The Design of an Effective Extreme_Controller_Mechanism Scheme for Software Defined Cognitive Radio Network

By: Brian Sibanda

Dissertation
submitted in fulfilment of the requirements for the degree
of
Masters in Computer Science
in the Faculty of Science and Agriculture (School of Mathematical and Computer Sciences)
at the
University of Limpopo
Turfloop Campus
Supervisor: Professor Mthulisi Velempini
2021

Acknowledgements

Firstly, my thanks to the Almighty God for his unconditional love, kindness and above all grace from the day I was formed in my mother's womb up to this day. His warm-hearted has guided me this far.

Secondly, my thanks go to my supervisor Professor Mthulisi Velempini, who introduced me to the world of Software Defined Cognitive Radio Network. He accompanied and encouraged me to complete this dissertation with perfection. Whenever I needed help, he was there for me: weekdays or weekends. He did not end there, but also extended his arm into teaching me on writing journal articles, conference papers, and impressing audiences in delivering presentations. Surely, these skills that I have learned from him will be part of my life forever and therefore my words cannot equate to express the appreciation I have towards him.

Thirdly, I am thankful to my CONERG colleagues without mentioning their individual names. This dissertations' perfection work would not be feasible without your critical constructive questions and suggestions. Indeed, our seminars were fruitful, and this dissertation bears testimony of it.

Finally, I am thankful to my wife – Melissa, family, and friends. Your support, encouragement, and prayers made it possible for me to complete this dissertation. You are my pillar of strength, in my highest and lowest points in time and I owe this dissertation to you!

About the Researcher

Brian Sibanda grew up in Zimbabwe during the year 2000s. From childhood, he really wanted to be a Computer Scientist. Driven by his passion for computers, he pursued a degree in Computer and Information Sciences at Monash University where he focused in Computer Networking, Business Systems and Computer Programming. Upon completion of his undergraduate degree, he advanced to complete his Honours Degree in Computer and Information Sciences at the same institution. Later on, he went to study for a Masters' Degree in Computer Science at the University of Limpopo.

Brian's interests span across several computer science disciplines such as network security, Computer Vision, machine learning, Robotics, Artificial Intelligence and Data Analytics. He prides himself as a young upcoming Computer Scientist of the 21st century.

List of Figures

List of Tables

List of Equations

List of Acronyms/Nomenclature

AES – Advanced Encryption Standard

ASVM – Advanced Support Vector Machine

AUC – Area Under Curve

CDLN – Deep Learning Convolution Network

CPU – Central Processing Unit

CRN – Cognitive Radio Network

CSS – Cooperative Spectrum Sensing

CU – Cognitive User

DDoS – Distributed Denial of Service

DR – Detection Rate

DSA – Dynamic Spectrum Access

DT – Detection Time

FAR – False Alarm Rate

FN – False Negative

FP – False Positive

FPR – False Positive Rate

ICASA – Independent Communications Authority of South Africa

IP – Internet Protocol

MATLAB – Matrix Laboratory

ML – Machine Learning

NN – Neural Network

OMNeT++ – Objective Modular Network Testbed in C++

OvA – One-vs-All

OvR – One-vs-Rest

PDF – Probability Density Function

PU – Primary User

PUE – Primary User Emulation

ROC – Receiver Operating Characteristics

RSS – Received Signal Strength

SDCRN – Software Defined Cognitive Radio Network

SDN – Software Defined Networking

SOM – Self Organising Maps

SU – Secondary User

TCP – Transmission Control Protocol

TP – True Positive

TPR – True Positive Rate

XCM – Extreme_Controller_Mechanism

Abstract

In Software Defined Cognitive Radio Network (SDCRN), network security is a significant issue. This issue arises when Software Defined Network (SDN) architecture integrates with the Cognitive Radio Network (CRN) technology. SDN is designed to improve network resource management, while CRN technology is meant at improving spectrum management. These technologies are vulnerable to several malicious attacks. These attacks include Distributed Denial of Service (DDoS) and Primary User Emulation (PUE). Both the DDoS and PUE can be disrupt services in the SDCRN. To curb these attacks, schemes which hardens the security of SDCRN need to be designed. Thus, in this study we propose a security mechanism called Extreme_Controller_Mechanism (XCM) that reduce the effects of DDoS and PUE. The proposed XCM scheme was designed and evaluated in three simulation environment, the OMNeT++, Octave, and MATLAB simulators. The SDCRN data set was generated using the Neural Network back propagation algorithms. The data set was then used in Matlab to evaluate the effectiveness of the prosed XCM scheme. XCM proved to be effective and efficient at detection and prevention of DDoS and PUE attacks in SDCRN. In terms of memory and processor utilisation, XCM proved to the best when compared to other schemes such as the Advanced Support Vector Machine (ASVM) and deep learning convolution network (CDLN). But in terms of detection time, the ASVM was found to be the best performing scheme. Regarding our test for detection rate, false positive and false negative, the XCM, ASVM and CDLM performed the same. The results of the XCM were therefore the best and superior to the ASVM and CDLM. This can be attributed to the fact that the XCM scheme is optimised for DDoS and PUE attacks. We can therefore conclude that our XCM scheme is the best performing scheme compared to the ASVM and CDLN schemes.

Keywords: Software Defined Cognitive Radio Network · Distributed Denial of Service · Primary User Emulation

Table of Contents

CHAPTER 1: INTRODUCTION AND BACKGROUND

1.1. Introduction

The Internet has created a digital world where communication can be done from any place [1]. The communication data is transferred from one user to the other using packets [2]. The authors in [2] described packets as the scrambled data bundled together and sent over a data network. However, these data networks are susceptible to different kinds of security attacks [3]. These different kinds of security attacks can harm any network, either a traditional or current network. However, both networks have the data and control plane bundled together inside the networking devices [1], [4]. This bundling together has made it difficult for the network personnel to configure or reconfigure the device to respond to faults, load and changes [1], [4], [5]. This eventually led to the emergence of Software Defined Networking (SDN) that separates the data and control plane (See Figure 1.1). The decoupling of the data and control plane makes it easier to manage and monitor the network [6-8].

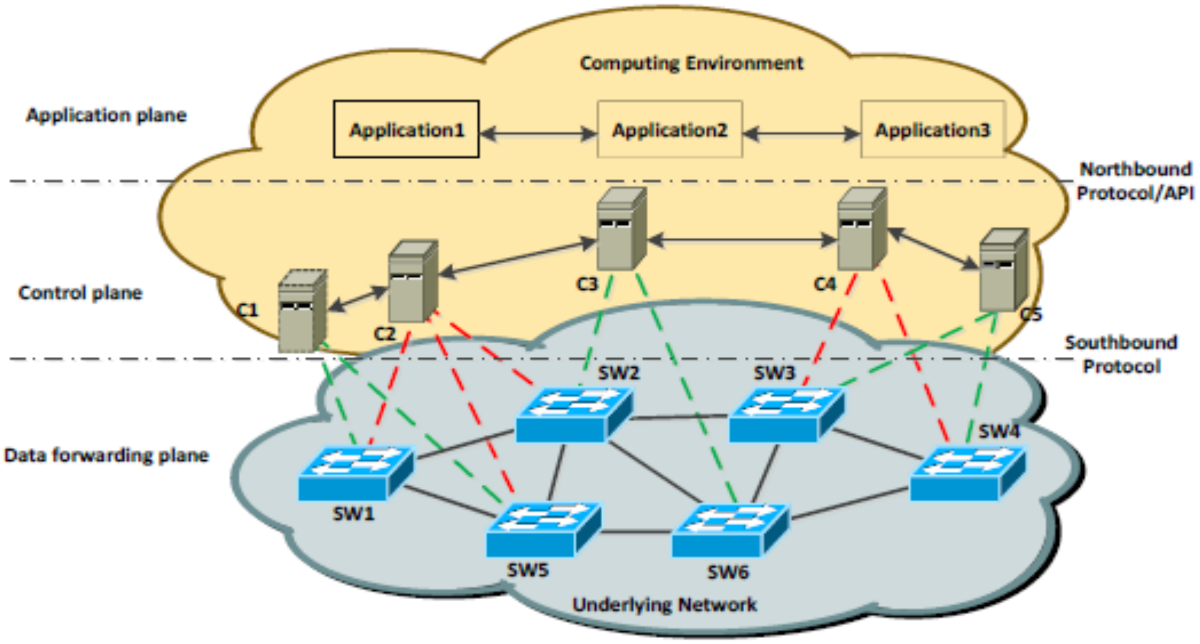


Figure 1.1: Software Defined Network Architecture

Source: Extracted from [6]

On the other hand, [9] reported that there had been a significant increase in the number of devices connected to the wireless networks each year. This has resulted in spectrum scarcity or shortage. According to [9], all the users of the wireless networks are classified into two categories, namely the primary users (PUs) and Secondary Users (SUs). PUs are referred to as licensed users in the wireless network, whilst the SUs are unlicensed users [10]. Authors of [10] went further in mentioning that the SUs are sometimes called Cognitive Users (CUs). However, both these two users use the spectrum differently and PUs have a higher priority over SUs. The PUs makes use of the licensed spectrum whilst the SUs makes use of the unlicensed spectrum [9-11]. The licensed spectrum is a reserved portion for the PUs alone and can only be opportunistically used by the SUs when it is not used by the PUs [11], [12]. However, the authors of [9] claimed that the licensed spectrum is not fully utilised by its sole owners, the PUs, as it is sometimes left being vacant. This finally led to the development of a technology called CRN that makes it possible for SUs to opportunistically sense and use the licensed spectrum if it becomes vacant [9], [10] (See Figure 1.2).

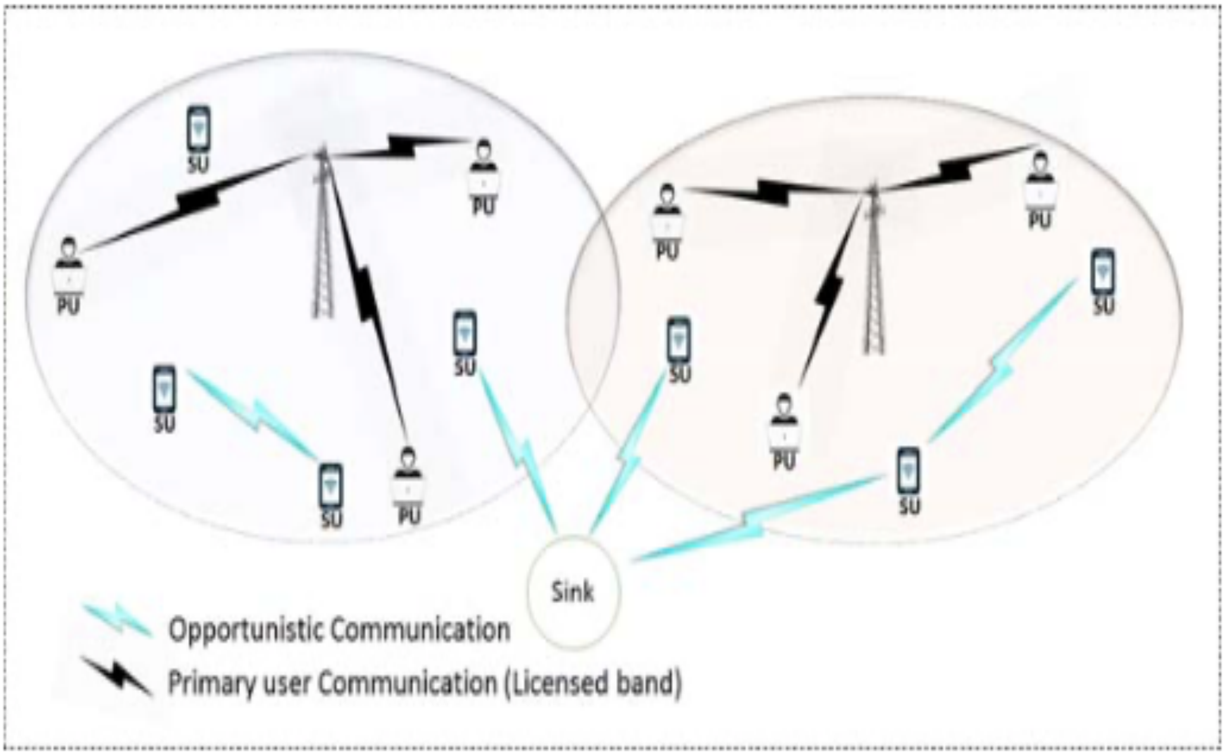

SU
PU
PU
SU
SU
PU
SU
PU
PU
SU
SU
PU
SU
Sink
Opportunistic Communication
Primary user Communication (Licensed band)

Figure 1.2: A Typical Cognitive Radio Network Architecture

Source: Extracted from [10]

Nonetheless, [6] and [9] mentioned that SDN and CRN bring in greater functionality for network resource management and dynamic spectrum management, respectively. Hence, the integration of these two amalgamates to the above-mentioned individual advantages. However, these advantages are more likely to be challenging to realise as the SDN architecture and CRN technology are prone to security attacks such as Distributed Denial of Service (DDoS) and Primary User Emulation (PUE) respectively [13], [14].

The first study on the integration of SDN and CRN was undertaken in 2010 [15]. Since then, there has been growing research interest in SDCRN. The SDCRN integrated environment is likely to be more susceptible to security attacks because it integrates vulnerable network technology. Hence, this study designed an Extreme_Controller_Mechanism (XCM) scheme which addresses the effects of DDoS and PUE attacks in SDCRN. This was motivated by the reality that these two attacks have not been addressed in SDCRN integrated environment, even though reports by authors such as [13], [14], and [16-18] revealed that these two attacks are the most severe in SDN and CRN respectively.

1.2. Prevention Mechanisms

A reasonable number of studies were conducted to counter the effects of DDoS and PUE attacks in SDN and CRN respectively. Some of these studies were such as by [5], [19], [20-23] for SDN and [14], [24-27] for CRN. These prevention mechanisms for the DDoS and PUE attacks are classified into two categories: the signature and anomaly techniques [19], [28]. The signature techniques are used to detect attacks through comparison of incoming traffic with those of the stored attack samples. Therefore, they are deemed unsuitable for the detection of new attacks, whilst the anomaly techniques detect attacks through application of statistical analysis or machine learning methods and are deemed effective [28], [29].

In their study, [20] proposed a threshold-based method to prevent DDoS attacks in SDN. The results confirmed the method as being useful in preventing the DDoS attacks in SDN. A feasible method of source-based Internet Protocol (IP) filtering technique was used in [21] to mitigate DDoS attacks in SDN. This method was found to be efficient only when the malicious traffic is low. In another different study, [5] proposed a filtering scheme to detect and reduce the effects of DDoS attacks on SDN. Using simulation, [5] proved this scheme as effective in reducing the effects of DDoS attacks on SDN. Similarly, a study by [22] showed that DDoS attacks can be prevented using an SDN-Oriented DDoS blocking scheme. Moreover, according to [19] machine learning-based techniques are also useful in reducing the effects of DDoS in SDN. In this regard [23] showed that lightweight method scheme is efficient in detecting DDoS attacks. The Self Organising Maps (SOM) technique, an unsupervised artificial NN, to prevent DDoS attacks on the network is used in this method.

On the other hand, [24] proposed a scheme that utilise energy localisation and variance mechanisms to prevent PUE attacks. The proposed scheme results confirm that the PUE can be mitigated in CRN environment. In support of these results [14] noted that filter-based techniques address the effects of PUE attacks. The results of the filter-based techniques showed these techniques outperform the Received Signal Strength (RSS) which is a localisation-based technique for handling detection and miss detection probabilities. However, the filtering technique has a weakness in that the initial coordinates of the PU cannot be identified in this technique. That means an attacker, which may be close to the PU may not be identified. Also, [25] investigated the PUE in Cooperative Spectrum Sensing (CSS) in CRN and revealed that PUE harms the operation of CSS in CRN. Malicious nodes prevented SUs from using vacant spectrum bands thus causing low utilisation of spectrum bands. The nodes broadcast reports that the primary users are busy whilst they are idle. To deal with this problem [26] proposed a Probability Density Function (PDF) based scheme. In this study, it was observed that an increase in the number of malicious users increases the probability of false alarms. Other studies such as [27] are of the view that encryption-based schemes can be useful. To prove this, [27] used the Advanced Encryption Standard (AES) scheme and their results indicated that the AES scheme can detect PUE attacks in CRN.

Given the various techniques designed to mitigate the effects of DDoS and PUE in SDN and CRN, the study designed an XCM scheme that incorporates the NN concepts to mitigate the effects of DDoS and PUE attacks in SDCRN environment.

1.3. Problem Statement

The DDoS is the most severe attack in SDN [13], [19], while the CRN is susceptible to the PUE attacks [14], [24]. The DDoS compromises the control plane in SDN [30] and the PUE interferes with sensing in the CRN [31]. The two attacks cause the unavailability of service [24], [30], [31]. In SDCRN integrated environment, the effects of these two attacks are likely to be compounded since the architecture and the technology are already vulnerable to the two attacks [9], [30], [31]. To the best of our knowledge, little research has been conducted on these attacks in SDCRN despite numerous studies conducted in SDCRN. However, this study designed a security scheme called XCM that effectively addresses the effects of DDoS and PUE attacks in SDCRN.

1.4. Research Aim

The study aimed to design a security scheme to address the effects of the DDoS and PUE attacks in SDCRN integrated environment.

1.5. Research Questions

In order to provide the answers for the study, the researcher of this study formulated three primary research questions as follows:

i. Which attributes of DDoS and PUE attacks can be detected and measured in SDCRN?
ii. What is the most effective technique that which address the DDoS and PUE attacks in SDCRN?
iii. What is the best strategy for optimising the most effective DDoS and PUE security scheme for efficient memory and Central Processing Unit (CPU)?

1.6. Research Objectives

In order to provide answers to our study and research questions, the following research objectives were formulated:

i. To investigate the network attributes of DDoS and PUE attacks which can be detected and measured.
ii. To explore the most effective technique which can address the DDoS and PUE attacks.
iii. To evaluate the efficiency of the XCM scheme in terms of memory and processor utilisation.
iv. To perform a comparative analysis of the XCM scheme compared to the existing DDoS and PUE schemes designed for SDN and CRN, respectively.

1.7. Research Hypothesis

The XCM scheme will effectively detect and protect the SDCRN from the effects of DDoS and PUE attacks.

1.8. Literature Review

The data and control plane, responsible for consulting the forwarding table to make new packet decisions and providing information for building a forwarding table, are bundled inside the networking devices [32]. This has complicated the matters in responding to faults, loads and changes and led to the emergence of the SDN paradigm, making the whole network managed and controlled [33], [34].

However, every country has its legislation or regulations that address radio frequency spectrum usage and availability. In South Africa, the Independent Communications Authority of South Africa (ICASA) is responsible for that legislation [35]. In [35], most of its radio frequency spectrums are already allocated to various governmental, corporate and academic organisations that are licensed users, leaving the few radio frequency spectrums for the unlicensed users continuously increasing daily. This has resulted in radio frequencies scarcity, mainly for unlicensed users. In order to ease this spectrum situation, a technology called CRN based on the Dynamic Spectrum Access (DSA) approach was developed [36], [37]. The CRN technology allows the licensed and unlicensed users to co-exist together by sharing the spectrum [38-40].
The integration of SDN architecture with CRN technology provides better network resource and dynamic spectrum management [6], [9]. However, the SDN and CRN

are susceptible to security attacks such as the DDoS [13], [19] and PUE [14], [24], respectively. As noted in [8], security is a significant concern in any network. Therefore, security mechanisms that can provide maximum protection to any network from these attacks are ideal.

Several studies such as [5], [19], [20-23] and [14], [24-27] described in the prevention mechanisms section above, have been proposed to detect and address the effects of the DDoS and PUE attacks in SDN and CRN respectively. These studies managed to successfully mitigate the DDoS and PUE attacks in SDN and CRN respectively. Unlike these studies, our study proposed a security prevention mechanism that detects and addresses these two attacks in SDCRN. Our proposed security mechanism incorporates NN concepts, which is an integral part of the machine learning techniques. This was mainly driven by the gaining of momentum in using machine learning techniques such as NN to prevent any security attacks in networking environments [19], [28].

1.9. Research Methodology

The primary tool used for the study was Objective Modular Network Testbed in C++ (OMNeT++). The use of OMNeT++ was primarily centred on its strong support for SDNs [41]. Also, OMNeT++ is considered as one of the most popular simulators used to test distributed protocols in practical wireless channels, radio models and node behaviour associated with radio access [42]. Since the study involved the detection and prevention of the DDoS and PUE attacks in the SDCRN integrated environment, OMNeT++ was found to be the suitable simulator that could allow the researchers to simulate these two attacks in a real networking integrated environment.

Moreover, the NN training and confirmation of results were evaluated in Octave and Matrix Laboratory (MATLAB), respectively. Both Octave and MATLAB were used as secondary simulators for the study. Octave is an open-source software which is readily available and can be used for simulation [43], whereas MATLAB is the most widely used simulator in network security [44].

Furthermore, the following metrics were considered in the evaluation of the efficiency of the designed scheme:

i. Detection time – the time taken for an attacker to be detected.

ii. Detection rate – the positive detection of malicious traffic.

iii. False positive – the amount of network traffics that are incorrectly detected and forwarded.

iv. False negative – the amount of network traffics that are incorrectly detected and dropped.

v. Memory Utilisation – the amount of memory used.

vi. CPU Utilisation – the required CPU processing time.

The comparative results were based on the metrics mentioned above. However, the XCM scheme was designed based on the NN concepts. This is because machine learning techniques such as NN are recently gaining momentum in detecting network attacks [19], [28]. Finally, the XCM scheme was optimised for high detection rate and low false alarm rate as per the recommendation in [23].

1.10. Significance and Outcomes

A practical, efficient, effective and lightweight security scheme was designed that consumes less memory and processing time. The scheme improves the detection rate and reduces false alarms in SDCRN integrated environment. Also, as a contribution to the body of knowledge, the designed XCM scheme could serve as a benchmark for future studies in SDCRN integrated environment.

1.11. Ethical Considerations

The study does not involve the use of human beings, animals or plants, and hence the study did not require ethical clearance.

1.12. Overview

The study managed to produce five chapters in total. Chapter 1 focused on introducing the introduction and background of the design of a practical scheme for SDCRN. This involved a summary of the introduction, background, problem statement, research aim, research questions, research objectives, research hypothesis, research methodology, study significance and ethical considerations. Chapter 2 necessitated the literature review of this research study, such as discussing the effects and prevention mechanism schemes of the DDoS and PUE attacks in SDCRN. Additionally, the conceptualised proposed framework for the SDCRN was presented. It was from this literature review discussion that the lacunae or gap of this study was established. Chapter 3 presented the research methodology describing the research design adopted and incorporated for the study. It also provided the simulation tools used, choice of selected network attributes, justification of NN concepts, and the study metrics. Chapter 4 offered trials and results, which analysed and reported on the findings of the study. Chapter 5 presented the study findings' discussion and conclusions and reported the significance of the study in terms of the contributions, recommendations, and areas of future studies. Finally, the research study also provided an appendix that shows other relevant and useful findings that complement this study results.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

The growth of SDCRN has attracted significant research and development activities in academia and industry [8], [10]. The integration of SDN with CRN alleviates network management and spectrum scarcity [6], [9]. However, as the SDCRN integrated environment matures, security considerations rise [21], [32]. Like any other network environment, a design of security mechanisms that prevent SDCRN from security attacks such as the DDoS and PUE is a sought-after solution to ensure that the benefits of easier network management and spectrum scarcity are realised to their full potential.

While several studies are starting to emerge on this topic in the literature, the fast pace of innovations in this domain mandates thorough, up-to-date designing of security mechanisms to prevent future security attacks on the SDCRN integrated environment [2]. Therefore, this chapter provided a thorough analysis of the existing security mechanisms designed to curb the DDoS and PUE attacks in SDN and CRN respectively. The conceptualised proposed framework which guided this study in the design of its security mechanism scheme.

2.2. Effects of DDoS and PUE Attacks in SDCRN

SDCRN was presented in the works such as [9], [45], [46]. SDCRN is an integration of SDN with CRN [9], [45], [46]. SDN is defined as a networking paradigm that decouples the data and control plane to make it easier to manage and control the network through programmability [2], [47-49]. Inversely, CRN is a technology used to alleviate spectrum scarcity in wireless networks [10], [50-52]. Hence, the integration of the SDN architecture with CRN technology provides a greater functionality in network management and efficient spectrum usage [6], [9], [45].

However, authors like [13] and [14] stated that the DDoS and PUE are the most severe SDN attacks and CRN attacks, respectively. Therefore, SDCRN will be susceptible to the DDoS and PUE attacks since the SDN architecture integrates on a vulnerable CRN technology. The DDoS attacks are reported in [13] and [53] as targeting the SDN controller to disrupt the whole SDN network, whilst [14] reported the PUE attacks as focusing on paralysing the physical layers of the CRN. Overall, both the DDoS and PUE attacks will result in the denial of services to the legitimate users in SDCRN integrated environment [53], [54]. Hence, this study focused on designing an effective security scheme to prevent the SDCRN from the DDoS and PUE attacks.

2.3. Prevention Mechanism Schemes for DDoS and PUE Attacks in SDCRN

Preventive mechanisms to address DDoS and PUE attacks have been proposed by a number of scholars. Scholars such as [19], [28] and [55] classified these techniques into two main categories, namely signature and anomaly techniques. The signature techniques involve detecting attacks through comparison of incoming traffic with those of the stored attack samples, whilst anomaly techniques involve the analysis of traffic through application of statistical or machine learning methods [19], [55]. Between these two techniques, the anomaly technique is reported as the most appropriate countermeasure mechanism for security attacks [55]. Therefore, our designed XCM scheme belongs to the anomaly technique as it implements NN concepts, which allow the analysis of traffic by applying machine learning methods.

Although numerous studies have been provided in the literature like our work, the effects of DDoS and PUE attacks in the SDCRN received little attention in literature. Therefore, for this study's purpose in addressing our related work, we primarily focused on the prevention mechanism schemes of DDoS attacks in SDN and PUE attacks in CRN. This was the only notable difference or distinguishing feature with our work that designed a security scheme which simultaneously detects and prevents these two attacks in SDCRN, instead of separate environments. Hence, the next two subsections discussed the prevention mechanism schemes for DDoS and PUE attacks in SDN and CRN respectively.

2.3.1. Prevention Mechanism Schemes for DDoS Attacks in SDN

In the literature, various methods had been employed for detection and prevention purposes of DDoS attacks in SDN [16]. Some of these methods are such as in [5], [16], [17], [19-23], and [54-67]. In [5], in which a filtering scheme that detect and reduce the effects of DDoS attacks in SDN was proposed and tested. The simulation results in this study proved the scheme to be effective. Then [19] used machine learning-based techniques in analysing the effects of DDoS attacks in SDN and proved the scheme to be effective. Equally so, a threshold-based method scheme was proposed in [20] and was found to be effective in detecting and dropping off packets from the attackers, thereby preventing DDoS attacks in SDN. Similarly, [21] proposed a feasible source-based IP filtering technique scheme but the results of the feasible method scheme were only efficient when the malicious traffic was low.

In their investigation, [22] proposed an SDN-Oriented DDoS blocking scheme and its results proved that the DDoS attacks could be prevented. Also, [23] weighed in by using a lightweight method scheme to detect DDoS attacks on the network. This method incorporated SOM technique which is an unsupervised artificial neural network. Results showed that the scheme was efficient in detecting the DDoS attacks. Likewise, a DDoS attack prevention mechanism scheme was designed in [55] and its results confirmed that the scheme could drop attack flows. Lately, the REsilient COntrol Network (RECON) scheme was used in [57] to mitigate DDoS attacks on the network. Its results illustrated that the scheme can lower DDoS attacks on the network. Also, [58] proposed an entropy method scheme for mitigating the DDoS attacks on SDN controllers. Their scheme was found to be effective, lightweight and yielding a very high detection rate. Moreover, [16] proposed an Advanced Support Vector Machine (ASVM) in the detection and mitigation of DDoS attack in SDN. It was found that their technique can detect the DDoS attack in SDN with a 97% and 3% of detection rate and false alarm rate, respectively.

A Domain Name System (DNS) based DDoS solution was proposed in [59] to mitigate the DDoS attacks in SDN and its results confirmed to be effective. Likewise, [60] proposed an SDN based proactive DDoS Defence Framework. In as far as mitigating the DDoS attacks in SDN, the defence mechanism was found to be effective. Authors in [54] proposed a model based on a Support Vector Machine (SVM) algorithm to detect DDoS attacks in the SDN environment. The results of their algorithm confirmed the detection and mitigation of the DDoS attacks in SDN with high efficiency.

Additionally, an SDN framework based on a machine learning technique using the SVM method was proposed in [61] to identify and defend DDoS attacks in SDN. Their machine learning technique results showed the effectiveness in preventing the DDoS attacks in SDN. The authors in [62] proposed a machine learning-based DDoS mitigation technique for protecting SDN against DDoS attacks. Results showed that their scheme could successfully protect the SDN from DDoS attacks. In [17], a multi-SDN based cooperation scheme was proposed to defend against DDoS attacks. Their scheme managed to attain a high detection accuracy and mitigate the DDoS attacks effectively. A security system that applies machine learning (ML) algorithms through periodic collection of network statistics from the forwarding elements was proposed in [63]. The proposed solution ensured that the SDN was secure against the DDoS attacks.

Authors in [64] also proposed a time and space-efficient solution for detecting DDoS attacks in SDN. The results of their solution proved to be efficient with appreciably good true positive and negative rates. Then, authors in [65] proposed a collaborative DDoS attack mitigation scheme using SDN and their scheme was found to be fast and reliable in efficiently mitigating the DDoS attacks in SDN. Equivalently, authors in [66] offered a lightweight and effective solution based on entropy method to detect DDoS attack in SDN, and a 96% detection rate was reported. Finally, authors in [67] implemented four machine learning methods, namely K-Nearest Neighbours (KNN), Artificial Neural Network (ANN), Naïve Bayes (NB), and Support Vector Machine (SVM), in order to detect DDoS attacks in SDN. Their test results revealed that these four machine learning methods can yield better results in the detection of DDoS attacks in SDN.

2.3.2. Prevention Mechanism Schemes for PUE Attacks in CRN

Different studies related to the prevention mechanism schemes for PUE attacks in CRN are also employed in the literature. Some of these studies are such as by authors in [14], [18], [24-27], and [68-73]. Authors in [14] employed a filtering-based technique to prevent the PUE attacks in CRN. The filtering technique was found to perform better than the RSS-based localisation technique in terms of probabilities of detection and miss detection. This is also supported by the findings of [24] in which their results were effective in detecting and controlling PUE attacks in a CRN environment. Equally, [25] proposed a protocol scheme to identify and eliminate the PUE attacks in CRN. The results proved that their protocol scheme indeed eliminates the PUE attacks in CRN. A PDF based scheme proposed by [26] however, revealed a positive correlation between the number of malicious users and the probability of false alarms.

Besides, research has also shown that encryption-based schemes are effective. In [27], an AES scheme was tested and PUE attacks in CRN were effectively detected. Also, [68] used a filtering algorithm scheme to detect the PUE attacks in CRN. Their scheme simulation results showed that it was reasonable in the detection of PUE attacks. Moreover, [69] used a channel impulse response scheme to detect PUE attacks in CRN. In a process, a modified channel estimation method was incorporated for an SU that did not have prior knowledge about the structure and content of the PU. Experimental results proved that their scheme performs well in the detection of PUE attacks in CRN.

Furthermore, [70] proposed the deep learning convolution network (CDLN) scheme, which incorporates semi-supervised machine learning techniques to detect PUE attacks in CRN. The scheme was found to lower the false alarm and improve the detection rate significantly. Then, [71] used an authentication mechanism scheme to mitigate the PUE attacks in CRN and its results confirmed the PUE attacks are mitigated successfully. Also, [72] used a radio frequency fingerprinting mechanism to mitigate the PUE attacks in CRN. Its results revealed that the PUE attacks could only be effectively mitigated in an ad-hoc CRN and not in the infrastructure-based CRN. Equivalently, [73] proposed a scheme based on the CR users' received power statistics and their scheme achieved a higher performance in the mitigation of PUE attacks in CRN. In conclusion, authors in [18] proposed an adaptive learning-based mechanism using cyclostationary features in mitigating the PUE attacks in CRN, and its results proved effective.

The above-discussed prevention mechanism schemes were all effective in detecting and addressing DDoS and PUE attacks in SDN and CRN respectively. However, in our work, we contribute to previous works, by designing a scheme that simultaneously detect and prevent DDoS and PUE attacks in SDCRN. Therefore, this study designed an effective security mechanism called XCM scheme which incorporated the NN concepts to detect and protect the SDCRN environment from the DDoS and PUE attacks as further presented in Chapter 3.

2.4. Proposed Conceptual Framework for SDCRN

The conceptualisation of our proposed SDCRN framework is based on the principle of commonalities of the SDN and CRN. In the past, the SDN and CRN were used as separate concepts but the advancements in technologies has created the opportunity for the SDN to integrate with CRN [9], [45], [46]. SDN is reported in [6] as having the aptitude to disaggregate traditional vertically integrated networking stacks with the purpose to tailor network operation for specialised environments. In other words, SDN allows the separation of the data and control plane in order to make it easier to manage and monitor the network. CRN, on the other hand, constitute the radio part that provides a promising solution for spectrum scarcity by using the DSA mechanisms [10]. This allows the PUs and SUs to co-exist together without interrupting each other. Therefore, for our study purpose and simplicity, we proposed a conceptualised SDCRN framework that merges the concepts of SDN easier network management and CRN efficient spectrum utilisation (See Figure 2.1).

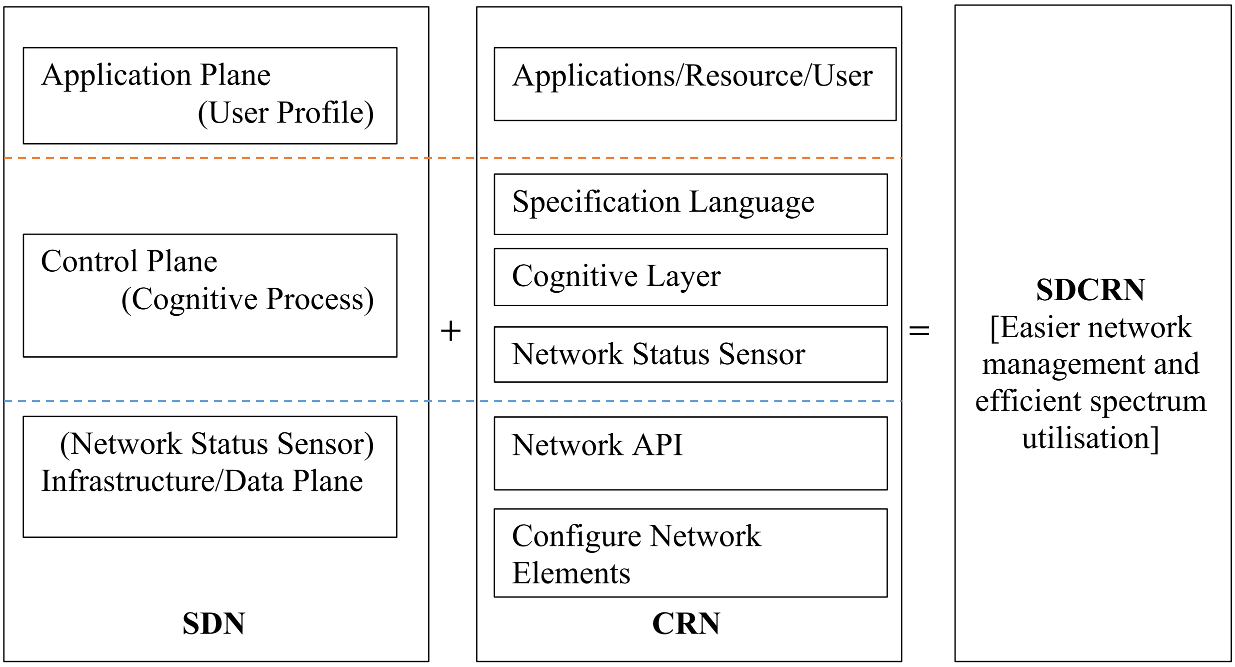


Figure 2.1: Conceptualised SDCRN Framework

2.5. Conclusion

The chapter managed to provide a comprehensive and insightful discussion on the literature review of designing security mechanism schemes for SDCRN against the DDoS and PUE attacks. This started with a discussion on the effects of the DDoS and PUE attacks in SDCRN integrated environment. This was followed by a well-detailed discussion on the security mechanism schemes that were deployed in the past to countermeasure the effects of the DDoS in SDN and PUE in CRN, respectively. This was because as per the researchers' knowledge none of the studies were found to be addressing the effects of both these two attacks simultaneously in SDCRN integrated environment. Hence, our study aim was unearthed based on this identified gap from the literature. Also, the proposed conceptual framework of our study was constructed and presented based on the principle of SDN and CRN commonalities. In conclusion, the next chapter provides the research methodology of the study.

CHAPTER 3: RESEARCH METHODOLOGY

3.1. Introduction

This chapter discusses the research methodology of the study. It starts by discussing the research design opted for this study. Then, the simulation tools that were used in the study are presented. In this study, three simulation tools were used namely OMNeT++, Octave and MATLAB. In addition, the reasons for the use of each simulation tool were provided. Also, the proposed network model for this study is provided. Furthermore, the proposed XCM scheme that was designed for this study is fully discussed in detail. These include the choice of the selected network attributes, justification of NN concepts' use, and implementation of the NN. Equally, the performance metrics that were used in the evaluation of our proposed XCM scheme are provided. Finally, the chapter concludes by providing a synopsis of the next chapter that follows in this study.

3.2. Research Design

Authors in [74] defined research design as “the overall strategy used in a research study to integrate different components coherently and logically to effectively address the research problem”. Uniformly, the authors in [75] weighed in by providing a definition of research design as “the blueprint for collecting, measuring and analysing data”. This study implemented an experimental research design. In [76], an experimental research design is defined as “a blueprint of the procedure that allows the researcher to control all variables that could influence the situation”. Additionally, scholars in [77] weighed in by defining an experimental research design as a process that, when executed, results in one and only one many observations. However, these observations are known as the experiment's outcomes and the collection of all those outcomes is referred to as a sample space. Therefore, the implementation of an experimental research design was motivated by its nature, giving the researchers a proper control of confounding variables that could potentially introduce bias. In the end, an experimental research design necessitated the study researchers to carry out network simulations to simultaneously detect and prevent the DDoS and PUE attacks in SDCRN.

3.3. Simulation Tools

In this study, simulation tools were used to help these study researchers design a network model, detecting attacks based on the network traffic attributes, training of the NN, and implementation of the designed XCM scheme. OMNeT++, Octave, and MATLAB were the three simulation tools used for this study. The subsections that follow provide the choice for each simulation tool use based on the study objectives.

3.3.1. OMNeT++

OMNeT++ is an open-source software initially developed in 2006 by OpenSim Limited in Germany. OMNeT++ was used as this study's primary simulator and its selection choice was solely centred on its ability to allow researchers create network simulations that are related to their studies [78]. OMNeT++ has also been reported as the most popular network simulator used in either wired or wireless areas, and as well as its robust support for SDN OpenFlow and other libraries [41]. Additionally, [42] weighed in by stating that OMNeT++ is one of the most renowned or widely acclaimed network simulators that can be used to test distributed protocols in practical wireless channels, radio models, and node behaviour associated with radio access. Since the study involved detection and prevention of the DDoS and PUE attacks in the SDCRN, OMNeT++ was the most suitable tool that allowed the researchers to simulate these two attacks real networking integrated environment.

3.3.2. Octave

OCTAVE is open-source software that is readily available and can be used for network simulations [43]. It was used as a supporting tool in this study specifically for the training of the NN.

3.3.3. MATLAB

MATLAB is reported in [44] as a widely used simulation tool in networking. In this study, MATLAB was used as a supporting tool. Its purpose was to confirm all the results of the study.

3.4. The SDCRN Model

In the quest to design an effective security scheme which can detect and protect SDCRN from DDoS and PUE attacks, an SDCRN model was proposed (See Figure 3.1). The SDCRN network model was used in the generation of traffic dataset. The traffic dataset is non-malicious traffic (normal) and malicious traffic (either DDoS or PUE or DDoS and PUE attacks). Algorithms 3.1 and 3.2 illustrate the DDoS and PUE attacks generation and detection in SDCRN, respectively. Additionally, the simulation parameters applied in the DDoS and PUE attacks generation and detection are also given in Tables 3.1 and 3.2, respectively.

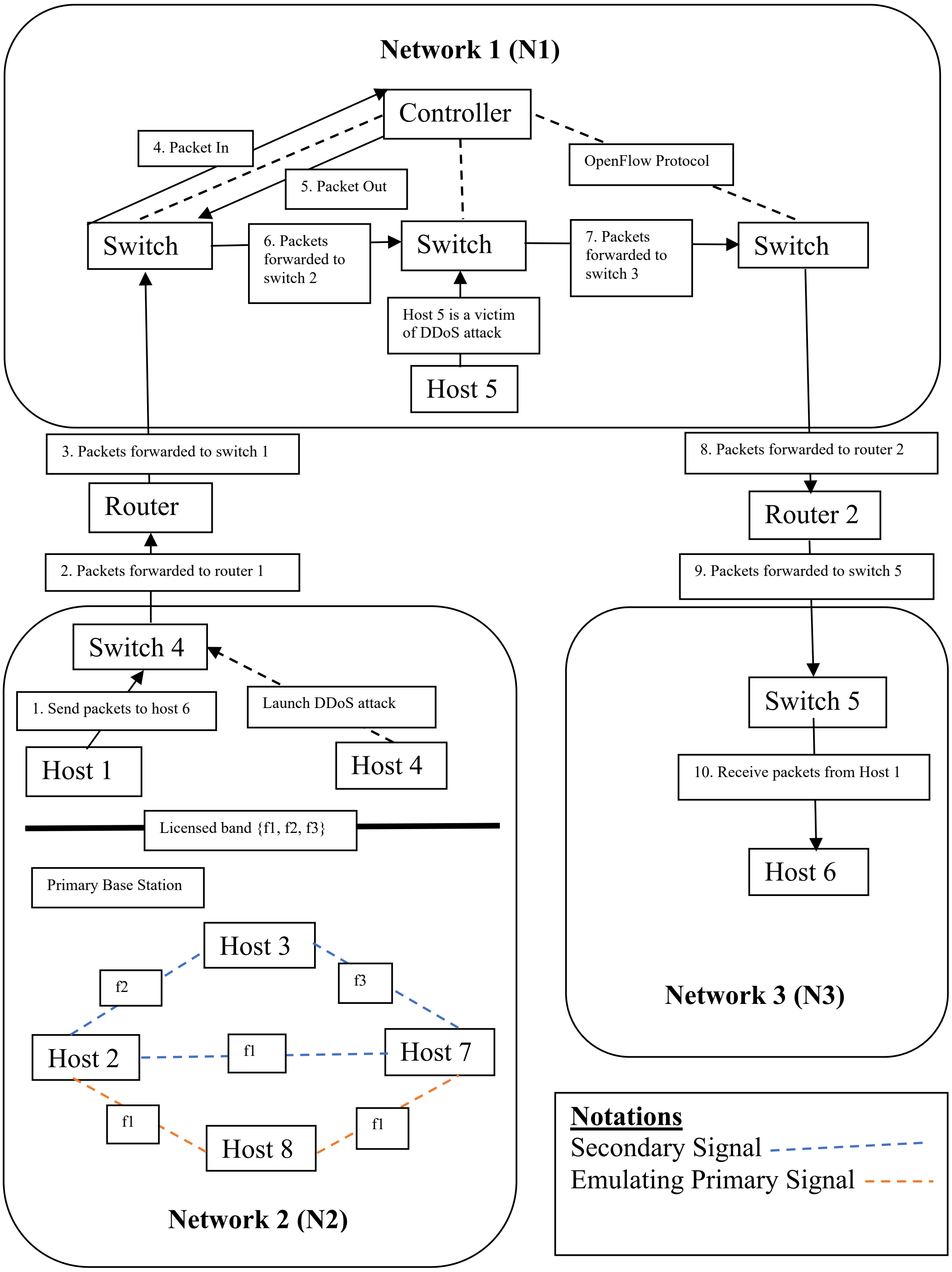

Network 1 (N1)
Controller
4. Packet In
5. Packet Out
OpenFlow Protocol
Switch
6. Packets forwarded to switch 2
Switch
7. Packets forwarded to switch 3
Switch
Host 5 is a victim of DDoS attack
Host 5
3. Packets forwarded to switch 1
8. Packets forwarded to router 2
Router
Router 2
2. Packets forwarded to router 1
9. Packets forwarded to switch 5
Switch 4
1. Send packets to host 6
Launch DDoS attack
Host 1
Host 4
Licensed band {f1, f2, f3}
Primary Base Station
Host 3
f2
f3
Host 2
f1
Host 7
f1
Host 8
f1
Network 2 (N2)
Switch 5
10. Receive packets from Host 1
Host 6
Network 3 (N3)
Notations
Secondary Signal
Emulating Primary Signal

Figure 3.1: The Proposed SDCRN Model

Algorithm 3.1: DDoS Attack Generation and Detection in SDCRN integrated environment.

1. Supposedly there are three networks: Network 1, Network 2, and Network 3. A communication link is then established between these networks.
2. Network 1 have only one user: Host 5; Network 2 has two users: Host 1 and Host 4; Network 3 has only one user: Host 6.
3. Host 4 is a malicious user (DDoS attacker) while Hosts 1, 5 (Victim) and 6 are all non-malicious users.
4. A DDoS attack is launched by Host 4 causing the controller to stop functioning. This results in Host 5 being unable to do any communication with the users in Networks 2 and 3.
5. Hosts 1 and 4 were used to generate non-malicious traffic (normal traffic) and malicious traffic (DDoS attack), respectively.

Table 3.1: Simulation Parameters for DDoS Attack in SDCRN

Parameter/Specification	**Input/Value**
Simulation Tool	OMNeT++
Name of Attack	DDoS
Protocol Applied	OpenFlow
Total Number of Nodes	4
Time of Simulation (seconds)	60
Simulation Area (m^2)	1 000 x 1 000
Transmission Distance (m)	300
Simulated Network Attributes	duration and src_bytes
Number of Packets per Second	120
Packet Size (bytes)	512
Traffic Connections	TCP
Maximum Speed (m/s)	30
Type of Node Movement	Random
Speed of Mobile Node (m/s)	40
Antenna Type	Omni
Queue Management Scheme	Drop Tail
Interface Queue Length (packets)	60
Radio Propagation Mode	2 Ray Ground
Height of Antenna (m)	1
Sensing Distance (m)	600

Algorithm 3.2: PUE Attack Generation and Detection in SDCRN integrated environment.

1. Supposedly there are three channels: f1, f2 and f3 in a licensed band.
2. We then assume the is no primary base station (BS) that is using any of the three channels to transmit to PU receivers. Thus, all the three channels are idle.
3. This permits Hosts 2, 3 and 7 to make transmissions using any of these three idle channels {f1, f2 and f3).

4. However, the presence of Host 8 (PUE attacker) mimicking the primary signal of f1 channel results into misleading Hosts 2 and 7 into vacating that channel, thus invoking a PUE attack.
5. Hosts 3 and 8 were used to generate non-malicious traffic (normal traffic) and malicious traffic (PUE attack), respectively.

Table 3.2: Simulation Parameters for PUE Attack in SDCRN

Parameter/Specification	**Input/Value**
Simulation Tool	OMNeT++
Name of Attack	PUE
Number of Channels	3
Total Number of Nodes	4
Time of Simulation (seconds)	60
Simulation Area (m^2)	2 000 x 2 000
Packet Size (bytes)	6 250
Bandwidth (Mbps)	2
SU Sensing Range (m)	200
Sensing Duration (ms)	1
Simulated Network Attributes	duration and src_bytes
Type of Node Movement	Random

However, the following set of assumptions were made in our network model regarding the PUE attacker:

i. The hardware and radio interference characteristics are similar for both our PUE attacker and the rest of the nodes.
ii. Our network model does not have any information or its strategy regarding the position of our PUE attacker.
iii. Our PU and PUE attacker do not have exactly the same radio behaviour.

Finally, we generated our dataset by using the OMNeT++ simulation tool. Our dataset is available upon request from Brian Sibanda (bsib1234@gmail.com).

3.5. Proposed XCM Scheme

The XCM scheme is a mechanism that was designed to effectively detect and prevent the DDoS and PUE in SDCRN. The XCM scheme was able to detect these two attacks based on the two network attributes selected for this study, which are the number of seconds of the connection (duration) and the number of data bytes from source to destination (src_bytes). The XCM scheme design was centred on multi-layer feed forward NN concepts. The dataset generated for non-malicious and malicious traffic in the SDCRN integrated environment was trained in Octave. This was done in order to pass intelligence to the XCM scheme so that it can be able in future to mitigate similar attacks on the same environment. Therefore, Figure 3.2 depicts the flowchart on the classification of attacks expected to be outputted by our XCM scheme, which are normal traffic (no attack), DDoS attack, PUE attack, and DDoS and PUE attacks.

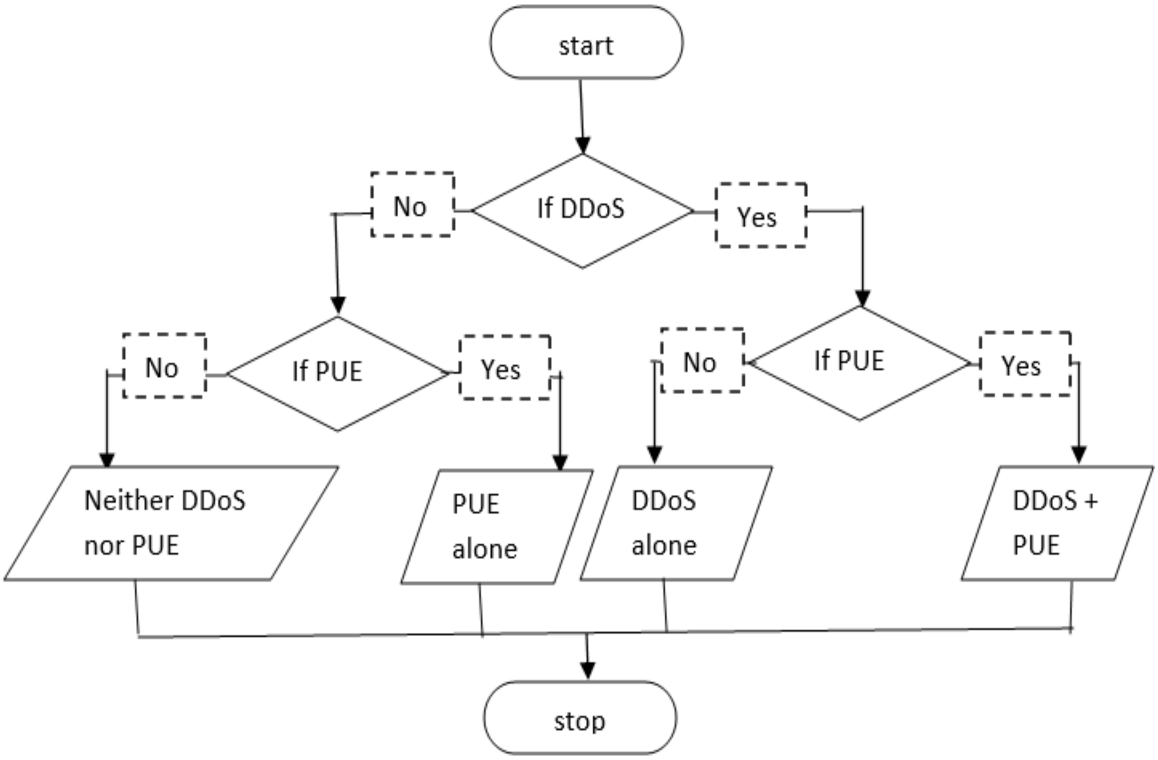
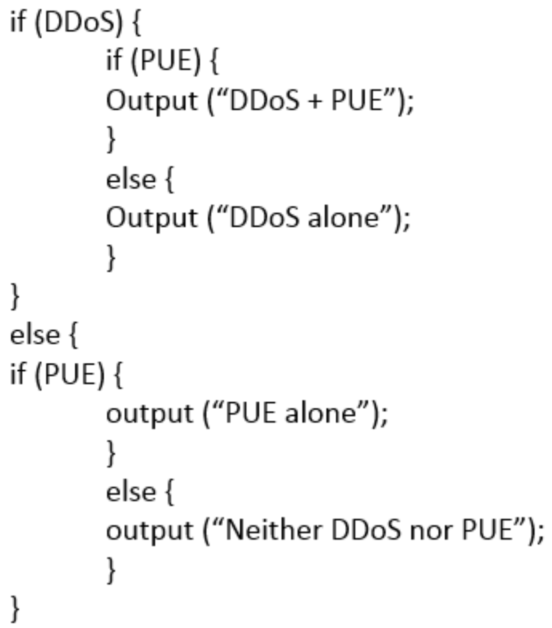


Figure 3.2: Flowchart on the Classification of Attacks

Furthermore, a pseudocode on the classification of attacks is presented as another way of representing this flowchart. It demonstrates how these expected outcomes can be illustrated in an English-like form of steps.

Pseudocode on the classification of Attacks



```
if (DDoS) {
          if (PUE) {
          Output ("DDoS + PUE");
          }
          else {
          Output ("DDoS alone");
          }
}
else {
if (PUE) {
          output ("PUE alone");
          }
          else {
          output ("Neither DDoS nor PUE");
          }
}
```

3.5.1. Justification of Network Attributes Selected

In our study, network attributes, duration and src_bytes were selected to capture non-malicious and malicious traffic measurements. Their selection was influenced by the works of authors in [79] and [80]. These authors regard these network attributes as the most effective attributes which can be used to capture any attack type in any networking environment.

3.5.2. Justification of NN Concepts

In this study, a scheme called an XCM was designed to detect and prevent DDoS and PUE attacks in SDCRN integrated environment. The XCM scheme was built centred on the multi-layer feed forward NN concepts. The NN concepts incorporation was mainly motivated by studies such as in [79], [81], [82]. In [81], NN has been used in network anomaly and misuse intrusion detection resulting in successful detection and security attacks. Also, [79] weighed in by affirming that the NN algorithms were reported as more effective in detecting attacks in different networking environments. Also, [82] confirmed that machine learning using NN concepts had been a powerful tool for modelling complex tasks, notably non-linear classifications. With all these mentioned, this made the NN concepts to be the best choice in this study.

This study involved detecting and preventing DDoS and PUE attacks, which are the most severe in SDCRN integrated environment. These two attacks are more difficult to detect and prevent in a real networking environment, and hence the suitability of NN concepts use as a mechanism for our XCM scheme. This allowed the researchers to design an effective XCM scheme to mitigate both DDoS and PUE attacks in SDCRN.

3.5.3. Implementation of NN

The NN implemented in this study is called the multi-layer feed forward NN (See Figure 3.3). The selection choice of this type of NN in our study is based on authors such as in [83], who reported them as the most common type of NN that has been applied to numerous applications, yielding successful results. In our study, the multi-layer feed forward NN has two input nodes, one layer of two hidden nodes, and one output node. The choice to use one hidden layer was influenced by most literature studies which suggest that a single hidden layer will provide a good approximation for most NN problems, and that adding extra hidden layers will yield little benefit [83].

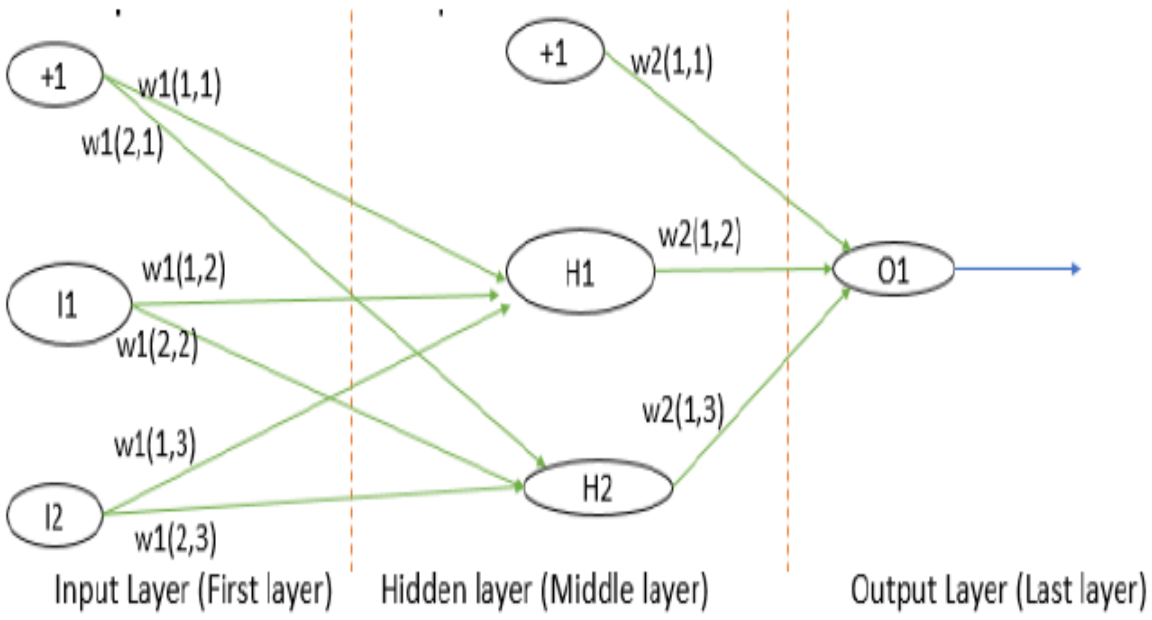

+1
w1(1,1)
w1(2,1)
+1
w2(1,1)
I1
w1(1,2)
w1(2,2)
H1
w2(1,2)
O1
w1(1,3)
w2(1,3)
H2
I2
w1(2,3)
Input Layer (First layer)
Hidden layer (Middle layer)
Output Layer (Last layer)

Figure 3.3: Multi-Layer Feed Forward NN Structure

However, in our situation, this one output node represents four class categories, as depicted in Table 3.3. The classification categories are assigned arbitrarily, as there are no rules to be followed in the assignment classification. This was just done so that the researchers could distinguish one class of attack from the rest of the other classes.

Table 3.3: Expected Detection Attacks Outcome

Multiclass Classification Category	**Detection Outcome**
1	Normal Traffic (no attack)
2	DDoS attack
3	PUE attack
4	DDoS and PUE attacks

In our case, class 1 represents neither DDoS nor PUE attacks (Normal Traffic), class 2 represents DDoS attack, class 3 represents PUE attack and class 4 represents both DDoS and PUE attacks. Therefore, Figure 3.4 represents the complete structure of our NN with the four class categories symbolising the multiclass classification known as the one-vs-all (OvA) or one-vs-rest (OvR). According to [84-86], a multiclass classification is suitable for non-linear models such as the NN, which results in more than two class outputs, and each sample is assigned to one and only one class output. Hence, the suitability of multiclass classification to be adopted in our study.

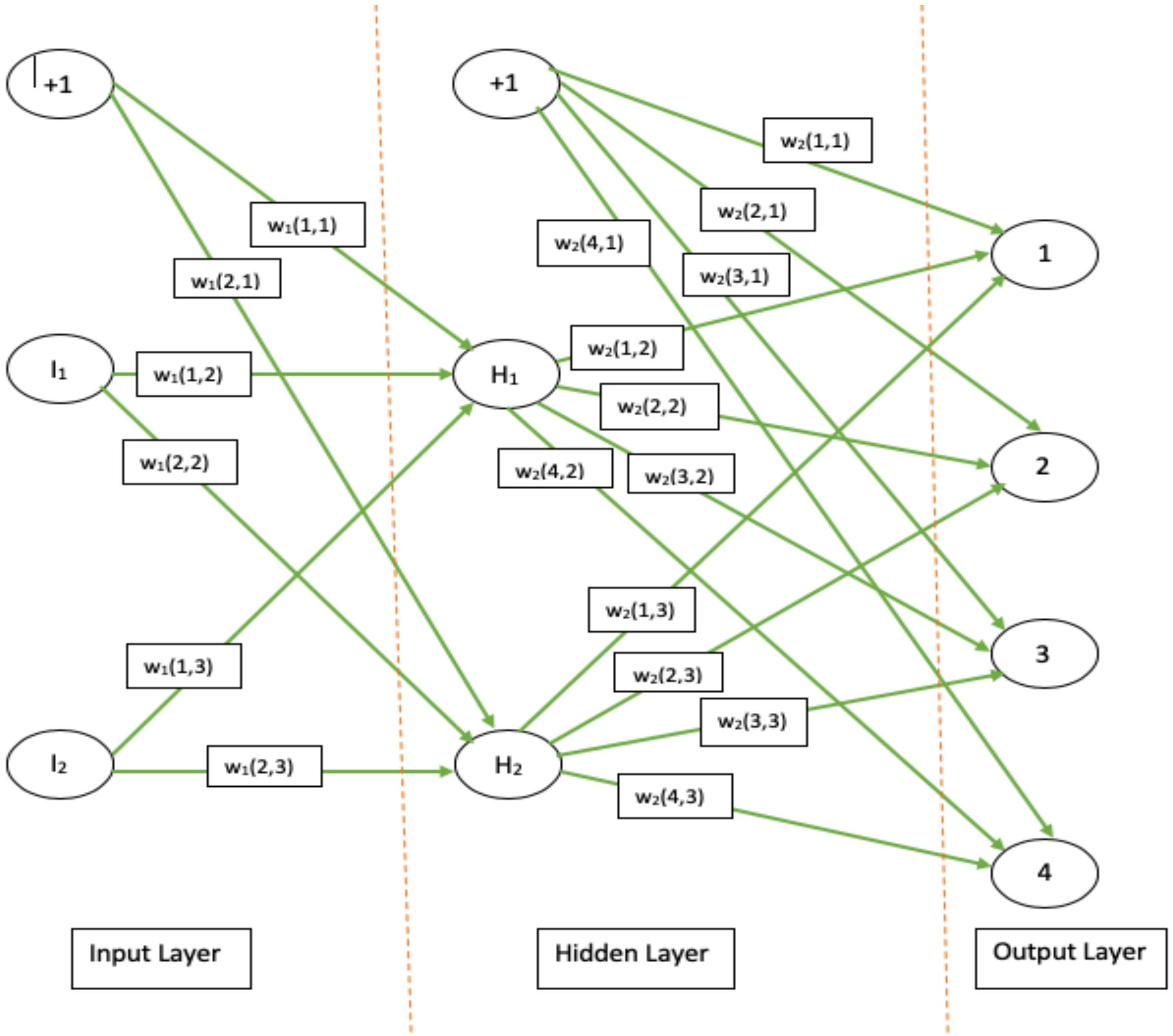


Figure 3.4: The Complete Multi-Layer Feed Forward NN Structure

Based on our NN structure in Figure 3.4, the following hypotheses were formulated:

- Let y = {1, 2, 3, 4}, where y represents the output classes 1, 2, 3 and 4 respectively.
- Let x represents the inputs with attributes x_1 = (I_1) = duration and x_2 = (I_2) = src_bytes.

$$h^1(x) = P(y = 1|x)$$

$$h^2(x) = P(y = 2|x)$$

$$h^3(x) = P(y = 3|x)$$

$$h^4(x) = P(y = 4|x)$$

Hence, our hypotheses formulation is: **$h^i(x) = P(y = i|x)$**, where (i = 1, 2, 3, 4)

Note: To predict a new x, we pick the class that maximises $h^i(x)$.

Our XCM scheme based on the NN concepts was trained in Octave version 4.2.2 with the aid of duration and src_bytes as network attributes. Our XCM scheme's training was done to pass some intelligence into the network traffic data. A standard method known as the logistic regression classifier or sigmoid function was applied to NN to help the scheme in its learning [16], [83], [87]. The sigmoid function restricts the range of our detection outcomes to be within the limits of 0 and 1 [87], [88]. Equation 3.1, as taken in [83], shows the sigmoid function expression.

$$f(x) = \frac{\mathbf{1}}{\mathbf{1}+ e^{-x}}$$

Equation 3.1: Sigmoid Function

Additionally, Figure 3.5 depicts the sigmoid function in a graphical representational notation.

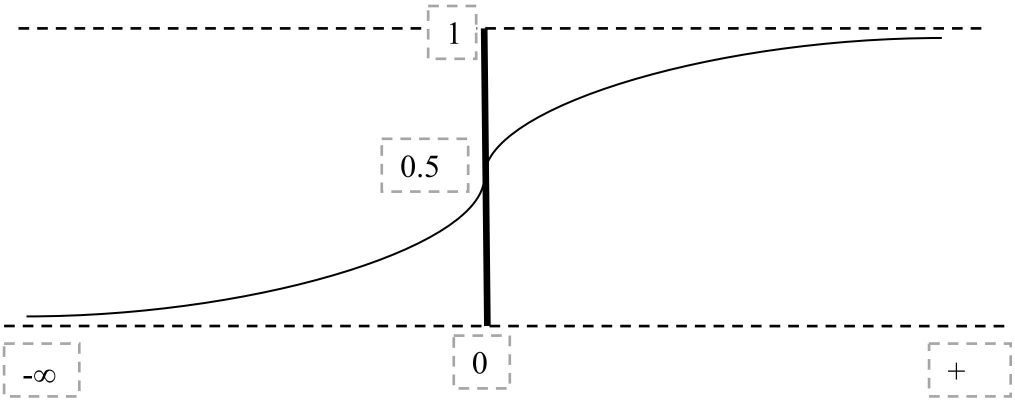


Figure 3.5: Graphical Representation of the Sigmoid Function

Source: Extracted from [87]

In the training of our NN, an algorithm known as the back propagation algorithm was applied. This was supported by authors such as [83], who noted that using a sigmoid function provides an advantage to a NN trained by a back propagation algorithm. Moreover, the connections used initially for the weights and biases were randomly assigned and it was necessary to adjust them. As reported in [88], the connections' adjustment is vital to obtain NN modelling's correct output. Hence, the NN training in our study can be summarised in the following three sequential steps:

i. Model Representation,
ii. Hypothesis Formulation, and
iii. Simulation Experiment - Multiclass Classification

Finally, after our NN training, we generated a new dataset from a similar SDCRN. We implemented our XCM scheme in MATLAB simulator environment to evaluate its performance in respect of detection time, detection rate, false positive, false negative, memory and CPU utilisation. The next section that follows discusses these six-evaluation metrics used for the study.

3.6. Study Evaluation Metrics

In our study, six metrics were considered to evaluate the performance of our proposed XCM scheme. These metrics were considered to evaluate our XCM scheme's performance because they are reported as the most widely used metrics in network attack detection [16], [89-93].

3.6.1. Detection Time

Detection Time (DT) is used in the study to represent the time taken for an attacker to be detected by our XCM scheme. According to [90], a shorter DT is necessary for any security scheme to be considered as a better performing scheme.

3.6.2. Detection Rate

Detection Rate (DR) refers to the correct rate for detecting malicious traffics [16]. Also, [16], [94] and [95] provided the formula used to calculate the DR for any attack networking scheme (See Equation 3.2). The DR of any security scheme should be higher for it to be deemed a better performing scheme [16].

$$\text{DR} = \frac{\boldsymbol{TP}}{\boldsymbol{FN+TP}} \text{x } 100\%$$

Equation 3.2: Detection Rate

where DR = Detection Rate, TP = True Positive and FN = false negative.

3.6.3. False Positive

False Positive (FP) is defined in [16] and [96] as the amount of network traffics that are incorrectly detected and forwarded. Moreover, in [91], a formula used in calculating the false positive rate for any security scheme has been provided (See Equation 3.3).

Lastly, [90] and [91] reported that any security mechanism scheme should produce a lower false positive rate for it to be considered as a better performing scheme.

$$\text{FPR} = \frac{FP}{FP + TN} \text{ x } 100\%$$

Equation 3.3: False Positive

where FPR = False Positive Rate, FP = False Positive, TN = True Negative.

3.6.4. False Negative

In [16] and [96], a false negative (FN) is defined as the amount of network traffic that is incorrectly detected and dropped. It was also defined as the scheme's failure to report an attacker when it appears or present [90]. Lower FN is deemed if any security mechanism scheme needs to be considered as a better performing scheme [89], [90].

3.6.5. Memory Utilisation

In the study, memory utilisation refers to space usage on the Random-Access Memory (RAM). Studies such as [92], [93], and [97] noted that the scheme should have a lower memory utilisation, since lower memory utilisation implies a lightweight scheme, i.e., utilises less space.

3.6.6. CPU/Processor Utilisation

In the study, CPU utilisation was used to refer to the computer processing time during programme execution. In [92] and [97], a lower CPU utilisation was reported as deemed appropriate for any security scheme to be considered as a better performing scheme.

3.7. Conclusion

The chapter managed to discuss the research methodology of the study. It provided a well-detailed analysis of the network simulators used and their choice of selection. It also furnished the network model designed and the algorithms used in the generation of attacks and non-attacks (normal traffics). Furthermore, it managed to provide the selection and justification of the network attributes incorporated in our XCM scheme.

The chapter also examined the justification for the inclusion of the NN concepts in our study. Finally, it discusses the evaluation performance metrics under consideration for the study. The next chapter is based on experimental results and analysis of this research study.

CHAPTER 4: EXPERIMENTS AND RESULTS

4.1. Introduction

This chapter presents the experimental results and analysis of our research. It discusses data availability and the determination of the ideal number of traffic data examples for our experimental simulations. Our experimental simulations were carried out in OMNeT++ and Octave environments. Results of our study were implemented and confirmed in the MATLAB simulator environment. Experimental results are presented in the form of the tables and graphs or charts.

4.2. Data Availability

There were no readily available datasets in our study, and the researchers generated their traffic dataset using OMNeT++ tool. Our dataset is available upon request from Brian Sibanda (bsib1234@gmail.com).

4.3. Dataset Size

A total of 300 traffic data examples, out of which 75 were genuine (normal traffic) and 225 were malicious, were used in the experiment. Our study used 300 traffic data examples consistent with a study in [16] which used the same dataset size. The study in [16] proposed an ASVM to detect and mitigate the DDoS attacks in SDN. Therefore, the study in [16] was used as a benchmark for our study's traffic dataset size. Table 4.1 shows the Experiment Data Division, which illustrates the representational split of our dataset. Our dataset was partitioned into the ratio of 70%:15%:15% for train, validation, and test, respectively. The ratio of 70%:15%:15% for train, validation, and test was applied in the study as it is the most generally acceptable partitioning criteria or rule used in most of the NN modelling studies [98].

Table 4.1: Experimental Dataset Division

Dataset	Examples					Percent
	Normal traffic	DDoS attacks	PUE attacks	DDoS and PUE attacks	Total	
All	75	75	75	75	300	100
Train	53	53	53	53	212	70
Validation	11	11	11	11	44	15
Test	11	11	11	11	44	15

4.3.1. Train Dataset

The train dataset is a dataset consisting of examples used during the learning process. The actual dataset is used to train the model by fitting the parameters such as the weights and biases in the NN modelling. Its primary purpose is to see and learn patterns from this data.

4.3.2. Validation Dataset

The validation data is a dataset consisting of examples used to tune the NN model's hyperparameters, or is the sample dataset used to provide an unbiased evaluation of the NN model fit on the training dataset whilst tuning model hyperparameters. It is sometimes known as the development set. Its main purpose is to understand model behaviour and generalizability on the unseen data and bring insights on how to tune the NN model based on the parameters.

4.3.3. Test Dataset

The test dataset is a dataset that is independent of the training dataset. It is a set of examples used only to assess the performance, i.e., a generalisation of the NN model or the sample of data used to provide an unbiased evaluation of the final NN model fit on the training dataset. Its primary purpose is to understand how the NN model can perform in a real-world scenario as it brings an entirely unbiased estimate of the NN model performance.

4.4. Experiments – Training, Validation and Testing

Our entire dataset had a total of 300 traffic data examples that were split into the training, validation and testing datasets. The training dataset was used to train our NN model, and the validation dataset was used in tuning our NN model hyperparameters in order to avoid overfitting. Also, the testing dataset was used to provide an unbiased evaluation of our NN model fit, thus whether our NN model can be generalised in the detection and prevention of similar attacks. Overall, it is from this entire dataset that our XCM scheme was designed in order to effectively address the effects of the DDoS and PUE attacks in SDCRN integrated environment.

In carrying out our experiments, the following two functions named as *XCMDeepLearning* and *XCMSigmoid* and as well as the two scripts named *XCMTrainingNetwork* and *XCMTestingDeepLearning* were developed. The subsections that follow discuss and present these functions and scripts in detail.

4.4.1. The XCMDeepLearning Function

The XCMDeepLearning function created was used to train the network of our research study. In training the network, the parameter values shown in Table 4.2 were applied.

Table 4.2: Parameter Values of the NN

Parameter	**Value**
Number of input layers	1
Number of neurons in the input layers	2
Number of hidden layers	1
Number of neurons in the hidden layers	2
Number of output layers	1
Number of neurons in the output layers	4
Activation function	f(x) = $\frac{\mathbf{1}}{\mathbf{1+}\ e^{-x}}$
Learning rate	0.01
Epoch	10000

The following code (algorithm 4.1) referred to here as XCMDeepLearning, was implemented in addition to the parameter values applied. This code was used to train the model based on the inputs fed into it and the set or initialised weights.

Algorithm 4.1: XCMDeepLearning Code

```
function [W1, W2] = XCMDeepLearning(W1, W2, i, c_O)
a = 0.01;

N = 212;
for j = 1:N
  transposed_I = i(j,:)';

  i_O_H_L = W1 * transposed_I;
  o_O_H_L = XCMSigmoid(i_O_H_L);

  i_O_O_N = W2 * o_O_H_L;
  f_O = XCMSigmoid(i_O_O_N);

  c_O_transpose = c_O(j,:)';
  e = c_O_transpose - f_O;

  d = e;

  e_O_H_L = W2' * d;
  d1 = (i_O_H_L > 0).*e_O_H_L;

  a_O_W2 = a * d * o_O_H_L;
  a_O_W1 = a * d1 * transposed_I;

  W1 = W1 + a_O_W1;
  W2 = W2 + a_O_W2;
end
end
```

4.4.2. The XCMSigmoid Function

The XCMSigmoid function was used as an activation function. The activation function was used to introduce non-linearities into our NN modelling, which enabled us to approximate arbitrarily complex functions. Algorithm 4.2 is the illustration of the code implemented to assist with that purpose.

Algorithm 4.2: XCMSigmoid Code

```
function y = XCMSigmoid(x)
  y = 1/(1+exp(-x));
end
```

4.4.3. The XCMTrainingNetwork Script

The XCMTrainingNetwork script was used to call the training function (XCMDeepLearning), train the network and save. Its implemented code is shown as algorithm 4.3.

Algorithm 4.3: XCMTrainingNetwork Code

```
i = [6.21 397;
     2.22 44;
     …;
     …;
     …;
    2.31 289;
    5.13 12716;
    ];

c_O = [1
        2
       …
       …
       …
       3
```

```
      4
      ];

W1 = 2*rand(1,2)-1;
W2 = 2*rand(1,2)-1;

for epoch = 1:10000
  [W1, W2] = Deep Learning(W1, W2, i, c_O);
end

save('XCMDeepNeuralNetwork.mat')
```

4.4.4. The XCMTestingDeepLearning Script

The XCMTestingDeepLearning script was used to load the trained network (XCMDeepNeuralNetwork.mat) and test our network's performance. Algorithm 4.4 illustrates the implementation of the code applied in this particular case.

Algorithm 4.4: XCMTestingDeepLearning Code

```
load('XCMDeepNeuralNetwork.mat')

i = [6.39 637;
     2.70 61;
     …;
     …;
     …;
     2.40 445;
     6.48 27145;
     ];

N = 88;
for j = 1:N
  transposed_I = i(j,:)';
  i_O_H_L = W1 * transposed_I;
  o_O_H_L = XCMSigmoid(i_O_H_L);

  i_O_O_N = W2 * o_O_H_L;
  f_O = XCMSigmoid(i_O_O_N);
end
```

After the training, validation, and testing phases, our XCM scheme was evaluated based on the evaluation metrics presented in Chapter 3. These metrics are DT, DR, FP, FN, memory, and processor utilisation. The following section presents the experimental results based on these metrics represented in tables and graphs or charts.

4.5. Experimental Results

Our traffic dataset, which was extracted, has more than two categories of output. Instead of y = {0, 1}, which is a normal binary classification, our data definition is y = {1, 2, 3, 4}, where y = 1 represents a class category 1 (no attack/normal traffic), y = 2 represents a class category 2 (DDoS attack), y = 3 represents a class category 3 (PUE attack) and y = 4 represents a class category 4 (both DDoS and PUE attacks). This categorisation is known as a multiclass classification or one-versus-all (O-v-A) or one-versus-rest (O-v-R) classification. Hence, our study has four binary classifiers to classify the non-malicious (legitimate) effectively and malicious from each category (see Table 4.3).

Table 4.3: Multiclass Classification Representation

Class Category	**Classification Training**
1 = normal traffic	1-vs-rest = 1 vs 2, 3 and 4
2 = DDoS attack	2-vs-rest = 2 vs 1, 3 and 4
3 = PUE attack	3-vs-rest = 3 vs 1, 2 and 4
4 = DDoS and PUE attacks	4-vs-rest = 4 vs 1, 2 and 3

We implemented the multiclass classification using the above-discussed functions and scripts until our NN had learned and outputted correct results. We then implemented our XCM scheme using a new dataset which we generated from the similar SDCRN integrated environment. Afterwards, we assessed our XCM scheme's performance in respect of DT, DR, FP, FN, memory, and CPU utilisation. These results are presented in the form of analytical calculations, tables such as frequency tables and confusion matrix, and graphs such as bar chart and Receiver Operating Characteristics (ROC).

4.5.1. Results and Analysis on XCM Scheme Performance Metrics

In this research study, six metrics were used to evaluate the effectiveness of the XCM scheme. These are DT, DR, FP, FN, memory, and processor utilisation. The following subsections present the results of these six-performance metrics.

4.5.1.1. Detection time

Detection Time (DT) is reported in [90] as "the time taken to detect a particular attack on a network". In other words, it is the interval between the time of attack launch and the time at which it was detected. The time taken to detect the DDoS attack, PUE attack, and DDoS and PUE attacks was recorded and tabulated in Table 4.4. The frequency table results show that it took the XCM scheme at most 5 microseconds (5 μs) to detect any of the class categories under consideration.

Table 4.4: Detection Times of Malicious Attacks

Class Category	**Time (μs)**
DDoS Attack	4.36
PUE Attack	4.71
DDoS and PUE Attacks	4.93

Furthermore, a bar chart (see Figure 4.1) was used to illustrate these malicious attacks' detection times. The results from this bar chart show that the detection time for a DDoS attack is lower than the PUE attack, and likewise, the detection time for PUE attack is also lower than the DDoS and PUE attacks. In essence, the detection time results for these attacks can be mathematically modelled as:

$$DT_{DDoS} < DT_{PUE} < DT_{DDoS\ and\ PUE}$$

where DT_{DDoS} = Detection Time for DDoS attack, DT_{PUE} = Detection Time for PUE attack and $DT_{DDoS\ and\ PUE}$ = Detection Time for DDoS and PUE attacks.

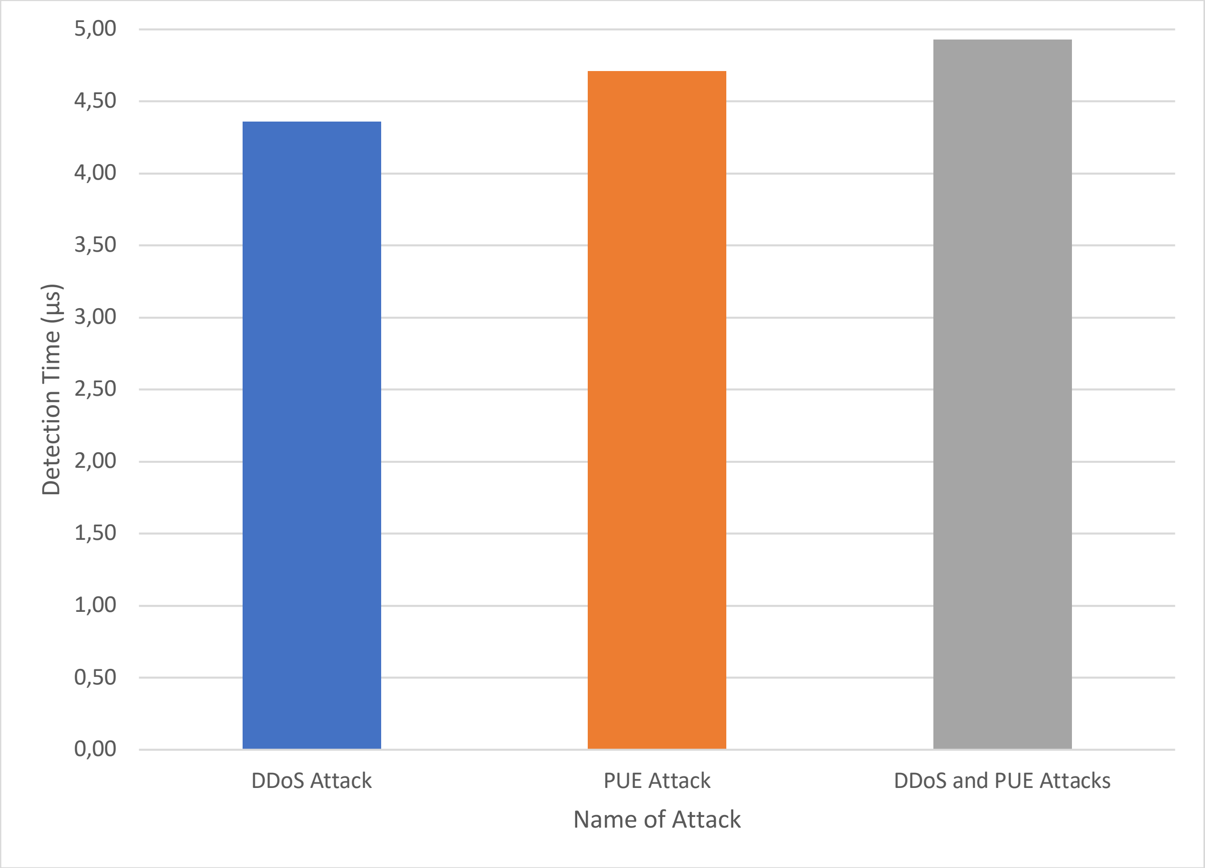


Figure 4.1: Detection Times of Malicious Attacks

On average, the DT of these attacks using the XCM scheme is approximately (4.36 + 4.71 + 4.93)/3 = 4.67 microseconds = 4.67μs (See table 4.4). However, to determine whether the XCM scheme has a shorter time in detecting these malicious attacks, a comparative analysis was carried out using two other related schemes that were also built through the incorporation of ML algorithms. These schemes are ASVM for DDoS [16] and deep learning convolution network (CDLN) for PUE [70]. Table 4.5 and Figure 4.2 show the schemes' comparative results on detection times recorded using ASVM for DDoS and CDLN for PUE against XCM for both DDoS and PUE.

Table 4.5: Schemes' Comparative Results based on Detection Times

Scheme Name	**Class Category**	**Time (μs)**
XCM	DDoS and PUE Attack	4.93
ASVM	DDoS Attack	4.42
PUE	PUE Attack	4.79

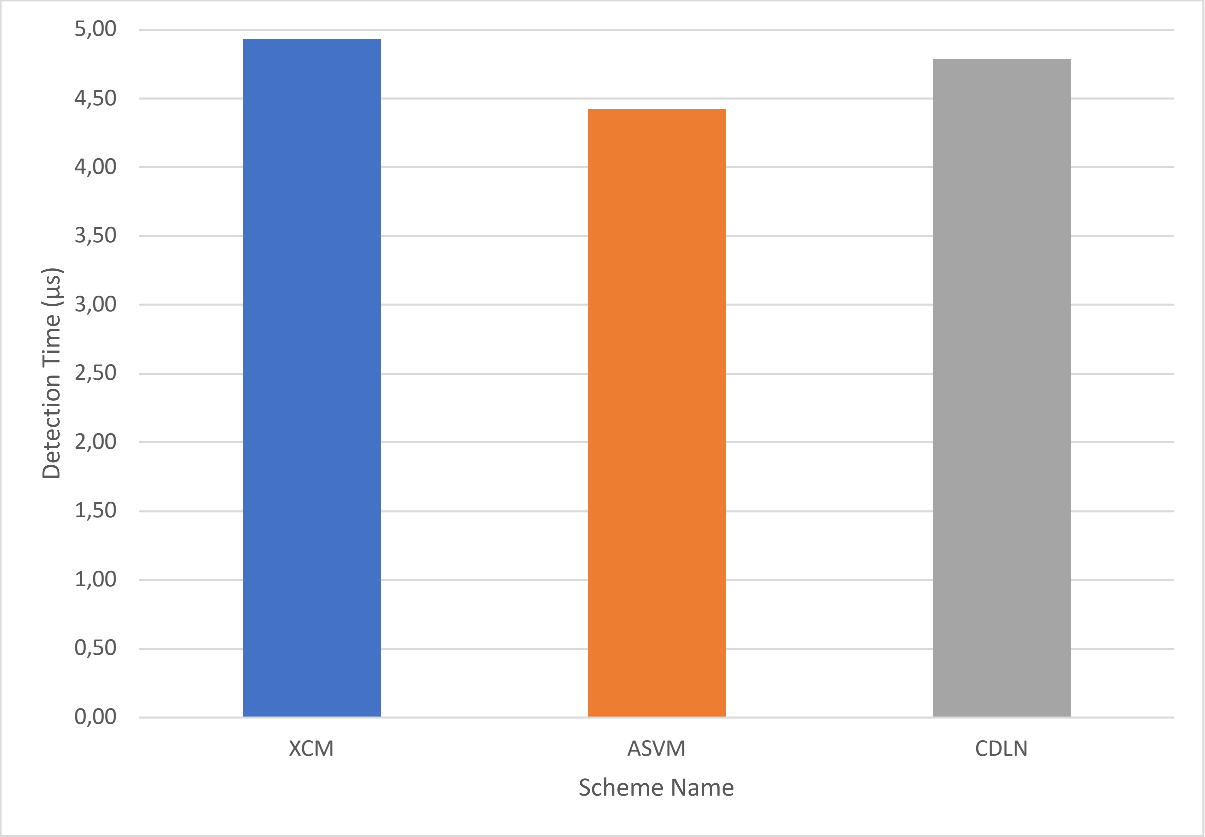


Figure 4.2: Detection Time based on each scheme

From the schemes' comparative results in Table 4.5 and Figure 4.2, XCM is found to take longer to detect DDoS and PUE, followed by CDLN to detect PUE and finally ASVM to detect DDoS. As stated in the works of [90], a shorter DT is required if any security scheme is to be considered as a better performing scheme. Thus, in this case, we can conclude that ASVM is the best one as it can detect DDoS attack with shorter time than CDLN for PUE and XCM for both DDoS and PUE. Furthermore, we can conclude that the CDLN scheme also performs better in terms of detection time than the XCM scheme.

4.5.1.2. Detection rate

Detection Rate (DR) is defined in [16] as "the correct rate for detecting malicious attacks/traffics". To determine the DR based on our XCM scheme, a confusion matrix with all four cases for this research study was generated and is depicted as shown in Figure 4.3. A confusion matrix is reported in [99] as "a table with rows that show the true class and columns that show the predicted class". The diagonal cells show where the true class and predicted class match. Thus, a confusion matrix presents off-diagonal elements as the percentage of incorrectly classified observations while diagonal elements as the percentage of those observations correctly classified in simple terms.

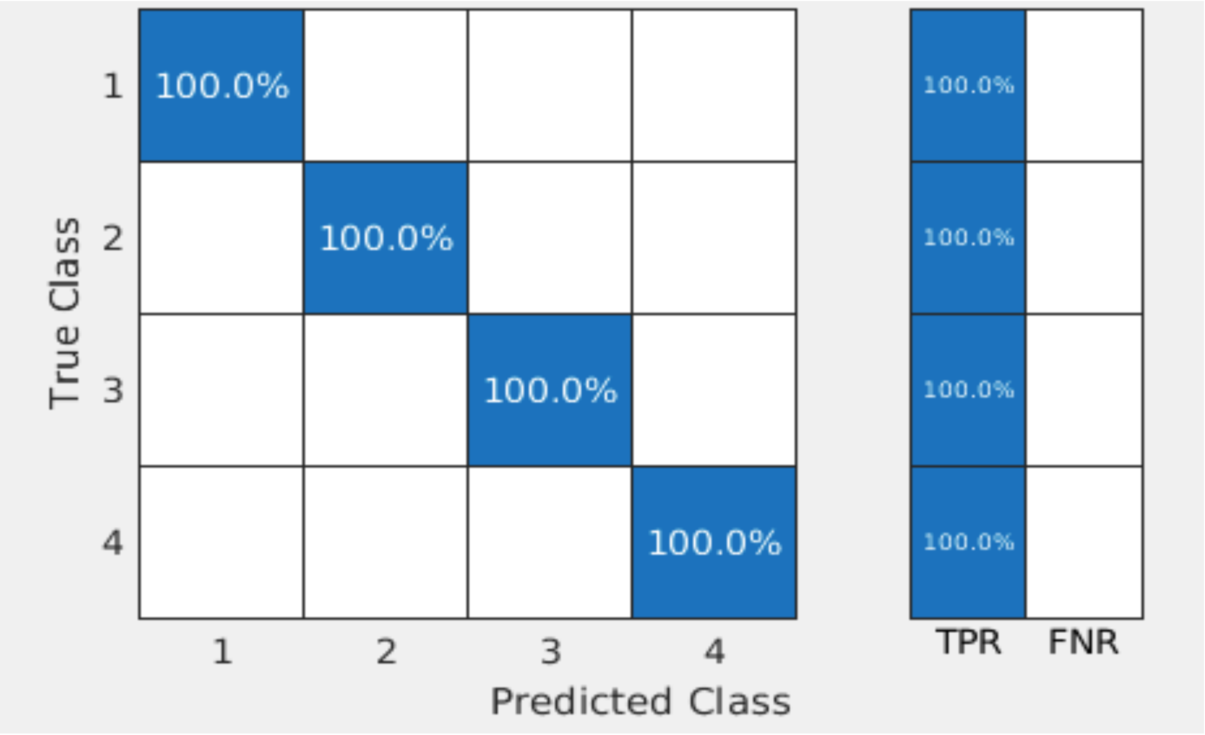


Figure 4.3: Confusion Matrix for Non-Malicious and Malicious Attacks

Based on Figure 4.3, we can observe that the DR rate for each case is 100%. This is concurred by the intersection of true class and predicted class for each own case as 100%. For example, the intersection of true class = 1 (normal traffic) versus predicted class = 1 (normal traffic) is 100%, and the intersection of true class = 2 (DDoS attack) versus predicted class = 2 (DDoS attack) is 100%. This is also similar for the intersection of true class = 3 (PUE attack) versus predicted class = 3 (PUE attack), and the intersection of true class = 4 (DDoS and PUE attacks) versus predicted class = 4 (DDoS and PUE attacks) respectively. Overall, these results from Figure 4.3 show that the XCM scheme performs very well in detecting non-malicious and malicious attacks correctly.

Furthermore, we also analytically calculated the DR for each case based on the DR formula (See Equation 4.1) as provided in studies such as [16], [94], and [95]. This DR formula can be used to calculate the DR for any security scheme, and the DR values should always be high for the scheme to be deemed the better performing one [16].

$$\text{DR} = \frac{\boldsymbol{TP}}{\boldsymbol{FN+TP}} \text{ x } 100\%$$

Equation 4.1: Detection Rate

where DR = Detection Rate, TP = True Positive and FN = false negative.

Cases 4.1, 4.2, 4.3 and 4.4 show the analytical DR results based on our study's 4 cases: no attack, DDoS attack, PUE attack, and DDoS and PUE attacks, respectively. The values for each case, TP and FN, were obtained from Figure 4.3.

Case 4.1: No Attack

$$\text{DR} = \frac{TP}{FN+TP} \text{ x } 100\% = \frac{100}{0+100} \text{ x } 100\% = 100\%$$

Case 4.2: DDoS Attack

$$\text{DR} = \frac{TP}{FN+TP} \text{ x } 100\% = \frac{100}{0+100} \text{ x } 100\% = 100\%$$

Case 4.3: PUE Attack

$$\text{DR} = \frac{TP}{FN+TP} \text{ x } 100\% = \frac{100}{0+100} \text{ x } 100\% = 100\%$$

Case 4.4: DDoS and PUE Attacks

$$\text{DR} = \frac{TP}{FN+TP} \text{ x } 100\% = \frac{100}{0+100} \text{ x } 100\% = 100\%$$

Overall, we can state that the DR based on our XCM scheme is very high (100%). As mentioned in [16], any security scheme that yields a high DR is deemed a better performing scheme. However, to determine whether our XCM scheme is the best performing scheme in terms of DR, the schemes' DR comparative analysis was performed as presented in section 4.5.2.1 through the use of Analysis of Variance (ANOVA) test (see Section 4.5.2.1).

4.5.1.3. False positive

FP is defined in [16] and [96] as "the amount of network traffic that is incorrectly detected and forwarded". In our study, FP results are presented using a ROC curve (Figure 4.4). A ROC curve is "a technique for visualising, organising and selecting classifiers based on their performance" [99]. Ideally, when the ROC curve is closer to the top-left corner, the better the performance. Results show that our XCM scheme managed to yield an FPR and TPR of 0% and 100%, respectively. This is seen with the point (0.00,1.00) on the ROC curve meaning FPR of 0.00 x 100% = 0% and TPR of 1.00 x 100% = 100%. This demonstrates that our XCM scheme managed to yield a perfect classification in non-malicious and malicious attacks. Furthermore, these results are also supported by the performance metric, Area Under Curve (AUC), which resulted in 100% since the AUC = 1.00 (See Figure 4.4), meaning AUC of 1.00 x 100% = 100%. Studies such as [99] noted that AUC values range between 0 and 1 (inclusive), and an AUC value that is closer to 100% illustrates the closeness to the perfection of a scheme.

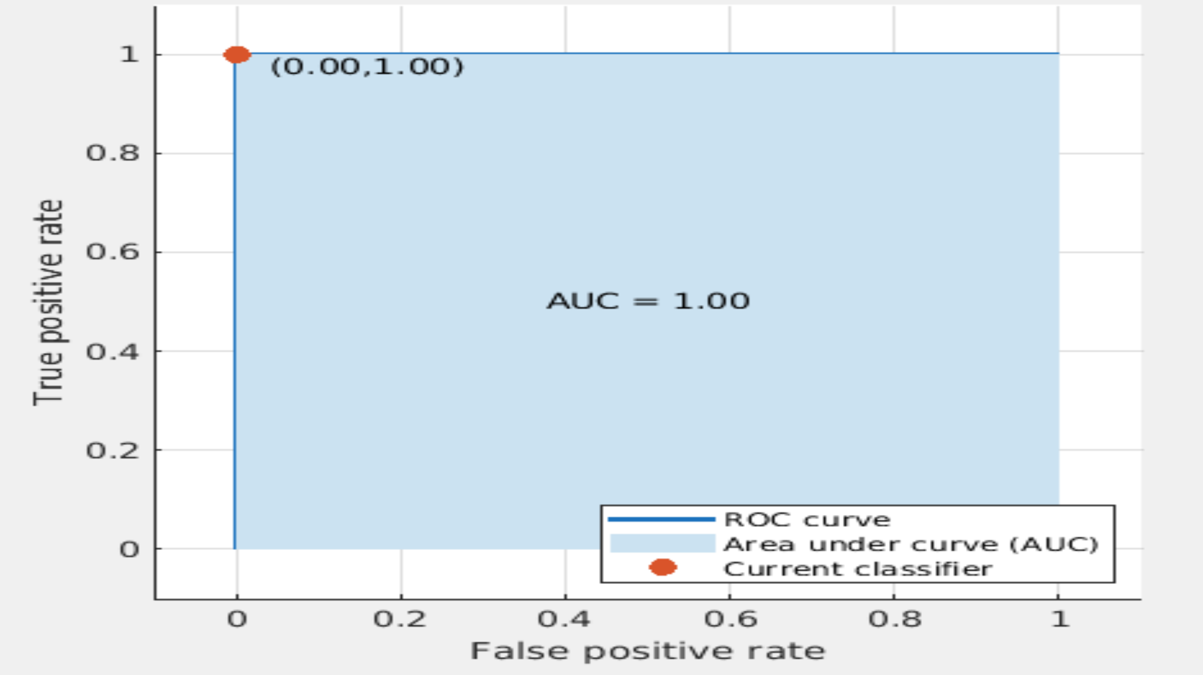


Figure 4.4: ROC Curve for Non-Malicious and Malicious Attacks

From Figure 4.4, based on the FPR and AUC results, we can conclude that our XCM scheme is consistent with the expected FPR and AUC results for any security scheme to be considered acceptable [90], [91]. Authors of [90] and [91] reported that any security scheme must produce low FPR and high AUC for it to be considered a good performing scheme. However, to determine whether our XCM scheme is performing well in terms of FP, an ANOVA test was carried out as presented in section 4.5.2.2 amongst the three schemes, namely the XCM ASVM and CDLN. (see Section 4.5.2.2).

4.5.1.4. False Negative

In [16] and [96], FN is defined as "the amount of network traffic that is incorrectly detected and dropped". It is also defined as a failure by the scheme to report an attacker when it appears or is present [90]. Low FN values are deemed appropriate on any security scheme to be considered a better performing scheme [89], [90]. In order to determine the FN results based on our XCM scheme, the FN formula given in [100] was used:

$$FN + TP = 100\%.$$

Equation 4.2: False Negative

where FN = false negative and TP = True Positive.

Cases 4.5, 4.6, 4.7 and 4.8 provide the analytical FN results based on our study's 4 cases: no attack, DDoS attack, PUE attack, and DDoS and PUE attacks, respectively. The values for each case TP were obtained from Figure 4.3.

Case 4.5: No Attack

$$FN + TP = 100\%$$

FN = 100% - TP, but TP = 100% for class 1 = no attack

$$FN = 100\% - 100\%$$

$$FN = 0.00\%$$

Case 4.6: DDoS Attack

$$FN + TP = 100\%$$

FN = 100% - TP, but TP = 100% for class 2 = DDoS attack

$$FN = 100\% - 100\%$$

FN = 0.00%

Case 4.7: PUE Attack

FN + TP = 100%

FN = 100% - TP, but TP = 100% for class 3 = PUE attack

FN = 100% - 100%

FN = 0.00%

Case 4.8: DDoS and PUE Attacks

FN + TP = 100%

FN = 100% - TP, but TP = 100% for class 4 = DDoS and PUE attacks

FN = 100% - 100%

FN = 0.00%

Overall, we can state that the FN based on our XCM scheme is low (0%). As mentioned in [89] and [90], any security scheme that yields a low FN is regarded as a good performing scheme. However, to determine whether our XCM scheme is the best performing scheme in terms of FN, our XCM scheme was compared against other two related schemes, ASVM and CDLN, via the ANOVA test (see Section 4.5.2.3). The results of this comparative analysis in terms of the FN metric are presented in section 4.5.2.3.

4.5.1.5. Memory utilisation

Authors in [92], [93] and [97] described memory utilisation as the space usage on the RAM. In these studies, the authors noted that the scheme should have a low memory utilisation as this signifies a lightweight scheme, i.e., utilises less space. Our study found that memory utilisation remained constant between 0.50% and 1.60% for any type of attack (see Figure 4.5).

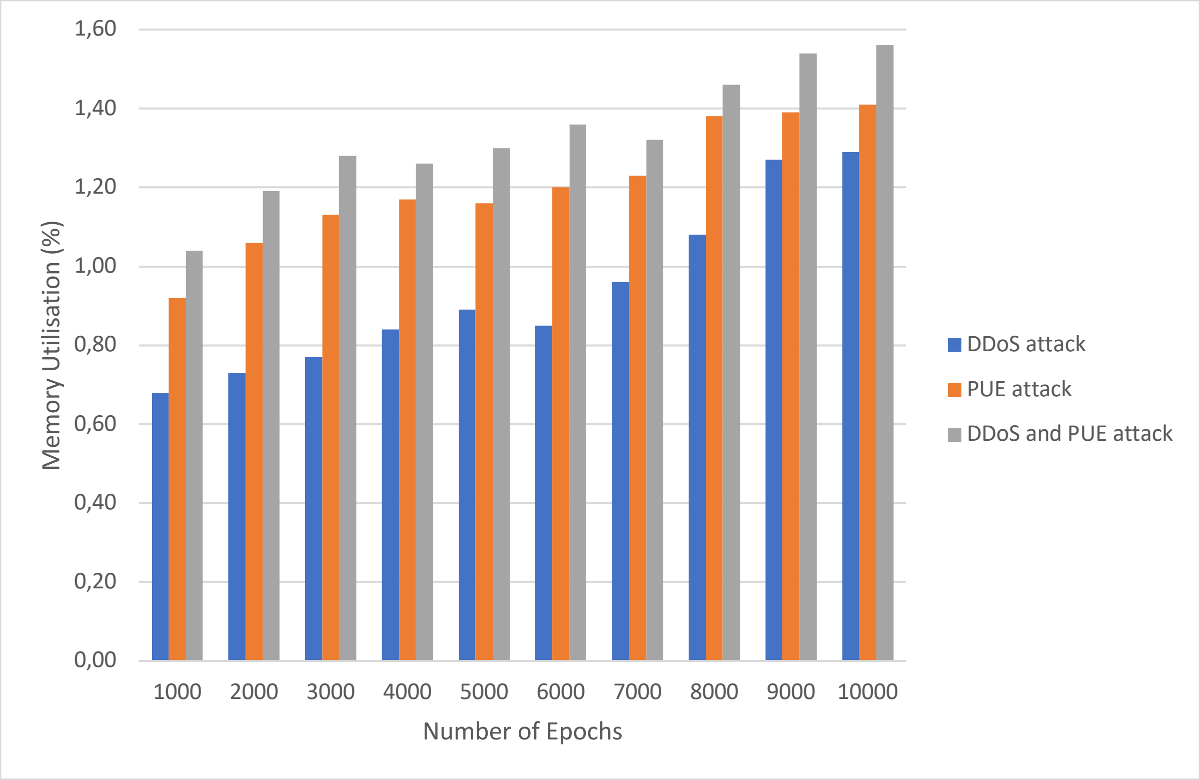


Figure 4.5: Memory Utilisation

Although the results based on Figure 13 remained constant within a range of 0.50% and 1.60%, thus symbolising low memory utilisation by our XCM scheme in the presence of DDoS and PUE attacks. It was essential to perform a comparative analysis amongst the XCM, ASVM and CDLN schemes to determine which scheme performs the best in detecting and preventing DDoS and PUE attacks while consuming less memory. The results based on this comparative analysis in terms of memory utilisation are in section 4.5.2.4 (see Section 4.5.2.4).

4.5.1.6. CPU/processor utilisation

CPU utilisation, sometimes known as processor utilisation, is noted in [92] and [97] as the computer's processing time during programme execution. Authors in [92] and [97] reported that any network security scheme should result in low CPU utilisation in order to be deemed reasonably better performing scheme. The CPU/processor utilisation results show that it remained between 0.60% and 2.00% for any attack type (see Figure 4.6).

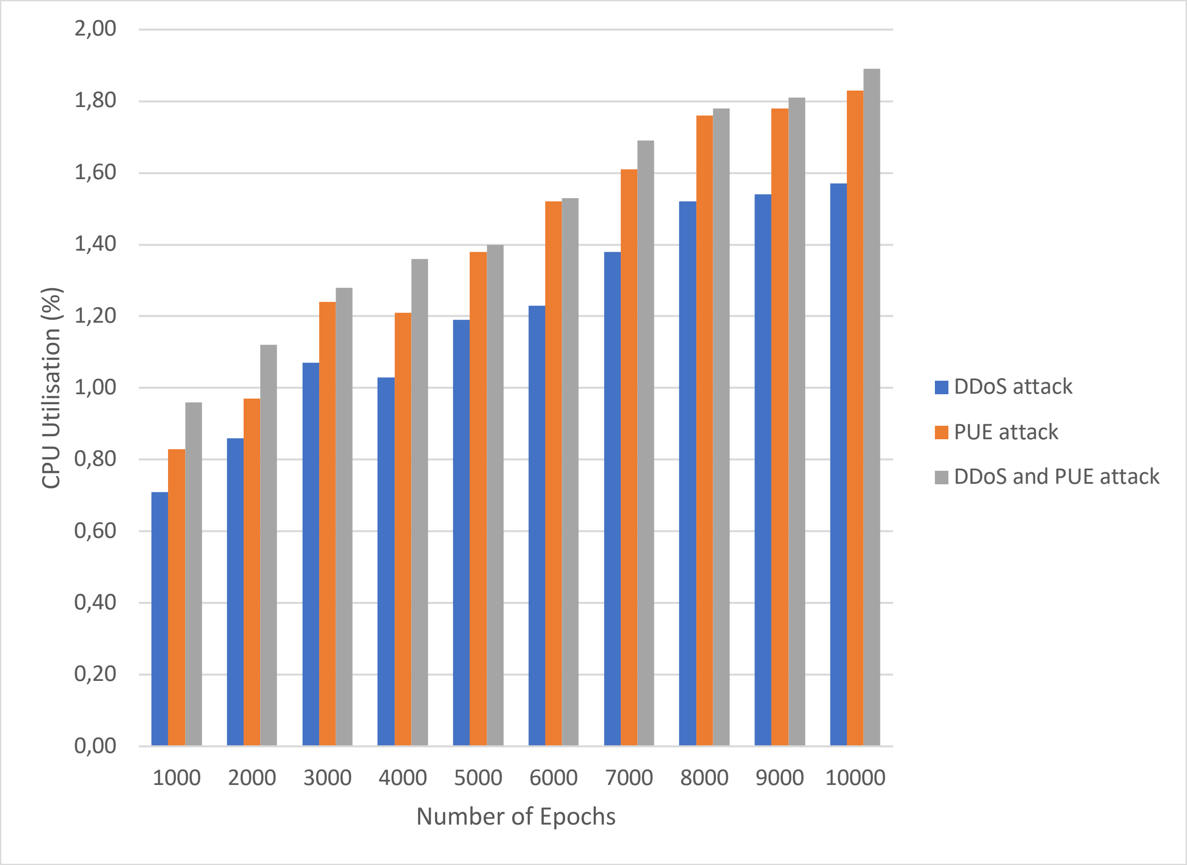


Figure 4.6: CPU/Processor Utilisation

Based on the CPU utilisation results, we can observe that our XCM scheme use low processing time (between 0.60% to 2.00%). However, to determine whether our XCM scheme has a low processor utilisation in terms of detecting the DDoS and PUE attacks, a comparative analysis was carried out against other two related schemes, which are ASVM and CDLN. This comparative analysis results in terms of CPU/processor utilisation in presence of DDoS and PUE attacks are presented in section 4.5.2.5 (see Section 4.5.2.5).

The next section that follows presented the comparative analysis and results of our XCM scheme against other two related schemes: ASVM [16] and CDLN [70] that were also built through the incorporation of ML algorithms. The comparative analysis was carried out using this study evaluation metrics, namely DR, FP, FN, memory and processor utilisation.

4.5.2. Schemes' Comparisons by Analysis of Variance Test

This section presents ANOVA test findings to determine whether our proposed XCM scheme is the best performing scheme as compared with the ASVM [16] and CDLN [70]. ANOVA test was applied because it is reported in [101] as the appropriate t-test method when comparing more than two populations. Furthermore, the authors in [101] also reported that the use of ANOVA allows testing several means simultaneously via a single F-test, thereby presenting a comparison of multiple populations at once. Due to having more than two schemes under consideration, it was found that the ANOVA is a suitable method to be applied in the schemes' comparisons.

Although the results (section 4.5.1) based on the XCM scheme can confirm the ability to detect and protect SDCRN from DDoS and PUE attacks, it was fundamental to this study to compare our XCM scheme against other related schemes such as ASVM [16] and CDLN [70]. These two schemes were selected because they were designed to detect and protect the SDN and CRN from DDoS and PUE attacks, respectively, in the same environment (SDCRN). Additionally, these two schemes were also built by incorporating ML algorithms, which is comparable to our XCM scheme built through the incorporation of ML algorithms in the form of NN concepts.

In order to conduct the ANOVA test, the ML algorithms of ASVM [16] and CDLN [70] were implemented, and their values based on the DR, FP, FN, memory and processor utilisation metrics were recorded against those of XCM (see tables 4.6, 4.8, 4.10, 4.12 and 4.15 respectively). The next five subsections that follow present the comparative analysis and results based on these three schemes, the XCM, ASVM and CDLN using DR, FP, FN, memory and processor utilisation metrics.

4.5.2.1. Schemes' Comparative Analysis and Results based on DR Metric

DR is defined in [16] as the correct rate for detecting malicious attacks/traffics. As mentioned in [16], any security scheme that yields a high DR is deemed a better performing scheme. To determine which is the best performing scheme in terms of DR metric, each Schemes' Algorithm at different split rate intervals were recorded and tabulated as in Table 4.6.

Table 4.6: Detection Rates (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	99	100	99
0.2	97	93	98
0.3	98	98	96
0.4	99	97	98
0.5	97	99	97
0.6	99	99	98
0.7	100	99	96
0.8	99	96	98
0.9	98	97	94

Furthermore, Figure 4.7 depict the same detection rates based on each Schemes' Algorithm at different split rate intervals. Figure 4.7 illustrate that all the three schemes managed to yield detection rates of at least 90% in the presence of the two attacks.

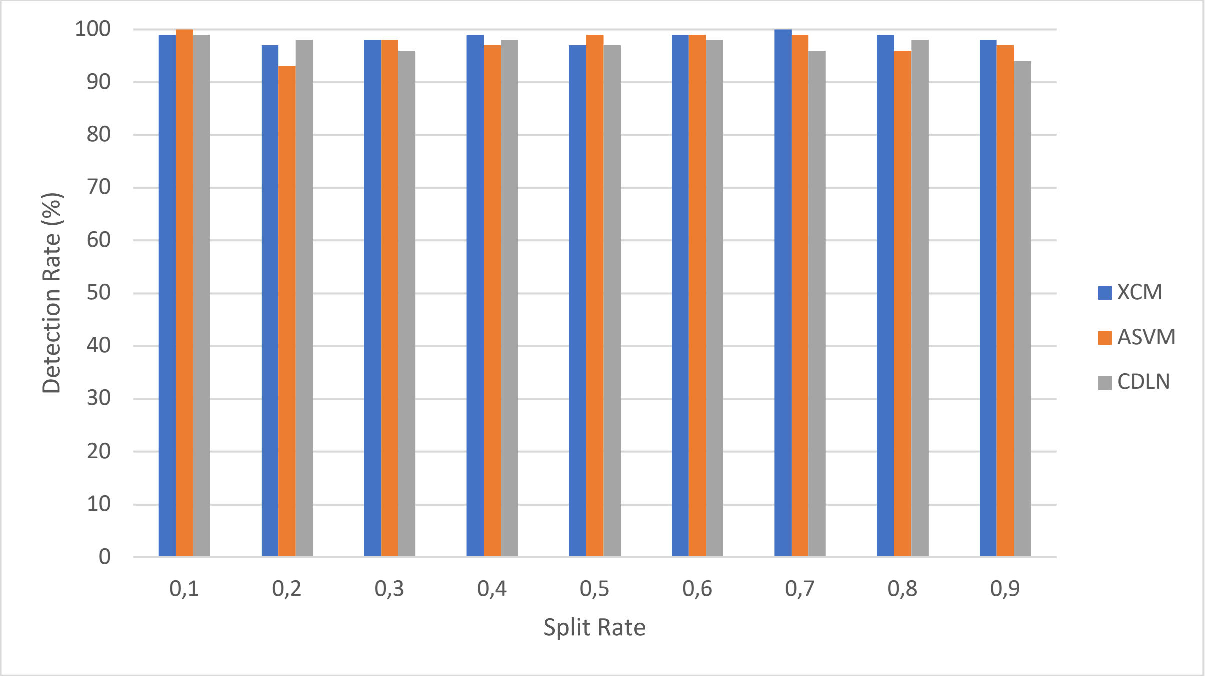


Figure 4.7: Detection Rates based on each Schemes' Algorithm

Based on the DR data in table 4.6 and Figure 4.7, an ANOVA test was performed, and its output results are as presented in Table 4.7.

Table 4.7: ANOVA Summary Output Results – Detection Rates

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	886	98.44444444	1.027777778
ASVM	9	878	97.55555556	4.527777778
CDLN	9	874	97.11111111	2.361111111

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.296296296	2	4.148148148	1.571929825	0.228284113	3.402826105
Within Groups	63.33333333	24	2.638888889			
Total	71.62962963	26				

From this ANOVA summary output results, a hypothesis testing presented as hypothesis testing 1 was carried out to compare the proposed XCM scheme with the ASVM scheme for DDoS and CDLN scheme for PUE.

Hypothesis Testing 1: Comparing Schemes' based on DR

Step 1: Formulation

H$_0$: μ_1 = μ_2 = μ_3

H$_1$: at least two μ's are different

- Let μ_1, μ_2 and μ_3 denote the DR averages of XCM, ASVM and CDLN schemes respectively.

Step 2: Test Statistic

$F_{\text{calculated}} = \frac{MS(between)}{MS(within)} = \frac{4.148148148}{2.638888889} = 1.572$

Step 3: P-Value

p-value = 0.228

Step 4: Decision

Reject H$_0$ if p-value < 0.05 (5% level of significance)

0.228 < 0.05 (False)

Step 5: Conclusion

We do not reject H$_0$ at 5% level of significance and conclude that there is insufficient evidence to suggest that at least two μ's are different.

Since we failed to reject H_0 and concluded that there is insufficient evidence to suggest that at least two μ's are different, this indicated that the three schemes do not differ in the detection and protection of SDCRN from DDoS and PUE attacks based on DR metric. Therefore, we can conclude that the XCM, ASVM and CDLN schemes all perform equally in detecting and protecting SDCRN from DDoS and PUE attacks based on DR metric.

4.5.2.2. Schemes' Comparative Analysis and Results based on FP Metric

FP is defined in [16] and [96] as "the amount of network traffic that is incorrectly detected and forwarded". As stated by authors in [90] and [91], any network security scheme must produce low FP for it to be considered a good performing scheme. To determine which is the best performing scheme in terms of FP metric, each Schemes' Algorithm at different split rate intervals were recorded and tabulated as in Table 4.8.

Table 4.8: False Positives (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	2	0	1
0.2	3	6	3
0.3	3	2	5
0.4	1	2	3
0.5	4	1	3
0.6	1	1	2
0.7	0	1	2
0.8	1	3	2
0.9	2	2	4

In addition, the same false positives recorded in Table 4.8 were also presented in form of a bar chart (see Figure 4.8). Figure 4.8 show that all the three schemes managed to yield false positives of at most 6% in the presence of DDoS and PUE attacks.

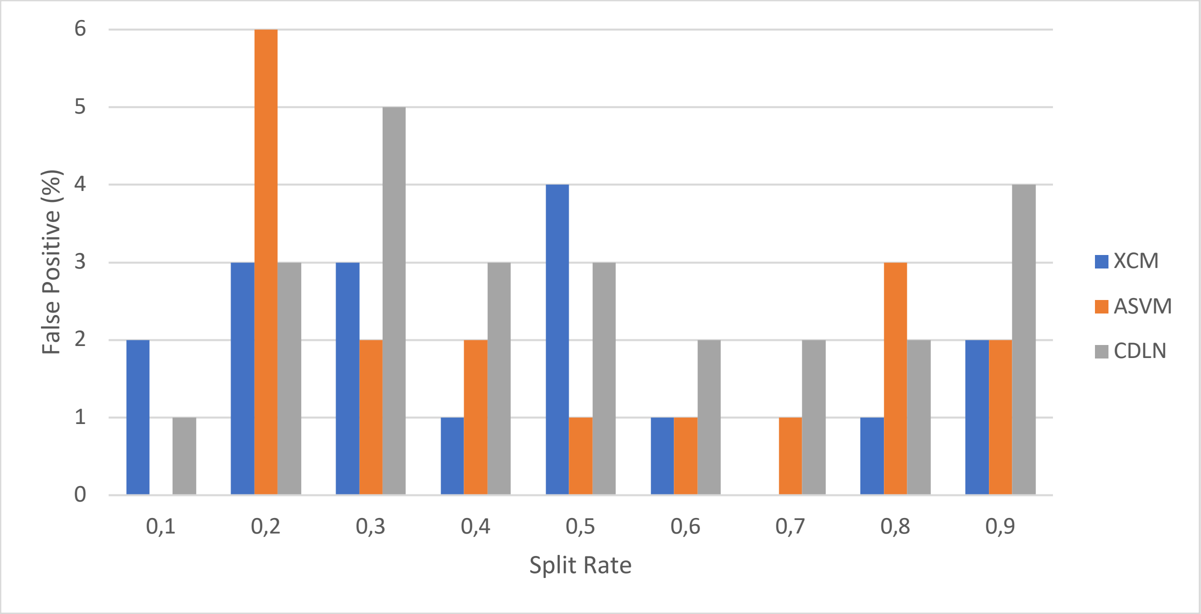


Figure 4.8: False Positives based on each Schemes' Algorithm

Based on the FP data in Table 4.8 and Figure 4.8, an ANOVA test was performed, and its output results is as presented in Table 4.9.

Table 4.9: ANOVA Summary Output Results – False Positives

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	17	1.888888889	1.611111111
ASVM	9	18	2	3
CDLN	9	25	2.777777778	1.444444444

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	4.222222222	2	2.111111111	1.04587156	0.366857812	3.402826105
Within Groups	48.44444444	24	2.018518519			
Total	52.66666667	26				

From this ANOVA summary output results, a hypothesis testing presented as hypothesis testing 2 was carried out to compare the proposed XCM scheme with the ASVM scheme for DDoS and CDLN scheme for PUE.

Hypothesis Testing 2: Comparing Schemes' based on FP

Step 1: Formulation

H$_0$: $\mu_4 = \mu_5 = \mu_6$

H$_1$: at least two μ's are different

- Let μ_4, μ_5 and μ_6 denote the FP averages of XCM, ASVM and CDLN schemes, respectively.

Step 2: Test Statistic

$F_{\text{calculated}} = \frac{MS(between)}{MS(within)} = \frac{2.111111111}{2.018518519} = 1.046$

Step 3: P-Value

p-value = 0.367

Step 4: Decision

Reject H$_0$ if p-value < 0.05 (5% level of significance)

0.367 < 0.05 (False)

Step 5: Conclusion

We do not reject H$_0$ at 5% level of significance and conclude that there is insufficient evidence to suggest that at least two μ's are different.

Since we failed to reject H$_0$ and concluded that there is insufficient evidence to suggest that at least two μ's are different, this indicated that the three schemes do not differ in the detection and protection of SDCRN from DDoS and PUE attacks based on FP metric. In other words, the XCM, ASVM and CDLN schemes all perform equally in the detection and protection of SDCRN from DDoS and PUE attacks based on FP metric.

4.5.2.3. Schemes' Comparative Analysis and Results based on FN Metric

In [16] and [96], FN is defined as "the amount of network traffic incorrectly detected and dropped". Also, authors in [89] and [90] weighed in by defining FN as a failure by the scheme to report an attacker when it appears or is present, and low FN values are deemed appropriate for any network security scheme. To determine whether our XCM scheme performed the best against ASVM and CDLN, FN values at different split rate intervals were recorded and tabulated as in Table 4.10.

Table 4.10: False Negatives (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	3	0	4
0.2	2	3	2
0.3	4	4	5
0.4	2	2	2
0.5	1	3	3
0.6	1	2	2
0.7	0	1	3
0.8	2	3	4
0.9	3	2	2

Moreover, the same false negatives recorded in Table 4.10 are presented graphically as a bar chart (see Figure 4.9). Figure 4.9 show that all the three schemes managed to yield false negatives of at most 5% in the presence of DDoS and PUE attacks.

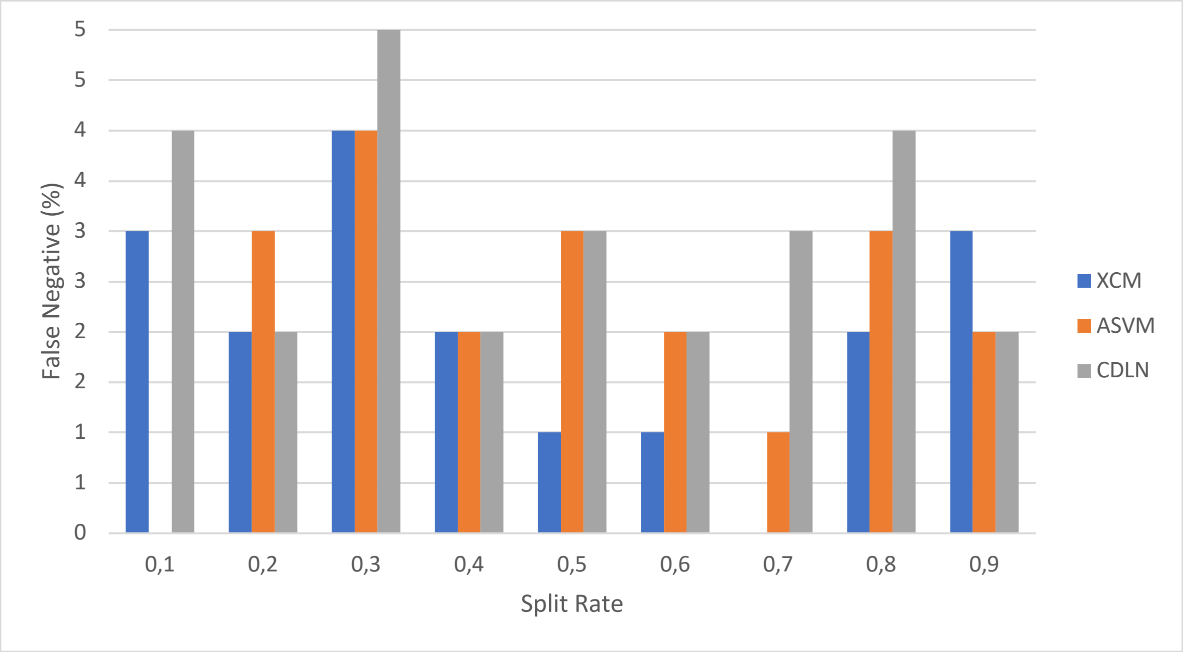


Figure 4.9: False Negatives based on each Schemes' Algorithm

Based on the FN data in table 4.10 and Figure 4.9, an ANOVA test was performed, and its output results are as presented in Table 4.11.

Table 4.11: ANOVA Summary Output Results – False Negatives

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	18	2	1.5
ASVM	9	20	2.222222222	1.444444444
CDLN	9	27	3	1.25

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	4.962962963	2	2.481481481	1.774834437	0.191046155	3.402826105
Within Groups	33.55555556	24	1.398148148			
Total	38.51851852	26				

From this ANOVA summary output results, a hypothesis testing presented as hypothesis testing 3 was carried out in order to provide a comparison of the proposed XCM scheme with the ASVM scheme for DDoS and CDLN scheme for PUE.

Hypothesis Testing 3: Comparing Schemes' based on FN

Step 1: Formulation

H_0: $\mu_7 = \mu_8 = \mu_9$

H_1: at least two μ's are different

- Let μ_7, μ_8 and μ_9 denote the FN averages of XCM, ASVM and CDLN schemes respectively.

Step 2: Test Statistic

$F_{\text{calculated}} = \frac{MS(between)}{MS(within)} = \frac{2481481481}{1.398148148} = 1.775$

Step 3: P-Value

p-value = 0.191

Step 4: Decision

Reject H_0 if p-value < 0.05 (5% level of significance)

0.191 < 0.05 (False)

Step 5: Conclusion

We do not reject H_0 at 5% level of significance and conclude that there is insufficient evidence to suggest that at least two μ's are different.

Since we failed to reject H_0 and concluded that there is insufficient evidence to suggest that at least two μ's are different, this indicated that the three schemes do not differ in the detection and protection of SDCRN from DDoS and PUE attacks based on FN metric. In essence, the XCM, ASVM and CDLN schemes all perform equally in detecting and protecting SDCRN from DDoS and PUE attacks based on FN metric.

4.5.2.4. Schemes' Comparative Analysis and Results based on Memory Utilisation

Authors in [92], [93] and [97] described memory utilisation as the space usage on the RAM. In these studies, the authors noted that the scheme should have a low memory utilisation as this signifies a lightweight scheme, i.e., utilises less space. To determine whether our XCM scheme performed the best against ASVM and CDLN, memory utilisation values at different epochs were recorded and tabulated as in Table 4.12.

Table 4.12: Memory Utilisations (%) based on each Schemes' Algorithm

Epochs	XCM	ASVM	CDLN
1000	1.04	1.41	1.47
2000	1.19	1.48	1.52
3000	1.28	1.54	1.65
4000	1.26	1.62	1.59
5000	1.30	1.57	1.68
6000	1.36	1.71	1.72
7000	1.32	1.73	1.81
8000	1.46	1.86	1.90
9000	1.54	1.82	1.89
10000	1.56	1.87	1.94

Furthermore, Figure 4.10 depict the same memory utilisations based on each Schemes' Algorithm at different epochs. Figure 4.10 show that all the schemes were able to yield less than 2% in memory utilisation.

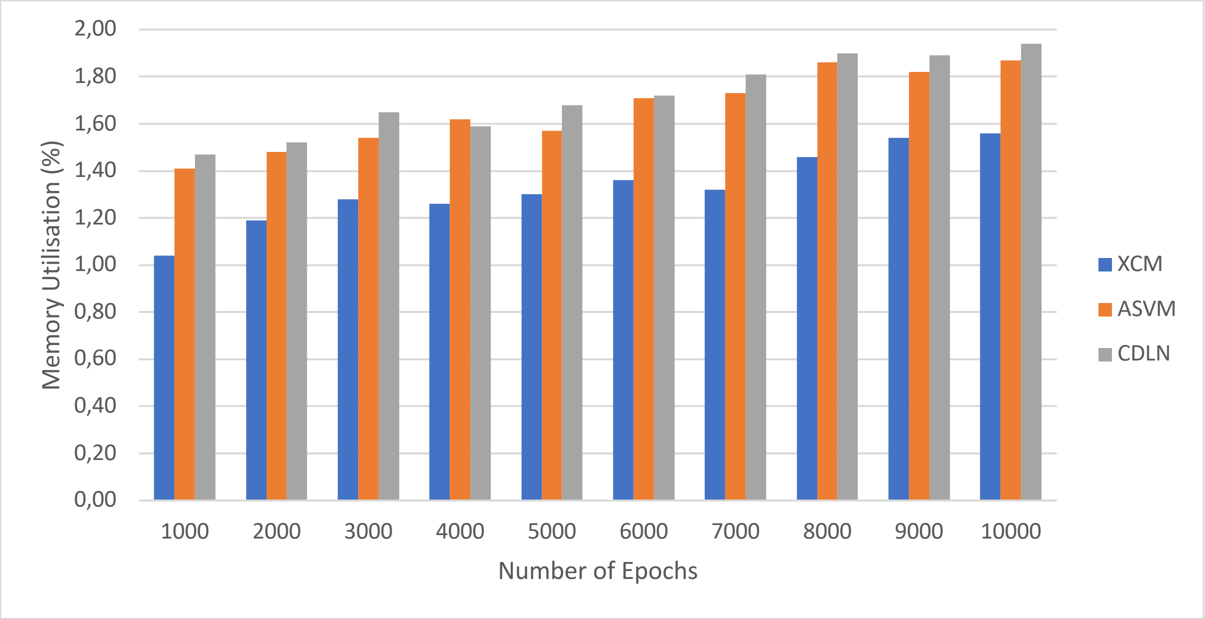


Figure 4.10: Memory Utilisations based on each Schemes' Algorithm

Based on the memory utilisation data in Table 4.12 and Figure 4.10, an ANOVA test was performed, and its output results are as presented in Table 4.13.

Table 4.13: ANOVA Summary Output Results – Memory Utilisations

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	10	13.31	1.331	0.02521
ASVM	10	16.61	1.661	0.026232222
CDLN	10	17.17	1.717	0.027067778

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.870106667	2	0.435053333	16.62412432	1.96862E-05	3.354130829
Within Groups	0.70659	27	0.02617			
Total	1.576696667	29				

From this ANOVA summary output results, a hypothesis testing presented as hypothesis testing 4 was carried out in order to provide a comparison of the proposed XCM scheme with the ASVM scheme for DDoS and CDLN scheme for PUE.

Hypothesis Testing 4: Comparing Schemes' based on memory utilisation

Step 1: Formulation

H$_0$: $\mu_{10} = \mu_{11} = \mu_{12}$

H$_1$: at least two μ's are different

- Let μ_{10}, μ_{11} and μ_{12} denote the memory utilisation averages of XCM, ASVM and CDLN schemes respectively.

Step 2: Test Statistic

$F_{\text{calculated}} = \frac{MS(between)}{MS(within)} = \frac{0.435053333}{0.02617} = 16.624$

Step 3: P-Value

p-value = 0.000

Step 4: Decision

Reject H$_0$ if p-value < 0.05 (5% level of significance)

0.000 < 0.05 (True)

Step 5: Conclusion

We reject H$_0$ at 5% level of significance and conclude that there is sufficient evidence to suggest that at least two μ's are different. This indicates that the three schemes differ in the detection and protection of SDCRN from DDoS and PUE attacks based on memory utilisation.

To further establish that a difference exists amongst the three schemes under consideration, a technique named the Tukey's test was performed. Authors in [101] stated that a Tukey's test is used to find the means (averages) significantly different from each other. Since in this study we were interested in determining which scheme is the best compared to the others in the detection of DDoS and PUE attacks in SDCRN, a Tukey's test was found to be suitable and applicable. The Tukey Test 1 presented to illustrate the steps, analysis, and results to determine which scheme is the best in terms of memory utilisation in detecting and preventing DDoS and PUE attacks in the SDCRN.

Tukey Test 1: Memory Utilisation

Step 1: List all the sample means in ascending order (see Table 4.14).

Step 2: Calculate absolute differences between pairs of means (see Table 4.14).

Table 4.14: Means and Absolute Differences – Memory Utilisation

		Absolute Differences		
	Means	**XCM**	**ASVM**	**CDLN**
XCM	1.331	0		
ASVM	1.661	0.330	0	
CDLN	1.717	0.386	0.056	0

Also, Figure 4.11 depict the averages for XCM, ASVM and CDLN in terms of memory utilisation. From these results, we can see that the XCM scheme has the lowest average memory utilisation, followed by the ASVM scheme and CDLN scheme.

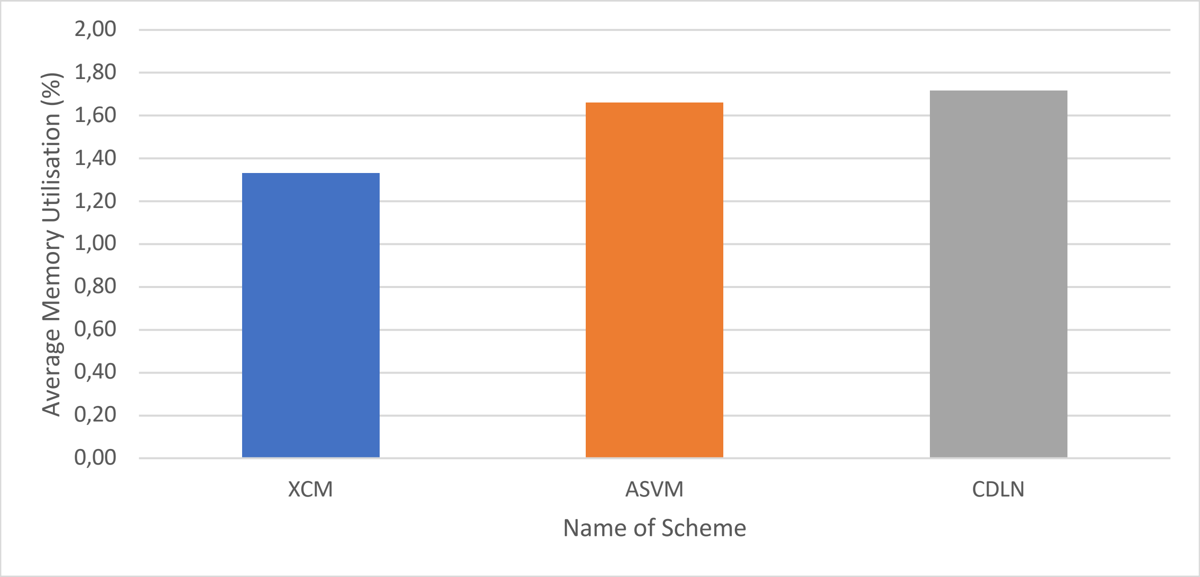


Figure 4.11: Average Memory Utilisation for XCM, ASVM and CDLN

Step 3: Find Q$\alpha_{,j,n-j}$ from Table of Percentage points of Studentised Range.

Q$\alpha_{,j,n-j}$ = $Q_{0.05,3,30-3}$ = $Q_{0.05,3,27}$ = 3.49, where α = level of significance, j = number of levels and n-j = df (within group variation).

Step 4: Calculate D = Q$\alpha_{,j,n-j}\sqrt{\frac{MS(within)}{nr}}$

D = $3.49\sqrt{\frac{0.02617}{10}}$ = 0.179, where nr = each category sample size.

Step 5: Report conclusion

Criteria: Any absolute difference value greater than D value (0.179) indicates a difference between the two schemes. In this case, we can conclude that a difference exists between XCM and ASVM (0.330 > 0.179) and XCM and CDLN (0.386 > 0.179). However, no difference exists between ASVM and CDLN (0.056 < 0.179). Thus, we can conclude that XCM is the best performing scheme in terms of memory utilisation in detecting and preventing DDoS and PUE attacks. Based on this result, we can state that our XCM scheme is lightweight as it utilises less memory than ASVM and CDLN schemes.

4.5.2.5. Schemes' Comparative Analysis and Results based on CPU/Processor Utilisation

CPU utilisation, sometimes known as processor utilisation, is noted in [92] and [97] as the computer's processing time during programme execution. Authors in [92] and [97] reported that any security scheme should result in low CPU utilisation to be regarded as a good performing scheme. To determine whether our XCM scheme performed the best against ASVM and CDLN, processor utilisation values at different epochs were recorded and tabulated as in Table 4.15.

Table 4.15: CPU/Processor Utilisation (%) based on each Schemes' Algorithm

Epochs	XCM	ASVM	CDLN
1000	0.96	1.43	1.47
2000	1.12	1.57	1.59
3000	1.28	1.59	1.63
4000	1.36	1.68	1.71
5000	1.40	1.75	1.78
6000	1.53	1.84	1.86
7000	1.69	1.87	1.92
8000	1.78	1.91	1.94
9000	1.81	1.93	1.97
10000	1.89	1.97	1.98

Figure 4.12 illustrates the same CPU utilisations based on each Schemes' Algorithm at different epochs. Figure 4.12 shows that all the schemes could yield less than 2% in terms of CPU/processor utilisation.

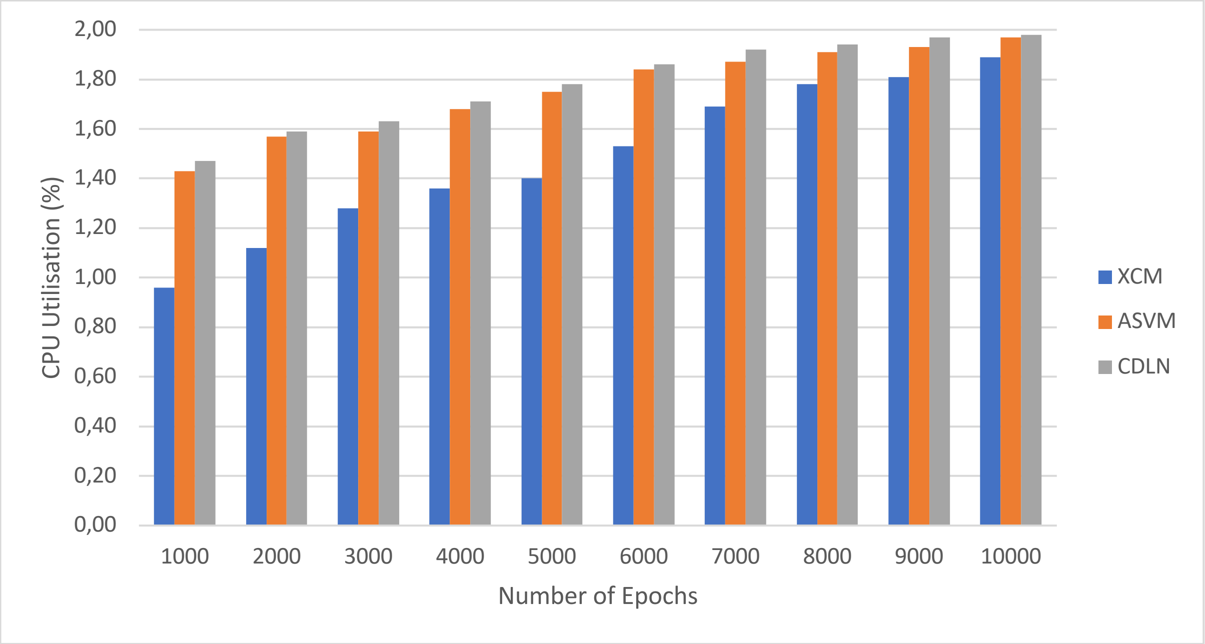


Figure 4.12: CPU/Processor Utilisations based on each Schemes' Algorithm

Based on the CPU/processor utilisation data in Table 15 and Figure 4.12, an ANOVA test was performed, and its output results are as presented in table 4.16.

Table 4.16: ANOVA Summary Output Results – CPU Utilisations

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	10	14.82	1.482	0.097151111
ASVM	10	17.54	1.754	0.032671111
CDLN	10	17.85	1.785	0.031894444

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.555846667	2	0.277923333	5.155745646	0.012692162	3.354130829
Within Groups	1.45545	27	0.053905556			
Total	2.011296667	29				

From this ANOVA summary output results, a hypothesis testing presented as hypothesis testing 5 was carried out to compare the proposed XCM scheme with the ASVM scheme for DDoS and CDLN scheme for PUE.

Hypothesis Testing 5: Comparing Schemes' based on processor utilisation

Step 1: Formulation

H_0: $\mu_{13} = \mu_{14} = \mu_{15}$

H_1: at least two μ's are different

- Let μ_{13}, μ_{14} and μ_{15} denote the processor utilisation averages of XCM, ASVM and CDLN schemes.

Step 2: Test Statistic

$F_{calculated} = \frac{MS(between)}{MS(within)} = \frac{0.277923333}{0.053905556} = 5.156$

Step 3: P-Value

p-value = 0.013

Step 4: Decision

Reject H_0 if p-value < 0.05 (5% level of significance)

0.013 < 0.05 (True)

Step 5: Conclusion

We reject H_0 at 5% level of significance and conclude that there is sufficient evidence to suggest that at least two μ's are different. This indicates that the three schemes differ in the detection and protection of SDCRN from DDoS and PUE attacks based on processor utilisation.

To further establish that a difference exists amongst the three schemes under consideration, a technique named the Tukey's test was performed. Authors in [101] stated that a Tukey's test is used to find the means (averages) significantly different from each other. Since in this study we were interested in determining which scheme is the best compared to the others in the detection of DDoS and PUE attacks in SDCRN, a Tukey's test was found to be suitable and applicable. The Tukey Test 2 presented illustrates the steps, analysis, and results to determine which scheme is the best in terms of processor utilisation in detecting and preventing DDoS and PUE attacks in the SDCRN.

Tukey Test 2: CPU/Processor Utilisation

Step 1: List all the sample means in ascending order (see Table 4.17).

Step 2: Calculate absolute differences between pairs of means (see Table 4.17).

Table 4.17: Means and Absolute Differences – CPU/Processor Utilisation

		Absolute Differences		
	Means	**XCM**	**ASVM**	**CDLN**
XCM	1.482	0		
ASVM	1.754	0.272	0	
CDLN	1.785	0.303	0.031	0

Additionally, Figure 4.13 depict the average CPU utilisation for XCM, ASVM and CDLN schemes. These results show that the CDLN scheme has the highest average CPU utilisation, followed by ASVM scheme and lastly by XCM scheme.

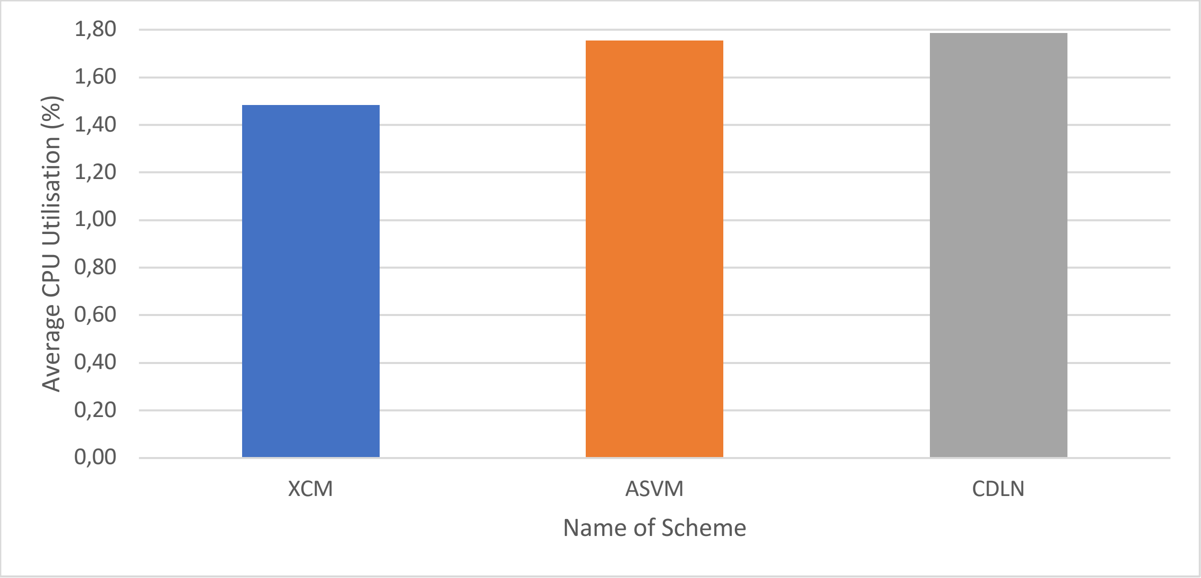


Figure 4.13: Average CPU Utilisation for XCM, ASVM and CDLN

Step 3: Find $Q\alpha_{,j,n-j}$ from Table of Percentage points of the Studentised Range.

$Q\alpha_{,j,n-j}$ = $Q_{0.05,3,30-3}$ = $Q_{0.05,3,27}$ = 3.49, where α = level of significance, j = number of levels and n-j = df (within group variation).

Step 4: Calculate D = $Q\alpha_{,j,n-j}\sqrt{\frac{MS(within)}{nr}}$

D = $3.49\sqrt{\frac{0.053905556}{10}}$ = 0.256, where nr = each category sample size.

Step 5: Report conclusion

Criteria: Any absolute difference value greater than D value (0.256) indicates a difference between the two schemes. In this case, we can conclude that a difference exists between XCM and ASVM (0.272 > 0.256) and XCM and CDLN (0.303 > 0.256). However, no difference exists between ASVM and CDLN (0.031 < 0.256). Therefore, we can conclude that XCM is the best performing scheme in terms of CPU/processor utilisation in detecting and preventing DDoS and PUE attacks. Based on this result, we can state that the XCM scheme is not CPU intensive than ASVM and CDLN schemes.

4.5.3. Summary of Comparative Results Among the Schemes

Table 4.18 summarises the comparative results of the XCM, ASVM and CDLN schemes based on the DT, DR, FP, FN, memory, and processor utilisation. These results are summary statistics from the findings in Sections 4.5.1 and 4.5.2. According to [101], the mean is a not useful statistic in the comparative analysis as outliers or extreme values influence it. Hence, this study used statistical methods and techniques such as ANOVA and Tukey, respectively, in a comparative analysis of this study as they are the most appropriate.

Table 4.18: Comparative Results of XCM, ASVM and CDLN

Scheme Name	**DT (µs) - Frequency**	**DR (%) - Average**	**FP (%) - Average**	**FN (%) - Average**	**Memory Utilisation (%) - Average**	**CPU Utilisation (%) – Average**
XCM	4.93	98.44	1.89	2.00	1.33	1.48
ASVM	4.42	97.56	2.00	2.22	1.66	1.75
CDLN	4.79	97.11	2.78	3.00	1.72	1.79
Method/Technique	Descriptive	Anova	Anova	Anova	Anova and Tukey	Anova and Tukey
Best Scheme	ASVM	All	All	All	XCM	XCM

The results confirm that our XCM scheme is superior to ASVM and CDLN in terms of memory and processor utilisations, whilst ASVM is superior to XCM and CDLN in terms of DT. However, no scheme is the best performing DR, FP, and FN, as all schemes perform equally in detecting and preventing DDoS and PUE attacks. Based on these findings, we can attest that our XCM scheme supersedes the ASVM and CDLN schemes since our XCM scheme is optimised for both DDoS and PUE attacks in SDCRN whilst the ASVM and CDLN schemes are optimised only for DDoS and PUE attacks in SDN and CRN respectively. Thus, our XCM scheme is effective, efficient, and lightweight for detecting and preventing DDoS and PUE attacks in SDCRN.

4.6. Conclusion

This chapter managed to provide the experimental results and analysis of this study. It began with data availability discussion, followed by determining an ideal number of traffic data examples – both legitimate and malicious traffics. It was then tailed to present experimental simulations and their corresponding results based on the six-performance metrics under study consideration: DT, DR, FP, FN, memory and CPU utilisation. The chapter concluded by carrying out ANOVA and Tukey's tests that produced results on which of the three schemes, namely XCM, ASVM and CDLN, would be the best way to detect and prevent DDoS and PUE in SDCRN. The next chapter that follows presents the discussion, conclusions and recommendations about this research study.

CHAPTER 5: DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

5.1. Introduction

This chapter summarises the research study, discussion of the research findings, conclusions, and recommendations based on the experimental simulations conducted in Chapter 4. This study investigated the design of an effective XCM scheme for SDCRN in addressing the effects of DDoS and PUE attacks. An effective, efficient and lightweight security scheme that can utilise less memory and processing time in the mitigation of DDoS and PUE attacks in SDCRN was designed and tested in this research study. The following section presented the summary of the research study.

5.2. Summary of the Research Study

The background of this research study was done by reviewing previous related studies from the literature. According to [6] and [9], SDN and CRN bring in greater functionality for managing networks and efficient use of spectrum, respectively. Thus, the integration of SDN with CRN amalgamates the advantages mentioned above. But the SDN architecture and CRN technology are both vulnerable to DDoS [19], [13] and PUE [14], [24] attacks. Therefore, an integration of SDN with CRN will likely result in the effects of these two attacks compounded as the architecture and technology are already vulnerable to these two attacks.

This research study designed and tested the effectiveness of the XCM scheme in addressing DDoS and PUE attacks in the SDCRN environment. This was inspired by the reality that these two attacks have not been addressed in SDCRN integrated environment whilst the authors such as [13], [14] and [16-18] revealed that these two attacks are the most severe in SDN and CRN.

The research questions of this study were:

i. Which attributes of DDoS and PUE attacks can be detected and measured in SDCRN?

ii. What is the most effective technique that can address the DDoS and PUE in SDCRN?
iii. What is the best strategy for optimising the most effective DDoS and PUE security scheme to efficiently utilise memory and CPU?

The objectives of this research study were:

i. To investigate the network attributes of DDoS and PUE attacks which can be detected and measured.
ii. To explore the most effective technique that can address the DDoS and PUE.
iii. To evaluate the efficiency of the XCM scheme in terms of memory and processor utilisation.
iv. To perform a comparative analysis of the XCM scheme compared to the existing DDoS and PUE schemes designed for SDN and CRN, respectively.

The hypothesis of this research study was:

- the XCM scheme will effectively detect and protect the SDCRN from the effects of DDoS and PUE attacks.

The literature review was provided in Chapter 2 and it looked at the effects of DDoS attacks on SDN as well as the effects of PUE attacks in CRN. The preventive security mechanisms proposed for mitigating the DDoS in SDN and PUE in CRN were thoroughly investigated. Also, the conceptual framework constructed for this study was presented. This conceptual framework was established based on the principle of SDN and CRN commonalities.

The research methodology used in this study was an experimental simulation and its approach was quantitative. The SDCRN network model was proposed, and algorithms developed for traffic dataset generation, i.e., non-malicious (normal) and malicious traffic, with duration and src_bytes as network attributes. We then trained the generated traffic dataset in Octave to teach our XCM scheme designed with the incorporation of NN concepts on detecting and preventing similar attacks in future in similar environments. During the classification phase, the XCM scheme produced four attack outcomes which are DDoS, PUE, DDoS and PUE, and neither DDoS nor PUE (normal traffic).

Following the NN training, the XCM scheme was implemented on the test dataset that was generated in a similar SDCRN environment in MATLAB. This was meant to evaluate its performance in terms of DT, DR, FP, FN, utilisation of memory and processor. These six metrics were used to evaluate the performance of the XCM scheme primarily because studies such as [16], [89-93] regard them as the most effective performance evaluation metrics.

The next section that follows presents our study findings' discussion based on the six-performance metrics cited above. The discussion of our study findings was centred on providing interpretations to our study results through comparing and contrasting them with the results from other related studies that have conducted in the past.

5.3. Results Discussion and Interpretation

In this section, our study findings are presented and discussed. Our study findings are compared and contrasted to other related studies' results. The performance of XCM scheme is displayed in the statistical results as shown in Table 5.1 using the six metrics under consideration for this study: DT, DR, FP, FN, memory, and CPU utilisation.

Table 5.1: Statistical Results for our proposed XCM Scheme Performance Metrics

Performance Metric	**Values**
DT (μs)	4.93
DR (%)	100
FP (%)	0
FN (%)	0
Memory Utilisation (%)	1.33
CPU Utilisation (%)	1.48

The findings indicate that our XCM scheme can detect the DDoS and PUE attacks in SDCRN within a short period (smaller than 5μs). This shows that our XCM scheme has a rapid response time in detecting DDoS and PUE attacks in the SDCRN. Hence, our DT findings are consistent with the works of [90] that also recorded a short period (less than 2 seconds) for PUE attack in CRN. Additionally, [90] noted that for a networking scheme to be classified as a suitable performing security mechanism, a lesser DT is expected.

Our findings also indicate that our XCM scheme achieved a high detection rate of 100%. This demonstrates that our XCM scheme performs very well on correctly detecting non-malicious (normal) and malicious (abnormal) traffics in SDCRN integrated environment. Henceforth, our studies' DR findings are consistent with the findings obtained in studies such as [16] and [70]. In these studies, detection rates of 99% for DDoS in SDN and 97% for PUE in CRN were achieved. Moreover, [16] added that for a security scheme to be considered a good one, the DR values achieved should always be high, thus closer to 100%.

Furthermore, in the presence of DDoS and PUE, the XCM scheme can achieve FP and FN rates of 0%. This shows that our XCM scheme does not misclassify these two attacks when subjected to them in SDCRN integrated environment. Our findings on FP and FN rates are consistent with the studies such as [90] and [91], who also obtained lower FP and FN rates in the presence of either DDoS in SDN or PUE in CRN. Studies as [89] confirmed that FP and FN rates should always be close to 0% for a network security scheme to be considered as a good scheme. Thus, our XCM scheme managed to achieve lower FP and FN rates as per the assertation of [89]. Therefore, we can conclude that our scheme is suitable for securing the SDCRN integrated environment against either DDoS or PUE attacks.

Additionally, lower memory and CPU utilisations were achieved in the studies of [92], [93] and [97]. These results are similar to the results obtained in our studies for memory and CPU utilisation. In our study, the XCM scheme was found to consume less resources in memory and processor time (less than 2%). Thus, our proposed XCM scheme can provide a secure SDCRN integrated environment from DDoS and PUE attacks with lower memory and processor utilisation.

Finally, a comparison was conducted among the XCM, ASVM and CDLN to determine which of them would be the best scheme in detecting and preventing DDoS and PUE attacks in SDCRN. Table 5.2 displays the summary of comparative results of the XCM, ASVM and CDLN schemes based on the DT, DR, FP, FN, memory and processor utilisation.

Table 5.2: Comparative Results of XCM, ASVM and CDLN

Scheme Name	**DT (µs) - Frequency**	**DR (%) - Average**	**FP (%) – Average**	**FN (%) - Average**	**Memory Utilisation (%) – Average**	**CPU Utilisation (%) - Average**
XCM	4.93	98.44	1.89	2.00	1.33	1.48
ASVM	4.42	97.56	2.00	2.22	1.66	1.75
CDLN	4.79	97.11	2.78	3.00	1.72	1.79
Method/Technique	Descriptive	Anova	Anova	Anova	Anova and Tukey	Anova and Tukey
Best Scheme	ASVM	All	All	All	XCM	XCM

The findings indicate that the XCM scheme is the best scheme under memory and processor utilisation, whilst ASVM scheme is the best scheme under DT. The findings also showed that all the three schemes performed the same in terms of DR, FP and FN. However, since our XCM scheme is optimised for both DDoS and PUE attacks, we can conclude that our XCM scheme is superior to ASVM and CDLN, optimised for DDoS in SDN and PUE in CRN.

Overall, our findings show that the results of our XCM scheme are consistent with results of studies by [5], [16] on DDoS attacks results in SDN as well as [24], [70] on PUE attacks result in CRN. Furthermore, since our XCM scheme was designed for the detection and prevention of DDoS and PUE attacks in SDCRN, our results provide evidence for us to conclude that our scheme performs better than other schemes optimised separately for DDoS in SDN and/or PUE in CRN. Hence, our proposed XCM scheme is effective in addressing the effects of the DDoS and PUE attacks in SDCRN integrated environment.

5.4. Future Work and Recommendations

In the future, it is hoped that the researchers of this study will advance the general knowledge about security mechanisms in the mitigation of security attacks in SDCRN by increasing the number of security attacks being investigated. It is also hoped that this research study could serve as a benchmark for future related studies in SDCRN integrated environment.

5.5. Final Conclusion

This study proposed an XCM scheme that effectively address DDoS and PUE attacks in the SDCRN. The results confirms that our XCM scheme took less time of about 4.93 μs in detecting both DDoS and PUE attacks in SDCRN. Also, the results confirm our XCM scheme managed to achieve a high DR of 100% for the DDoS and PUE in SDCRN. Furthermore, the results confirm our XCM scheme accomplished low rates of 0% in terms of FP and FN in the presence of either DDoS or PUE attacks. In terms of resource usage, the results confirm our XCM scheme is lightweight as it uses less resources of 1.33% and 1.48% for memory and processor time, respectively. Additionally, when compared to ASVM optimised for DDoS and CDLN optimised PUE, the results prove our XCM performance supersedes their performance. Finally, our results can attest XCM scheme is effective in detecting and preventing DDoS and PUE attacks in SDCRN integrated environments. The results of the XCM were therefore the best and superior to the ASVM and CDLM. This can be attributed to the fact that the XCM scheme is optimised for DDoS and PUE attacks.

APPENDICES

Appendix A: XCMDeepLearning Function

```
function [W1, W2] = XCMDeepLearning(W1, W2, i, c_O)
a = 0.01;

N = 212;
for j = 1:N
  transposed_I = i(j,:)';

  i_O_H_L = W1 * transposed_I;
  o_O_H_L = XCMSigmoid(i_O_H_L);

  i_O_O_N = W2 * o_O_H_L;
  f_O = XCMSigmoid(i_O_O_N);

  c_O_transpose = c_O(j,:)';
  e = c_O_transpose - f_O;

  d = e;

  e_O_H_L = W2' * d;
  d1 = (i_O_H_L > 0).*e_O_H_L;

  a_O_W2 = a * d * o_O_H_L;
  a_O_W1 = a * d1 * transposed_I;

  W1 = W1 + a_O_W1;
  W2 = W2 + a_O_W2;
end
end
```

Appendix B: XCMSigmoid Function

```
function y = XCMSigmoid(x)
  y = 1/(1+exp(-x));
end
```

Appendix C: XCMTrainingNetwork Script

```
i = [
      6.21        397;
      8.46        568;
      5.41        438;
      5.17        195;
      9.07        513;
      9.44        663;
     10.01        234;
      9.43        310;
      8.39        557;
      9.48        439;
      9.77        110;
      8.99        564;
      6.21        438;
      4.27        336;
      6.97        565;
      7.32        837;
      7.83        130;
      7.74        455;
      5.94        280;
      4.09        790;
      4.07        807;
      9.48        454;
      8.25        633;
      6.90        223;
      6.72        892;
      5.97        321;
      6.06        908;
      4.56        326;
      3.81        397;
      3.69        263;
     10.07        408;
      5.32        677;
      8.55        640;
```

9.49	99;
8.79	447;
5.90	363;
6.13	805;
3.51	543;
5.51	808;
6.53	139;
9.34	580;
3.63	650;
9.04	725;
4.50	242;
9.65	487;
9.35	427;
4.17	258;
6.85	178;
4.73	111;
7.05	422;
9.27	78;
3.94	538;
7.02	771;
2.22	44;
2.29	69;
2.01	60;
1.68	92;
2.49	49;
2.23	95;
3.00	57;
2.23	80;
2.07	88;
2.77	124;
2.18	27;
2.15	133;
1.63	162;
2.87	52;
2.87	84;
1.69	123;
2.37	153;
2.14	48;
2.99	22;
2.85	28;
1.85	147;
1.37	88;
1.99	44;

1.44	103;
2.96	84;
1.50	19;
2.64	9;
2.03	71;
1.61	98;
1.51	57;
2.52	42;
1.81	37;
2.80	149;
2.29	114;
1.74	126;
2.87	112;
2.54	82;
2.30	116;
2.22	131;
2.87	28;
1.67	5;
2.72	60;
1.82	24;
1.54	158;
1.38	32;
2.46	48;
2.40	4;
2.16	165;
2.49	76;
1.61	73;
1.71	73;
1.46	147;
2.12	43;
2.31	289;
2.84	476;
3.84	538;
2.09	147;
3.42	455;
2.97	373;
2.55	433;
2.34	171;
1.90	144;
4.01	255;
2.87	504;
2.18	185;
2.93	38;

2.84	565;
3.13	100;
3.08	160;
3.21	519;
2.61	296;
3.66	97;
2.75	153;
3.44	479;
3.50	111;
3.43	89;
2.04	227;
4.06	492;
2.97	362;
2.00	443;
2.97	162;
2.00	518;
2.56	76;
3.80	561;
4.10	4;
1.90	19;
3.77	431;
2.30	25;
3.81	326;
3.98	538;
2.44	124;
4.17	219;
3.50	302;
3.48	394;
2.43	75;
2.10	138;
3.20	535;
4.10	93;
3.16	15;
2.87	61;
2.05	406;
2.35	307;
2.65	564;
3.06	517;
3.19	547;
2.49	251;
5.13	12716;
6.50	32844;
7.72	32280;

3.51	13524;
8.52	22295;
6.62	35435;
7.65	24681;
5.22	13680;
3.93	12672;
11.11	31620;
6.26	13608;
4.69	24605;
4.78	6156;
8.15	29380;
8.98	8400;
5.21	19680;
7.61	79407;
5.59	14208;
10.94	2134;
7.84	4284;
6.36	70413;
4.80	9768;
6.83	3916;
2.94	23381;
12.02	41328;
4.46	6878;
5.28	3987;
6.03	11502;
3.22	50764;
3.87	4332;
9.58	23562;
7.42	148;
5.32	2831;
8.63	49134;
4.00	3150;
10.93	36512;
10.11	44116;
5.61	14384;
9.26	28689;
10.05	8456;
5.81	1970;
6.61	4500;
3.82	3312;
4.93	84530;
5.66	2976;
7.77	720;

```
    6.89      244;
    4.43    66990;
    5.85    23332;
    4.27    41172;
    5.23    37741;
    4.66    80409;
    5.28    10793;
];

c_O = [
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
```

1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

2

3

3

3

3

3

3

3

3

3

3

3

3

3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
4
4

4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4

```
    4
    4
    4
    4
    4
    4
    4
    4
];

W1 = 2*rand(1,2)-1;
W2 = 2*rand(1,2)-1;

for epoch = 1:10000
  [W1, W2] = Deep Learning(W1, W2, i, c_O);
end

save('XCMDeepNeuralNetwork.mat')
```

Appendix D: XCMTestingDeepLearning Script

```
load('XCMDeepNeuralNetwork.mat')

i = [
    6.39        637;
    9.41        168;
    6.84        524;
    9.18         92;
    3.44        513;
    3.77        782;
    6.81        457;
    6.85        855;
    3.55        429;
    9.44        779;
    8.85        503;
    9.71        359;
    4.29        546;
    5.51         94;
    9.84        518;
```

3.51	917;
4.15	648;
9.06	688;
6.33	360;
5.39	766;
7.27	586;
6.14	777;
2.70	61;
1.90	89;
2.04	142;
2.58	130;
3.03	141;
2.63	59;
2.01	153;
1.40	57;
1.55	160;
1.98	57;
2.11	94;
1.69	118;
2.21	130;
2.63	105;
2.45	114;
2.71	104;
1.48	168;
2.98	42;
1.53	20;
1.86	38;
2.83	15;
3.00	145;
2.40	445;
3.15	479;
2.49	558;
2.97	51;
4.00	35;
2.25	360;
3.41	507;
1.83	292;
2.85	457;
2.08	275;
3.68	94;
2.25	1;
3.41	389;
3.28	99;

```
     2.72         8;
     4.17       431;
     3.37       207;
     2.80        85;
     2.46       156;
     3.21       168;
     1.99       279;
     3.04       352;
     6.48     27145;
     5.99     42631;
     5.08     79236;
     7.66      6630;
    12.12      4935;
     5.92     21240;
     6.85     77571;
     2.56     16644;
     4.42     73120;
     4.12     15675;
     7.76      8836;
     3.80       118;
     7.54     50570;
     8.63     10395;
     6.66       912;
    11.30     44824;
     4.99     34776;
     8.34      3570;
     3.76      3120;
     5.97      6384;
     5.63      4185;
     9.12     51040;
];

N = 88;
for j = 1:N
  transposed_I = i(j,:)';

  i_O_H_L = W1 * transposed_I;
  o_O_H_L = XCMSigmoid(i_O_H_L);

  i_O_O_N = W2 * o_O_H_L;
```

```
 f_O = XCMSigmoid(i_O_O_N);
end
```

Appendix E: Detection Rates (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	99	100	99
0.2	97	93	98
0.3	98	98	96
0.4	99	97	98
0.5	97	99	97
0.6	99	99	98
0.7	100	99	96
0.8	99	96	98
0.9	98	97	94

Appendix F: Analysis of Variance – Detection Rates

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	886	98.44444444	1.027777778
ASVM	9	878	97.55555556	4.527777778
CDLN	9	874	97.11111111	2.361111111

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.296296296	2	4.148148148	1.571929825	0.228284113	3.402826105
Within Groups	63.33333333	24	2.638888889			
Total	71.62962963	26				

Appendix G: False Positives (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	2	0	1
0.2	3	6	3
0.3	3	2	5
0.4	1	2	3
0.5	4	1	3
0.6	1	1	2
0.7	0	1	2
0.8	1	3	2
0.9	2	2	4

Appendix H: Analysis of Variance – False Positives

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	17	1.888888889	1.611111111
ASVM	9	18	2	3
CDLN	9	25	2.777777778	1.444444444

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	4.222222222	2	2.111111111	1.04587156	0.366857812	3.402826105
Within Groups	48.44444444	24	2.018518519			
Total	52.66666667	26				

Appendix I: False Negatives (%) based on each Schemes' Algorithm

Split Rate	XCM	ASVM	CDLN
0.1	3	0	4
0.2	2	3	2
0.3	4	4	5
0.4	2	2	2
0.5	1	3	3
0.6	1	2	2
0.7	0	1	3
0.8	2	3	4
0.9	3	2	2

Appendix J: Analysis of Variance – False Negatives

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	9	18	2	1.5
ASVM	9	20	2.222222222	1.444444444
CDLN	9	27	3	1.25

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	4.962962963	2	2.481481481	1.774834437	0.191046155	3.402826105
Within Groups	33.55555556	24	1.398148148			
Total	38.51851852	26				

Appendix K: Memory Utilisations (%) based on each Schemes' Algorithm

Epochs	XCM	ASVM	CDLN
1000	1.04	1.41	1.47
2000	1.19	1.48	1.52
3000	1.28	1.54	1.65
4000	1.26	1.62	1.59
5000	1.30	1.57	1.68
6000	1.36	1.71	1.72
7000	1.32	1.73	1.81
8000	1.46	1.86	1.90
9000	1.54	1.82	1.89
10000	1.56	1.87	1.94

Appendix L: Analysis of Variance – Memory Utilisations

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	10	13.31	1.331	0.02521
ASVM	10	16.61	1.661	0.026232222
CDLN	10	17.17	1.717	0.027067778

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.870106667	2	0.435053333	16.62412432	1.96862E-05	3.354130829
Within Groups	0.70659	27	0.02617			
Total	1.576696667	29				

Appendix M: CPU Utilisations (%) based on each Schemes' Algorithm

Epochs	XCM	ASVM	CDLN
1000	0.96	1.43	1.47
2000	1.12	1.57	1.59
3000	1.28	1.59	1.63
4000	1.36	1.68	1.71
5000	1.40	1.75	1.78
6000	1.53	1.84	1.86
7000	1.69	1.87	1.92
8000	1.78	1.91	1.94
9000	1.81	1.93	1.97
10000	1.89	1.97	1.98

Appendix N: Analysis of Variance – CPU Utilisations

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
XCM	10	14.82	1.482	0.097151111
ASVM	10	17.54	1.754	0.032671111
CDLN	10	17.85	1.785	0.031894444

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.555846667	2	0.277923333	5.155745646	0.012692162	3.354130829
Within Groups	1.45545	27	0.053905556			
Total	2.011296667	29				

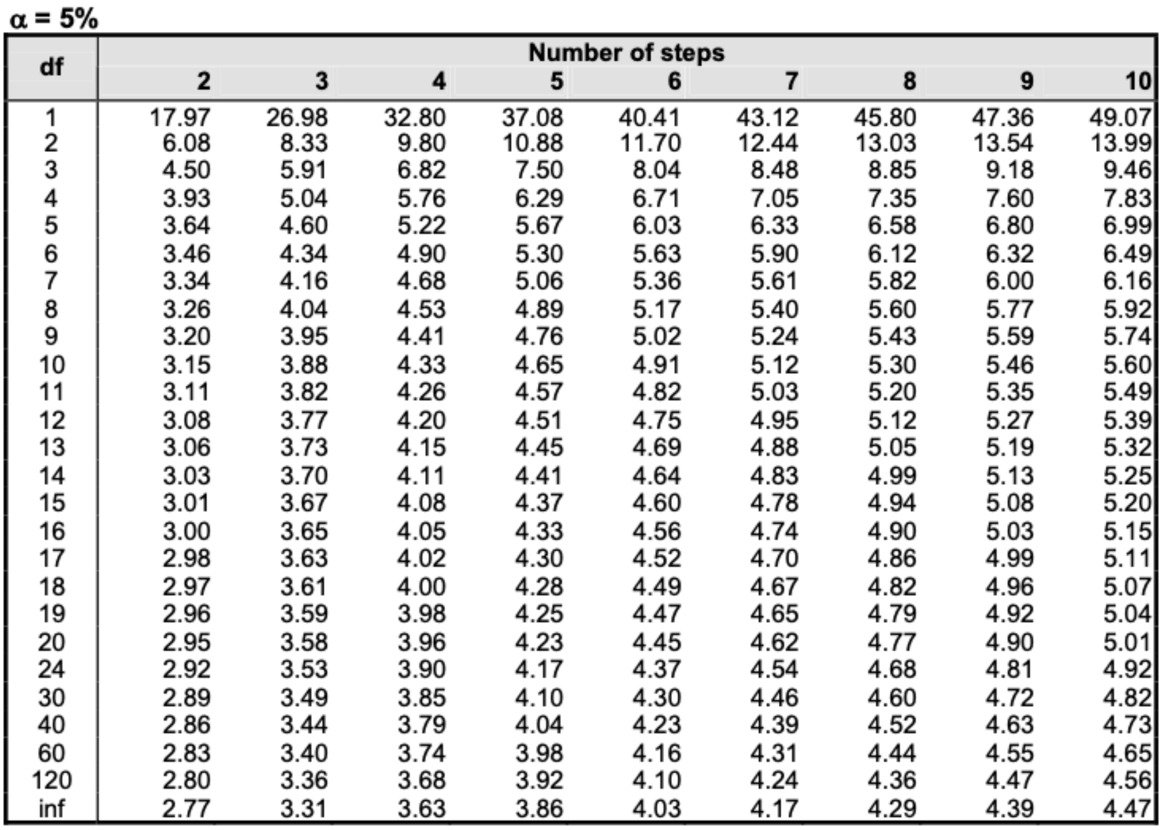
Appendix O: Tukey's Test Analysis – Memory Utilisation

		Absolute Differences		
	Means	**XCM**	**ASVM**	**CDLN**
XCM	1.331	0		
ASVM	1.661	0.330	0	
CDLN	1.717	0.386	0.056	0

Appendix P: Tukey's Test Analysis – CPU Utilisation

		Absolute Differences		
	Means	**XCM**	**ASVM**	**CDLN**
XCM	1.482	0		
ASVM	1.754	0.272	0	
CDLN	1.785	0.303	0.031	0

Appendix Q: Percentage Points of the Studentised Range



α = 5%

df	Number of steps								
	2	3	4	5	6	7	8	9	10
1	17.97	26.98	32.80	37.08	40.41	43.12	45.80	47.36	49.07
2	6.08	8.33	9.80	10.88	11.70	12.44	13.03	13.54	13.99
3	4.50	5.91	6.82	7.50	8.04	8.48	8.85	9.18	9.46
4	3.93	5.04	5.76	6.29	6.71	7.05	7.35	7.60	7.83
5	3.64	4.60	5.22	5.67	6.03	6.33	6.58	6.80	6.99
6	3.46	4.34	4.90	5.30	5.63	5.90	6.12	6.32	6.49
7	3.34	4.16	4.68	5.06	5.36	5.61	5.82	6.00	6.16
8	3.26	4.04	4.53	4.89	5.17	5.40	5.60	5.77	5.92
9	3.20	3.95	4.41	4.76	5.02	5.24	5.43	5.59	5.74
10	3.15	3.88	4.33	4.65	4.91	5.12	5.30	5.46	5.60
11	3.11	3.82	4.26	4.57	4.82	5.03	5.20	5.35	5.49
12	3.08	3.77	4.20	4.51	4.75	4.95	5.12	5.27	5.39
13	3.06	3.73	4.15	4.45	4.69	4.88	5.05	5.19	5.32
14	3.03	3.70	4.11	4.41	4.64	4.83	4.99	5.13	5.25
15	3.01	3.67	4.08	4.37	4.60	4.78	4.94	5.08	5.20
16	3.00	3.65	4.05	4.33	4.56	4.74	4.90	5.03	5.15
17	2.98	3.63	4.02	4.30	4.52	4.70	4.86	4.99	5.11
18	2.97	3.61	4.00	4.28	4.49	4.67	4.82	4.96	5.07
19	2.96	3.59	3.98	4.25	4.47	4.65	4.79	4.92	5.04
20	2.95	3.58	3.96	4.23	4.45	4.62	4.77	4.90	5.01
24	2.92	3.53	3.90	4.17	4.37	4.54	4.68	4.81	4.92
30	2.89	3.49	3.85	4.10	4.30	4.46	4.60	4.72	4.82
40	2.86	3.44	3.79	4.04	4.23	4.39	4.52	4.63	4.73
60	2.83	3.40	3.74	3.98	4.16	4.31	4.44	4.55	4.65
120	2.80	3.36	3.68	3.92	4.10	4.24	4.36	4.47	4.56
inf	2.77	3.31	3.63	3.86	4.03	4.17	4.29	4.39	4.47

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey”, *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[2] R. Sathya and R. Thangarajan, “Efficient Anomaly Detection and Mitigation in Software Defined Networking Environment”, in *2015 2nd International Conference on Electronics and Communication Systems,* Coimbatore, 2015, pp. 479-484.

[3] Q. Liu, L. Xing, and C. Wang, "Framework of Probabilistic Risk Assessment for Security and Reliability", in *2017 IEEE Second International Conference on Data Science in Cyberspace*, Shenzhen, 2017, pp. 619-624.

[4] Y. Zhang, B. Zhu, Y. Fang, S. Guo, A. Zhang, and S. Zhong, “Secure Inter-domain Forwarding Loop Test in Software Defined Networks”, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-18, July 2017.

[5] X. Huang, X. Du, and B. Song, “An Effective DDoS Defense Scheme for SDN”, in *2017 IEEE International Conference on Communications,* Paris, 2017, pp. 1-6.

[6] X. Fan, Z. Lu, L. Ju, and D. Mu, “The Research on Security SDN South Interface Based on OTR Protocol”, in *2016 16th International Symposium on Communications and Information Technologies,* Qingdao, 2016, pp. 629-633.

[7] P. Zhang, H. Wang, C. Hu, and C. Lin, “On Denial of Service Attacks in Software Defined Networks”, *IEEE Network*, vol. 30, no. 6, pp. 28-33, Nov-Dec. 2016.

[8] P. Rengaraju, S. S. Kumar, and C. Lung, “Investigation of Security and QoS on SDN Firewall Using MAC Filtering”, in *2017 International Conference on Computer Communication and Informatics*, Coimbatore, 2017, pp. 1-5.

[9] Y. Jararweh, M. Al-Ayyoub, A. Doulat, A. Al Abed Al Aziz, H. A. BanySalameh, and A. A. Khreishah, “Software Defined Cognitive Radio Network Framework: Design and

Evaluation", *International Journal of Grid and High Performance Computing*, vol. 7, no. 1, pp. 15-31, 2015.

[10] J. Maisuria and S. Mehta, "An Overview of Medium Access Control Protocols for Cognitive Radio Sensor Networks", in *4th International Electronic Conference on Sensors and Applications*, 15-30 November 2017 [Online]. Available: http://sciforum.net/conference/ecsa-4 [Accessed: 10 May 2018].

[11] D. Kiwan, A. El Sherif, and T. ElBatt, "Stability Analysis of a Cognitive Radio System with a Dedicated Relay", in *2017 International Conference on Computing, Networking and Communications*, Santa Clara, 2017, pp. 750-756.

[12] S. Bhunia, E. Miles, S. Sengupta, and F. Vazquez-Abad, "CR-Honeynet: A Cognitive Radio Learning and Decoy Based Sustenance Mechanism to Avoid Intelligent Jammer", *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 567-581, Sept. 2018.

[13] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey", *IEEE Communication Surveys and Tutorials*, vol. 17, no. 4, pp. 2317-2346, Fourthquarter 2015.

[14] Z. El Mrabet, Y. Arjoune, H. El Ghazi, B. Abou Al Majd, and N. Kaabouch, "Primary User Emulation Attacks: A Detection Technique Based on Kalman Filter", *Journal of Sensor and Actuator Networks*, vol. 7, no. 26, pp. 1-14, 2018.

[15] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "An Architecture for Software Defined Cognitive Radio", in *IEEE Symposium on Architectures for Networking and Communications Systems*, La Jolla, 2010, pp. 1-12.

[16] M. M. Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking", *Journal of Computer Networks and Communications*, vol. 2019, pp. 1-12, Mar. 2019.

[17] B. He, F. Zou, and Y. Wu, “Multi-SDN Based Cooperation Scheme for DDoS Attack Defense”, in *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications*, Shanghai, 2018, pp. 1-7.

[18] S. Arun and G. Umamaheswari, “An Adaptive Learning-Based Attack Detection Technique for Mitigating Primary User Emulation in Cognitive Radio Networks”, *Circuits, Systems and Signal Processing*, vol. 39, pp. 1071-1088, 2020.

[19] J. Ashraf and S. Latif, “Handling Intrusion and DDoS Attacks in Software Defined Networks Using Machine Learning Techniques”, in *2014 National Software Engineering Conference*, Rawalpindi, 2014, pp. 55-60.

[20] D. Hyun, J. Kim, D. Hong, and J. Jeong, “SDN-based Network Security Functions for Effective DDoS Attack Mitigation”, in *2017 International Conference on Information and Communication Technology Convergence*, Jeju, 2017, pp. 834-839.

[21] N. N. Dao, J. Park, M. Park, and S. Cho, “A feasible method to combat against DDoS attack in SDN Network”, in *2015 International Conference on Information Networking*, Cambodia, 2015, pp. 309-311.

[22] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, “A SDN-Oriented DDoS Blocking Scheme for Botnet-Based Attacks”, in *2014 Sixth International Conference Ubiquitous and Future Networks*, Shanghai, 2014, pp. 63-68.

[23] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow”, in *IEEE Local Computer Network Conference*, Denver, 2010, pp. 408-415.

[24] R. Sultana and M. Hussain, “Mitigating Primary User Emulation Attack in Cognitive Radio Network Using Localization and Variance Detection”, in *Proceedings of First International Conference on Smart System, Innovations and Computing*, 9 January 2018 [Online]. Available: SpringerLink, https://link.springer.com [Accessed: 10 May 2018].

[25] R. D. Kadu, P. P. Karde, and V. M. Thakare, “Performance of CSS Cognitive Radio Networks Under Primary User Emulation Attack”, in *2017 International Conference on Trends in Electronics and Informatics*, Tirunelveli, 2017, pp. 19-24.

[26] H. Sharma and K. Kumar, “Primary User Emulation Attack Analysis on Cognitive Radio”, *Indian Journal of Science and Technology*, vol. 9, no. 14, pp. 1-6, Apr. 2016.

[27] A. Alahmadi, M. Abdelhakim, J. Ren, and T. Li, “Defense Against Primary User Emulation Attacks in Cognitive Radio Networks Using Advanced Encryption Standard”, *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 772-781, May 2014.

[28] P. V. Trung, T. T. Huong, D. V. Tuyen, D. M. Duc, N. H. Thanh, and A. Marshall, “A Multi-Criteria-based DDoS-Attack Prevention Solution using Software Defined Networking”, in *2015 International Conference on Advanced Technologies for Communications,* Ho Chi Minh City, 2015, pp. 308-313.

[29] M. M. Oo, S. Kamolphiwong, and T. Kamolphiwong, “The Design of SDN Based Detection for Distributed Denial of Service (DDoS) Attack”, in *2017 21st International Computer Science and Engineering Conference*, Bangkok, 2017, pp. 1-5.

[30] I. Alsmadi and D. Xu, “Security of Software Defined Networks: A survey”, *Computers and Security*, vol. 53, pp. 79-108, Sept. 2015.

[31] S. Rizvi, N. Showan, and J. Mitchell, “Analyzing the Integration of Cognitive Radio and Cloud Computing for Secure Networking”, *Procedia Computer Science*, vol. 61, pp. 206-212, 2015.

[32] S. Padmaja and V. Vetriselvi, “Mitigation of Switch-DOS in Software Defined Networking”, in *2016 International Conference on Information Communication and Embedded Systems*, Chennai, 2016, pp. 1-5.

[33] L. F. Carvalho, G. Fernandes, J. J. P. C. Rodrigues, L. S. Mendes, and M. L. Proenca, “A Novel Anomaly Detection System to Assist Network Management in SDN

Environment", in *2017 IEEE International Conference on Communications*, Paris, 2017, pp. 1-6.

[34] W. Li, W. Meng, and L. F. Kwok, "A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures", *Journal of Network and Computer Applications*, vol. 68, pp. 126-139, June 2016.

[35] "ICASA", www.icasa.org.za, 2018. [Online] Available: https://www.icasa.org.za/legislation-and-regulations/spectrum-usage-and-availability-q1-2019. [Accessed: 25 June. 2018].

[36] G. Elanagai and C. Jayasri, "Implementation of Network Security Based Data Hauling by Collaborative Spectrum Sensing in Cognitive Radio Network", in *2017 International Conference on Innovations in Information, Embedded and Communication Systems*, Coimbatore, 2017, pp. 1-5.

[37] K. Tang, R. Shi, and E. Luo, "Secure Beamforming Design with Bidirectional Secondary Transmissions in Wireless-Powered Cognitive Radio Networks", in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications*, Guangzhou, 2017, pp. 428-432.

[38] W. El-Hajj, H. Safa, and M. Guizani, "Survey of Security Issues in Cognitive Radio Networks", *Journal of Internet Technology*, vol. 12, no. 2, pp. 181-198, Mar. 2011.

[39] R. D. Kadu and P. P. Karde, "Improving Performance of CSS Cognitive Radio Networks Under Jamming Attack", in *2017 2nd International Conference on Communication Systems, Computing and IT Applications*, Mumbai, 2017, pp. 213-217.

[40] S. Raj and O. P. Sahu, "Countermeasures to Security Threat/Attacks on Different Protocol Layers in Cognitive Radio Networks: An Overview", in *2017 International Conference on Smart Technology for Smart Nation*, Bangalore, 2017, pp. 1076-1082.

[41] I. H. Abdulqadder, D. Zou, I. T. Aziz, B. Yuan, and W. Li, “SecSDN-Cloud: Defeating Vulnerable Attacks through Secure Software-Defined Networks”, *IEEE Access*, vol. 6, pp. 8292-8301, Mar. 2018.

[42] A. Diaz and P. Sanchez, “Simulation of Attacks for Security in Wireless Sensor Network”, *Sensors 2016*, vol. 11, pp. 1-27, Nov. 2016.

[43] “GNU Octave”, gnu.org, 2018. [Online] Available: https://www.gnu.org/software/octave/. [Accessed: 28 Sep. 2018].

[44] M. Amjad, M. H. Rehmani, and S. Mao, “Wireless Multimedia Cognitive Radio Networks: A Comprehensive Survey”, *IEEE Communications Surveys and Tutorials*, vol. 20, no. 2, pp. 1056-1103, Secondquarter 2018.

[45] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, “Towards software defined cognitive networking”, *2015 7th International Conference on New Technologies, Mobility and Security*, Paris, 2015, pp. 1-5.

[46] G. Sun, G. Liu, and Y. Wang, “SDN architecture for cognitive radio networks”, *2014 1st International Workshop on Cognitive Cellular Systems*, Rhine River, 2014, pp. 1-5.

[47] S. Jero, W. Koch, R. Skowyra, H. Okhravi, C. Nita-Rotaru, and D. Bigelow, “Identifier binding attacks and defences in Software-Defined Networks”, *26th Security Symposium*, pp. 415-432, 2017.

[48] S. Xiang, X. Wu, H. Zhu, W. Xie, L. Xiao, and P. C. Vinh, “Modeling and Verifying Basic Modules of Floodlight”, *Mobile Networks and Applications*, pp. 1-15, 2018.

[49] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, Y. Sun, “A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning”, *2019 IEEE Access*, vol. 7, pp. 95397-95417, July. 2019.

[50] H. Idoudi, K. Daimi, and M. Saed, “Security challenges in cognitive radio networks”, *Proceedings of the World Congress on Engineering*, London, 2014, pp. 2-4.

[51] W. Alhakami, A. Mansour, and G. A. Safdar, “Spectrum sharing security and attacks in CRNs: A Review”, *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 76-87, 2014.

[52] J. Li, Z. Feng, Z. Feng, and P. Zhang, “A survey of security issues in cognitive radio networks”, *China Communications*, vol. 12, no. 3, pp. 132-150, 2015.

[53] P. Dong, X. Du, H. Zhang, and T. Xu, “A Detection Method for a Novel DDoS Attack Against SDN Controllers by Vast New Low-Traffic Flows”, in 2016 IEEE International Conference on Communications, Kuala Lumpur, 2016, pp. 1-6.

[54] D. Li, C. Yu, Q. Zhou, and J. Yu, “Using SVM to Detect DDoS Attack in SDN Network”, in 2018 *2nd Annual International Conference on Cloud Technology and Communication Engineering*, Nanjing, 2018, pp. 1-7.

[55] P. Van Trung, T. T. Huong, D. Van Tuyen, D. M. Duc, N. H. Thanh, and A. Marshall, “A multi-criteria-based DDoS-attack prevention solution using software defined networking”, *2015 International Conference on Advanced Technologies for Communications*, Ho Chi Minh City, 2015, pp. 308-318.

[56] P. Dong, X. Du, H. Zhang, and T. Xu, “A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows”, *2016IEEE International Conference on Communications*, Kuala Lumpur, 2016, pp. 1-6.

[57] F. Gillani, E. Al-Shaer, and Q. Duan, “In-design resilient SDN control plane and elastic forwarding against aggressive DDoS attacks”, *Proceedings of the 5th Association for Computing Machinery Workshop on Moving Target Defense*, Toronto, 2018, pp. 80-89.

[58] S. M. Mousavi and M. St-Hilaire, “Early detection of DDoS attacks against SDN controllers”, *2015 International Conference on Computing, Networking and Communications*, Garden Grove, 2015, pp. 77-81.

[59] S. Saharan and V. Gupta, “Prevention and Mitigation of DNS based DDoS Attacks in SDN Environment”, in *2019 11th International Conference on Communication Systems and Networks*, Bengaluru, 2019, pp. 571-573.

[60] N. Z. Bawany, J. A. Shamsi, and K. Salah, “DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions”, *Arabian Journal for Science and Engineering,* vol. 42, pp. 425-441, Feb. 2017.

[61] L. Yang and H. Zhao, “DDoS Attack Identification and Defense Using SDN Based on Machine Learning Method”, in *15th International Symposium on Pervasive Systems, Algorithms and Networks*, Yichang, 2018, pp. 174-178.

[62] S. S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Lee, F. Hussain, C. A. Kerrache, E. Barka, and M. Z. A. Bhuiyan, “A New Machine Learning-Based Collaborative DDoS Mitigation Mechanism in Software Defined Network”, in *14th International Conference on Wireless and Mobile Computing, Networking and Communications*, Limassol, 2018, pp. 1-8.

[63] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, and D. Huang, “A Defense System for Defeating DDoS Attacks in SDN Based Networks”, in *MobiWac’ 17*, Miami, 2017, pp. 83-92.

[64] S. Ali, M. K. Alvi, S. Faizullah, M. A. Khan, A. Alshanqiti, and I. Khan, “Detecting DDoS Attack on SDN Due to Vulnerabilities in OpenFlow”, in *2019 International Conference on Advances in the Emerging Computing Technologies*, Medina, 2020, pp. 1-6.

[65] S. Hameed and H. A. Khan, “SDN Based Collaborative Scheme for Mitigation of DDoS Attacks”, *Future Internet MDPI*, vol. 10, no. 23, pp. 1-18, Feb. 2018.

[66] S. M. Mousavi and M. St-Hilaire, “Early Detection of DDoS Attacks Against SDN Controllers”, in *2015 International Conference on Computing, Networking and Communications, Communications and Information Security Symposium*, California, 2015, pp. 77-81.

[67] H. Polat, O. Polat, and A. Cetin, “Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models”, *Sustainability MDPI*, vol. 12, no. 3, pp. 1-16, Feb. 2020.

[68] P. Kai, Z. Fanzi, and Z. Qingguang, “A new method to detect primary user emulation attacks in cognitive radio networks”, *3rd International Conference on Computer Science and Service System*, Bangkok, 2014, pp. 674-677.

[69] Q. Jiang, H. Chen, L. Xie, and K. Wang, “On detecting primary user emulation attack using channel impulse response in the cognitive radio network”, *Frontiers of Information Technology and Electronic Engineering*, vol. 18, no. 10, pp. 1665-1676, Oct. 2017.

[70] S. Srinivasan, K. B. Shivakumar, and M. Mohammad, “Semi-supervised machine learning for primary user emulation attack detection and prevention through core-based analytics for cognitive radio networks”, *International Journal of Distributed Sensor Networks*, vol. 15, no. 9, pp. 1-12, 2019.

[71] A. Jayapalan and T. Karuppasamy, “Authentication Scheme to Combat a Primary User Emulation Attack Against Cognitive Radio Users”, *Security and Communication Networks*, vol. 8, pp. 4242-4253, 2015.

[72] S. U. Rehman, K. Sowerby, and C. Coghill, “Radio-Frequency Fingerprinting for Mitigating Primary User Emulation Attack in Low-end Cognitive Radios”, *IET Communications*, vol. 8, no. 8, pp. 1274-1284, 2014.

[73] M. Ghaznavi and A. Jamshidi, “Defence Against Primary User Emulation Attack Using Statistical Properties of the Cognitive Radio Received Power”, *IET Communications*, vol. 11, no. 9, pp. 1535-1542, 2017.

[74] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*, 4th ed., London: Pearson Education Limited, 2016, p. 131.

[75] U. Sekaran and R. Bougie, *Research Methods for Business: A Skill-Building Approach*, 6th ed., New York: Wiley, 2013, p. 95.

[76] R. E. Kirk, *Experimental design: Procedures for the behavioral sciences*, 4th ed., Thousand Oaks, CA: Sage, 2013, p. 11.

[77] D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, "*Mathematical Statistics with Applications"*, 7th ed., Belmont, CA: Cengage Learning, 2008, p. 27.

[78] T. Chamberlain, "*Learning OMNeT++*", Packt Publishing Ltd, Birmingham, UK, 2013.

[79] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computer Systems*, vol. 82, pp. 761-768, May. 2018.

[80] W. Wang, S. Gombault, and T. Guyet, "Towards fast detecting intrusions: using key attributes of network traffic", *ICIMP 2008 – Third International Conference on Internet Monitoring and Protection*, Bucharest Romania, Jun. 2008, pp. 86-91.

[81] A. Naik and S. W. Ahmad, "Data Mining Technology for Efficient Network Security Management", *International Journal of Computer Science Trends and Technology*, vol. 3, pp. 7-12, June. 2015.

[82] D. Bienstock, G. Munoz, and S. Pokutta, "Principled deep neural network training through linear programming," *arXiv preprint arXiv*, vol. 1810, pp. 1-26, Oct. 2018.

[83] M. Alkasassbeh, G. Al-Naymat, A. B. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques", *International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 436-445, Jan. 2016.

[84] K. K. Mohbey, “Multi-class approach for user behaviour prediction using deep learning framework on twitter election dataset”, *Journal of Data, Information and Management*, vol. 2, pp. 1-14, Oct. 2019.

[85] D. Silva-Palacios, C. Ferri, and M. J. Ramirez-Quintana, “Improving Performance of Multiclass Classification by Inducing Class Hierarchies”, *International Conference On Computational Science*, Zurich Switzerland, Jun. 2017, pp. 1692-1701.

[86] Y. Wang, E. Coiera, W. Runciman, and F. Magrabi, “Using multiclass classification to automate the identification of patient safety incident reports by type and severity”, *BioMed Central Medical Informatics and Decision Making*, vol. 17, no. 84, pp. 1-12, Jun. 2017.

[87] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Elsevier, MA: Morgan Kaufmann Publishers, 2011, pp. 227-236.

[88] N. Meti, D. G. Narayan, and V. P. Baligar, “Detection of Distributed Denial of Service Attacks using Machine Learning Algorithms in Software Defined Networks”,*2017 International Conference on Advances in Computing, Communications and Informatics*, Udupi India, Sept. 2017, pp. 1366-1371.

[89] J. Benabbou, K. Elbaamrani, and N. Idboufker, “Security in OpenFlow-based SDN, opportunities and challenges”, *Photonic Network Communications*, vol. 37, no. 1, pp. 1-23, Sept. 2018.

[90] S. Chen, K. Zeng, and P. Mohapatra, “Hearing is Believing: Detecting Mobile Primary User Emulation Attack in White Space”, *2011 Proceedings IEEE INFOCOM*, Shanghai China, Apr. 2011, pp. 1-5.

[91] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, “XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-based Cloud”, *2018 IEEE International Conference on Big Data and Smart Computing*, Shanghai China, Jan. 2018, pp. 251-256.

[92] S. Hameed and H. A. Khan, “SDN Based Collaborative Scheme for Mitigation of DDoS Attacks”, *Future Internet*, vol. 10, pp. 1-18, Feb. 2018.

[93] W. R. Ghanem, R. E. Mohamed, M. Shokair, and M. I. Dessouky, “Particle Swarm Optimization Approaches for Primary User Emulation Attack Detection and Localization in Cognitive Radio Networks”,*2018 35th National Radio Science Conference*, Cairo Egypt, Mar. 2018, pp. 1-16.

[94] J. N. Bakker, B. Ng, and W. K. G. Seah, “Can Machine Learning Techniques be Effectively used in Real Networks Against DDoS Attacks”, in *2018 27th International Conference on Computer Communication and Networks*, Hangzhou, 2018, pp. 1-6.

[95] F. Gharvirian and A. Bohlooli, “Neural Network Based Protection of Software Defined Network Controller against Distributed Denial of Service Attacks”, *International Journal of Engineering*, vol. 30, no. 11, pp. 1714-1722, Nov. 2017.

[96] A. AlEroud and I. Alsmadi, “Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach”, *Journal of Network and Computer Applications*, vol. 80, pp. 152-164, 2017.

[97] A. S. Pimpalkar and A. R. B. Patil, “Defense against DDoS Attacks Using IP Address Spoofing”, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, pp. 1919-1926.

[98] Y. Xu and R. Goodacre, “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning”, *Journal of Analysis and Testing*, vol. 2, pp. 249-262, Oct. 2018.

[99] H. M. Furqan, M. A. Aygul, M. Nazal, and H. Arslan, “Primary User Emulation and Jamming Attack Detection in Cognitive Radio via Sparse Coding”, *EURASIP Journal on Wireless Communications and Networking*, vol. 141, pp. 1-19, Jul. 2020.

[100] D. Chicco and G. Jurman, “The Advantages of the Matthews Correlation Coefficient (MCC) over F1 score and accuracy in binary classification evaluation”, *BMC Genomics*, vol. 21(6), pp. 1-13, Jan. 2020.

[101] G. Keller, “*Statistics for Management and Economics*”, 11th ed., Boston, MA: Cengage Learning, 2017, pp. 517-590.