RF model was evaluated based on accuracy, precisions, recall, F1-score, AUC, etc. The aim was to utilize a fault-intrusion mechanism to minimize failure among proposed controllers in the SDWSN. In any based SDN-WSN computer system, fault tolerance and intrusion or attacks are critical components. As SDWSNs adopt SDN and WSN paradigms, fault tolerance is crucial since the network is made up of sensor nodes. If controller-based FT and intrusion detection frameworks are not well designed and implemented, this will lead to a single point of failure. Therefore, we have proposed a good design and implementation that addresses this problem aligned with the research objectives stated in Chapter 1. After running simulations, the generated data was collected, analysed, and evaluated both quantitatively and qualitatively.

5.3 Results and Analysis

This section presents obtained experimental results performed in FaToIDM separately as FaToM and IDM. It further presents the process involved in simulations performed to evaluate the effectiveness and performance of the integrated SDWSN model designed in Chapter 4. Subsections 5.4.1 and 5.4.2 outline simulation results.

Based on the results obtained in both FaToM and IDM, a comparative analysis was done. According to FaToM, the aim was to prevent the centralized controller from being a single point of failure due to faults and intrusions. The mechanisms that can lead to the deployment of logically centralized but physically distributed controllers were designed. In this context, to analyse the network handling capability of proposed controllers, the evaluation metrics specified in Table 5.3 were considered and we varied the network topology size by adding 12 hosts separately so that readings can be noted. Thus, the traffic was analysed by the Wireshark tool and ping commands and similar network types were designed to perform packet transmission under the control of default, and floodlight controllers. Moreover, to consistently keep a functional network latency or average delay, the controller does not need to be on the HIGH mode. In this case, throughput was used to measure the transmitted data due to traffic flows. In essence, the RF-based ML model performed very well among other algorithms, and it was modelled using the orange tool in the DDoS dataset [138]. Thus, the simulation results for the FaToIDM were performed in Linux-Ubuntu 18.04.5 LTS, 64-bit operating system and further discussion is outlined in sub-section 5.4.1 and 5.4.2.

The sub-section that follows presents the actual results of the FaToM identification using Wireshark and IDM; using the RF model as discussed in Chapter 4. Under the FaToM phase, we used network latency and throughput as metrics, while under the IDM phase, the RF-based ML model was used to check if it was able to detect intrusions or attacks.

5.3.1 FaToM performance

The performance of our proposed fault-tolerant controllers' mechanism was implemented in the proposed FaToM and evaluated using Wireshark in Mininet. Results were generated for throughput and latency/delay metrics using analysed traffic flows in the network as shown in Figure 5.1. Moreover, this approach was evaluated with 3 controllers (pox, default, and floodlight) connected to 9 sensors as shown in Tables 5.4 and 5.5.

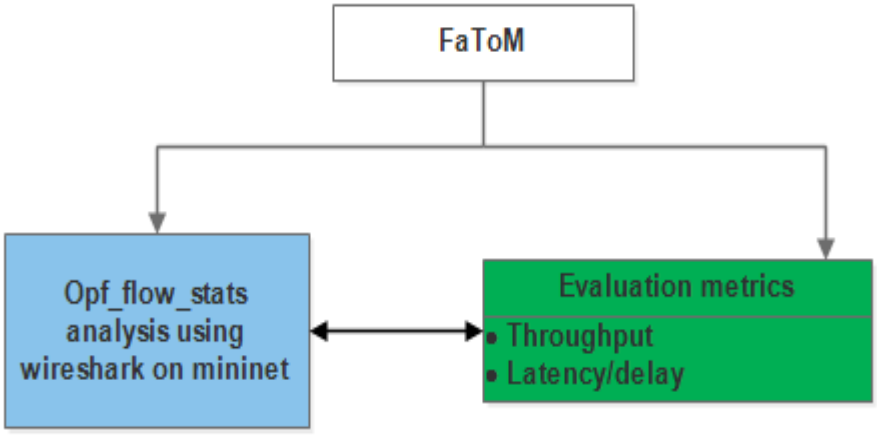

Figure 5.1: FaToM evaluation process

However, based on the policy or rules and actions highlighted, results for the FaToM were obtained in alignment with the following hypothesis:

i. If the priority for controller fault is LOW, the controller will continue to operate.
ii. If the priority for controller fault is MODERATE, an alert will be given.
iii. If the priority for controller fault is HIGH, action will be taken immediately to avoid failure in the network.

A. **Network latency -** The latency measures interval fault errors between inputs or outputs of transmitted data. That is, the aim of reducing network latency will help to detect failures or faults at an early stage. However, this will also help to enhance the SDWSN lifetime. According to the FaToM results, it is discovered that the floodlight controller is 2 times faster in terms of identifying faults by showing better results compared to default and Pox. Moreover, nine test input time intervals are chosen for both controllers. Therefore, Table 5.4 and Figure 5.1 summarize the output time interval and time delay fault error.

Table 5.1 Network latency output data

Controller (s)	No of nodes	Latency output time interval (s)
Pox	24	36.0
OpenFlow (Default)		35.0
Floodlight		34.0
Pox	24	38.0
OpenFlow (Default)		40.0
Floodlight		35.0
Pox	24	42.0
OpenFlow (Default)		45.0
Floodlight		37.0

Figure 5.2 shows that the delay or latency of Pox and the default controller is higher compared to the floodlight controller. Thus, the floodlight controller is more robust and can withstand faults.

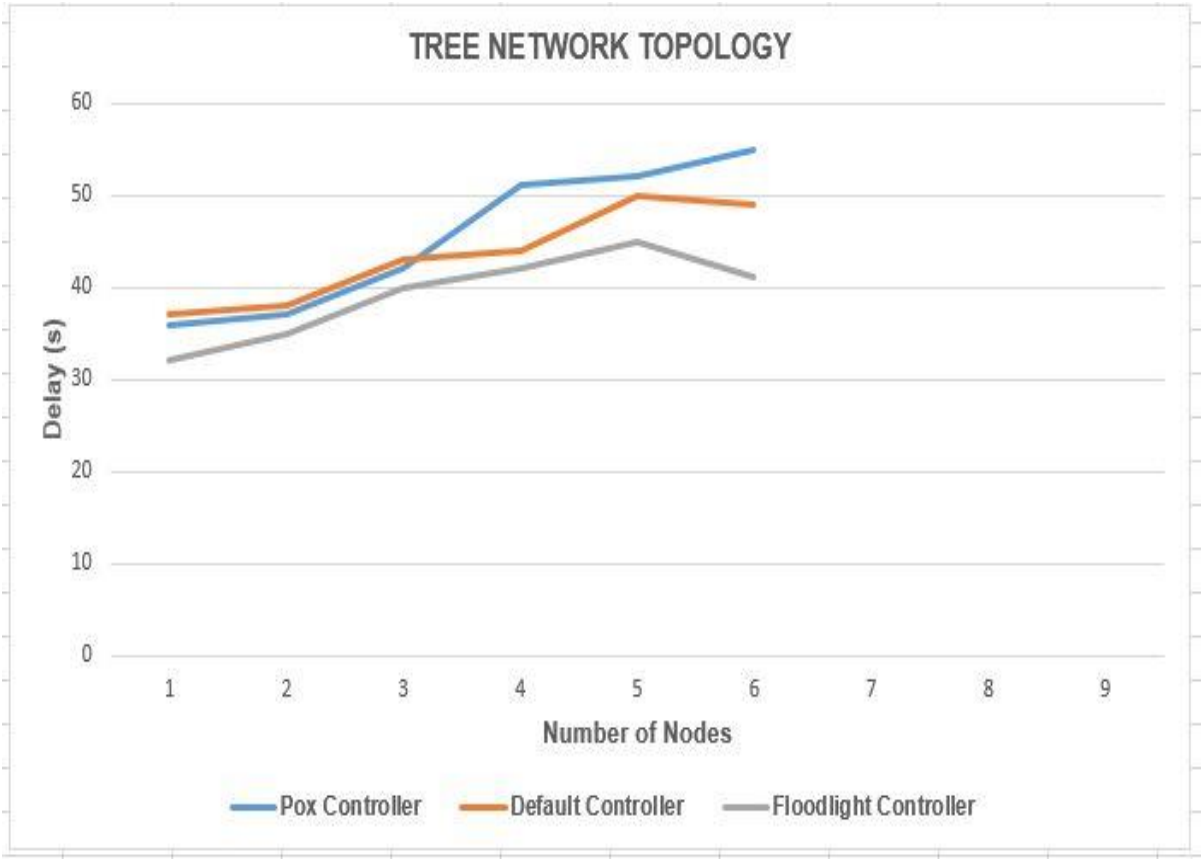

Figure 5.2: Network Latency

B. **Network throughput –** In essence, FaToM will decrease throughput to violate the high-fidelity level required by applications from the application plane. However, the goal is to sufficiently exert the failing controller or sensor node by alleviating packet congestion, this is to improve throughput in the SDWSN. The throughput was tested on the fixed-selected controllers as shown in Figure 5.2 versus the 24 sensor nodes. In addition, six test input data setups were chosen, and test data was transmitted. Therefore, Table 5.2 and Figure 5.3 summarize results for throughput measured in three controllers which are both slightly different in terms of faults errors.

Table 5.2 Throughput output data

Controller (s)	No of nodes	Throughput output data (kbps)
Pox OpenFlow (Default) Floodlight	24	3.00 6.00 8.00
Pox OpenFlow (Default) Floodlight	24	7.52 12.6 40.70
Pox OpenFlow (Default) Floodlight	24	14.80 24.50 81.00

According to Figure 5.2, the flow rate denotes that according to packets captured using the Wireshark. Both floodlight and pox controllers outperform the default controller.

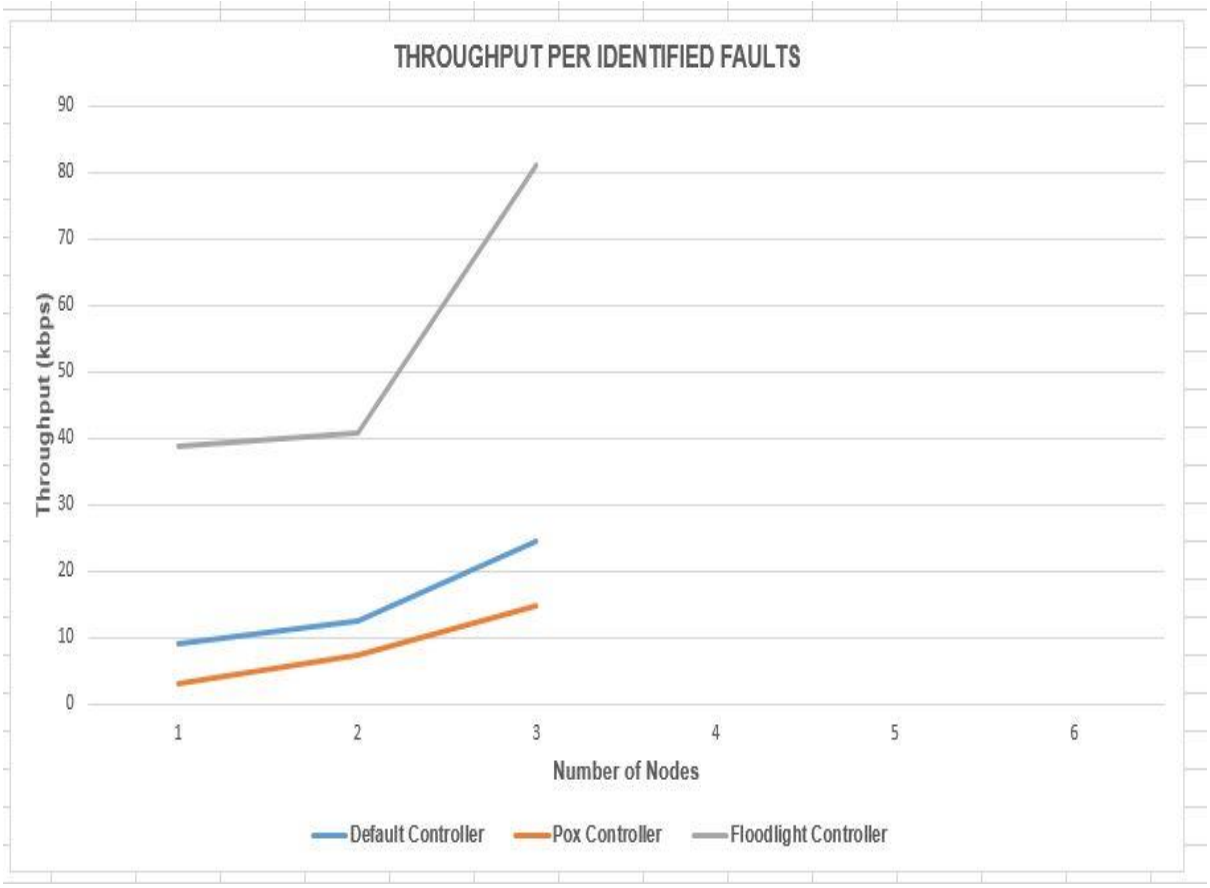

Figure 5.3: Network throughput

According to Figures 5.2 and 5.3, the throughput of the floodlight controller is higher at 8.00, 40.70 and 81.00 kbps compared to the pox and default controller. Therefore, results imply that with 80 % coverage for FT, controllers are exchanging operations for a certain period to avoid a single point of failure.

5.3.2 IDM performance

This sub-section presents obtained IDM results using the rapid miner tool with the best-selected ML algorithm, i.e., the RF model. We initially employed Wireshark to capture and analyse the traffic or packet flows for intrusions. However, no attacks were found to enable the evaluation of the IDM performance and effectiveness. Therefore, due to time constraints and other resources, we applied the CIC-DDoS2019 dataset [138] to evaluate the RF model's effectiveness to determine its capability of detecting intrusions or attacks. Figure 5.4 shows the process involved in obtaining the results.

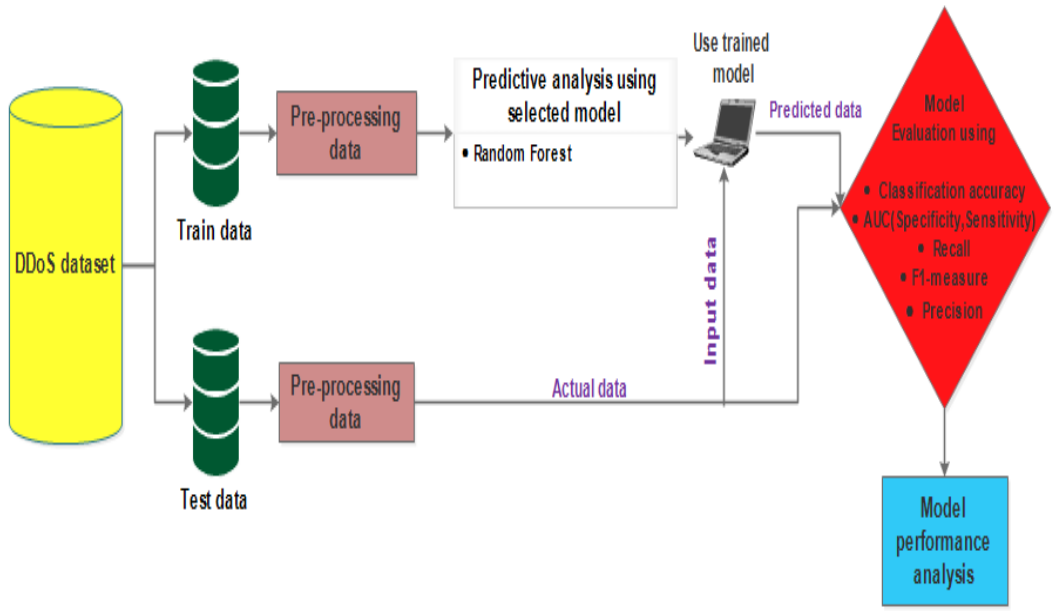

Figure 5.4: IDM evaluation process

A. DDoS dataset

To test the performance and effectiveness of the IDM, we utilized the CIC-DDoS2019 dataset [138]. The dataset has been used by several researchers to evaluate IDS's effectiveness in detecting DDoS attacks. The CIC-DDoS2019 dataset is publicly available at (http://www.unb.ca/cic/datasets/CICDDoS2019) and it consists of over 80 features and about 12 DDoS attacks such as NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN and TFTP for training and 7 attacks such as PortScan, NetBIOS, LDAP, MSSQL, UDP, UDPLag and SYN for testing [138]. However, to determine the performance and effectiveness of the IDM, we used the trained CIC-DDoS2019 dataset [138] in Orange and

we applied a 5-fold cross-validation. The model performance was evaluated using CA, AUC (Specificity, Sensitivity), F measure, Precision and Recall metrics.

B. RF Performance

To determine the performance of the IDM, we employed the trained CIC-DDoS2019 dataset and applied a 5-fold cross-validation for evaluation. Performance analysis of the IDM shows a good detection ability of the RF Model. However, the model is not 100 % accurate but still outperformed other models discussed in Chapter 4. Accordingly, given an ID-based SDWSN identification problem, prediction based on accuracy outcome is represented well. Table 5.3 presents simulated results on the RF-based IDM. The analysis shows an accuracy of 98.0 %. This indicates that this model is more accurate in terms of classifying intrusions or attacks such as DDoS. Moreover, it also shows that RF has an AUC of 100.0 %, a precision of 97.0 %, a recall of 97.0 %, and an F-measure of 97.0 %. This also shows that the RF model is not vulnerable to classifying features. Therefore, in conclusion, based on the experimental results retrieved, there is an indication that RF is the best-performing model due to its accuracy, and efficiency.

Table 5.3 RF-based ID's recorded performance metrics

RF Model	**Performance**
Class	Intrusion- DDoS attacks
Accuracy (%)	98.7
AUC (Specificity, Sensitivity) (%)	99.9
Precision (%)	97.0
Recall (%)	97.0
F1 -measure (%)	96.8

Based on the hypothesis, obtained results show that the RF model has the capability of detecting attacks before the damage of three proposed multiple controllers can be experienced. The hypothesis is "If the TPR of the correct number of features in the DDoS dataset is more than 60 %, then alternately, apply an efficient RF model for better detection. The accuracy was analysed based on criteria compared with [11] and also supported by Figure 5.5 such as accurately identifying packet flows and the performance reflecting the entire packet flow.

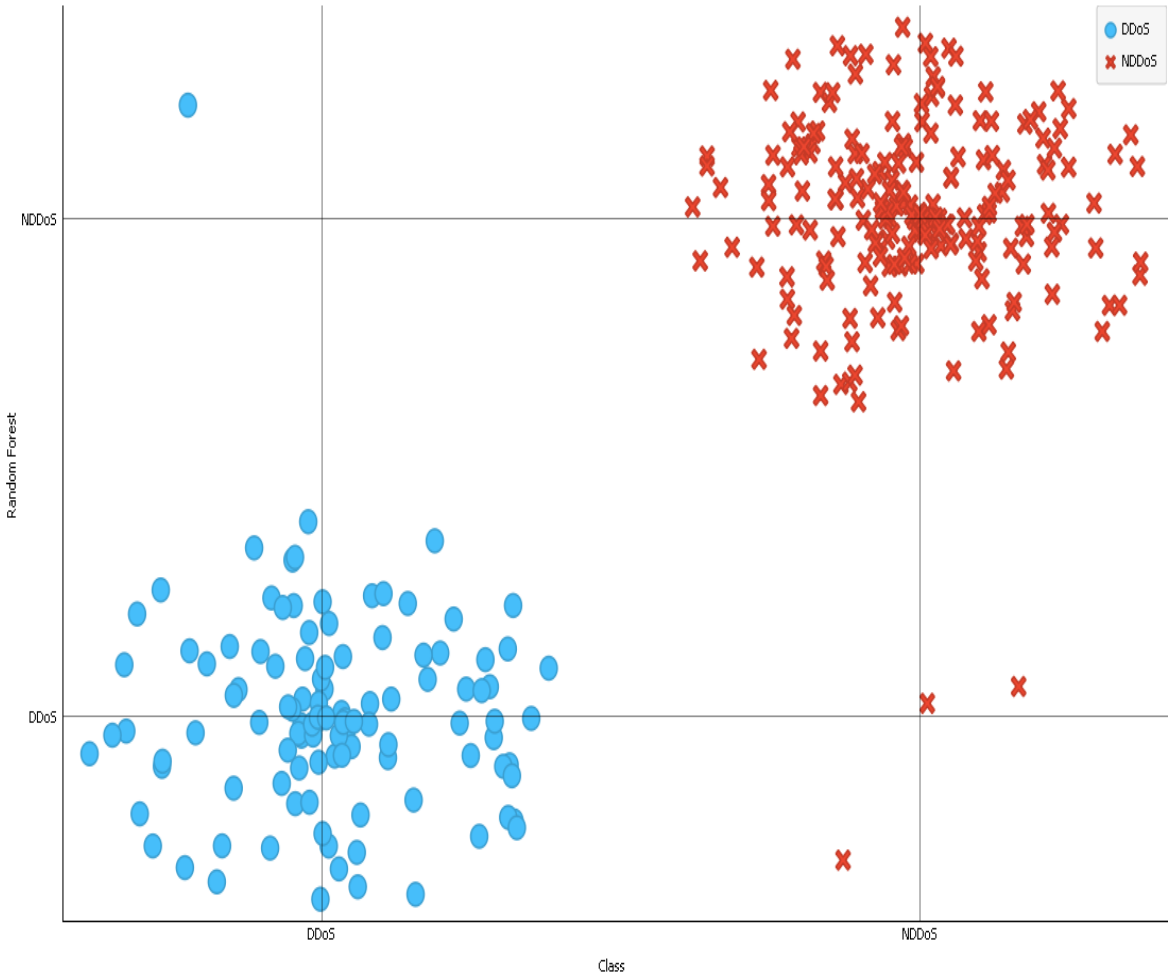

Figure 5.5. RF Model performance

Furthermore, Figure 5.6 shows that the RF model has a probability threshold of 0.033 in terms of precision (y-axis) versus recall (x-axis) and this shows the best results with slight errors.

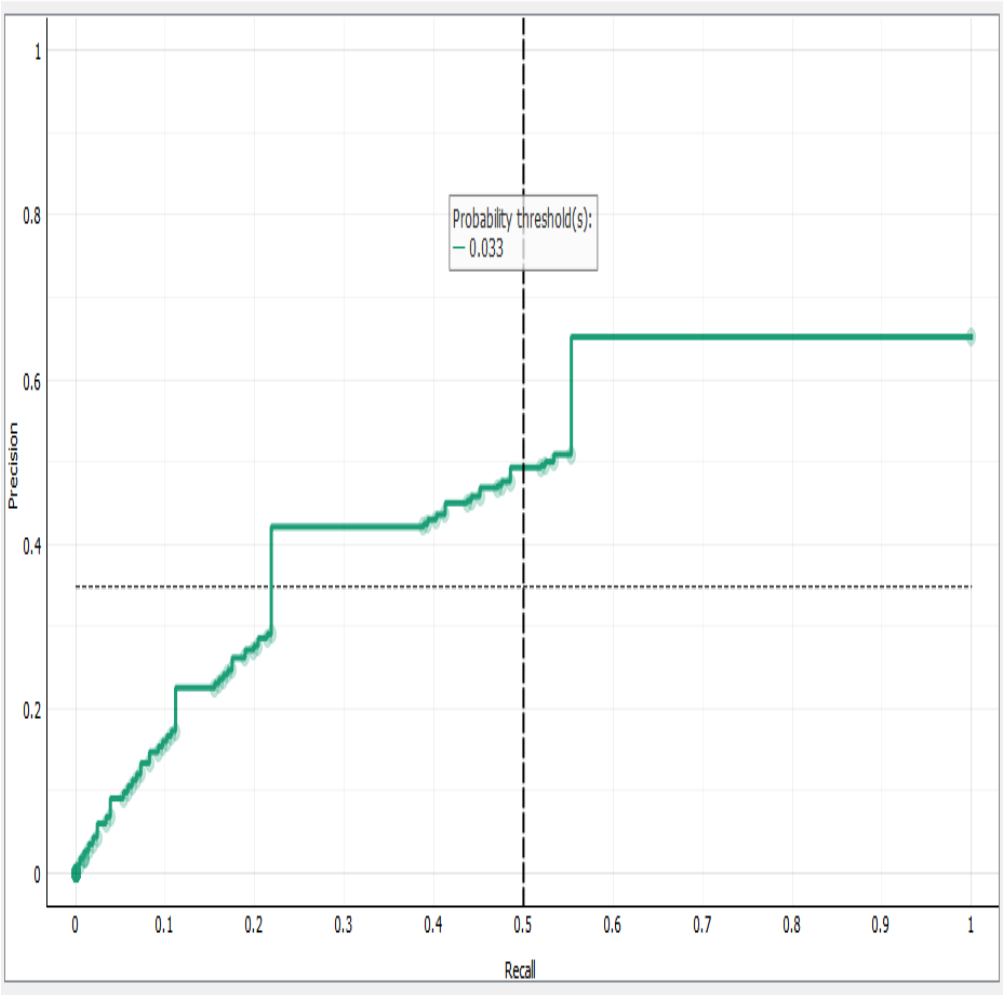

Figure 5.6. RF Model's probability

The RF model outperformed other selected algorithms in Chapter 4 and got a TPR (sensitivity) of 0.500, and based on Figure 5.7, it has 0.556. This shows that the DDoS dataset accommodates the RF model very well while orange experiments are faster compared to the rapid miner.

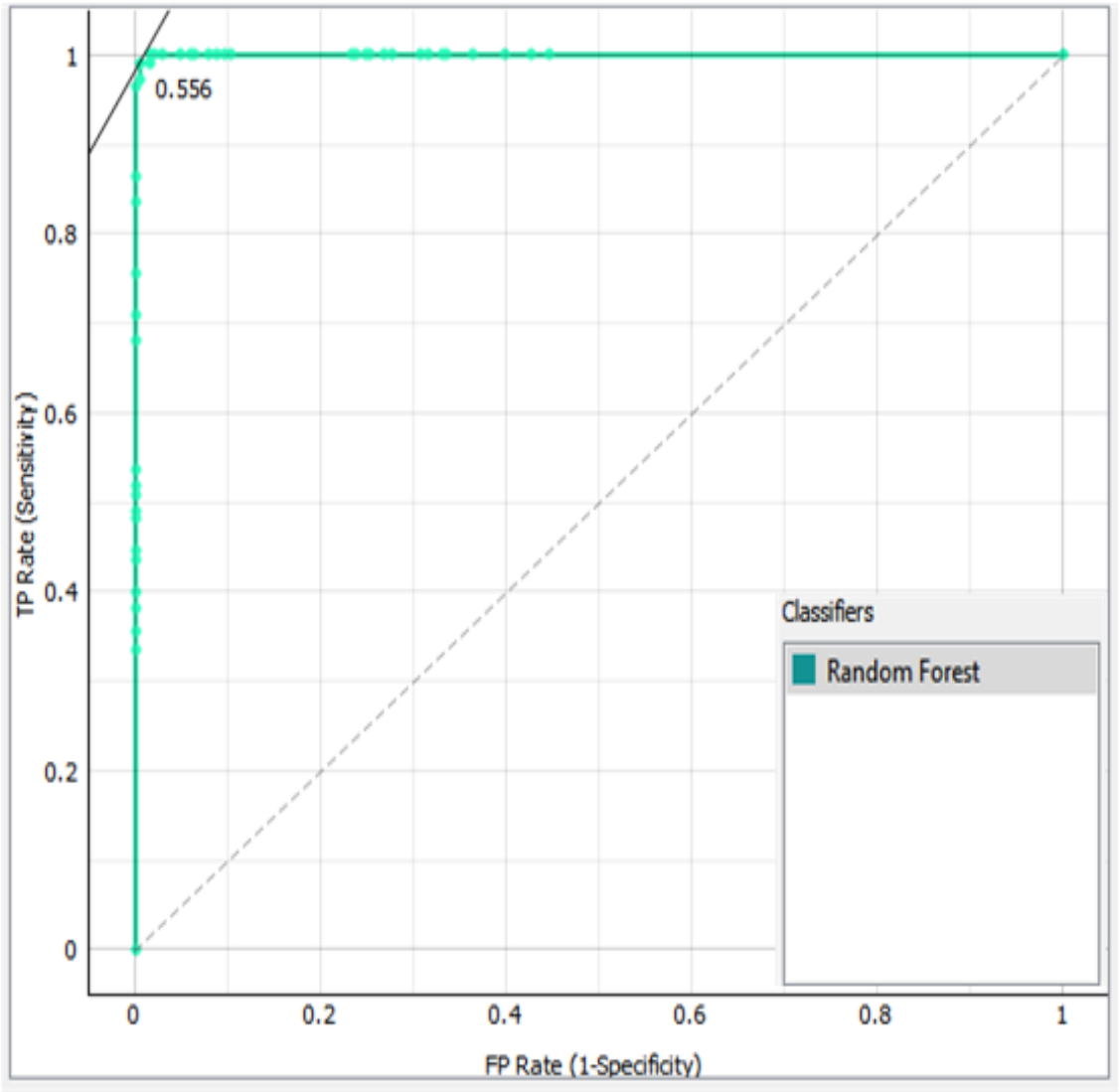

Figure 5.7. ROC based on the RF model

5.4 System Implementation

In this section, we have implemented the SDWSN system and studied its performance. However, as shown in Figure 5.8, the system includes FaToM, IDM with Pox, and Default and Floodlight controllers (DMCs). Furthermore, this section presents the visualized SDWSN system shown as proof of the conceptual model already shown in Figure 4.14 in Chapter 4. The FaToM manages faults while the IDM or IDS identifies intrusions or attacks in controllers. That is the DMCs program the SDWSN operation when flow-based *opf_flow_stats* flows. However, after controllers received *opf_flow_stats*, the DMCs change the SDWSN data plane.

In this case, the IDM is used to send an alert or alarm if intrusions or attacks occur. Furthermore, for performance evaluation, this system used two machines (Ubuntu VM 1 and 2). Ubuntu virtual machine 1 (VM) is configured as the host, and it runs using Mininet and Rapid Miner which deploys network domain with host and OpenFlow sensors. Moreover, it runs OpenFlow protocol and belongs to the data plane. In contrast, Ubuntu VM 2 is configured

as the internal network. That is, Ubuntu VM 2 emulates the network domain where faults and intrusions can be initiated. The OpenFlow sensors send *opflow_stats* to the IDM or IDS through bi-directional link 1.

Therefore, based on the connection, bi-directional link 1 between OpenFlow sensors and SDWSN-based controllers, packets are exchanged in the control and data plane, uni-directional link 2 between OpenFlow sensors and FaToM and IDM or IDS all *opf_flow_stats* are sent for analysis. Uni-directional link 3 enables FaToM to identify faults in controllers while link 4 enables IDM or IDS to notify controllers (Pox, default, and Floodlight) about intrusions using the RF model.

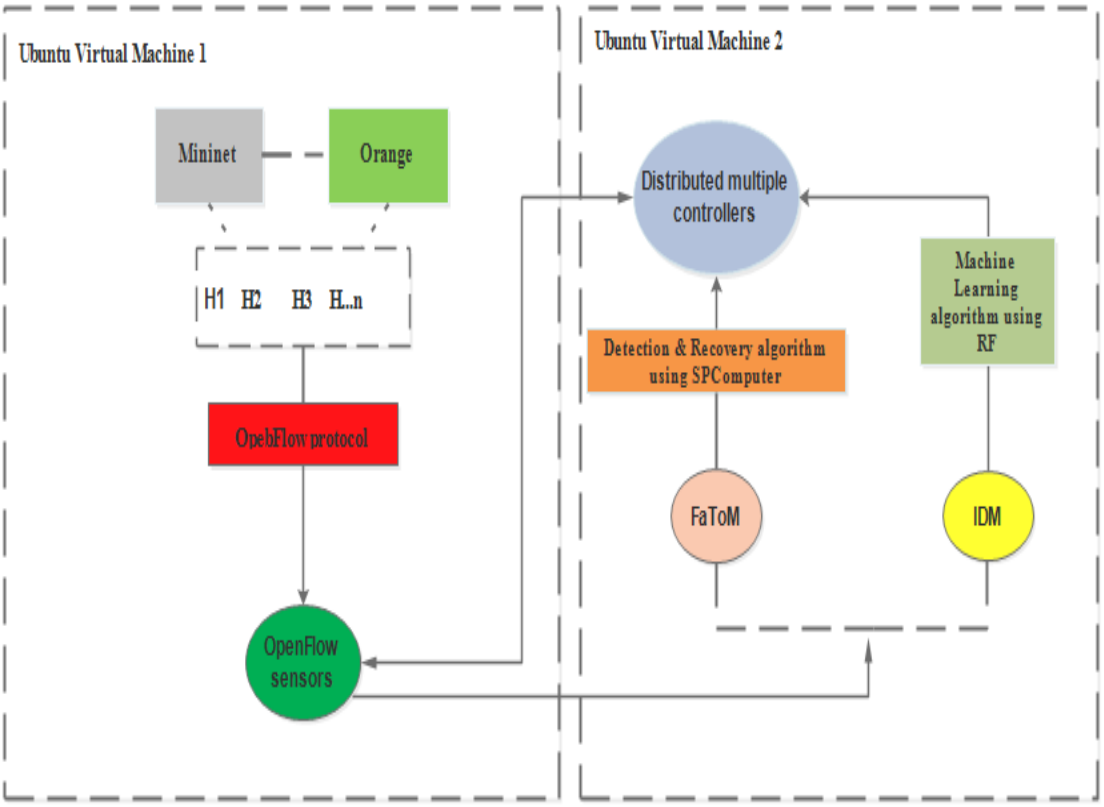

Figure 5.8: SDWSN system implementation

5.5 Discussions

This section presents detailed results on FaToIDM and its comparative analysis. Accordingly, results indicate that proposed DMCs (Floodlight, Pox, and Default) are reliable on the SDWSN. The aim was to design fault-intrusion tolerant DMCs with the best-chosen FT parameters and best-performing ML model. Regarding results obtained from the simulation performed, it is shown that the throughput of the floodlight controller is higher with 8.00, 40.70 and 81.00 kbps compared to pox and default controller. Therefore, it is concluded that controllers are exchanging operations for a certain period to avoid a single point of failure. Since RF outperformed other ML models with 98.7 % accuracy, it is a reliable model to be chosen for detecting catastrophic intrusions in SDWSNs

Finally, in this research with reference results obtained under the FaToM phase, a replication-based mechanism was utilized on our proposed controllers Pox, Default and Floodlight using the Wireshark tool to capture network packets and throughput and latency. As shown in Tables 5.4 and 5.5, simulations gave good results in terms of identifying faults among these proposed DMCs. Moreover, in the IDM phase, usage of the ML algorithm was extended to detect intrusions before entering controllers. In particular, the RF model is classified in terms of accuracy and effectiveness. According to Table 5.6, the RF model performed well and is considered for detecting intrusions or identifying attacks such as DDoS in the aspect of securing a proposed FaToIDM for the SDWSN.

5.5.1 Comparative Analysis with Existing Models

As presented in subsection 5.4.1, the researcher presented experimental results on DMC‘s latency and throughput for FT and intrusion detection or classification using the LR model. The aim was to achieve the best-performing model with resilient controllers to avoid faults and which can also identify intrusions in the SDWSN. Reasonably, due to training and testing using the Orange tool on the DDoS dataset, the RF model showed impressive results.

Moreover, a comparative analysis was done to comprehensively assess the model performance in terms of evaluation metrics specified in Table 5.4 and 5.5. In terms of analysis done on controllers to attain better FaToM Li et.al. [139], showed that floodlight controller has better performance. Overall, based on the analysis of floodlight, default controllers have shown reliable experimental performance by achieving 24 and 81 kbps of throughput to avoid a single point of failure or faults. Furthermore, in IDM, Belavagi et al. [66], illustrated that RF has the highest accuracy and it outperformed other used ML models. Generally, Figure 5.2 shows a graph that concludes that Floodlight and default controllers are reliable in identifying faults at an early stage. Figure 4.13 in Chapter 4 represents a graph showing predictive accuracy in comparison to four selected ML models. That is, RF has 98.7 % of correctly classified instances and a 1.3 % of detection error or misclassification. Therefore, this indicates that according to experimental results, RF has high accuracy compared to other models and can detect intrusions or attacks in controllers for the SDWSN.

Table 5.4: Theoretical existing FT models evaluation

Study	FT mechanism used	Tool used	Effectiveness
[118, 140]	Distributed fault-tolerant clustering algorithm (DFCA)	Cluster heads (CHs)	DFCA uses (CHs) for cluster formation of sensor nodes. Therefore, this mechanism presents distributed run time recovery from experienced faults due to the sudden failure of CHs.
[103]	Static protection module Restoration Module	iPerf 2019, Wireshark Mininet	Wireshark showed good experimental results by analysing network traffic packets compared to iPerf 2019.
[119]	Light synchronization Virtual controller redundancy-based VRRP protocol	Cbench Mininet	Performance tests in C-bench on these two mechanisms showed that light synchronization will impose important performance penalties in new high controller traffic flow.
[141]	FCAFT faced based	Matlab	Based on the simulation results, the proposed FCAFT provided more quality coverage than the face-based technique. Therefore, the conclusion is that FCAFT accomplished self-healing capability compared to the existing faced-based method because it does not apply a node substitution during failure recovery in the network.
[142]	Routing algorithm using Fuzzy logic	TOSSIM Simulations	A fault is based on the connectivity of the node, however, physical connectivity between the nodes is breached due to interferences from outside the network. Finally, an attack in a sensor network where the network has experienced an attack due to a node or few nodes which are not available to the network. Therefore, their algorithm performed well and had some drawbacks.
FaToM	Wireshark, Replication	Mininet	FaToM-based SDWSN aims to deploy central but physically distributed controllers, it is a defending mechanism which will prevent controller faults or failures.

In this subsection, the researcher evaluated results obtained in this research with existing results from other researchers in the literature that used FT and ID mechanisms. Based on the experiment performed, the researcher used a DDoS dataset to evaluate the RF model concerning its effectiveness in identifying or detecting intrusions. To identify the effectiveness of the model, the discussion is done as follows:

Table 5.5: Theoretical ID model evaluation

Study	ID mechanism used	Classifiers used	Dataset	Effectiveness
[61]	ML	Logistic regression, SVM, Random Forest	NSL-KDD	In terms of the effectiveness of the algorithms, it was discovered that RF outperformed other techniques in identifying either network traffic as normal or an anomaly. However, SVM estimated intrusion with the lowest probability. In conclusion, RF has the lowest FPR and highest TPR in identifying network intrusions and showed good results.
[100]	ML	Random forest Decision tree	KDD 99	The authors proposed a threat-aware system aimed to detect malicious traffic. Therefore, according to the evaluation, RF presented the best experiment.
[143]	Hybrid ML technique	K-Means clustering SVM	NSL-KDD	The aim is to reduce FPR, and FNR and improve DR. However, for improvement in classification performance, some steps in the dataset were considered. Therefore, after classification using specified tools or algorithms was performed, the results showed that the hybrid ML technique achieved a positive DR and reduced the false alarm rate.
[144]	ML	KNN SVM PCA-KNN PCA-SVM	KDD 99	The usage of supervised learning obtained better accuracy with the help of KNN. However, principal component analysis (PCA) was applied for the feature vector dimensionality reduction and obtained the best accuracy. Therefore, it was concluded that the PCA-KNN classifier performs better.
[145]	ML	RF	KDD Cup 1999 and NSL-KDD	The reliability of the hybrid detection system was measured regarding the delay, delivery ratio, drop overhead, energy consumption and throughput. According to the authors, the proposed methodology showed an increased overall system efficiency in comparison to the system performance with KNN classifier-based IDS system.
IDM	ML	RF	CIC-DDoS2019 dataset	IDM-based SDWSN aims to apply an ML algorithm which will detect intrusions and prevent these catastrophic attacks from entering the controllers in SDWSN.

5.5.2 Comparing Research Findings with Existing Literature Findings

Based on existing results authors proposed a new algorithm for FT with coverage preservation in face-based WSN by selecting substitute nodes to replace the failing ones. They also presented a distributed algorithm for failure recovery, which is in conjunction with repairing and restoring the network structure. In comparison to other authors, from existing techniques, the performance of the FCAFT scheme was evaluated through simulations and showed effective results of FCAFT in handling failures by restoring the connectivity of the face-structured WSN. Further statement shows that existing SDN-based data plane FT mechanisms can be categorized into reactive and passive which may or may not depend on the controller. However, these mechanisms are still at an early stage because they have partial solutions.

From the ID perspective, the NSL-KDD dataset was proposed to develop an efficient IDS. However, the authors applied adaptive ensemble learning techniques involving DT, RF, K-nearest, and deep neural networks. Thus, regarding the results acquired, the DT as a stand-alone algorithm showed tremendous accuracy compared to the adaptive algorithms. Anomaly-based attacks are detected when a deviation from normal network behaviour is based on the controller-OpenFlow flow statistics request and reply approach via deep inspection using the NN model in real-time for traffic classification. However, for performance, evaluation was done using open daylight (ODL) controller, running on the Ubuntu 16.04 operating system and Open Virtual Switches which is based on the protocol of OpenFlow. Therefore, the experiment was simulated using Mininet and 7 features selected from the NLS KDD dataset. Therefore, obtained results show promising improvement in the detection rate, achieving an accuracy of over 97 %.

In particular, to the above comparative findings, simulation results obtained for FaToM showed an improved throughput and latency in the controllers deployed while the IDM revealed the effectiveness of the deployed RF in terms of CA using the CIC-DDoS2019 dataset.

The results showed an outstanding performance in Pox, default controllers compared to floodlight controllers for the SDWSN in terms of throughput and latency or delay. Moreover, the IDM showed about 98.7 % detection accuracy, 99.9 % specificity and sensitivity, 97 % precision and recall and 96.8 % F1-measure by the RF-based IDS model. This shows that FaToM is a mechanism that can strongly defend against faults experienced in controllers and IDM can also safeguard against intrusions or attacks in SDWSN.

5.6 Chapter Summary

This chapter presented a comprehensive analysis and gave detailed results obtained from the experiment. Chapter 5 of this research also gives detailed test results based on latency, delay, and throughput using the Wireshark network packet analyser and in perspective of NB, LR, RF and SVM. Moreover, results show that RF is an effective model while NBs are efficient. Therefore, based on low latency, delay, high throughput and high efficiency, latency, delay, throughput, and RF are chosen in the implementation of the integrated FaToIDM in SDWSN.

Chapter 6

Summary, Conclusion and Recommendations

6.1. Summary

In this research, the aim was to design and implement a FaToIDM for SDWSNs to effectively increase reliability and security. Chapter 1 introduced the proposed topic, and gave a background of the research area. In essence, it outlined the problem statement, the aim of the study, research questions and objectives. Additionally, research questions were constructed to achieve the research aim. Finally, Chapter 1 outlined the research methodology adopted in this research project.

Chapter 2 allowed the researcher to investigate existing literature based on the proposed research topic. This included an investigation of security catastrophes experienced in SDN, WSN and SDWSN. In this case, mechanisms used by other researchers were studied to solve security issues experienced by networks. Chapter 2 explained the appropriate FT and ID mechanisms to protect the entire network. In Chapter 3, the research methodology and design used to achieve the research aim were outlined and discussed. Chapter 4 outlines the integrated model design known as FaToIDM with its components for the SDWSN. Also, the Wireshark tool and SVM, NB, LR, and RF-based ML models were explained and evaluated in Chapter 4.

Furthermore, Chapter 5 presented the evaluation and results, implemented the SDWSN system and we theoretically evaluated it based on what other researchers have done in the literature study under Chapter 2. The proposed FaToIDM operates by using Wireshark to identify faults through packet or message analysis in proposed multiple controllers (Pox, Default and Floodlight and an effective RF model to classify intrusions or attacks such as DDoS). Moreover, replication with a light synchronization mechanism is proposed to design a fault-tolerant controller which incorporates security.

In FaToM, the usage of Wireshark packet flows was analysed through selected parameters highlighted in Chapter 5 in terms of fault identification. From IDM, the RF model was used to effectively identify intrusion or attacks such as DDoS. Therefore, the main objective of this research study was to design fault and intrusion-tolerant mechanisms to have resilient controllers. This research's goal was achieved by:

i. Following objectives stated in Chapter 1
ii. Conducting a comprehensive literature review on SDN, WSN, especially security-related studies, FT and IDS shown in Chapter 2
iii. Designing an integrated model for the SDWSN system as shown in Chapter 4.
iv. Implementing an integrated model for the SDWSN system and conducting theoretical evaluation by assessing its performance as shown in Chapter 5.

The aim of the research was achieved by answering the RQs as stated in Chapter 1:

RQ1: "How can we prevent the centralized controller from being a single point of failure due to faults and intrusions in the SDWSNs?"

This RQ was answered by comprehensively conducting the literature review based on existing work in SDN, WSN and SDWSN outlined in Chapter 2. Security issues and other mentioned countermeasures experienced in SDNs, WSNs and SDWSNs were outlined in the literature. Fault tolerance and intrusion detection approaches and mechanisms in specified networks were studied to help in designing an integrated FaToIDM for SDWSN.

RQ2: "How can we design and develop an efficient model that incorporates both fault tolerance and intrusion detection or identification mechanisms for a reliable and secured SDWSN?"

RQ2 was answered in Chapter 4 where the researcher designed disjointed models namely: FaToM and IDM in Mininet and evaluated them using Wireshark and RF for self-healing capability in controllers. However, this was to design the integrated model for SDWSN which will identify faults and detect intrusions or attacks through network packet flows. Therefore, if faults are experienced, FaToM should alternate controllers and if intrusions are experienced, IDM using the RF model should take mitigation action.

RQ3: "How can we evaluate the developed model for effective reliability and security performance?"

RQ3 is answered by performing theoretical evaluations of the proposed FaToIDM in Chapter 5 to assess effectiveness, reliability and security based on the Wireshark tool (FaToM) and RF model (IDM).

6.2. Conclusion

SDWSN is a network paradigm which applies SDN strategies to improve outgrowing technological applications in the realm of WSNs. However, issues faced with this novel network model are faults, intrusions or security attacks in controllers and are caused by challenges inherent in the SDN and WSN. Therefore, this research study was conducted to address those challenges by designing fault and intrusion-tolerant controllers that are both attack-aware and can eliminate a single point of failure in a single deployed controller. This study proposed and designed an efficient model incorporating FT and ID mechanisms to secure SDWSN. The proposed system takes advantage of the controller-OpenFlow statistics REQUESTS and REPLIES to effectively detect faults and malicious intrusions in the SDWSN. Both FT and IDS mechanisms were modelled independently and functionalities and integrated to form FaToIDM. To evaluate the performance and effectiveness of the FaToIDM, maximum throughput and network latency or delays were used as metrics using three controllers. Under the IDM phase, the model utilized is RF with CA, AUC, precision, recall, and F-measure.

The simulation results obtained for FaToM showed an improved throughput and latency in the controllers deployed while the IDM revealed the effectiveness of the deployed RF in terms of CA using the CIC-DDoS2019 dataset. However, due to time constraints, the designed FaToIDM was not fully implemented as an integrated system. The results showed an outstanding performance in Pox, default controllers compared to floodlight controllers for the

SDWSN in terms of throughput and latency or delay. Moreover, the IDM showed about 98.7 % detection accuracy, 99.9 % specificity and sensitivity, 97 % precision and recall and 96.8 % F1-measure by the RF-based IDS model. This shows that FaToM is a mechanism that can strongly defend against faults experienced in controllers and IDM can also safeguard against intrusions or attacks in SDWSN. Therefore, for SDWSN to be resilient, a model that incorporates both faults and attack detection must be in place to protect the network from all malicious attacks and unexpected faults that can result in access to network-sensitive resources and even failure of the entire network. We believe that, if FaToID architecture is integrated into the SDWSN model, it could go a long way to make the network dependable and resilient.

6.3. Recommendations and Future Works

This section presents the possible recommendations and future works based on this research study.

1) The design or evaluation metrics FaToIDM are not related and their usage for analysis approach in this research is not correlated. Therefore, the proposed SDWSN-based self-healing and IDM is to apply or use active replication techniques together with an anomaly detection approach to reveal the relationship between these metrics. Based on the FaToM approach, both throughput and latency showed better results in terms of fault detection and recovery in centralized but logically distributed multiple controllers DMCs. Moreover, in IDM, RF has shown the best performance in detecting intrusions or attacks.
2) Due to time constraints and a lack of important resources, the designed FaToIDM was not fully implemented to ascertain its performance and effectiveness. Therefore, to further improve the proposed self-healing and intrusion detection model designed in Chapter 4, FaToIDM must be implemented as a single integrated model and more features or functionalities such as combining active replication and anomaly detection. This will provide a hybrid mechanism that will actively block catastrophic faults and intrusions in the network at an early stage.

References

1. Letswamotse, B.B., et al., *Software defined wireless sensor networks (SDWSN): a review on efficient resources, applications and technologies.* Journal of Internet Technology, 2018. **19**(5): p. 1303-1313.
2. Kobo, H.I., A.M. Abu-Mahfouz, and G.P. Hancke, *A survey on software-defined wireless sensor networks: Challenges and design requirements.* IEEE access, 2017. **5**: p. 1872-1899.
3. Kocakulak, M. and I. Butun. *An overview of Wireless Sensor Networks towards internet of things*. in *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*. 2017. Ieee.
4. Luo, H., et al., *Software-defined architectures and technologies for underwater wireless sensor networks: A survey.* IEEE Communications Surveys & Tutorials, 2018. **20**(4): p. 2855-2888.
5. Mohapatra, H. and A.K. Rath, *Fault tolerance in WSN through uniform load distribution function.* International journal of sensors wireless communications and control, 2021. **11**(4): p. 385-394.
6. Mohapatra, H. and A.K. Rath, *Survey on fault tolerance-based clustering evolution in WSN*. IET networks, 2020. **9**(4): p. 145-155.
7. Miyaji, A. and K. Omote, *Self-healing wireless sensor networks.* Concurrency and Computation: Practice and Experience, 2015. **27**(10): p. 2547-2568.
8. Diaz, S., D. Mendez, and R. Kraemer, *A review on self-healing and self-organizing techniques for wireless sensor networks.* Journal of Circuits, Systems and Computers, 2019. **28**(05): p. 1930005.
9. Guo, L., Y. Li, and Z. Cai, *Minimum-latency aggregation scheduling in wireless sensor network.* Journal of Combinatorial Optimization, 2016. **31**(1): p. 279-310.
10. Deshpande, P. and M.S. Madankar. *Techniques improving throughput of wireless sensor network: A survey*. in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. 2015. IEEE.
11. Ravipati, R.D. and M. Abualkibash, *Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper.* International Journal of Computer Science & Information Technology (IJCSIT) Vol, 2019. **11**.
12. Jurado-Lasso, F.F., et al., *A survey on Machine Learning Software-Defined Wireless Sensor Networks (ML-SDWSNs): Current status and major challenges.* IEEE Access, 2022. **10**: p. 23560-23592.
13. Pritchard, S.W., G.P. Hancke, and A.M. Abu-Mahfouz. *Security in software-defined wireless sensor networks: Threats, challenges and potential solutions*. in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. 2017. IEEE.
14. Upadhyay, R., U.R. Bhatt, and H. Tripathi, *DDOS attack aware DSR routing protocol in WSN.* Procedia Computer Science, 2016. **78**: p. 68-74.
15. Isong, B., et al., *Comprehensive review of SDN controller placement strategies.* IEEE Access, 2020. **8**: p. 170070-170092.
16. Hassan, M.A., Q.-T. Vien, and M. Aiash, *Software defined networking for wireless sensor networks: a survey.* Advances in Wireless Communications and Networks, 2017. **3**(2): p. 10-22.
17. Cooley, D., *Wireless Sensor Networks Evolve to Meet Mainstream Needs.* RTC Magazine, 2012.

18. Ndiaye, M., G.P. Hancke, and A.M. Abu-Mahfouz, *Software defined networking for improved wireless sensor network management: A survey.* Sensors, 2017. **17**(5): p. 1031.
19. Jacobsson, M. and C. Orfanidis. *Using software-defined networking principles for wireless sensor networks*. in *SNCNW 2015, May 28–29, Karlstad, Sweden*. 2015.
20. Chiang, M. and T. Zhang, *Fog and IoT: An overview of research opportunities.* IEEE Internet of things journal, 2016. **3**(6): p. 854-864.
21. Kreutz, D., et al., *Software-defined networking: A comprehensive survey.* Proceedings of the IEEE, 2014. **103**(1): p. 14-76.
22. Scott-Hayward, S., S. Natarajan, and S. Sezer, *A survey of security in software defined networks.* IEEE Communications Surveys & Tutorials, 2015. **18**(1): p. 623-654.
23. Gao, Z., C. Cecati, and S.X. Ding, *A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches.* IEEE transactions on industrial electronics, 2015. **62**(6): p. 3757-3767.
24. Ahmed, M., A.N. Mahmood, and J. Hu, *A survey of network anomaly detection techniques.* Journal of Network and Computer Applications, 2016. **60**: p. 19-31.
25. Ashraf, J. and S. Latif. *Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques*. in *2014 National software engineering conference*. 2014. IEEE.
26. Abduvaliyev, A., et al., *On the vital areas of intrusion detection systems in wireless sensor networks.* IEEE Communications Surveys & Tutorials, 2013. **15**(3): p. 1223-1237.
27. da Costa, K.A., et al., *Internet of Things: A survey on machine learning-based intrusion detection approaches.* Computer Networks, 2019. **151**: p. 147-157.
28. Narayanaraju, S., S. Umar, and R. Kumar, *A Review of Wireless Sensor Networks: Attacks and Countermeasures.* International Journal of Science, Engineering and Computer Technology, 2013. **3**(12): p. 466.
29. Anwar, R.W., et al., *Security issues and attacks in wireless sensor network.* World Applied Sciences Journal, 2014. **30**(10): p. 1224-1227.
30. Modieginyane, K.M., et al., *Software defined wireless sensor networks application opportunities for efficient network management: A survey.* Computers & Electrical Engineering, 2018. **66**: p. 274-287.
31. Azzabi, T., H. Farhat, and N. Sahli. *A survey on wireless sensor networks security issues and military specificities*. in *2017 International conference on advanced systems and electric technologies (IC_ASET)*. 2017. IEEE.
32. Kocher, I.S., et al., *Threat models and security issues in wireless sensor networks.* International Journal of Computer Theory and Engineering, 2013. **5**(5): p. 830-835.
33. Zhu, W.T., et al., *Detecting node replication attacks in wireless sensor networks: a survey.* Journal of Network and Computer Applications, 2012. **35**(3): p. 1022-1034.
34. Chelli, K. *Security issues in wireless sensor networks: Attacks and countermeasures*. in *Proceedings of the world congress on engineering*. 2015.
35. Mohan, S. and S.G.D.S. Ramya, *A Survey on Wireless Sensor Network Security.*
36. Giruka, V.C., et al., *Security in wireless sensor networks.* Wireless communications and mobile computing, 2008. **8**(1): p. 1-24.
37. Xia, W., et al., *A survey on software-defined networking.* IEEE Communications Surveys & Tutorials, 2014. **17**(1): p. 27-51.
38. Luo, T., H.-P. Tan, and T.Q. Quek, *Sensor OpenFlow: Enabling software-defined wireless sensor networks.* IEEE Communications letters, 2012. **16**(11): p. 1896-1899.

39. El-Mougy, A., M. Ibnkahla, and L. Hegazy. *Software-defined wireless network architectures for the Internet-of-Things*. in *2015 IEEE 40th local computer networks conference workshops (LCN Workshops)*. 2015. IEEE.
40. Choi, Y., Y. Choi, and Y.-G. Hong. *Study on coupling of software-defined networking and wireless sensor networks*. in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2016. IEEE.
41. Chaudet, C. and Y. Haddad. *Wireless software defined networks: Challenges and opportunities*. in *2013 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2013)*. 2013. IEEE.
42. Raza, M.H., et al., *A comparison of software defined network (SDN) implementation strategies.* Procedia Computer Science, 2014. **32**: p. 1050-1055.
43. Akhunzada, A., et al., *Secure and dependable software defined networks.* Journal of Network and Computer Applications, 2016. **61**: p. 199-221.
44. Zhang, H., et al., *A survey on security-aware measurement in SDN.* Security and Communication Networks, 2018. **2018**.
45. El Moussaid, N., A. Toumanari, and M. El Azhari. *Survey of Security in Software-Defined Network*. in *International Conference on Advanced Information Technology, Services and Systems*. 2017. Springer.
46. Thupae, R., et al. *Software defined wireless sensor networks mangement and security challenges: A review*. in *IECON 2018-44th annual conference of the IEEE Industrial Electronics Society*. 2018. IEEE.
47. Miyazaki, T., et al. *A software defined wireless sensor network*. in *2014 International Conference on Computing, Networking and Communications (ICNC)*. 2014. IEEE.
48. Jurado Lasso, F.F., et al., *A Survey on Software-Defined Wireless Sensor Networks: Current status, machine learning approaches and major challenges.* TechRxiv, 2021.
49. O'Shea, D., V. Cionca, and D. Pesch. *The presidium of wireless sensor networks-a software defined wireless sensor network architecture*. in *International Conference on Mobile Networks and Management*. 2015. Springer.
50. Dhamecha, K. and B. Trivedi, *Sdn issues-a survey.* International Journal of Computer Applications, 2013. **73**(18).
51. Bakshi, K. *Considerations for software defined networking (SDN): Approaches and use cases*. in *2013 IEEE Aerospace Conference*. 2013. IEEE.
52. Botelho, F., et al. *On the design of practical fault-tolerant SDN controllers*. in *2014 third European workshop on software defined networks*. 2014. IEEE.
53. Botelho, F.A., et al. *On the feasibility of a consistent and fault-tolerant data store for SDNs*. in *2013 Second european workshop on software defined networks*. 2013. IEEE.
54. Jerlin, C.A. and N. Rajkamal, *Fault tolerance in wireless sensor networks.* International Journal of Innovative Research in Advanced Engineering, 2015. **2**.
55. Duran, C.M., E.A. Leal, and J.F. Botero. *Improving fault tolerance in critical networks through OpenFlow*. in *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*. 2017. IEEE.
56. Fonseca, P., et al. *A replication component for resilient OpenFlow-based networking*. in *2012 IEEE Network operations and management symposium*. 2012. IEEE.
57. Fonseca, P., et al. *Resilience of sdns based on active and passive replication mechanisms*. in *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013. IEEE.
58. Pfeiffenberger, T., et al. *Reliable and flexible communications for power systems: Fault-tolerant multicast with SDN/OpenFlow*. in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. 2015. IEEE.

59. Qiu, M., et al. *A novel energy-aware fault tolerance mechanism for wireless sensor networks*. in *2011 IEEE/ACM International Conference on Green Computing and Communications*. 2011. IEEE.
60. Baraneetharan, E., *Role of machine learning algorithms intrusion detection in WSNs: a survey.* Journal of Information Technology, 2020. **2**(03): p. 161-173.
61. Zwane, S., P. Tarwireyi, and M. Adigun. *Performance analysis of machine learning classifiers for intrusion detection*. in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. 2018. IEEE.
62. Tang, T.A., et al. *Deep learning approach for network intrusion detection in software defined networking*. in *2016 international conference on wireless networks and mobile communications (WINCOM)*. 2016. IEEE.
63. Maleh, Y., et al., *A global hybrid intrusion detection system for wireless sensor networks.* Procedia Computer Science, 2015. **52**: p. 1047-1052.
64. Sedjelmaci, H., S.M. Senouci, and M. Feham, *An efficient intrusion detection framework in cluster-based wireless sensor networks.* Security and Communication Networks, 2013. **6**(10): p. 1211-1224.
65. Ha, T., et al., *Suspicious traffic sampling for intrusion detection in software-defined networks.* Computer Networks, 2016. **109**: p. 172-182.
66. Belavagi, M.C. and B. Muniyal, *Performance evaluation of supervised machine learning algorithms for intrusion detection.* Procedia Computer Science, 2016. **89**: p. 117-123.
67. Simeone, O., *A very brief introduction to machine learning with applications to communication systems.* IEEE Transactions on Cognitive Communications and Networking, 2018. **4**(4): p. 648-664.
68. Alsheikh, M.A., et al., *Machine learning in wireless sensor networks: Algorithms, strategies, and applications.* IEEE Communications Surveys & Tutorials, 2014. **16**(4): p. 1996-2018.
69. bhai Gupta, A.R. and J. Agrawal. *A comprehensive survey on various machine learning methods used for intrusion detection system*. in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*. 2020. IEEE.
70. Dwivedi, R.K., S. Pandey, and R. Kumar. *A study on machine learning approaches for outlier detection in wireless sensor network.* in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. 2018. IEEE.
71. Husain, R. and D.R. Vohra, *A survey on machine learning in wireless sensor networks.* International Education And Research Journal, 2017.
72. Abhale, A.B. and S. Manivannan, *Supervised machine learning classification algorithmic approach for finding anomaly type of intrusion detection in wireless sensor network.* Optical Memory and Neural Networks, 2020. **29**(3): p. 244-256.
73. Taher, K.A., B.M.Y. Jisan, and M.M. Rahman. *Network intrusion detection using supervised machine learning technique with feature selection*. in *2019 International conference on robotics, electrical and signal processing techniques (ICREST)*. 2019. IEEE.
74. Negandhi, P., Y. Trivedi, and R. Mangrulkar, *Intrusion detection system using random forest on the NSL-KDD dataset*, in *Emerging Research in Computing, Information, Communication and Applications*. 2019, Springer. p. 519-531.
75. Farnaaz, N. and M. Jabbar, *Random forest modeling for network intrusion detection system.* Procedia Computer Science, 2016. **89**: p. 213-217.

76. Subba, B., S. Biswas, and S. Karmakar. *Intrusion detection systems using linear discriminant analysis and logistic regression*. in *2015 Annual IEEE India Conference (INDICON)*. 2015. IEEE.
77. Dina, A.S. and D. Manivannan, *Intrusion detection based on machine learning techniques in computer networks.* Internet of Things, 2021. **16**: p. 100462.
78. Rashid, S., et al. *Wireless sensor network for distributed event detection based on machine learning*. in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*. 2014. IEEE.
79. Ali, W.A., et al., *A review of current machine learning approaches for anomaly detection in network traffic.* Journal of Telecommunications and the Digital Economy, 2020. **8**(4): p. 64-95.
80. Pachauri, G. and S. Sharma, *Anomaly detection in medical wireless sensor networks using machine learning algorithms.* Procedia Computer Science, 2015. **70**: p. 325-333.
81. Dey, S.K. and M.M. Rahman, *Effects of machine learning approach in flow-based anomaly detection on software-defined networking.* Symmetry, 2019. **12**(1): p. 7.
82. Mousavi, S.M. and M. St-Hilaire. *Early detection of DDoS attacks against SDN controllers*. in *2015 international conference on computing, networking and communications (ICNC)*. 2015. IEEE.
83. David, J. and C. Thomas, *DDoS attack detection using fast entropy approach on flow-based network traffic.* Procedia Computer Science, 2015. **50**: p. 30-36.
84. Qin, Y., J. Wei, and W. Yang. *Deep learning based anomaly detection scheme in software-defined networking*. in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2019. IEEE.
85. Magán-Carrión, R., J. Camacho, and P. García-Teodoro, *Multivariate statistical approach for anomaly detection and lost data recovery in wireless sensor networks.* International Journal of Distributed Sensor Networks, 2015. **11**(6): p. 672124.
86. Sahoo, K.S., M. Tiwary, and B. Sahoo. *Detection of high rate DDoS attack from flash events using information metrics in software defined networks*. in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*. 2018. IEEE.
87. Banitalebi Dehkordi, A., M. Soltanaghaei, and F.Z. Boroujeni, *The DDoS attacks detection through machine learning and statistical methods in SDN.* The Journal of Supercomputing, 2021. **77**(3): p. 2383-2415.
88. Sathya, R. and R. Thangarajan. *Efficient anomaly detection and mitigation in software defined networking environment*. in *2015 2nd international conference on electronics and communication systems (ICECS)*. 2015. IEEE.
89. Neu, C.V., et al. *An approach for detecting encrypted insider attacks on OpenFlow SDN Networks*. in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2016. IEEE.
90. Manu, B. and A.K. Koundinya. *Intrusion Tolerant Architecture for SDN Networks Through Flow Monitoring*. in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. 2017. IEEE.
91. Sahoo, K.S., et al., *An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics.* Future Generation Computer Systems, 2018. **89**: p. 685-697.
92. Thupae, R., et al. *Machine learning techniques for traffic identification and classifiacation in SDWSN: A survey*. in *IECON 2018-44th annual conference of the IEEE Industrial Electronics Society*. 2018. IEEE.

93. Kim, J., S. Yu, and J. Lee. *Short paper: Wireless sensor network management for sustainable Internet of Things*. in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. 2014. IEEE.
94. Ortega, C., et al. *Improving WSN application QoS and network lifetime management using SOA strategies*. in *2011-MILCOM 2011 Military Communications Conference*. 2011. IEEE.
95. Kim, H., et al. *Coronet: Fault tolerance for software defined networks*. in *2012 20th IEEE international conference on network protocols (ICNP)*. 2012. IEEE.
96. ElDefrawy, K. and T. Kaczmarek. *Byzantine fault tolerant software-defined networking (SDN) controllers*. in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*. 2016. IEEE.
97. Liu, H., et al., *A scale-free topology model with fault-tolerance and intrusion-tolerance in wireless sensor networks.* Computers & Electrical Engineering, 2016. **56**: p. 533-543.
98. Song, S., et al., *Control path management framework for enhancing software-defined network (SDN) reliability.* IEEE Transactions on Network and Service Management, 2017. **14**(2): p. 302-316.
99. Cascone, C., et al., *Fast failure detection and recovery in SDN with stateful data plane.* International Journal of Network Management, 2017. **27**(2): p. e1957.
100. Song, C., et al. *Machine-learning based threat-aware system in software defined networks*. in *2017 26th international conference on computer communication and networks (ICCCN)*. 2017. IEEE.
101. Bhuiyan, M.Z.A., et al. *Local monitoring and maintenance for operational wireless sensor networks*. in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. 2013. IEEE.
102. Sarkar, R. and J. Gao, *Differential forms for target tracking and aggregate queries in distributed networks.* IEEE/ACM Transactions on Networking, 2012. **21**(4): p. 1159-1172.
103. Yamansavascilar, B., et al., *Fault tolerance in SDN data plane considering network and application based metrics.* Journal of Network and Computer Applications, 2020. **170**: p. 102780.
104. Hu, T., et al., *FTLink: Efficient and flexible link fault tolerance scheme for data plane in Software-Defined Networking.* Future Generation Computer Systems, 2020. **111**: p. 381-400.
105. Bysani, L.K. and A.K. Turuk. *A survey on selective forwarding attack in wireless sensor networks*. in *2011 International Conference on Devices and Communications (ICDeCom)*. 2011. IEEE.
106. Panda, M., A. Abraham, and M.R. Patra, *A hybrid intelligent approach for network intrusion detection.* Procedia engineering, 2012. **30**: p. 1-9.
107. Mourabit, Y.E., et al. *Intrusion detection system in Wireless Sensor Network based on mobile agent*. in *2014 Second World Conference on Complex Systems (WCCS)*. 2014. IEEE.
108. Gao, X., et al., *An adaptive ensemble machine learning model for intrusion detection.* IEEE Access, 2019. **7**: p. 82512-82521.
109. Al-issa, A.I., et al. *Using machine learning to detect DoS attacks in wireless sensor networks*. in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. 2019. IEEE.
110. Abubakar, A. and B. Pranggono. *Machine learning based intrusion detection system for software defined networks*. in *2017 seventh international conference on emerging security technologies (EST)*. 2017. IEEE.

111. Chen, X.-F. and S.-Z. Yu, *CIPA: A collaborative intrusion prevention architecture for programmable network and SDN.* Computers & Security, 2016. **58**: p. 1-19.
112. Carvalho, L.F., et al., *An ecosystem for anomaly detection and mitigation in software-defined networking.* Expert Systems with Applications, 2018. **104**: p. 121-133.
113. Jeong, C., et al. *Scalable network intrusion detection on virtual SDN environment*. in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*. 2014. IEEE.
114. Zanna, P., et al. *Adaptive threat management through the integration of IDS into software defined networks*. in *2014 International Conference and Workshop on the Network of the Future (NOF)*. 2014. IEEE.
115. Sayeed, M.A., M.A. Sayeed, and S. Saxena. *Intrusion detection system based on Software Defined Network firewall*. in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. 2015. IEEE.
116. Chen, P.-J. and Y.-w. Chen. *Implementation of SDN based network intrusion detection and prevention system*. in *2015 International Carnahan Conference on Security Technology (ICCST)*. 2015. IEEE.
117. Ajaeiya, G.A., et al. *Flow-based intrusion detection system for SDN*. in *2017 IEEE Symposium on Computers and Communications (ISCC)*. 2017. IEEE.
118. Mohapatra, H. and A.K. Rath, *Fault-tolerant mechanism for wireless sensor network.* IET Wirel. Sens. Syst., 2020. **10**(1): p. 23-30.
119. Sidki, L., Y. Ben-Shimol, and A. Sadovski. *Fault tolerant mechanisms for SDN controllers*. in *2016 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*. 2016. IEEE.
120. Vilchez, J.M.S. and D.E. Sarmiento. *Fault tolerance comparison of onos and opendaylight sdn controllers*. in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. 2018. IEEE.
121. Gite, P., et al., *ML Based Intrusion Detection Scheme for various types of attacks in a WSN using C4. 5 and CART classifiers.* Materials Today: Proceedings, 2023. **80**: p. 3769-3776.
122. Gandhimathi, L. and G. Murugaboopathi, *A Novel Hybrid Intrusion Detection Using Flow-Based Anomaly Detection and Cross-Layer Features in Wireless Sensor Network.* Automatic Control and Computer Sciences, 2020. **54**: p. 62-69.
123. Arkan, A. and M. Ahmadi, *An unsupervised and hierarchical intrusion detection system for software-defined wireless sensor networks.* The Journal of Supercomputing, 2023: p. 1-27.
124. Creswell, J.W. and J.D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. 2017: Sage publications.
125. Bisandu, D.B., *Design science research methodology in computer science and information systems.* International Journal of Information Technology, 2016. **5**(4): p. 55-60.
126. Kankam, P.K., *The use of paradigms in information research.* Library & Information Science Research, 2019. **41**(2): p. 85-92.
127. Geerts, G.L., *A design science research methodology and its application to accounting information systems research.* International journal of accounting Information Systems, 2011. **12**(2): p. 142-151.
128. Dresch, A., D.P. Lacerda, and J.A.V. Antunes, *Design science research*, in *Design science research*. 2015, Springer. p. 67-102.
129. Richey, R.C. and J.D. Klein, *Design and development research: Methods, strategies, and issues*. 2014: Routledge.

130. Peffers, K., et al., *A design science research methodology for information systems research.* Journal of management information systems, 2007. **24**(3): p. 45-77.
131. DeFranzo, S.E., *What's the difference between qualitative and quantitative research.* Retrieved from SnapSurveys: https://www. snapsurveys. com/blog/qualitative-vs-quantitative-research, 2011.
132. Peng, H., et al., *A detection method for anomaly flow in software defined network.* IEEE Access, 2018. **6**: p. 27809-27817.
133. Tan, X., et al., *Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm.* Sensors, 2019. **19**(1): p. 203.
134. Wu, T., et al., *Intrusion detection system combined enhanced random forest with SMOTE algorithm.* EURASIP Journal on Advances in Signal Processing, 2022. **2022**(1): p. 1-20.
135. Meena, G. and R.R. Choudhary. *A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA*. in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*. 2017. IEEE.
136. Thomas, R. and D. Pavithran. *A survey of intrusion detection models based on NSL-KDD data set*. in *2018 Fifth HCT Information Technology Trends (ITT)*. 2018. IEEE.
137. Bukar, U.A. and M. Othman, *Architectural design, improvement, and challenges of distributed software-defined wireless sensor networks.* Wireless Personal Communications, 2022. **122**(3): p. 2395-2439.
138. Sharafaldin, I., et al. *Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy*. in *2019 International Carnahan Conference on Security Technology (ICCST)*. 2019. IEEE.
139. Li, Y., et al. *Performance analysis of floodlight and Ryu SDN controllers under mininet simulator*. in *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*. 2020. IEEE.
140. Azharuddin, M., P. Kuila, and P.K. Jana. *A distributed fault-tolerant clustering algorithm for wireless sensor networks*. in *2013 International conference on advances in computing, communications and informatics (ICACCI)*. 2013. IEEE.
141. Al Aghbari, Z., P.R. PV, and A.M. Khedr, *A Robust Fault-Tolerance Scheme with Coverage Preservation for Planar Topology based WSN*. 2021.
142. Parwekar, P. and S. Rodda. *Fault Tolerance in Wireless Sensor Networks: Finding Primary Path*. in *Proceedings of the Second International Conference on Computer and Communication Technologies*. 2016. Springer.
143. Mohamad Tahir, H., et al., *Hybrid machine learning technique for intrusion detection system.* 2015.
144. Kumar, I., et al., *Development of IDS using supervised machine learning*, in *Soft computing: Theories and applications*. 2020, Springer. p. 565-577.
145. Indira, K. and U. Sakthi, *A Hybrid Intrusion Detection System for SDWSN using Random Forest (RF) Machine Learning Approach.* International Journal of Advanced Computer Science and Applications, 2020. **11**(2).