

A FUSED MACHINE LEARNING INTRUSION DETECTION MODEL IN MANETS.

KIRORI GATHUO MINDO

**A Research Thesis Submitted to the Institute of Postgraduate Studies in Partial Fulfillment
for the Requirements of the Doctor of Philosophy in Information Technology Security and
Audit of Kabarak University.**

KABARAK UNIVERSITY

NOVEMBER 2019

DECLARATION

I hereby declare that this research thesis is my original work and has not been submitted to any other University or College for purposes of examination or academic award. Any information given is my entire work and all the relevant sources are quoted and acknowledged accordingly.

Sign _____ Date _____

Kirori Gathuo Mindo

GDS/M/0163/01/17

RECOMMENDATION

To the Institute of Postgraduate Studies and Research:

The research thesis entitled “**A Fused Machine Learning Intrusion Detection Model For The Provision Of Smart Health Care In Manets.**” written by Kirori Gathuo Mindo is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the thesis and recommend it to be accepted in partial fulfillment of the requirement for the Doctor of Philosophy in Information Technology Security and Audit.

Signature _____

Date _____

Prof. Simon M. Karume

Department of Computer Science

Kabarak University

Signature _____

Date _____

Dr. Moses M. Thiga

Department of Computer Science

Kabarak University

COPYRIGHT

© 2019

Kirori Gathuo Mindo

All rights reserved. No part of this thesis may be reproduced or stored in any retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, and recording without prior written permission of the author or Kabarak University on that behalf.

ACKNOWLEDGMENTS

This is to acknowledge the Almighty God for His Provision and sustenance throughout my studies at Kabarak University. For our light and momentary troubles are achieving for us eternal glory that far outweighs them all. 2 Corinthians 4:16-18. I also acknowledge my supervisors Prof Simon M Karume and Dr Moses M. Thiga for their wise counsel, guidance and support throughout this research. They have shaped, guided, advised, corrected and led this research for the long period since problem identification to completion. Their input has really been immense and overwhelming. I also acknowledge my colleagues and classmates for the encouragement through this journey.

DEDICATION

To my father Duncan Mindo, for he gave me the best gift anyone could give another person, he believed in me. He brought me to life, nurtured me, taught me, dressed me, clothed me, fought for me, treated me, held me and most importantly he loved me unconditionally. He is truly what a powerful and positive influence he continues to be. My siblings Shiru, Gathuo, Gatimu, Wambui, Wanjugu and Wangari and my mother Lucy Njoki as well.

ABSTRACT

Mobile Ad-Hoc Networks – MANETs are prevalent in healthcare monitoring of high blood pressure, high cholesterol levels and various heart conditions and cardiac misnomers like syncope, third murmurs and atrial fibrillation. These irregularities that cause mysterious fainting, unexplained stroke, heart palpitations and atrial fibrillation need to be monitored remotely, accurately and effortlessly. However, the growth and provision of MANETs in smart healthcare monitoring has faced various security obstacles, primarily security. The characteristic mobility of these health monitoring devices as well as their inherently dynamic network topology, causes the connectivity structure to change frequently and unpredictably. Further, these smart devices have limited resources in storage, processor capability and memory, thus these weaknesses and inherent nature makes them subject to attacks like Denial of Service (DoS) attacks. These attacks on MANETs can reduce or mask the monitoring of health deterioration which can in turn lead to death, immobility or temporary functional disability. There is need to provide resilient security methodologies that do not require enormous computing resources. While entry prevention is the most viable disposition, it is not always possible to stop unauthorized access. Thus, it is critical to investigate use of machine learning based intrusion detection to buttress and provide sufficient security against DOS and other attacks in MANETs. Various anomaly-based intrusion detection systems employ varying techniques to identify anomalies in the context of diverse and valid variables. Most of these techniques however fail to capture and take account the physiognomies of MANETs. In the intervening time, usage of internet of things in the provision of smart healthcare is expanding and the inherent risks snowballing. Attacks aimed at MANETs are increasing to an alarming extent. This study employed a fusion of machine learning techniques through both simulation and a running prototype to achieve a more resilient intrusion detection system. The study was implemented and evaluated on a MANET environment on both Linux NS 2 and further implemented on a network of Smart wearable devices and Raspberry Pi. The results sowed that it is possible to identify and reduce cases of DDOS and blackhole attacks on MANETs by using intrusion detection system improved through machine learning. This study contributed to the body of knowledge in the field intrusion detection systems through ubiquitous learning.

Keywords: MANET, Smart Healthcare, Intrusion Detection Systems, Machine Learning, Fused, Internet of Things.

TABLE OF CONTENTS

DECLARATION.....	i
RECOMMENDATION.....	ii
COPYRIGHT.....	iii
ACKNOWLEDGMENTS.....	iv
DEDICATION.....	v
ABSTRACT.....	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	xiii
LIST OF FIGURES.....	xiv
ABBREVIATIONS.....	xvi
DEFINITION OF TERMS.....	xviii
CHAPTER ONE.....	1
INTRODUCTION.....	1
1.1. Introduction.....	1
1.2. Background of the Study.....	1
1.3. Statement of the Problem.....	8
1.4. Device Objectives of the Study.....	9
1.4.2. Overall Objective.....	9
1.4.2. Specific Objectives.....	9
1.5. Research Questions.....	10
1.6. Significance of the Study.....	10
1.7. Scope of the Study.....	11
1.8. Justification for this Study.....	11
1.9. Limitations.....	12

1.10.	Conclusion.....	12
CHAPTER TWO		13
LITERATURE REVIEW		13
2.1.	Introduction.....	13
2.2.	Security Obstacles in the Application of MANET for Provision of Smart Health Care ...	13
2.1.1.	Review of MANETs in General.....	13
2.1.2.	History of the Internet of Things.....	17
2.1.3.	General Global Usage of the Internet of Things	21
2.1.4.	MANET in Smart Healthcare.....	24
2.1.5.	Security Obstacles in MANETs	26
2.3.	Security Vulnerabilities in MANETs.....	27
2.4.	Common Attacks in MANETs.....	28
2.5.	MANET Security Models for.	30
2.5.1.	What is a Security Model	30
2.5.2.	Types of Security Models	31
2.6.	Intrusion Detection Models for the MANET.....	33
2.6.1.	Implementation of MANET Anomaly-Based Intrusion Detection Model for Health Care.	40
2.6.2.	Evaluation, Validation and Verification of Related works in	50
2.6.3.	Hybrid and Fused approaches for ABIDS in MANETS	52
2.7.	Types of Intrusion Detection Systems in MANETs	56
2.7.1.	Classification Based on Mode of Operations	57
2.7.2.	Classification Based on Detection Algorithms Used in the IDSs	60
2.8.	Machine Learning	64
2.8.1.	Concept of machine learning.....	65

2.8.2.	Brief History of ML	66
2.8.3.	Game of Checker Players.....	66
2.8.4.	The Perceptron	67
2.8.5.	The Nearest Neighbor Algorithm.....	68
2.8.6.	Deep learning	68
2.8.7.	General Applications of Machine Learning.....	69
2.8.8.	Machine Learning Models and their Application in MANETS.....	72
2.8.9.	Research Gap in General ABIDS Techniques for MANET	83
2.8.10.	Conceptual Framework.....	85
CHAPTER THREE		88
RESEARCH DESIGN AND METHODOLOGY		88
3.1	Introduction.....	88
3.1.1	General Literature Review Methodology.....	88
3.1.2	Research Philosophy	88
3.1.3	Proof of Concept Methodology.....	89
3.1.4	Design of a MANET ABID Model for Smart Healthcare.	90
3.1.5	PPDIOO methodology Research Design for Model Design	90
3.1.6	The Preparation Stage	90
3.1.7	The Planning Stage.....	91
3.1.8	The Design Stage	92
3.1.9	Implementation of the Fused Machine Learning Intrusion Detection Model.....	94
3.1.10	Simulation on NS 2 on Linux.....	94
3.1.11	Implementation Smart Healthcare Network on Raspberry Board Microcontroller	95
3.1.12	Evaluation of the Smart Health Care MANET Anomaly-Based Intrusion Detection Model.	97

3.1.13	Evaluation of the Finite State Automata	97
3.1.14	Evaluation of the Network Simulator 2 Environment.....	98
3.1.15	Evaluation of the Smart Health Care Network on Raspberry Board Microcontroller ...	99
3.1.16	Equipment to be used in the Experiment	100
CHAPTER FOUR.....		101
DATA ANALYSIS, PRESENTATION AND DISCUSSION		101
4.1	Introduction.....	101
4.1.1	Weaknesses and Security Obstacles in The Application Of MANETs For Provision of Smart Health Care.....	101
4.1.2	Methodology for the Identification of Existing Weaknesses and Security Obstacles.	102
4.1.3	The Weaknesses and Security Obstacles In The Application Of MANETs For Provision of Smart Health Care.	102
4.1.4	Weakness 1: Distributed Operation.....	103
4.1.5	Weakness 2: Multi-Hop routing.....	103
4.1.6	Weakness 3: Light Weight Terminals.....	103
4.1.7	Weakness 4: Shared Physical Medium	104
4.1.8	Weakness 5: Limited bandwidth.....	104
4.1.9	Weakness 6: Dynamic topology.....	104
4.1.10	Weakness 7: Routing Overhead.....	104
4.1.11	Weakness 8: Hidden terminal problem	105
4.1.12	Weakness 9: Wireless Radio	105
4.1.13	Weakness 10: Mobility.....	105
4.1.14	Weakness 11: Battery constraints	105
4.1.15	Auxilliary Security Weakness	106

4.1.17.	Validation of the identified Weaknesses and Security Obstacles in the Application of MANETs for Provision of Smart Health Care.....	108
4.1.18.	Design Recommendations to Address the identified weaknesses and Security Obstacles In The Application Of MANETs For Provision Of Smart Health Care.....	111
4.1.19.	Conclusion.....	112
4.2.	Design of A MANET Anomaly-Based Intrusion Detection Model for Smart Health care.	113
4.2.1.	Design of Logical Topology.	114
4.2.2.	Design of a Finite State Machine to describe Normal and Anomalous Events within a MANETs.....	120
4.2.3.	Design of the Integrated Network Connectivity within a MANET	130
4.2.4.	The Model Design for the Fused Machine Learning Intrusion Detection Model for The Provision Of Smart Health Care in MANETS.....	133
4.3.	Implementation of The Fused Machine Learning Intrusion Detection Model for The Provision of Smart Health Care in MANETS.....	134
4.3.1.	Implementation of the MANET IDS on Linux using NS 2	134
4.3.2.	Implementation and Testing of the MANET IDS on Linux and NS 2 against TCP SYN Flood, Blackhole and malicious traffic	144
4.3.3.	Implementation of MANET IDS prototype on Raspberry Pi and Generic Smart-watch...	147
4.4.	Performance of The Fused Anomaly-Based Intrusion Detection Model For MANET In Smart Healthcare.....	171
4.4.1.	Network Performance Test Results.....	172
CHAPTER FIVE		180
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS		180
5.1.	Introduction.....	180

5.2. Summary	180
5.2.1. Review security weaknesses in the application of MANETs for provision of smart health care.	181
5.2.2. Design a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.	182
5.2.3. Implementation of a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.	182
5.2.4. Evaluation of the Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.	183
5.3 Recommendations.....	184
5.4 Policy Recommendations.....	184
5.5 Recommendations for Future Research.....	184
REFERENCES.....	186

LIST OF TABLES

Table 1: Research Gap in General ABIDS Techniques for MANET	84
Table 2: Research Gap in Machine Learning based Techniques for MANET	84
Table 3: Summary of MANET Vulnerabilities that propagate security threats.	107
Table 4: Malicious Packets permeated into the MANET successfully.....	110
Table 5: Categorization of packet flow to interest data	117
Table 6: Logical and Data Propagation Scenarios in a FSM	121
Table 7: State transition variables that identify invalid inputs within a TCP session.....	128
Table 8: State logic assignment for TCP activity between TCP devices	129

LIST OF FIGURES

Figure 1: Enabling technologies for successful deployment of the Internet of Things (Source: Khan, Chen, & Hulin, 2014).....	17
Figure 2: Device-to-Device Communication Underlying Cellular Communications Systems (Source: Jani et al., 2009).	19
Figure 3: Conceptual Framework (Source: Author)	87
Figure 4: An open vulnerable MANET	109
Figure 5 : The Logical Topology for Data Capture from the smart watch Bluetooth device (Source: Author)	116
Figure 6 : Data Classifier Model (Source: Author)	117
Figure 7: SVM into ANN integration (Source: Author).....	119
Figure 8: Finite State Machine Design and Construction Process (Source: Author)	120
Figure 9: FSM for Normal Transition Activity in MANETs (Source: Author)	123
Figure 10: Shows the Normal and Anomalous FSM for TCP sessions in MANETS (Source: Author).....	124
Figure 11: Design of a Formal Notation in Voluminous Amplification in UDP and Denial of Sleep.....	126
Figure 12: Minimization of Finite State Machines	127
Figure 13: Model Design for the Fused Machine Learning Intrusion Detection Model For The Provision Of Smart Health Care In MANETS.	132
Figure 14: MANET with RFDs, FFDs and Malicious radios.	136
Figure 15: Normal Data Propagation Mode.....	138
Figure 16: MANET With Malicious Data is Propagating	138
Figure 17: Topology of a blackhole attack.	142
Figure 18: Implementation of IDS script for blackhole learning environment	142
Figure 19: MANET topology with IDS algorithm injected into the FFD	144
Figure 20: Smart Watch Physical Address	148
Figure 21: Smart Watch Collecting Readings	149

Figure 22: Raspberry Pi 3Model B+ (Source: Buy a Raspberry Pi 3 Model B – Raspberry Pi. (n.d.). Retrieved from https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/) ...	151
Figure 23: Snort analyzer sniffing packets on the MANET	153
Figure 24: MANET under manipulated DDOS attack	155
Figure 25: Output.....	161
Figure 26: KNN accuracy of Testing and Training Data.....	162
Figure 27 IDS introduced into the MANET detects anomalous activity (DDOS)	167
Figure 28: IDS Fusion Framework	168
Figure 29: Average delay comparison with Fused IDS and malicious traffic	174
Figure 30: Average throughput on the MANET while under attack with Fused IDS	177
Figure 31: DDOS Packet Delivery Ratio with the Fused IDS	179

ABBREVIATIONS

ABIDS	Anomaly Based Intrusion Detection System
AES	Advanced Encryption System
ANN	Artificial Neuron Networks
CCTV	Closed Circuit Television System
CPU	Central Processing Unit
D2D	Device to Device
DDOS	Distributed Denial of Service
DNA	Deoxyribonucleic Acid
DOS	Denial of Service
FFD	Fully Functional Device
FIRE	Fuzzy Intrusion Recognition Engine
HIOT	Health Internet of Things
IDS	Intrusion Detection System
IEEE	Institute of Electronic and Electric Engineers
IOT	Internet of Things
LoRa	Low Power Radio
MAC	Media Access Control
MANETs	Mobile Ad hoc Networks
NSA	National Security Agency
PPDIO	Prepare Plan Deploy Implement Optimize
RFD	Reduced Functional Device
RFID	Radio Frequency Identifier
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time

SABIDS	Statistical Anomaly Based Intrusion Detection System
SDN	Software Defined Networking
SMAC	Sequential Multiple Analysis Computer
SSL	Secure Socket Layer
TDMA	Time Division Multiplexing
UNIX	Unix Operating System
WIFI	Wireless Fidelity
WSN	Wireless Sensor Network

DEFINITION OF TERMS

- Anomaly-based** This is a type of intrusion detection system that identifies malicious activity and intrusions by monitoring system activity and classifying it as either normal or anomalous (Kondaiah and Sathyanarayana, 2018).
- Internet of Things.** This is a network of physical devices that are embedded with computing device such as software, sensors, actuators, and radios which enables these things to connect, collect and exchange data (Karlsson, Dooley and Pulkkis., 2018).
- Intrusion Detection Systems** This is a software system that monitors a network so as to identify and warn of malicious activity or policy violations (Spanos., 2018).
- MANET** A mobile ad hoc network (MANET) is a wireless ad hoc network or ad hoc wireless network that is self-configuring, self-healing and infrastructure-less network of mobile devices that have been interconnected wirelessly through radio technology. They are characterized by limited bandwidth, shared transmission channels and lack centralized processing unit of administration (Gammar, Amine & Kamoun, 2010)
- Smart Healthcare** Use and implementation of various technologies so as to achieve and leverage on better diagnostic tools, better treatment for patients, and devices and consequently improve the quality of life for all (Solanas et al., 2014).

CHAPTER ONE

INTRODUCTION

1.1. Introduction

This chapter gives a brief background of the main concepts and problems informing this study specifically on the need to deliver a secure smart health care Intrusion Detection System (IDS) primarily based on the fusion of contemporary machine learning techniques. It further proceeds to state the research problem, outlines the research objectives and defines the scope, significance and the expected outcomes of the study.

1.2. Background of the Study

The use of smart devices in provision of healthcare provides numerous benefits. Use of technology in the healthcare profession has not only led to easier diagnosis but has also enhanced the accuracy, lowered the costs, promoted the health of workers and researcher(s) collaboration, and facilitated the development of efficient and effective healthcare systems as well (Reddy et al, 2018). The provision of smart healthcare services is dependent on the Internet of Things that run on MANETs. While it is particularly indispensable, security of the systems and data remains a critical challenge that hinders the accelerated adoption of smart health care (Iyengar, Kundu & Pallis, 2018).

A Mobile Ad Hoc Network (MANET) is a collection of wireless mobile devices that create a temporary network without the need of a pre-existing network infrastructure backbone or centralized administration (Pullin, Pattinson, & Kor, 2018). It is decentralized, self-healing and adaptive gathering of independent mobile devices. These MANETs communicate over wireless links. Each device in a MANET is both a network user and a network router. Due to the

high mobility of these devices and their owners, the network topology quickly changes frequently and unpredictably (Pullin, Pattinson, & Kor, 2018).

Provision of smart healthcare services is dependent on technologies such as MANETs and the Internet of Things (IOT). IOT is a highly interconnected network of a loose collection of disparate, purpose-built networks (Yang, 2010). The IOT supports deployment of wireless sensors and distributed applications in communication, health, security among other fields, based on ubiquitous computing and convergent networks (Gope, Hwang, 2016). It is predicted that there will be over 50 billion Internet of Things (IoT) devices by the year 2020 (Cisco, 2011). A typical family home may comprise of more than 500 smart devices by 2022 (Gartner News, Sep 8, 2015).

While their adoption for domestic purposes may experience an upsurge, smart devices have limited resources in low storage, limited processor capability thus are vulnerable to eavesdropping, malicious attacks, packet sniffing and other security threats, which in turn affect the security of the entire IOT ecosystem (Li et al 2016). It is also predicted that more than 25% of identified cyber security attacks in enterprises will involve the Internet of Things (Gartner News, Sep 8, 2015). IOT smart devices in healthcare generally revolve around a MANET formation since the users are mobile. The incentives for MANET's use in the health and medical fraternity as a whole has also been on usability, financial and life-saving capabilities of the technology making the technology acquire its own path in MANET development, there has been various acronyms for these technologies like Internet of Medical Things (IoMT) and Healthcare Internet of Things (HIOT) (Patel et al., 2010).

MANETS have various protocols that perform various functions. An Ad Hoc On-Demand Distance Vector (AODV) routing protocol primarily supports mobile ad hoc networks. This protocol is responsible for establishing routes to destinations when requested and propagates unicast and multicast routing. This AODV protocol was jointly built by Nokia, the University of California and the University of Cincinnati (Hassan et al., 2018). The Temporally Ordered Routing Algorithm (TORA) is an algorithm that enables routing of data over Wireless and Mobile ad hoc networks. This algorithm reduces the number of required control messages within a network (Patel & Tripathi, 2018).

The Destination Sequenced Distance Vector (DSDV) is a distance vector protocol that implores devices to intermittently update routing information. Devices on the network build a routing table that corresponds to ports, network and the distance to each of them. The protocol thus avoids the formation of routing loops that eat up resources on the network (Singh & Dhir, 2018). The Optimized Link State Routing (OLSR) protocol is a link state algorithm created to meet requirements of a mobile adhoc network. The protocol minimizes the message data by enabling devices to resend packets. The difference in this protocol is the ability to propagate, unlike classical link state algorithms, temporary link state information within the network (Poularakis, Iosifidis, & Tassiulas, 2018).

It is reported that smart healthcare devices experience twice the number of cyber security attacks as opposed to other industries. FortiGuard Labs that provides cyber security defense mechanisms reported that the healthcare sector experienced an average of 32,000 intrusion cyber security attacks per day in 2017. This is in comparison to 14,300 attacks in other industries (Adefala, 2018). In a recent cyber-attack, regarded as the biggest distributed denial-of-services

(DDoS) attack ever experienced, a botnet of thousands of hacked IOT smart devices redirected traffic to a European based webserver in 2018 with packets exceeding one terabit per second (Liu, Jin, Hu, & Bailey, 2018).

It was recently reported by Ars Technica (Urquhart & McAuley, 2018) that hackers wrestled control over various IOT devices including cameras, routers and other IOT devices. They then initiated several DDoS attacks, which propagated data exceeding 300 Gbps. In 2014, a children's Boston Hospital was a victim of a consistent DDoS attack, whereby hackers against Justina Pelletier at that time withheld at the hospital in Boston against her parents' wishes, were seeking her release. (Hongach, 2018). Recently a National Security Agency (NSA) which is an American national intelligence agency of the United States Department of Defense cyber weapon - WannaCry was spread across the world, it infected at least 200,000 Windows based machines, which included systems at more than 45 hospitals in the United Kingdom. Various medical devices and technology-based healthcare devices were affected too (Kao & Hsiao, 2018). Orangeworm hackers also attacked X-Ray and MRI Machines by targeting critical systems executed by major international healthcare companies based in the United States, Europe, and Asia with a key focus on the healthcare devices (Arapi, 2018).

Any MANET implementation, therefore, needs to embed security and privacy by design which should be part of any MANET project, use case or deployment. Leveraging on MANET and bio-medical data aims to improve and reduce errors and costs. Making sure data and devices do not get exposed or used for the wrong reasons is a key proposition for any Smart Health implementation. The personal or confidential nature of health data makes it considerably

challenging when implementing MANET as the threshold for security and privacy is much higher, even supported by regulation (Munns & Basu, 2017).

Denial of Service attacks are common threats in MANETs that deny users from accessing the system(s) and information when and if they require it. In particular, a DOS/DDOS attack targets a device by using malicious unwanted response requests thereby draining resources and rendering the device unable to respond to genuine user requests (Hui, Kim, & Wang, 2017). Two strategies can be employed to identify such DoS attacks; Signature detection and Anomaly detection. The Signature analysis techniques employ the tactic of consolidating information in a manner similar to expert systems methodology. The signature approach however uses the information that is consolidated in a different way, by deciphering and breaking down data into a series of appraised events thereby decreasing the alarm threshold of the intrusion system (Kumar, Mangathayaru, & Narsimha, 2016). Efficiency is the utmost trait of this signature based analysis technique, which has enable its implantation in the market as a viable enterprise security system. The solution however has a major bottleneck in the requirement for regular updates in order to protect the network from newly discovered threats (Iqbal et al., 2016).

An anomaly-based intrusion detection system, monitors and alerts intrusions and misuse by observing activity that falls out of normal system operation. This is opposed to signature-based systems, which can only detect attacks for which a signature has previously been created. Anomaly based intrusion detection has the capability to identify unknown intrusions as well as a zero day assaults. The strength of this emerges from the capability of ABIDS to model standard operation disposition of a network and further identify deviations from the baseline. ABIDS can also be specifically configured to suit a particular network thus making it challenging for a

previously successful attack in one network to be replicated in a unique setting (Gai, Qiu, Tao, & Zhu, 2016). Anomaly Based intrusion detection systems can implement different methodologies in either; an artificial intelligent knowledge-based detection, statistical anomaly detection, data-mining based detection or a machine learning based detection algorithm. This study will put into consideration, the following five case scenarios that a good intrusion detection model should seek to satisfy as the following;

- i. Which are the most appropriate attributes and measures that can best describe a series of logical events?
- ii. What is a good model of anomalous activity within an ad-hoc wireless network that signifies an attack?
- iii. Which modeling rule should be integrated so as to identify anomalies within logical events without having false alarms?
- iv. Which is the most optimal model for an intrusion detection system that would best protect ad-hoc wireless networks?
- v. How can the security system acquire this knowledge ubiquitously?

Anomaly Based intrusion detection systems can implement different methodologies in either; an artificial intelligent knowledge-based detection, statistical anomaly detection, data-mining based detection or a machine learning based detection algorithm. These MANET systems can also be specifically configured to suit a particular network thus making it challenging for a previously successful attack in one network to be replicated in a unique setting (Gai, et al., 2016).

There are various divergent anomaly-based intrusion detection techniques used depending on the type of data processing related as well as behavioral model. These techniques include: Statistical, Operational or threshold metric model, markov model, statistical moments or mean/standard deviation model, multivariate model, univariate model, cognition based, time series model, finite state automata, descriptive scripts, adept techniques, artificial intelligent machines, Bayesian techniques, genetic algorithms, neural artificial networks, outlier intrusion detection systems, fuzzy logic, computer immunology based and user intention based techniques (Jyothsna, Prasad, & Prasad, 2011). This study will delve into the above techniques and propose a fused system by employing beneficial attributes of these models.

The Bayesian Approach uses illustrational diagrams to reveal possible interactions between various devices and resources in a network. Where data flow is unpredictable in nature and intrusion case scenarios cannot be pre-determined (Kabir, Onik, & Samad, 2017). This technique is great for consolidation of an existing intelligent formation (Khan, et al, 2016). Implementing this approach enables the capability to anticipate consequent user actions by any device or user in the network. Neural networks in the construction of intrusion detection systems reveal an efficient substitute to statistical methodologies. Neural networks are commonly found in anomaly-based intrusion detection systems (Roy et al, 2017). This can be used with data mining so as to diminish amounts of the input data as well as to choose occurrences which expose traffic anomalies (Atre & Singh, 2016). Effective for identifying innovative and new network attacks. Fuzzy systems have the ability to dynamically consolidate data inputs fed from variable devices in the network. Fuzzy logic techniques enable quick construction of “if-then” criteria that can mimic and recognize unauthorized intrusions (Mkuzangwe & Nelwamondo, 2017). The accuracy and effective results generated enables a classification and ranking. The

better ranked domain applications edge out moribund and programs deemed less effective. Sequentially high performing applications outlast those with a low accuracy, ultimately only the strong survive (Bhattacharjee, Fujail & Begum, 2017).

A benchmark intrusion detection dataset (ISCX) was used to evaluate the efficiency of the kM-RF, and a deep analysis is conducted to study the impact of the importance of each feature defined in the pre-processing step (Soheily-Khah, Marteau & Béchet, 2017). A machine learning neural network algorithm was used to characterize legitimate communications and to identify suspicious scenarios by profiling and monitoring of the Radio Signal Strength Indication (RSSI) (Roux, Alata, Auriol, Nicomette & Kaâniche, 2017). Sultana, Chilamkurti, Peng, and Alhadad, (2018) defined a networking (SDN) technology as a platform using ML/DL approaches to detect vulnerabilities and anomalies. In addition, anomaly detection in a network was achieved by a learning machine detection and analysis of seasonal network patterns for anomaly detection which may construct a model of normal network behavior, to detect data points that deviate from the norm (Sartran et al., 2017)

1.3.Statement of the Problem

Security of smart devices used in the provision of healthcare is extremely vital. These devices are such that they provide mission critical support in healthcare through supporting, monitoring vital signs, deviations from norm, lifestyles and body metabolism. Devices in smart health include but are not limited to wearable heart monitors, body sensors and pacemakers. These devices are primarily adorned so that deteriorating patient's medical condition is observed and identified or alerted in time. However, these devices, which run on MANETs, are such that their physiognomy lacks the adequate capability to devise robust systems to shield themselves against

eavesdropping, malicious attacks, packet sniffing and other security threats, especially DDOS. DDOS attacks can conceal deteriorating health risks from discovery by both the patient and health specialist thus can lead to death, immobility, permanent or impaired disability. A patient's worsening condition might not be alerted to both the patient and health specialist as envisaged. Medical concerns within hospitals include unexplained fainting, unexplained stroke, heart palpitations and atrial fibrillation that need to be monitored remotely, accurately and effortlessly. Cases of DDOS attacks against smart health devices are on a rapid rise. Most IDS techniques fail to capture and take account of the characteristics of MANETs, which malicious attacks exploit.

1.4. Device Objectives of the Study

1.4.2. Overall Objective

To design, implement and evaluate a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

1.4.2. Specific Objectives

- i. Review Security Weaknesses and Obstacles in the Application Of MANETS for Provision of Smart Health Care.
- ii. Design a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.
- iii. Evaluate the Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

1.5. Research Questions

- i. What are the Security Weaknesses and Obstacles in the Application Of MANETS for Provision of Smart Health Care?
- ii. How can a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS be Designed?
- iii. How can a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS be implemented?
- iv. How can the Fused Machine Learning Model for the Provision of Smart Health Care in MANETS be evaluated?

1.6. Significance of the Study

This study contributed greatly to the pool of knowledge in MANET by bolstering the deployment and application of MANET technologies to solve healthcare with minimal privacy and security concerns. Security of these devices will help close healthcare access gap and aid in improvement of the overall human development. Improvement in security will accelerate the adoption of heart monitors, body sensors, pacemakers that will greatly increase and improve quality of healthcare. This will greatly assist in elongating life and improving quality of life in people suffering from heart conditions, high blood pressure, cholesterol issues and cardiac misnomers like syncope, atrial fibrillation and other irregularities. Results of this study will help minimize privacy concerns and reduce technology stigma. It will also offer endless opportunities for development of ICTs in healthcare.

1.7. Scope of the Study

This study focused only on monitoring indefinite data and addressing anomalies for packet propagation from within an MANET network and observing anomalies therein to detect an intrusion. A model was only designed to guide the general structure of the intrusion detection system for MANET. The research was performed by having a simulation of the MANET environment on Network Simulator 2, then evaluate the same using Linus NS 2 machine learning environment by integrating various. Further, smart wearable devices and Raspberry Pi Boards were implemented to replicate a smart healthcare environment.

1.8. Justification for this Study

The current methods used in the industry like electrocardiography are extremely expensive and disruptive. This study helped in identifying and monitoring cases like myocardial infarction, a third heart sound, cardiac murmurs other findings suggestive of a structural heart disease that can cause shortness of breath, fainting or collapse, seizures or heart attacks remotely, effortlessly and accurately. There was need to also monitor the effects of a medication on the heart as well as assess severity of electrolyte abnormalities, such as hyperkalemia by using MANETs. Protection and safeguarding of patients confidential data and information will prop adoption of accurate technology for health monitoring while incurring minimal costs. The improvement of data security will assist in the adoption of smart healthcare among the populace thus improve production and reduce the overall cost of acquiring and implementing secure smart healthcare systems. The benefits of usage of secure smart health devices, through intrusion detection in MANETs is that the general public will be more assured of the safety of their data, while service providers and users will be confident of the security of both systems, data and general health eco-system.

1.9. Limitations

This study was limited to studying and implementing wireless technologies that propagate MANET standards. The study was limited to devices currently used within the healthcare sector primarily for heart monitoring only and can be implemented in Kenyan hospitals. This study only focused on the design of a prototype model to demonstrate the working of the proposed secure MANET system.

1.10. Conclusion

This chapter gave a brief background of the main concepts and problems informing this study specifically on the need to provide a secure smart health care ecosystem through provision of an anomaly-based intrusion detection system. It further proceeded to state research problem, research objectives, defined the scope, significance and the expected outcomes of the study.

CHAPTER TWO

LITERATURE REVIEW

2.1.Introduction

This chapter will present a discussion on the internet of things examples, uses of MANET, protocols, smart homes, industries and smart health in focus with a bias on wireless network security, types of attacks, security models and related works in healthcare provision and the various security threats that target such systems. In addition, it reviews various intrusion detection techniques in use in the industry. The theoretical and conceptual frameworks for the study will also be presented and discussed.

2.2. Security Obstacles in the Application of MANET for Provision of Smart Health Care

MANETs are by their very nature mobile and dynamic. Thus, inherently, they possess neither a fundamentally uniformed coordination nor rigid hierarchical topology architecture. This, coupled with lack of a centrally coordinated security system, makes these devices especially vulnerable to attacks as opposed to wired networks. Smart devices have limited resources in low storage, low memory and limited processing power. Their inherent nature incapacitates their ability to shield themselves against eavesdropping, malicious attacks, packet sniffing and other security threats. The confidentiality, integrity or availability of systems and data becomes compromised as a result.

2.1.1. Review of MANETs in General

Mobile Ad Hoc Networks (MANET) is a network of physical electronic devices that are embedded with electric and electronic components, software, sensors, actuators, radios and rooted within everyday tools or machinery like home appliances, vehicles, wearable devices like

watches, doors, traffic lights, switches and other items (Alagoz et al., 2017). These devices have radios that enable connectivity to networks which permit these devices to connect and exchange data with other devices or networks (Vermesan, et al., 2011). The Internet of Things system allows interrelated computing devices, mechanical and digital machines, objects, animals or people that have unique addresses that gives them the ability to transfer data over a network effortlessly without human-to-human or human-to-computer intervention (Agrawal & Vieira, 2013).

While each of the internet of thing is uniquely identified using layer 2 addresses, within its embedded computing, firmware or operating system, the device also has the ability to inter-operate within the Internet infrastructure as well as local network (Savolainen, Soinen, & Silverajan, 2013). The MANET enables devices to be sensed, connected, communicated or controlled remotely over the existing wireless network infrastructure (Hsiao, Lian, & Sung, 2016). This provides an enabling opportunity for more direct integration with the physical world and computer-based systems. These healthcare systems in MANETs result in improved ease, efficiency, accuracy and economic advantage as well as reduced human involvement in their control and usage (Nigam, Asthana & Gupta, 2016). Once MANET is amplified with sensors and actuators, this wireless technology becomes an exquisite and ubiquitous manifestation of the common cyber-physical computerized systems commonly found in working, business, industrial or agricultural fields.

These MANET are also encompassed and adopted by other technologies such as smart homes, smart grids, smart cities, virtual power plants, smart agriculture, smart weather and intelligent transportation (Nigam, Asthana, & Gupta, 2016). These "Things", refer to a wide

assortment of electronic devices such as wearable technologies, health and heart monitoring implants, biochip transponders on wildlife and domesticated animals, CCTV cameras that stream audio visual data, car tracks, DNA analyzing devices in the environment, food, disease, pathogens, buildings, and even field operation devices used in firefighting, search and rescue operations (Bedi, Venayagamoorthy, & Singh, 2016). These things are a tangled mix of hardware, software, data and service. Consequently, the consequences of embedding the internet of things with minuscule addressable devices or machine-readable sensors would be transformation of business, security and daily activities.

The capability to interact remotely with devices based on a person's immediate needs, greatly eases and improves quality of life (Stankovic, 2014). The interconnection of these devices allow for generation of data from remote devices to other objects includes the notion of the connecting the physical world with a virtual world into a multi-level oriented architecture with the nature and devices at the bottom level, the Internet, sensor network, and mobile network, and intelligent human-machine communities at the top level. This architecture disperses users and enables them to accomplish tasks ubiquitously as well as solve everyday problems by using the neural ad-hoc network. This network enables the active flow of data, information, knowledge, material, energy and services in the global troposphere (Wu, Meng, & Gray, 2017). This gravitating superlative model envisioned the development and growth of the Internet of Things (Al-Fuqaha et al., 2015). The figure 4 below by Khan et al, demonstrates the enabling environment and technologies for successful deployment of the Internet of Things.

A wireless sensor network (WSN) consists of independent objects with embedded software, electronic and electric capability. These objects ubiquitously monitor the physical and

environmental conditions like temperature, sound, wind, pressure, motion, energy or pollutants, at various facets and locations (Joshi et al., 2012). The actuator sensor nodes are at the very foundation of these devices (Yusuf, 2014). Devices in these Low power Wireless Personal Area Networks are applications under the IEEE 802.15.4 specification which broadcast radio frequency within a short area. This specification provides a foundation upon which smart devices can propagate data through the Internet Protocol under IEEE 802.15.4 networks. These Low power Wireless Personal Area Networks integrate various compression methods which allow communication between devices over 802.15-based wireless networks. Applications used in the Low power Wireless Personal Area Networks are closely related to internet communication (Ana de Pablo Escolà, 2009). Figure one (1) below illustrates the various enabling technologies necessary for the deployment of internet of things and accompanying technologies.

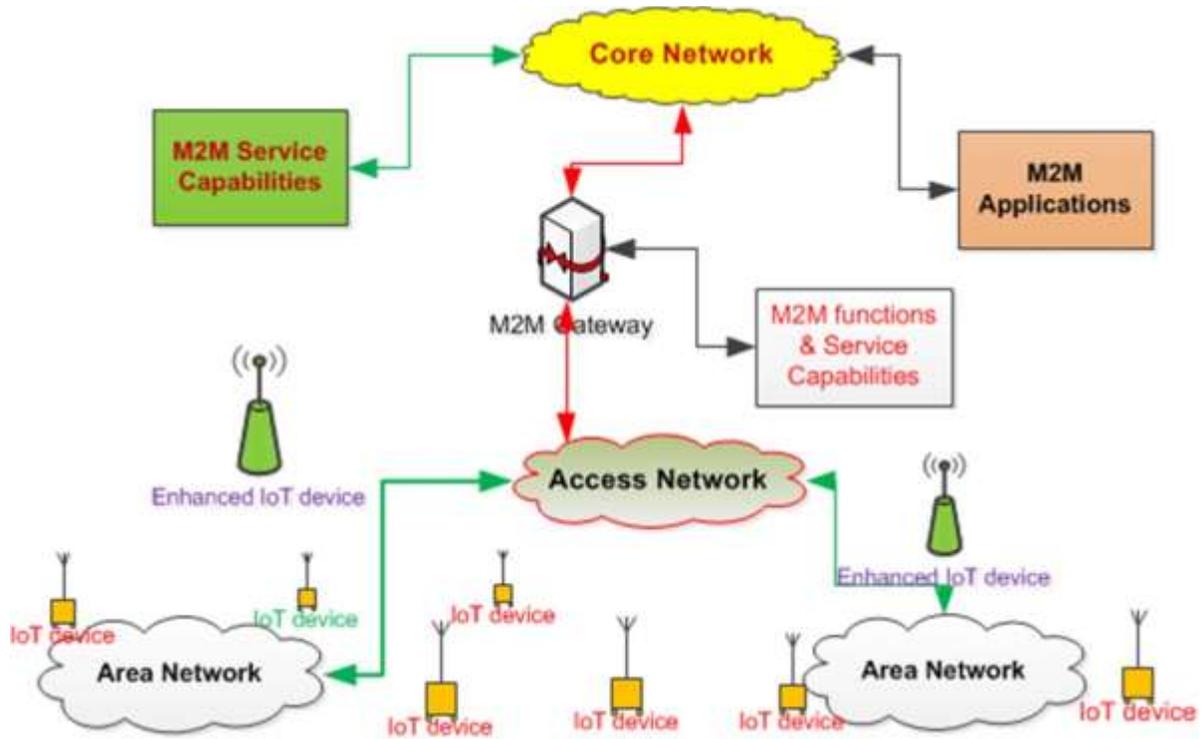


Figure 1: Enabling technologies for successful deployment of the Internet of Things (Source: Khan, Chen, & Hulin, 2014)

2.1.2. History of the Internet of Things

Kevin Ashton, a technology pioneer and assistant brand manager at Procter & Gamble (Mehta, Kale, & Utage, 2017), coined the terminology “the Internet of things” in 1999. The AutoID Center consortium at MIT formally began research on the Internet of Things (Michael, 2017) The Internet of things industry has then evolved into a convergence of numerous communication and wireless technologies, that enable ubiquitous wireless communication, embedded sensors, real-time analytics, artificial intelligence and machine learning, and embedded systems (Norman, 2017).

The genesis enabler for the MANET concept was that of a network of smart devices which was in 1982, where a modified Coke dispensing machine at Carnegie Mellon University was extraordinarily the first device to have an Internet connection (Saha, Mandal, & Sinha, 2017). This appliance provided information on available stock count and whether sodas were chilled (Breur, 2015). Mark Weiser's vision in 1991 presented through a seminar paper titled "The Computer of the 21st Century" drew ideas on the ability to have ubiquitous computing (Kušen, & Strembeck, 2017). The dream was further driven through various academic venues such as UbiComp and PerCom which enabled a contemporary vision of the Internet of Things (Kaur, & Saini, 2017).

Later in late 1994 Reza Raji described the concept of moving small packets of data between a large set of nodes enabling home automation, through home appliances as well as automation of entire factories. Further, several companies between 1993 and 1996 anticipated solutions in the MANET spectrum. Nonetheless, it's only in 1999 that the MANET industry started gathering traction. In 1999 Bill Joy presented to the World Economic Forum his revolutionary Device to Device (D2D) communication technology that allowed exchange of data between two devices, at Davos (Borgohain, Kumar, & Sanyal, 2015). The figure 2 below shows the underlying communications systems for Device-to-Device Communication.

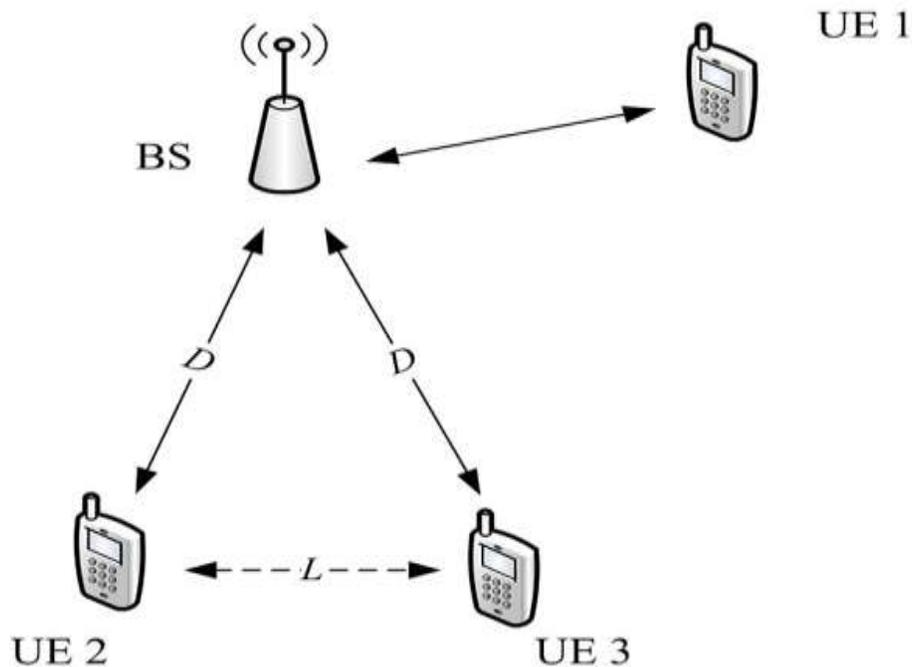


Figure 2: Device-to-Device Communication Underlying Cellular Communications Systems

(Source: Jani et al., 2009).

Device-to-Device also referred to as D2D communication is a common logical topology found in mobile and global systems for mobile (GSM) networks which enables direct communication between two mobile devices without assistance from the Base Station or telecommunication service provider network. This type of D2D communication is generally not visible to the telecommunication service provider network and occurs on the unlicensed spectrum.

The concept of the Internet of things was popularized in the early 1999, at the MIT's Auto-ID Center and further through related market-analysis reports and publications (Sundmaeker, Guillemin, Friess & Woelfflé, 2010). Use of Radio-frequency identification (RFID) made a great contribution where one of the founders of the Auto-ID Center - Kevin Ashton espoused and broke ground for the Internet of things by providing and leading in MANET research. Kevin Ashton phrased MANET as a scenario where all objects, machines and

people in daily life were embedded with identifiers, thus allowing computers to manage and audit them (Ashton, 2011). In addition to the use of RFID in tagging most of the internet of things, this phenomenon could be accomplished through technologies such as barcodes, near field communication, Quick Response codes and digital watermarking (Vongpradhip & Rungrungsilp, 2012).

Gartner projected that 6.4 billion Internet of Things will be in use in 2017. This forecasts that the number will grow tremendously three-fold to 21 billion by the year 2020. (Morgan, 2014). Yu et al., (2015) estimated that the number of MANET devices deployed will grow exponentially from 5 Billion in 2015 to 25 Billion by the year 2020. Other studies in 2013 estimated that 9 billion MANET devices were in use and forecasted the numbers to grow beyond 24 billion 2020 (Gubbi, Buyya, Marusic & Palaniswami, 2013). Other experts have put evaluations on the MANET indicators with the ecosystem consisting of 30 billion objects by 2020, and further a 17 to 32 percent annual growth, thus the MANET industry will grow to be more than a trillion-dollar market before 2020 Sheth, Jaimini, and Yip (2018).

While various radio wavelength technologies are in use, the most common are Light-Fidelity, Near-field communication (NFC), QR codes and barcodes, Radio-frequency identification (RFID), Thread, Wi-Fi, Z-Wave and ZigBee. By 2011, there were 2 billion Wifi certified devices in use for various connectivity purposes (Bartoli et al., 2011). Bluetooth has the lions share with 3 billion Bluetooth enabled devices in the market in 2014, in addition over 10 billion Bluetooth devices will be available on the global market by 2018.(Chang, 2014).

Wireless Sensor Networks mainly exhibit a small packet data size that propagates a packet size of 127 Bytes and 81 octets for data packets. WSN support both 16-bit short as well as the

IEEE 64-bit extended MAC addressing scheme. WSN also have low throughput and low consumption of bandwidth as well, with data rates of 250 kbps, 40 kbps, and 20 kbps for the physical layers. These technologies support logical topologies like star topology; however the most commonly logical topology is the mesh. WSN also support high device density, with up to 60,000 wireless devices connecting in a network. In addition, the devices use low power by muting its sending/receiving capability when not required. Devices in such networks are classified either into a full function device (FFD) or reduced function devices (RFD) (Ana de Pablo Escolà, 2009).

2.1.3. General Global Usage of the Internet of Things

Global usage and application of the internet of Things is widespread and broad. There exist numerous fields and classifications from a consumer, enterprise, government, institutional and infrastructure applications as well. The Internet of things is regarded as the subsequent stage of the information age and referenced the inter-connectivity of all industries from smart health, education, urban transport and domestic devices (Duan, Chen & Xing, 2011).The processing ability, memory and power on these devices enable the MANET to interconnect and communicate in all possible fields and tenets. These objects in diverse dispositions and natural ecosystems to buildings and industries (Olufisoye, 2016) ubiquitously, reliably, remotely and effortlessly harness the collection of data and information.

2.1.3.1.Consumer Applications

There is a rapidly blossoming adoption of MANET devices specifically for consumer use. These include of consumer home applications such as self-driven and connected cars, home and mobile entertainment, wearable technology, smart home automation, connected health, and

domotic appliances like television, cookers, ovens, dryers, washing machines, air vacuums cleaners, air purifiers, refrigerators and freezers that are remotely connected for control and monitoring (Wang, Lei, Wan, Zhang & Li, 2017).

2.1.3.2.Smart Homes

Smart Homes unveils numerous advantages including the ability to create an amicable and environmentally friendly home by pushing mundane tasks and duties beyond the dweller. This is achieved through automation of some functions such as switching off and on lights and electronics are turned off. One of the major obstacles to obtaining smart home technology is the high initial cost. MANET domotic devices have played a great part in enabling home automation in our everyday home activities (Batalla, Mastorakis, Mavromoustakis & Zurek, 2016).

2.1.3.3.Business and Enterprise Usage of MANET

Use of MANET in running business and enterprises is estimated to account for nearly 40% or 9.1 billion devices by 2019. MANET is in use in insurance aiding the automobile insurance claims resolution process. A sensor-enabled car notifies the owner by alerting their smartphone, about a car incident and further contacts an insurance claims representative (Manral, 2015).

2.1.3.4.Usage of MANET in Broadcasting and Media

The fourth estate has embraced smart reporting and the Internet of things through marketing and studying consumer habits. Through behavioral targeting, these objects collect and stream numerous actionable facts of information about millions of individuals. These connected profiles are consumed by media producers who parade display advertising according to the consumer's

recognized habits at a time and location so as to maximize its effect. Further information is collected by tracking how consumers interact with the given content (Lee & Cho, 2015).

2.1.3.5.MANET in Smart Infrastructure Management

MANET is crucial in efficient and remote monitoring and controlling of operations in both urban and rural infrastructures like roads, bridges, railways, solar and wind turbines. MANET can be useful in infrastructure monitoring for occasions or changes in structural conditions that can compromise safety and increase risk of such infrastructure (Eriksson et al., 2008).

2.1.3.6.MANET in Smart Manufacturing

MANET has enabled smart manufacturing through remote equipment control and management of manufacturing apparatus, tools management, workflow management, manufacturing process management through MANET thereby encompassing industrial applications into the smart everything fold. Intelligent MANET systems facilitate rapid manufacturing of machinery-made products, syncing with dynamic response to manufacturing-process product demands, automatic response to supply chain networks as well as real-time optimization of manufacturing production (Li & Kara, 2017).

2.1.3.7.MANET use in Smart Agriculture

There has been a considerable contribution of creative and innovative agriculture through adoption of MANET. Due to a high demand of food by the world's populace as well as challenging global weather and climate change, farmers have found it necessary to adopt the use of MANET (Cao et al., 2016).

2.1.3.8.MANET in Smart Energy Management

The ability to remotely control and interact with installations in the home, office and general hospitality industries by switching on/off lights, accessing heating systems, closing/opening water taps, controlling ovens though MANET is critical today. Integration of sensors and actuator systems within installations enables optimization of energy consumption in various ways (Sheikhi, Cimellaro & Mahin, 2016).

2.1.3.9.MANET in Smart Environment Management

Environmental monitoring applications collect data on the quality of air and water so as to enable better environmental protection (Jaiswal, Liu & Ling, 2016). Monitoring atmospheric and soil conditions is also crucial especially to check against erosion and pollution. Ability to remotely monitor wildlife migration, feeding, movements and mating activities is crucial (Elias, Golubovic, Krintz & Wolski, 2017).

2.1.3.10. MANET in Smart Cities and Metropolitans

Songdo in South Korea will most likely be the first of its kind fully equipped and wired smart city in the world. This will provide a constant flow of city data that is monitored and analyzed automatically and remotely by an array of intelligent systems with little human intervention (Muralidharan, Roy & Saxena, 2016).

2.1.4. MANET in Smart Healthcare

MANET devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as

pacemakers, Fitbit electronic wristbands, or advanced hearing aids (Shinde & Prasad, 2017) and (Koshti, Ganorkar, & Chiari, 2016). Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up (Elsokah and Zerek (2019). It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses.

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. Other consumer devices to encourage healthy living, such as, connected scales or wearable heart monitors, are also a possibility with the MANET. More and more end-to-end health monitoring MANET platforms are coming up for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements (Konstantinidis et al., 2017). The Research & Development Corporation (DEKA), a company that creates prosthetic limbs, has created a battery-powered arm that uses myoelectricity, a device that converts muscle group sensations into motor control. The arm is nicknamed Luke Arm after Luke Skywalker (Star Wars) (Benz, Yao, Rose, Olgac, Kreutz, Saha & Civillico, 2016).

The internet of things refers to a network of devices that work together to achieve a common goal in a given organization. The devices that comprise of this network of devices may include computers, vehicles and phones. The devices are fitted with sensors and software that enable them transfer data effectively. The MANET can be controlled remotely by those individuals tasked with that responsibility. Every item that is a component of the MANET in a given

organization can be uniquely identified, this means when data is being transmitted it is easy to identify the actual device that has transmitted it and its location. (Gubbi et al., 2013).

2.1.5. Security Obstacles in MANETs

Smart Devices lack the requisite ability to provide adequate security and privacy capability to guard against intrusions and thus MANET and wireless sensor networks cannot be assumed to have the capability to maintain confidentiality, authentication and integrity. Studies have shown that up to 70% of the MANET devices and networks are very easy to intrude and access without authorization (Lee & Lee, 2015). MANET devices face various challenges including;

2.1.5.1. Privacy of Data and Information

There lacks a universal standard and protocol for a comprehensive provision of privacy and confidentiality of data and information for MANET devices and networks. Some vendors ship devices with a backdoor ability to access and collect user information for future enterprise intelligence. There is a huge challenge in the privacy of data collected by the MANET devices, as well as protection and privacy of data stored or being transmitted by the wireless sensor networks (Sheng et al., 2013).

2.1.5.2. Data Security

MANET devices have a challenge of providing security of data collected and during transmission from attackers or un-authorized users as well as other devices that could be compromised on the wireless sensor network (Farooq et al., 2015).

2.3. Security Vulnerabilities in MANETs

There are numerous security vulnerabilities that threaten overall security of an MANET eco-system (Jeyanthi, Abraham and Mcheick, 2019). These vulnerabilities take advantage of weaknesses in the system, applications or physical devices on the wireless sensor network. These are elucidated as follows;

2.3.1.1. System Security Vulnerabilities

This System security looks at the ability of an MANET eco-system to identify, isolate, prescribe and proactively protect itself from vulnerabilities. There is a requirement to design, build a diverse security model and framework that can deliver an appropriate security for an MANET eco-system (Jeyanthi, Abraham and Mcheick, 2019).

2.3.1.2. Application Security Vulnerabilities

Application Security provides protection for upper layer 7 MANET applications so as to enable the MANET applications identify, isolate and protect themselves against attackers (Alam, 2019).

2.3.1.3. MANET Device Vulnerabilities

The wireless network ensures all communication between devices, actuators, gateways and other end and intermediary devices is protected from all forms of attack.

2.4. Common Attacks in MANETs

Common attacks in MANETs seek to compromise the CIA Triad. Thereby, Confidentiality, Availability and Integrity of data and systems is affected. Existing literature presents numerous IoT attacks as illustrated in the subsections below;

2.4.1.1. Leakage of Information which compromises confidentiality

Data and information collected and transmitted by the smart devices within an MANET wireless sensor network is susceptible to leakage (Rath et al, 2018). Data and information from these devices is easily leaked since there lacks sufficient data encryption that is applied either between gateway and sensors or between the sensors themselves. In addition, user authentication to prevent un-authorized access and/or enable detection of unwanted and unauthorized parties is often weakly implemented (Rath et al, 2018).

2.4.1.2. Denial of Service and/or Distributed Denial of Service which compromises availability

This is a common attack that denies users from accessing the system(s) and information when and if they require it. This DOS/DDOS attack targets a device by using malicious unwanted response requests thereby draining resources and rendering the device unable to respond to genuine user requests. While no data is leaked or exposed, it is very disastrous as it makes systems unusable and renders data/information un-useful (Dhindsa & Bhushan, 2019).

2.4.1.3. Falsification of data which compromises integrity

This attack happens when a wireless device is in communication with the gateway and the attacker successfully captures the collect packets in transition and alters the fields containing

routing information. As a result, the attacker can access the information therein and alter, leak or destroy the data/information as a whole. Most SSL mechanisms have the capability to protect against this type of attack, while unauthorized devices that gain access should be entirely blocked. Most of these attacks happen by rogue devices posing as passive and silent observers which actually eavesdrop and perform traffic analysis. These hostile and malicious devices, silently listen the communication (Ngomane, Velempini & Dlamini 2018)

When the WSN is not secure, there are numerous kinds of attacks that can be experienced on the MANET eco-system. Data and information on this WSN is then vulnerable to many kinds of attacks. It is necessary to implement security measures that will prevent the attacks and limit the impact caused by any kind of attack aimed at MANET Devices. Data modification is an example of such an attack (Mishra, Li, Pan, Kuhnle, Thai & Seo, 2017). This attack takes place especially when the attacker successfully manages to find his or her way in the eco-system. The attacker then goes on to modify the data that is being sent from one person to another without their knowledge. (Stallings, 2007). Sniffer attack is another example of network attack. A sniffer can be an application or device. This application/device can be able to capture data and even read transmitted data. A sniffer attacker can be able to analyze information on the WSN and enable them access the eco-system. A sniffer attack can be prevented by encrypting the data that is being transmitted from one point to another.

Another attack is known as compromised-key attack. A key can be a code that is used on the MANET system when one a user want to access information when this key is gets to the attacker it becomes a compromised key (Yeh, Tsaur & Juang, 2016). The attacker may use it to gain information from a secure communication. An application – layer attack is another example of attack on devices on a wireless sensor network. This attack mainly focuses on the applications

installed on the devices that propagate data on the MANET network. For example the attack can attack the firmware and operating systems that are installed on the gateways. With this attack, the attacker can be able to bypass the normal access controls that as set to secure the devices and MANET system. This simply implies that the data and information being shared over the MANET system will be at risk and therefore vulnerable.

The other types of attacks include denial-of-service attack (Yan, Yu, Gong & Li, 2016). This is the most difficult attack to handle because of the impact that it can cause. The attackers using this kind of threat can easily launch this type of an attack on the MANET network without being easily identified. The attacker posts as a legit user and the moment he or she gets to the system, the attack is launched and the system administrator can have a difficult to identify and prevent further attacks. Eavesdropping is another type attack that can be launched. (Ma, Wang, Lei, Xu, Zhang & Li, 2016). However this kind of attack is rare, its impact can be of much impact. Brute force attack is another type of an MANET network attack. In this kind of attack, the attacker launches his or her attack right on the front door of the MANET eco-system (Ghanem & Ratnayake, 2016). The attackers try to get onto the WSN by finding the password through a trial and error technique. The attacker may spend weeks trying to figure out the password when he or she gets it, he will be able to use it effectively (Friedrichs et al., 2002).

2.5.MANET Security Models for.

2.5.1. What is a Security Model

A security model refers to a symbolic representation of a particular security policy. It elucidates the requirements of the policy architects. A model is broken into a set of rules that should be adhered to within a computing system. A security model integrates various security requirements and delivers the obligatory arithmetic formulas, relationships, and domain structure

that must be adhered to so as to accomplish this security objective (Almorsy, Grundy & Müller, 2016).

2.5.2. Types of Security Models

There are various security models used to provide security in MANETs. They are categorized into either mathematical, data based, high level or logical.

2.5.2.1.State Machine Models

State machine models enable verification of the state of security within a system by taking into account all current instances of devices accessing the network resources. The State Machine presents a correlation by describing the abstract objectives and that if the devices can only access resources by means that are concurrent with the security policy, then the system is deemed to be secure (Amalfitano, Fasolino & Tramontana, 2015). State machines deliver a simple basis for significant security models. If error messages are experienced either on boot-up or during data propagation then the transition is deemed illegitimate and thus declared insecure. Such scenarios can be experienced forcing the system to halt, thereby providing protection for itself and data (Chung et al., 2015).

2.5.2.2.Bell-LaPadula Model

This model inherits its nature from the U.S. military which implemented mathematical time-sharing mainframe systems providing a secure domain without unauthorized access of confidential information (Ely, 2017). This Bell-LaPadula mathematical model was developed to negate unauthorized access by implementing multilevel security policies that defines the grand notion of a secure state machine. The Bell-LaPadula model's main objective is to avert leakage of confidential data from being accessed by unauthorized users (Wen, Cala, Watson &

Romanovsky, 2016). Systems that implement the Bell-LaPadula model are normally classified as multilevel security systems since users with diverse authorizations partake the model while the systems data is processed with different variations. This model is referred to as an information flow security model on the basis that information is not propagated in an insecure manner (Domingo & Wietgreffe, 2015).

2.5.2.3.The Biba Model

The Biba security model while similar was developed however after the Bell-LaPadula security model. It implements a state machine model that upholds the integrity of information being propagated under possible threats or on exposure (Bertino, 2015). The Biba Security Model can access and read data at lower security levels. This model segregates data by separating information in the integrity level from access by any device in a higher integrity level. The Biba security model is mostly used to enforce data integrity. It does put much focus on the flow of data, from one level to another (El Hassani et al., 2015).

2.5.2.4.The Clark-Wilson Security Model

The Clark-Wilson model was developed with an altered objective of protecting data integrity by preventing system devices from accessing, reading and making unauthorized modification of data, as well as prevention of fraudulent actions while mitigating against commercial transactional errors (Nazareth & Choi, 2015). The Clark-Wilson security model, operates on applications that restrict which actions users can and cannot perform on networked devices. The model provides an elaborate mechanism to mitigate corruption of data as well as buttress data integrity by both authorized and un-authorized users (Peters & Panayi, 2016). This

particular rule, guarantees that any crucial undertaking cannot be performed by only one entity (Bann, Singh & Samsudin, 2015).

2.5.2.5.Information Flow Security Model

The Information Flow Security Model to enhance security for information flow, without regard on the directional flow of data and information. The Information Flow Security Model guarantees data security at informational flow between devices whose data at the risk on the same level and the flow between different levels as well (Baek, Vu, Liu, Huang & Xiang, 2015). A system is deemed secure and free from fraud when there is no illegitimate flow of information. This model enables flow of data and information only from one security level to another as well as flow from one device to another.

2.6.Intrusion Detection Models for the MANET

Intrusion Detection System is a security measure that can be installed on a network to prevent potential attacks from taking place. The IDS allows network administrators to detect individuals trying to compromise the system so that they retrieve information from it. There are various activities that the administrators can implement in order to identify attacks. These includes security policies violation (Chaudhary and Shrimal, 2019). The IDS works best because it designed in a manner that enable it to detect the vulnerabilities on the system in which it is installed. For example, it can work on the basis of previous attacks that affected the network and work backwards to eliminate the chances of another similar attack.

According to (Rafsanjani, 2009; Kumar & Dutta, 2016), an intrusion detection system (IDS) is a software or hardware, or a combination of both, that monitors activities in a network or an information system with the aim of identifying and reporting malicious activities. Intrusion

detection is the process performed by this security system. According to Kumar and Dutta (2016), intrusion detection refers to the process of observing the activities taking place in a network or a system, analyzing them to establish any signs of security policy and standards violation, and notifying the relevant authorities of the identified violations. Therefore, the IDS automates the intrusion detection process in a network or system environment with the primary goal of finding and reporting activities that try to compromise the availability, confidentiality, and integrity of a network or system (Kumar et al., 2013). Therefore, it is a necessary second-line defense mechanism for a Mobile Ad-hoc Network.

To accomplish its goal, the IDS performs specific activities in sequence. IDS begins by monitoring network traffic or system activities. During this process, IDS compares all the activities or traffic with the system or network rules respectively. Secondly, IDS automatically establishes unauthorized, suspicious, or malicious activities or traffics that breaks the set rules. Lastly, the system triggers an alarm to notify the relevant authorities of the recognized traffic or activities (Kumar and Dutta, 2016; Kumar et al., 2013). As stated, the detection of malicious activities or traffic is achieved by comparing them with the existing policy rules. According to Kumar and Dutta (2016), the rules include interval rule, retransmission rule, delay rule, integrity rule, jamming rule, and radio transmission range. These rules are used as the basis against which the network traffic or system activities are compared.

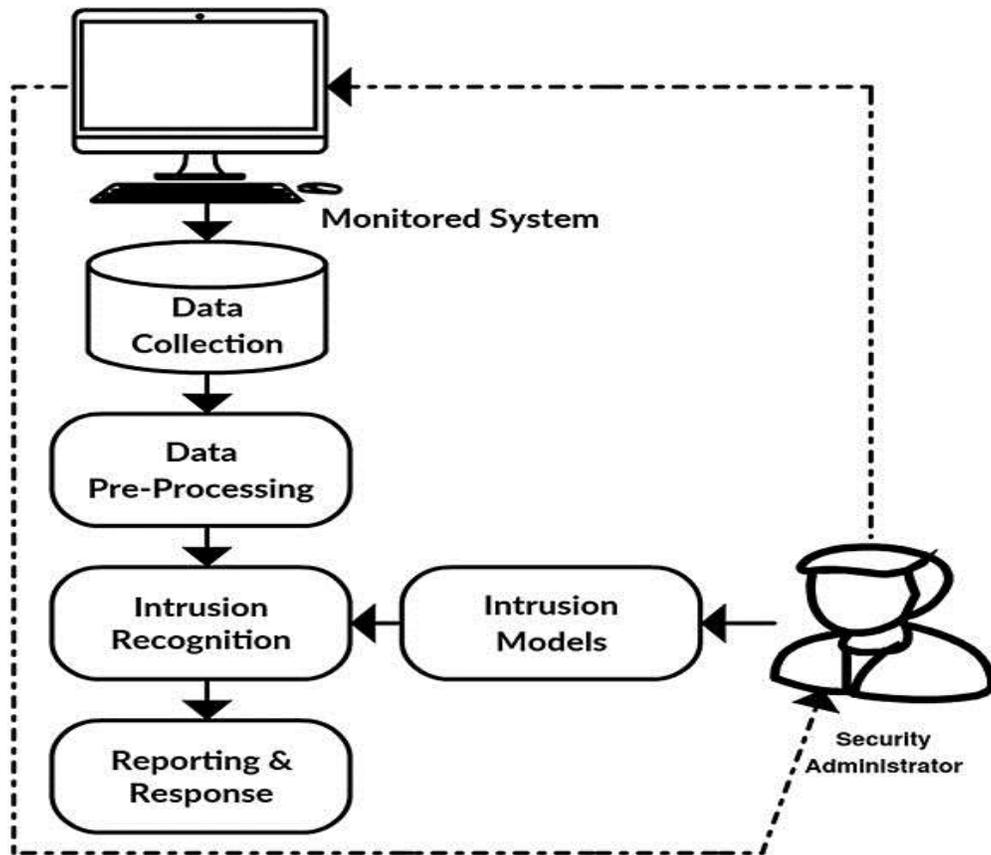
Using the set security policy rules, IDS conducts four major functions that include data collection, data pre-processing, intrusion detection, and reporting (Amir et al., 2013). The data collection component of the IDS collects the audit data from the traffic or system activities and send it to the pre-processing module (Kumar & Dutta, 2016). In the pre-processing component,

the collected audit data is converted into the most appropriate format before being released into the subsequent components (Amiri et al., 2013; Kumar et al., 2013). The converted data is then sent to the data detection/recognition module, the audit data is analyzed by comparing it against the detection rules to identify any suspicious or malicious activities or traffic. The intrusion component creates the profile of the identified benign activity subject to the rules (Kumar & Dutta, 2016). When the intrusion is recognized, the Response/reporting module of the IDS generates an alarm to alert the relevant authority of the intrusion (Dorri, Kamel, & kheyrkhah, 2015). This process takes place every time a new activity or traffic is initiated. As a result, Rafsanjani (2009) explains that the expansion of the network increases the system activities or network traffic hence weakening the intrusion detection system. IDS is more effective when the system or network is small.

The IDS architecture determines how the nodes in the MANET conducts the detection process, and how they respond to the identified suspicious activities or traffics in the system or network. According to Amiri et al. (2013), the available IDS architectures for Mobile Ad-hoc Networks are groups into three categories namely hierarchical, cooperative, and stand-alone. In the stand-alone architecture, each node conducts the IDS processes locally without collaboration (Kumar et al., 2013). This means that each node has its IDS. Besides, the nodes do not have an idea of what is going on in the other nodes (Rafsanjani, 2009). The alerts remain in that particular node without crossing the network. However, this architecture has its limitations. According to Amiri et al., (2013), the IDS under this architecture is not accurate for new attacks.

In the cooperative IDS architecture, each node has its IDS but the detection process is done in collaboration. This architecture is sometimes referred to as distributed architecture

(Rafsanjani, 2009). After recognizing an intrusion, the nodes share the established information or data, degree of risks, asset attacks, and the appropriate actions with each other (Mitrokotsa, Tsagkaris, & Douligeris, 2008). This idea means that the detection process and decisions are executed globally by all nodes in collaboration. The integration technique is utilized in analyzing the identified attack case or detected malicious traffic. Lastly, the hierarchical IDS architecture utilizes a multilayer technique that involves subdividing the whole network into subdivisions or clusters. In every cluster, a particular node is chosen and made the cluster-head. Cluster-heads perform different functions in the intrusion detection process. This architecture is more preferred because it facilitates the effective utilization of constrained resources (Amiri et al., 2013). However, the drawback of this architecture is the fact that the MANET's mobility makes it hard to create and utilize clusters and cluster-heads which are necessary for the detection process to take place (Rafsanjani, 2009). This type of IDS infrastructure is suitable for communicating during military operations.



(Source: Kumar & Dutta, 2016)

The IDS can detect attacks using several methods. For instance, detection can be achieved through signature-based detection. These patterns are studied and compared to previous events or attacks and then identifies new threats. Subsequently, the system administrators can be able to identify any threats on the network. An IDS is made of three basic components that include Network Intrusion Detection System (NIDS), Network Node Intrusion Detection System (NNIDS) and Host Intrusion Detection System. Each of these components plays a vital role in securing networks (Benkhelifa, Welsh and Hamouda, 2018).

The Network intrusion Detection System works by initially analyzing the traffic on the network. It then identifies possible threats with those attacks that are already registered on its library. The Host Intrusion Detection System on the other hand captures the image of the entire

system file set and then compares it with the previous picture. If there is a difference at all then it alerts the system's administrators who then comes and stops the possible attack. There is also Cloud Intrusion Detection system that is used for public environments.

There are two general types of Intrusion detection Systems; Host based IDSs and network based intrusion systems (NIDS). Each of those two systems has sensors that are aligned to the type of intrusion system. There are sensors on a network based IDS that monitor streams of traffic (Kenkre, Pai & Colaco, 2015). The network based IDS have various advantages and disadvantages. For instance, implementation of NIDS by corporations and governments translates to a lower cost of ownership. A relatively lower cost with NIDS arises from the fact that the traffic on the network is monitored as a whole (Gai, Qiu, Tao & Zhu, 2016). This means that the need for loading software on each host on the network is omitted. Additionally, deployment of NIDS is much easier primarily because the installation of that network does not affect the existing infrastructure. Detecting network based attack on this kind of network is easier. This is because the network based IDS have sensors that check all the packets and identify any threat that may exist on the network. Using a network based IDS is also advantageous because it has real time detection and quick responses to any kind of attack that might face the network.

Host based intrusion detection systems also have numerous advantages (Liu et al, 2018). According to Liu et al (2018), HIDS are capable of giving feedback on whether an attack has been a failure or a success. This is because the Host based intrusion detection system contains logs of all activities that have taken place. This kind of IDS can also monitor the activities that affect a given network where it is installed. Additionally, the host based IDS is capable of detecting the attacks that have been caused on the Network based IDS and gone unnoticed.

Network based IDS sensors cannot, for example, detect when an unauthorized user makes changes on a network. This turns to be quite important to administrators because they can be able to handle attacks way before they are actually launched (Jose et al, 2018).

The host based Intrusion detection sensors are installed inside the host servers or machines that play host to them. This means that there is no additional hardware that is required in order to install this kind of IDS. When a comparison is made between the host and network based sensors, the host based sensors are way cheaper. This means that the cost of entry to this kind of IDS is cheaper. From the sound of the advantages of these two kinds of IDS, we are prompt to assume that every institution out there needs either of these IDS. However each of these IDS comes with disadvantages that might limit its performance and it is important to know each of them (Marteau, 2019).

The IDS technology is advancing on a daily basis and therefore organizations that acquire either of them should ensure that their system is up to date so that it can be able to handle even the most recent kinds of threats that can be launched on a given network. Having an IDS system on a network is not the solution to preventing all kinds of attacks. The success of these healthcare systems in MANETS depend heavily on the way the IDS sensors are deployed on a given network. Therefore system administrators should ensure that the deployment procedure is done and achieved in the manner that they are supposed to take place. The IDS technology also is a reactive activity not a proactive. This simply means that the IDS technology heavily relies on previous attack patterns. The technology cannot work independently. However, The IDS technology is very important for any organization that seeks to secure itself right from the network level (Taher, Jisan and Rahman, 2019). A lot of information can be secured through the

process and this is what each organization seeks to achieve at the end of the day. It is important to put in place better identification and strong authentication processes.

2.6.1. Implementation of MANET Anomaly-Based Intrusion Detection Model for Health Care.

Designing, developing, and implementation of IDS for MANETs is faced by a variety of challenges. The network's characteristics make the direct implementation of IDS ineffective. As a result, different factors must be considered before IDS implementation in the MANETs. These factors are related to the functional features of the network and they include the following.

- i. Absence of central point: MANETs lack a centralized management area like the one in the wired network or switched wireless networks. As a result, the network lacks gateways and routers for facilitating centralized network security management. This idea means that the implementation of IDS in MANETs must consider distribution and node cooperation. On the other hand, the distribution is limited to factors like low energy and limited bandwidth.
- ii. Mobility: MANETs are dynamic in nature. This means that the network's topology is subject to frequent changes. The high mobility in the network makes it hard for the IDS to establish if the out-of-date information suggests that the network has been intruded or whether it simply means that the node is awaiting some update data. Kumar and Dutta (2016), therefore, suggest that the IDS implementation must consider a flexible architecture.
- iii. Wireless links: As opposed to wired networks, wireless networks are associated with limited bandwidths. As a result, an increase in the intrusion detection process would create congestion in the network hence hindering the normal traffic flow. As a result, Subba, Biswas, and Karmakar (2016) suggest that MANETs IDS needs to consider data flow

minimization to prevent network congestion. However, this move may mean compromising IDSs' performance.

- iv. Insecure communication links: Insecure links MANETs prone to different passive attacks like interference and eavesdropping. Therefore, IDS implementation in this network needs to ensure that encryption mechanisms are kept in place to ensure that attackers do not learn of the working mechanisms of the MANET IDS (Subba, Biswas, & Karmakar, 2016). However, this mechanism is hard to implement in MANETs because of energy limitations. Authentication and cryptography require a large amount of energy which is not available in MANET's nodes.
- v. Limited resources: MANETS are heterogeneous networks with mobile devices that exhibit different energy resources and computational capabilities. This idea means that IDS design is subjected to constraints like energy and memory (Rafsanjani, 2009). On the other hand, MANET IDS requires optimization to control energy consumption and network congestion.

While developing IDSs for MANETs, the above considerations must be considered. Besides, Kumar and Dutta (2019) suggest that IDSs implementation MANETs must meet other important requirements. First, the introduction of the IDSs should avoid, by all means, introducing new vulnerabilities in the network. Also, the IDS should achieve self-management to facilitate the automatic detection of any changes in the hardware and software environment of the MANET. This means that the IDS must be designed to achieve self-adaption when changes in configurations take place. Thirdly, the IDS should be economic in terms of resource utilization

when performing its functions. In summary, Kumar and Dutta (2019) suggest that the IDS in MANET must be optimized to overcome all the challenges from the network's architecture.

An anomaly-based intrusion detection system, monitors and alerts intrusions and misuse by observing activity that falls out of normal system operation. This is as opposed to signature-based systems, which can only detect attacks for which a signature has previously been created. Anomaly based intrusion detection has the capability to identify unknown intrusions as well as zero day assaults. The strength of this emerges from the capability of ABIDS to model standard operation disposition of a network and further identify deviations from the baseline. ABIDS can also be specifically configured to suit a particular network thus making it challenging for a previously successful attack in one network to be replicated in a unique setting (Gai, Qiu, Tao, & Zhu, 2016). Anomaly Based intrusion detection systems can implement different methodologies in either; an artificial intelligent knowledge-based detection, statistical anomaly detection, data-mining based detection or a machine learning based detection algorithm.

2.6.1.1. Statistical ABIDS Model (SABIDS)

A Statistical intrusion detection model is a common technique for identifying attacks and intrusions in a network. Statistical based anomaly detection techniques employ statistical values and statistical assessments to conclude whether the observed performance departs considerably from the expected norm (Zaidi et al., 2016). These statistical anomaly detection systems rely on the basis of a quasi-stationary activity, which is rare for most of the data processed by anomaly detection methods. Further the SABIDS learning process is long and takes before attaining accuracy and effectiveness. In addition, SABIDS have a conundrum in setting

the right alarm volume. A volume too high might not identify attacks, while a low alarm volume might result to numerous false notifications (Moustafa, Creech & Slay, 2017).

2.6.1.2.Operational Model or Adaptive Threshold Model

Operational Model also referred to as the Adaptive Threshold model is founded on the hypothesis that abnormal activity can be recognized by comparing a stream of activity against a predefined limit. On the basis of observed activity over a phase of time, an alarm can be raised. The methodology is applied particular statistics are commonly related to network intrusions. An Adaptive Threshold Algorithm can be combined in this case scenario as a child model. A child model is such that it derives various or all properties and characteristics from another parent model thus it becomes a submodel also referred to as a derived model, extended model, nested model or child model. The Adaptive threshold algorithm is a simple and straightforward methodology, which evaluates whether the amount of data over a given interval meets a set threshold (Saikumar et al., 2017).

It is vital to note that the difference between adaptive threshold model and this study, is that the study does not inherit properties or characteristics, rather, it marries two models to fuse and create a hybrid.

2.6.1.3.Markovian Process Model or Marker Model

The Markovian/Marker methodology is co-joined together with data values so as to conclude on the regularity of a specific occurrence, on the basis of prior events. This model symbolizes every captured data as an isolated case and exploits a state transition method to establish whether the observed occurrence is normal based on prior events. This model is principally advantageous

if the sequence of occurrences is predominantly significant. This model is based on two major procedures the Markov chains and the Markov models (Almusallam, Tari & Zomaya, 2017).

2.6.1.4. Statistical Moments or Mean and Standard Deviation Model

The Statistical Moments or Mean and Standard Deviation Model implements statistical prediction and evaluation from the norm on the basis of current values measured against a spread of possible scenarios. The statistical moments sub-model, evaluates and concludes that a particular occurrence, which goes beyond a set alarm value is anomalous. This method comprehends device instances without the prior knowledge of the devices traffic behaviors. This methodology is very flexible and has ability to determine anomalous activity without prior briefing or configurations. It is however a very complex model to implement and build (Kumar, & Venugopalan, 2017).

2.6.1.5. Multivariate Model

The Multivariate Model is used to monitor and detect intrusions based on two or more behavioral occurrences. The Multivariate Model thrives in occurrences of two or more behavioral occurrences that allow identification of possible irregularities in cases of complex conditions with multiple constraints. This Multivariate Model when augmented with statistical methods like chi-square produces improved results with low false alarm experienced as well as a high detection rate. In this methodology anomalous activity is identified fast. This model however is compute intensive with large volumes of statistical processes required so as to evaluate events accurately (Kolhe, Bhosale, Lathe, Mane & Bhattad, 2016).

2.6.1.6. Time Series Model

The TimeSeries Model identifies intrusions through a process of evaluating the sequence and time taken to perform various tasks in a networked system. An occurrence is marked as normal if the metrics are lower than the threshold, while an occurrence is marked as anomalous if the metrics are observed to be higher than the threshold. The Time Series methodology is flexible since it adapts and modifies itself on the basis of user actions. The alarm is set off by activities that exhibit substantial departure from the regular disposition (Wang et al., 2016).

2.6.1.7. Data Mining Based Approach

The Data Mining Based Approach is useful when used to identify external malicious traffic coming into the network. It however is a weak mechanism for identifying internal cases of attacks. This data mining method is crucial in excluding regular occurrences from raising the alarm thus enabling security admins to only dedicate time to managing actual network attacks. The data mining approach has the capability of identifying false alarms as well as irregular signatures thus enabling only the actual anomalous activity being identified and acted upon. This Data Mining based method uses two major techniques; clustering of traffic into groupings and identifying normal occurrences while facilitating discovery of attacks. (Sahasrabudde et al., 2017).

2.6.1.8. Association Rule Discovery

The Association Rule discovery is a common methodology while albeit slow uses correlation between various elements to identify anomalous activity. This method is used in a “market basket analysis” case scenario that identifies irregularities in the purchasing conduct of

supermarket clients. Also referred to as Boolean association rules, this technique tries to identify various arrays of items within the market that consumers regularly buy consequently in every purchase. Disadvantages of this methodology are that it exponentially proliferates the occurrences as the number of elements grows (Kong & Ryang, 2016).

2.6.1.9. Knowledge Based Detection Technique

Knowledge based detection methodology is a flexible technique which is applied in both anomaly based intrusion detection systems as well as signature based systems. This technique captures and stores known intrusions and network threats. These stored information is then employed to mitigate future intrusions as well as raise the threshold alarm. Occurrences that do not trigger the threshold alarm are treated as safe events (Kevric, Jukic & Subasi, 2017).

2.6.1.10. State Transition Analysis

The State transition analysis methodology is an open source technique that reviews possible intrusions through objectives and transitions. These state transition diagrams provide pictorial illustrations of events that an attacker can successfully perform in order to attack a network. The sequence of occurrences undertaken during an attack towards a network is recorded for every compromised state. State transition illustrations recognize the necessities for every intrusion as well as causes for the attack success. These illustrations enable the identification of key occurrences that enable an intrusion possible (Le Dang, Le & Le, 2016).

2.6.1.11. The Expert System

An experts system is a form of artificial intelligence, through which a computer system uses to imitate the decision-making capability of a human. This technique integrates a knowledge-

based intrusion-detection methodology. The expert system contains a set of rules that describe attacks. Audit events are then translated into facts carrying their semantic signification in the expert system, and the inference engine draws conclusions using these rules and facts. This method increases the abstraction level of the audit data by attaching semantic to it. Expert systems are integrated in both signature based and anomaly based intrusion detection systems as well (Folorunso, Ayo & Babalola, 2016).

2.6.1.12. Signature Analysis

The Signature analysis techniques employ the tactic of consolidating information in a manner similar to that adopted by expert systems. The signature approach however uses the information that is consolidated in a different way, by deciphering and breaking down data into a series of appraised events thereby decreasing the alarm threshold of the intrusion system. Efficiency is the utmost trait of this signature based analysis technique that has enabled its implantation in the market as a viable enterprise security system. The solution however has a major bottleneck in the requirement for regular updates in order to protect the network from newly discovered threats (Iqbal & Calix, 2016).

2.6.1.13. Machine Learning Based Detection Technique

Machine learning is the capability of an application or network system to acquire and advance its security capability by consolidating data and information and building a concrete algorithm as a result. Machine learning based detection is reliant on construction of a system that expands and progresses the ability to protect the network on the basis of improving on prior performance. This technique enables the machine-based system usable in wide and varying case scenarios. However, this machine learning intrusion detection system gobbles up system

resources thus can partake a huge amount of memory, bandwidth and CPU time (Buczak & Guven, 2016). Machine learning IDS use in either artificial intelligent fuzzy logic, artificial neural networks, bayesian techniques, genetic algorithms, support vector machines or Bayesian systems.

The Bayesian Approach technique uses illustrational diagrams to reveal possible interactions between various devices and resources in a network. This system can be used in cases where data flow is unpredictable in nature and intrusion case scenarios cannot be pre-determined (Kabir, Onik & Samad, 2017). While the Bayesian Approach is a relatively new technique in the field of intrusion detection, its grouping and combination with statistical methodologies have risen to efficient network security solutions. The only disadvantage with this technique, is that it is compute intensive in nature and can only work best when integrated with a Pseudo-Bayes technique so as to enhance the anomalous intrusion detection system's capability to identify new types of intrusions with a low threshold. The Bayesian Approach has the ability to spontaneously accommodate traffic streams with missing entries that would otherwise make it difficult to decipher an attack. This technique is superlative for consolidation of an existing intelligent formation and actual traffic flow (Mir, Khan, Butt & Zaman, 2016).

2.6.1.14. Neural Networks Technique

A neural network is an interconnection of various devices within a network, where the computational output of one device forms the input of another. Intrusion detection systems implementing the neural network approach have the capability to anticipate consequent user actions by any device or user in the network. This technique has the capability of conceptualizing an occurrence that results in anomaly detection. Neural networks in the construction of intrusion

detection systems reveal an efficient substitute to statistical methodologies. Neural networks are commonly found in anomaly based intrusion detection systems (Roy et al., 2017).

2.6.1.15. Fuzzy Logic Approach

Fuzzy logic methodologies have the capability to handle significantly huge amounts of traffic and domain constraints especially in a scenario with cases of data approximation. Fuzzy logic can be amalgamated with data mining so as to diminish amounts of the input data as well as to choose occurrences which expose traffic anomalies (Atre & Singh, 2016). The fuzzy logic technique is extremely effective for purposes of identifying innovative and new network attacks. A fuzzy logic methodology studies and evaluates the frequency of activities, CPU activity and device session connection durations. Fuzzy systems have the ability to dynamically consolidate data inputs fed from variable devices in the network. Fuzzy logic techniques enable quick construction of “if-then” criteria that can mimic and recognize unauthorized intrusions (Mkuzangwe & Nelwamondo, 2017).

2.6.1.16. Genetic Algorithms Technique

Genetic algorithms methodology has been used in computational biology by naturally selecting and evaluating the evolution of domains. The Genetic Algorithms initiates an indiscriminate generation of an enormous number of potential applications. The accuracy and effective results generated by each domain enables a classification and ranking. The better ranked domain applications edge out moribund and programs deemed less effective. Sequentially high performing applications outlast those with a low accuracy, ultimately only the strong survive (Bhattacharjee, Fujail & Begum, 2017).

2.6.2. Evaluation, Validation and Verification of Related works in

There are several implementations that have been evaluated on various levels and using different metrics to ascertain their ability and robustness in provision of security.

2.6.2.1.Support Vector Machine I

Eskin and Honig implemented a support vector machine that operate by mapping input data streams against a greater domain value and consequently calculate the ideal alarm threshold. This is achieved by using an assessment test that is determined by consolidating route-path vectors instead of evaluating the total number of traffic streams received (Honig, Howard, Eskin & Stolfo, 2016).

2.6.2.2.Wisdom and Sense

Wisdom and Sense is an anomaly detection system built and operated by Safeguards and Security Group at Los Alamos National Laboratory. This was a conglomerate with Oak Ridge National Laboratory. Wisdom and Sense is an intrusion detection solution that identifies statistical occurrences of anomalies in user activity. This solution creates a set of rules that statistically define the performance users based on prior recordings of user actions. Then ongoing network activity is compared to the set of rules in order to identify any presence of inconsistencies (Singh & Singh, 2016).

2.6.2.3.Computer Watch

AT and T's ABDIS – the Computer Watch is an intrusion detection system solution which was built using open source UNIX/MLS multilevel security operating system. This system evaluates user action and compares it against a set of rules that define appropriate network data policy. It further exposes user action that goes against the norm. This technique while useful in

policy based profiles, is less effective in comparison to statistical techniques especially when processing large amounts of traffic (Beigh & Peer, 2011).

2.6.2.4.Haystack

Haystack is an anomaly based intrusion detection system that detects intrusions in multiuser the American Air Force systems. Haystack has the ability to rummage through enormous amounts of traffic data, and further reduce these streams to diminutive digests of user action, anomalous occurrences as well as instances where there is a breach of security especially from within its own network (Somwanshi & Joshi, 2016).

2.6.2.5.Petri Nets - IDMANET

IDMANET is a knowledge-based intrusion detection system that was created and run by Purdue University, using Colored Petri Nets (CPN). The major advantages of using Colored Petri Nets is that they have a general ability to review all types of data, are very simple to apply and run and can represent data streams in the form of illustrations. Users can also configure instances and lines signatures of known network intrusions and further integrate them onto IDMANET. In addition complex signatures can be easily written in Colored Petri Nets. Unfortunately, complex signatures are compute intensive (Sethuraman, 2017).

2.6.2.6.Fuzzy Intrusion Recognition Engine (FIRE)

Dickerson et al. implemented a Fuzzy Intrusion Recognition Engine (FIRE) through the application of fuzzy sets and fuzzy rules. The Fuzzy Intrusion Recognition Engine (FIRE) is an anomaly based intrusion detection solution that applies fuzzy logic to evaluate and identify intrusions and malicious traffic streaming into a wireless network eco-system. FIRE implements

a simple data mining methodology so as to propagate network input data and identify data that is leaning towards anomalous activity (Elhag et al., 2019).

2.6.2.7. Genetically Programmed ABIDS

Crosbie and Spafford (2017) to identify traffic anomalies in a network implemented a genetically programmed algorithm for sparse trees to identify traffic anomalies in a network. The objective was to decrease the incidences of false alarms by incorporating human input within a feedback iteration. This study demonstrates an advanced design technique of providing security within a network established on genetic algorithm optimization. This solution was capable of guaranteeing efficient performance of the network while significantly reducing costs associated with running it. This methodology can be also used in producing safety systems for fire signaling (Bakour, Daş and Ünver, 2017).

2.6.3. Hybrid and Fused approaches for ABIDS in MANETS

This study also reviews other research in the area of fusion of various machine learning techniques. This section introduces the various sampled methods of fused machine learning for intrusion detection.

There are various methods that employ a hybrid model in MANET including a detection system for flooding attacks known as FADA in MANETs which employed machine-learning mechanism. This technique employs route discovery historical data by the devices on the MANET to identify devices that match normal and those that have anomalous similarities. This research introduced a routing protocol that prevents flood attacks christened FAPRP Flood Attack Prevention Protocol. This new technique improves on the existing AODV protocol by adding and fusing a new technique as described above. This mechanism has the ability to provide

intrusion detection with 99% accuracy in Network Simulator 2. It also provides improved results as opposed previous protocols used in detection and prevention of flood attacks (Luong, Vo and Hoang, 2019).

Another hybrid model employed artificial neural networks techniques to classify data in MANETs. This provided an express technique for identifying data, highlighting, blacklisting, compromised device healing and intrusion detection. This technique through artificial neural networks classified devices and data by comparing packets propagated by each. This mechanism further isolated and healed compromised devices successfully. This mechanism had an curate ability to provide protection at above 88% with much easier and less cumbersome tasking process.

This research provides an avenue for future exploration into faster and easier threat identification, process and anomalous activity isolation and ability to be backward compatible with existing intrusion detection mechanisms (Sowah et al., 2019). Mishra and Naik introduced a technique based on a combination of fuzzy clustering and particle swarm optimization to provide a hybrid machine learning technique for intrusion detection from captured data within a MANET. The study entailed taking fuzzy data sets to categorize anomalous activities separately from normal traffic flows. This mechanism christened “Particle swarm optimization” has the ability to accurately perform data classification from various data flows. In addition, the resultant algorithm does not consume device resources in comparison with competing mechanisms (Mishra and Naik, 2019).

Another hybrid mechanism introduced by Ye (2019) involved a key feature recognition technique that evaluated anomalous activity from data propagated within a MANET. This

technique involved fusing artificial neural networks and a support vector machine. An artificial neural network technique for identifying and labeling patterns of intrusions is fed into a Support Vector Machine, which further evaluates this data to reduce the possibility of false positives. This combination ensures great accuracy in identification and reduces the positive alarm as a consequence. Ye (2019) was able to provide this mechanism after propagation of data for 3.18 ms (Ye, 2019).

Another fusion of various machine learning technique is a hybrid of feature selection, anomaly detection and data classification. In this technique, a unique algorithm is introduced – the intelligent flawless feature selection algorithm (IFLFS). This technique involves the selection of data patterns that most likely represent anomalous activity. This unique contribution of this method was its ability to select the finest combination of data flows that best describes an unauthorized intrusion. Thereafter, the identified data set would be fed into an outlier detection segment to further separate and optimize the classifications. This technique enjoyed a more accurate intrusion detection with a few false positives. (Kovaras and Rajkumar, 2019).

An innovative biologically based intrusion detection technique involved a combination of feature selection and nature based mechanism so as to ease anomalous activity identification as well as optimize on search algorithms. This technique employs evolutionary search methods as ant search and bee search, genetic search and particle swarm search techniques (Komninos and Procopiou, 2019). Another hybrid technique involved combination of network operations so as to achieve data flow measurements propagated by devices in the MANET network. The resulting technique reviews times spent by each node and performs comparisons with set limits. If the data sets exceed the limits an alarm is raised. (Hoffmann and Holland, 2019). Ciocarlie and

others fused various techniques to identify anomalous network activity by abstracting and modeling scenarios that best depict an intrusion (Ciocarlie et al., 2019)

Elwahsh et al (2018) introduced a neutrosophic mechanism which used an arithmetic algorithm for complex data flows within a MANET. This neutrosophic method employs symbolic artificial intelligence to determine whether data flows resemble an attack. Devices in a MANET are categorized into members and nonmembers and data flows from or to the devices are symbolically rather than numerically compared from what is regarded as member norm. This study introduces an artificially intelligent engine that uses a combination of fused approaches to determining classification of data Elwahsh et al (2018). This neutrosophic variables and training data in input into the intelligence engine and sample attacks initiated therein. This neutrosophic mechanism is great for importance in identifying unknown attacks in MANETs.

A research by Vimala, Khanaa and Nalini (2018) introduced a novel method, which provided a unique mobile agent based intrusion detection system. The network first provided classification of various network attacks through the TSVID Classification method. This algorithm uses RBF various iterative learning methods. Consequently, data is classified through NNIDS, which employ neural networks to achieve this goal. This particular mechanism has the ability to consume qualitative and quantitative datasets. Thereafter, by employing an iterative learning method, the resultant classified data in provided for comparison purposes. Graphs of the classified data is compared with normal and anomalous datasets thereby reducing the false alarms and greatly providing real time results with a humble consumption of network resources as well as working with great accuracy and efficiency (Vimala, Khanaa and Nalini, 2018).

Another method was introduced to protect against jellyfish attacks by combining support vector machines (SVM) and data from authenticated routing packets. A machine learning mechanism that studied packet propagation was employed. This mechanism selected trusted devices by having a hierarchical tree with the most trustworthy device atop of the triangle. Various intrusion detection algorithms were employed to study packet delays, packet drops and data throughput. The results provided the express ability to detect and raise alarms for Jellyfish attacks within MANETs (Doss et al., 2018)

Another hybrid machine-learning mechanism introduced detection in two segments - local and global detection. Network data propagated by devices was harnessed and classified at the local segment. This local data was studied and a comparison was run between network and device disposition. Based on the data collected and comparison thereof, normal and anomalous behavior was detected. The local segments involved dedicated sniffers (DS) which employed supervised learning mechanisms using decision trees in order to provide accurate behavior instances. The global segment inherited these results and further created a time-based profile indicating which devices were regarded as normal and anomalous. These results were tested thrice to ensure accuracy and a low level of false positives (Amouri, Alaparthi and Morgera, 2018). The table 1 below shows various differences in the implementation of machine learning techniques.

2.7.Types of Intrusion Detection Systems in MANETs

For many years, scholars and researchers have been working had to perfect the application of IDSs in Mobile Ad-hoc Networks by working on the limitations of these networks. As a result, different studies have been published to explain the different types of intrusion

detection systems that can be used to secure MANETs. Therefore, this section looks at some of the suggested types of IDS that have been proposed and suggested as appropriate in the Mobile Ad-hoc Networks. These IDPs are classified according to the mode of operation and the type of algorithm used in detecting and reporting harmful and suspicious activities in the system or traffic in the network.

2.7.1. Classification Based on Mode of Operations

Subba, Biswas, and Karmakar (2016) explain what types of IDSs in MANETs can be classified according to their mode of operations. In this classification, the IDSs are categorized into anomaly-based intrusion detection systems, Signature-based intrusion detection systems, and specification-based intrusion detection systems. The operations of the two types determine how detection and reporting of malicious traffics in MANETs are conducted.

2.7.1.1. Anomaly Based Intrusion Detection Systems for MANETs

This type of IDS uses a predefined behavior of the network to test and establish the existence of abnormal traffics. The IDS learns the normal behavior of the network and tests the existing traffic against the learned behavior. If the traffic goes against the predetermined normal behavior, an alarm is raised and the IDS alerts the network administrator of the suspicious traffic (Subba, Biswas, & Karmakar, 2016). As a result, this IDSs carries out detection in two phases; training and testing. During the training phase, the network's normal profile is developed. The profile is developed by learning about all the network specifications and rules that have been set by the administrators. These are considered acceptable network behavior. According to Jyothsna, Prasad, and Prasad (2011), the IDS engine plays a critical role in the training phase by cutting through the different protocols at the different network levels. The engine needs to process and

understand the goal of every protocol in the network. While the protocol analysis process consumes a lot of resources, it is vital in eliminating the number of false-positive alerts during the testing and detection processes.

During the testing phase, the IDS analyzes the existing traffics using learned profiles of the network's normal behavior (Jyothisna, Prasad, & Prasad, 2011). The learned model is, therefore, used as the basis for examining and determining the signs of traffic misbehavior in the network. Upon detection, the IDS creates an alert immediately. The IDS uses different techniques to test the traffic against the acceptable standards. Some of the techniques include machine learning, data mining, and statistical methods (Subba, Biswas, & Karmakar, 2016). This type of MANET IDS is usually preferred because of its ability to establish and detect the malicious activities were previously unknown and had not been encountered during the training phase. This advantage is crucial because it helps the IDS to detect new worms in the network (Jyothisna, Prasad, & Prasad, 2011). The major demerit of this type of IDS is the existence of a huge number of false-positive alarms. Also, the IDS encounters a problem during the definition of the ruleset. The effectiveness of this IDS depends on protocol implementation and testing (Subba, Biswas, & Karmakar, 2016). If the process is not conducted effectively, some abnormal behavior may pass unnoticed.

2.7.1.2. Signature-Based Intrusion Detection System for MANETs

This MANET intrusion detection systems utilize a database of predetermined attack signatures to detect malicious traffics in the network. The traffic data is compared against the database, and an alert is created if traffic is found to match the recorded attack signatures (Subba, Biswas, & Karmakar, 2016). The effectiveness of this type of IDS is its high rate of detection for the known attack signatures. However, the IDS does not detect new attack signatures that have

not been recorded in the database (Yeo, Che, & Lakkaraju, 2017). This acts as one of the limitations of this type of IDSs.

As opposed to the implementation of the protocol for testing which consumes a lot of time and resources, the creation of a database of attack signatures is easier. The creation depends on the available knowledge of the attack signatures (Jyothsna, Prasad, & Prasad, 2011). The IDS also records all the known buffer overflows and other vulnerabilities thus making the process easier. The signature-based IDS not only informs of the detected malicious traffic but also states the specific cause of the alert. As stated, the main limitation of this IDSs is the ability to detect only the attack signatures that had been previously stored in the database. As a result, it is hard for the system to detect new attacks. Besides, the fact that the technique uses string matching and regular expressions to detect malicious traffics makes it vulnerable to deception (Subba, Biswas, & Karmakar, 2016). Nonetheless, this type of IDSs does not function properly when the device is based on advanced technologies like payload encoders and nop generators (Jyothsna, Prasad, & Prasad, 2011). These technologies decrease the efficiency of the signature-based IDSs hence making them ineffective.

2.7.1.3. Specification-Based Intrusion Detection System for MANETs

This type of intrusion detection system works by specifying a set of constraints on the protocols or network traffic, and any traffic that violates these constraints is considered an intrusion (Subba, Biswas, & Karmakar, 2016). This type of IDS is preferred because it is capable of detecting both known and new attacks with the lowest rates of false-positive (Berthier, & Sanders, 2011). However, the main limitation of this type of IDSs is the fact that it requires much

time and effort to come up with a detailed specification that defines a secure network behavior. As a result, the process is computationally expensive and consumes a lot of time.

Specification-based IDS is considered to combine the advantages of anomaly detection and those of misuse. The specifications that define the behavior of a health network are created manually. These specifications are used as the basis for detecting any attacks or suspicious network traffics (Subba, Biswas, & Karmakar, 2016). Traffic that shows any sign of specification deviation raises an alarm and the system creates an alert immediately. The fact that the IDS detects intrusions from deviations from the normal network behavior makes it appropriate for detecting even the unknown attacks in the network (Berthier, & Sanders, 2011). This idea also makes it reduce the number of false-positive alarms.

2.7.2. Classification Based on Detection Algorithms Used in the IDSs

Apart from the mode of operations, the types of IDSs in MANETs can also be classified based on algorithms used in executing the detection process. IDSs use different algorithms to analyze the system activities or network traffic and detect attacks. In MANETs, the IDSs used can also be classified according to the algorithms they use to evaluate the network traffic and detect suspicious or harmful traffic. Amiri et al. (2013) give different types of algorithms that IDSs in MANETs utilize. Some of these IDSs are as explained below.

2.7.2.1. Danger Theory-Based Algorithm IDS for MANETs

This type of intrusion detection system uses the danger theory-based algorithm in the analysis of network traffic and detection of attacks. The specific algorithm used by the IDSs in this category is referred to as the dendritic cell algorithm (DCA), one of the danger theory-based algorithms (Abdelhaq, Alsaqour, & Abdelhaq, 2015). The IDS that utilize this theory is used in

MANETs to track and recognize sleep deprivation attack. According to Amiri et al. (2013), the working of this IDS is dependent on the idea that every node in the mobile ad hoc network needs to protect itself from any form of danger without relying on the assistance of mobile agents. The IDS's dendritic cell algorithm works over another type of algorithm termed as a mobile dendritic cell algorithm, MDCA (Abdelhaq, Alsaqour, & Abdelhaq, 2015). As a result, the architecture for this category of MANET IDS is also termed as MDCA.

The two main components of MDCA are adaptive and innate subsystems. Amiri et al. (2013) explain that the working of this algorithm begins by verifying the packet ID of every packet that has been entered in the memory. The packet ID is then compared to the packets from previous attackers. If it is found in the memory, the algorithm automatically knows that the packet is originating from the attacker. The next step taken by MDCA is rejecting the packet before erasing its data from the network's routing table. After this step, the algorithm sends a repeat alert message (Amiri et al., 2013). On the other hand, the availability of the packet ID in the list of alarmed packets is considered by the algorithm to be an attack that originates for a different node. This detection forces the algorithm to reject the packet immediately. Otherwise, the packet needs to be analyzed by the packet analyzer which removes the needed antigens from the network's routing table. The limitations of this algorithm include limitations in the network's bandwidth and high rate of power consumption.

2.7.2.2. Classification Algorithm Intrusion Detection Systems in MANETs

This type of IDSs for MANETs was designed and evaluated by Mitrokotsa, Tsagkaris, and Douligeris (2008). The IDS works by utilizing supervised classification algorithms. This IDS utilizes hierarchical architecture. This architecture is made up of many IDS agents that perform

the detection function together. The IDS agents utilize Linear Model, multilayer perception (MLP), SVM, Naive Bayes, Gaussian Mixture, models for performing classification. The creation of these models requires data for label training (Amiri et al., 2013). This architecture uses the advantage of multiple local intrusion detection systems agents that analyze the MANETs traffic and detect malicious traffic locally. Together, these IDSs makes up the MANET's detection system.

Each intrusion detection system in this multilayered architecture is composed of various components. The data collector is responsible for collecting audit data from the network traffic and activity logs. From the collector, the audit data is sent to the intrusion detection engine. This engine is responsible for analyzing the local audit data for intrusion detection. In the local IDS, the classification algorithm helps in the intrusion detection process. Finally, the response engine gives the intrusion alert in case of any intrusion detection (Mitrokotsa, Tsagkaris, & Douligeris, 2008). This type of IDSs for MANETs uses the classification neural networks to analyze and detect different attacks in the mobile ad hoc networks. The detected attacks include forging packets, black hole, flooding, and dropping attacks (Mitrokotsa, Tsagkaris, & Douligeris, 2008). It is crucial to note that the intrusion detection system agents work together but the detection is performed locally.

2.7.2.3. Zone-Based IDS for MANET's

Another crucial type of IDSs for mobile ad hoc networks is the Zone-Based intrusion detection system. This classification is included in this category because of the type of algorithms used for the intrusion detection process. According to Amiri et al. (2013), the IDSs in this category uses the zone-based algorithms to analyze the traffic data and detect malicious

intrusions in MANETs. The design was explained by Sun, Wu, and Pooch, (2006), to explain how IDSs can be used to eliminate the disruption and routing attacks from mobile ad-hoc networks. The IDS design for this category uses cooperative architecture (Amiri et al., 2013).

The network that utilizes this type of IDS is subdivided into non-overlapping zones. Concerning the nodes, they are divided into two categories. The node that physically connects with a node in another zone is termed as the gateway zone. The other type of node called the intra-zone node. The gateway nodes collect all the locally produced alerts from the intra-zone nodes. After the collection, the gateway node conducts correlation and aggregation tasks to eliminate the false positive alerts produced by them locally. The Zone-Based IDSs are preferred in MANETs because of their mode of operations. The different modules that perform the detection process are Local Aggregation and Correlation Engine-LACE, Global Aggregation and Correlation Engine-GACE, and the Intrusion Response Module (IRM) (Sun, Wu, & Pooch, 2006). The LACE aggregates and correlates the intrusion detection results of the detection engines locally. On the other hand, the GACE aggregates and correlates the detection outcomes of the nodes in the local zone (Amiri et al., 2013). The IRM's responsibility is to take the appropriate corresponding measures following the detection of an attack. Some of the immediate steps taken include identification of the intruders, exclusion of the compromised nodes from the MANETs, and reinitiating the disrupted channels of communication in the network. This process marks the complete process of the intrusion detection process.

2.7.2.4.BeeID Intrusion Detection Systems for MANET

This type of intrusion detection system uses a hybrid approach to the idea of the negative selection (NS) and the artificial bee colony or simply (ABC) algorithms to conduct the

instruction recognition process. The two algorithms are together referred to as the BeeID algorithms, hence the name BeeID intrusion detection system (Amiri et al., 2013). These types of IDSs are used in the AODV-based mobile ad-hoc networks. The detection process in this IDSs happens in three phases namely training, recognition, and updating.

During the training phase, a niching type of the BeeID algorithm (NicheNABS) executes the NS algorithm severally to formulate a collection of negative detectors to cover the available space (Singh & Bedi, 2015). These mature negative detectors are used in the detection phase to differentiate between malicious and normal network traffics (Amiri et al., 2013). Finally, in the updating phase, the detectors are updated using total or partial update mechanisms. A complete process involving the three steps ensures that the AODV-based MANETs are protected. This type of IDSs uses stand-alone architecture during its design. Through the BeeID method, the IDSs detect different types of attacks in mobile ad-hoc networks (Singh & Bedi, 2015). The examples of attacks detected using this type of IDSs include a black hole, worm hole, flooding, rushing, and neighbor attacks.

2.8. Machine Learning

Machine learning is one of the major advancements in modern technology that have facilitated the development of intelligent systems capable of solving problems that could traditionally be assigned to humans only. Particularly, machine learning is basically a section of the larger field of artificial intelligence. Before delving deeper into machine learning and its applicability in MANETS, it is necessary to initially develop a coherent understanding of what the term actually means. Consulting a variety of sources reveals definitions that slightly differ but ultimately reveal a common theme.

2.8.1. Concept of machine learning

Malhotra (2015) and El Naqa and Murphy (2015) define Machine Learning (ML) as the study of computer algorithms and statistical & computational models that are used by computer systems to execute specific tasks without necessarily relying on explicit instructions, programs, set patterns, or interventions from humans. While Malhotra (2015) and El Naqa and Murphy (2015) consider ML to be ‘a study,’ Libbrecht and Noble (2015) have a slightly different definition. Explicitly, Libbrecht and Noble (2015) describes machine learning as a set of artificial intelligence algorithms designed to solve particular logical and computational problems. ML offers an opportunity for machines to gain new knowledge without being unequivocally programmed, and based on the concept of learning from data.

According to Qiu et al (2016), Machine learning is a field of research which primarily focuses on the properties, performance, and theory of learning algorithms and systems. In the context of the study by Marblestone, Wayne and Kording (2016), learning algorithms or systems are modestly computer models that are capable of using a limited amount of information fed into them to develop inferences for solving future problems. While the concept of ML originated from computational logics, it is currently regarded as a highly interdisciplinary field borrowing ideas from different fields such as engineering, finance, biology, cognitive science, and statistics (Libbrecht and Noble, 2015). Furthermore, machine learning has application in different areas such as physical security, information security, manufacturing (e.g. industrial robots), control engineering, transportation (e.g. level 5 autonomous vehicles), and landscaping. Considering its application in a wide variety of areas, the concept of machine learning has significantly transformed modern science and society.

2.8.2. Brief History of ML

Over the years, as the main subset of AI, ML has consistently used different learning algorithms to allow computers to learn and improve themselves without being programmed or instructed. As such, it is evidence that ML has a permanent history that constitutes its gradual evolution. AI is increasingly becoming ‘more intelligent’, and ML is crunching businesses and industries. In fact, some futurists predict the possibility of continued advancements in machine learning (and the general field of artificial intelligence) resulting in adverse consequences. For instance, the concept of ‘technological singularity’ has become a buzzword in the 21st century, implying the advancement of machine intelligence beyond untamable levels. However, at least in the foreseeable future, machine learning should play a great role in improving the overall quality of life for the human race. The section, therefore, offers a brief overview or summary of the critical stages in the history of ML. The section is divided into several stages.

2.8.3. Game of Checker Players

In 1952, Samuel’s Checkers Player became the first widely recognized learning system that obtained public approval. The process pioneered decisive ideas regarding the evolution of machine learning (Samuel, 1988). The Checkers Player was, for the first time, designed to remember positions that were frequently encountered in the course of the play (Samuel, 1988). Therefore, the simple form of *Rote learning* allowed the systems to save time and search more in-depth in the subsequent games, whenever a stored position was encountered on the board or in some line of calculations (Samuel, 1988). The next component that the Checker Player featured was Reinforcement Learning that was used to examine weights for its evaluation function and realize a win (Samuel, 1988). Checker Players could train itself against its stable copy (Samuel, 1988).

The development of computer programs for playing games by IBM was limited in terms of memory. Therefore, Samuel invested alpha-beta pruning, which entailed a scoring function using a specified position on the pieces of aboard (Samuel, 1988). The scoring function in the Checkers Player tried to evaluate the probability of winning for each side (Samuel, 1988). The program was autonomous and could automatically select its next winning strategy through the use of a minimax approach, which evolved to be a minimax algorithm (Rajasekar & Subramani, 2016). To enhance its performance, Samuel designed a different mechanism that allowed the program to become better and called it Rote learning (Samuel, 1988). The program could remember and record positions seen and combined them with values of the expected reward function (Samuel, 1988). Therefore, it was such advancements in technology that Arthur Samuel, IBM scientist coined the term “Machine Learning” in 1952 (Samuel, 1988). Instead of manually programming the 500 quintillion in 2 potential scenarios from the checkboard into a computer, Samuel instructed the system to react based on the previously played games, and then weigh provided factors, calculate the risk, and plan the next strategy (Samuel, 1988).

2.8.4. The Perceptron

Frank Rosenblat developed the Perceptron in 1957, depicting the second stage in the evolution of ML (Kapoor, Wiebe & Svore, 2016). The Perceptron refers to an algorithm that was developed to provide classified potential outcomes in computing (Tang, Deng & Huang, 2015). It presented another significant advancement of how ML algorithms would gather and generate data (Kapoor, Wiebe & Svore, 2016). Designed in the US lab research center, Perceptron was a simple model imitating biological neurons in an artificial neural network in supervised learning for binary classifiers (Tang et al., 2015). Perceptron was meant to classify visual inputs, categorize subjects into one of the two types and conduct image processing (Nishani & Biba,

2016). The perceptron, therefore, advanced the evolution of ML by allowing patterns and group classification through a linear separation by arbitrary assigning them numeric visual inputs (Kapoor, Wiebe & Svore, 2016). Compared to a single checker player, perceptron was able to classify linearly inseparable data, thereby solving problems that a single-player algorithm could not solve (Tang et al., 2015).

2.8.5. The Nearest Neighbor Algorithm

The development represented the third stage in ML evolution. In 1967, the Nearest Neighbor Algorithm was initially conceptualized and it signaled a paradigm shift to the development of pattern recognition algorithms (Rajasekar & Subramani, 2016). The Nearest Neighbor Algorithm was used to map routes and became the first technology to be used to offer a solution to the traveling salesperson problem by deciding the most efficient path (Chen & Hao, 2017). The stage later paved a way to the invention of Multilayers in 1967 and allowed extensive research in neural networks (Chen & Hao, 2017). The use of multiple layers led to the introduction of feedforward neural networks and Backpropagation in the 1970s and is currently used to train deep neural networks (Chen & Hao, 2017). In 1990, Boosting algorithms were invented to reduce bias during supervised learning (Nishani & Biba, 2016).

2.8.6. Deep learning

The stage marks the most advanced phase in the evolution of ML. Deep learning is the 21st-century invention that refers to ML algorithms utilizing multiple layers to provide the highest levels of precision/accuracy from raw data input (Zhang et al., 2017; Marblestone, 2016). The previous innovations in ML have led to the creation of Speech Recognition and Facial Recognition systems (Zhang, Tan, Han & Zhu, 2017). For instance, deep learning explains how in image processing, the lower layers seem to identify edges, whereas the higher

layers identify concepts that are relevant to humans, such as digits, faces, and even letters (Zhang et al., 2017).

As one of the advanced stages of ML, deep learning uses brain-inspired architectures to train computers on how to learn by examples (Nishani & Biba, 2016). The technology is currently dominating the autonomous cars, by allowing them to recognize stop signage or make a difference between pedestrian and lampposts (Marblestone., 2016; Zhang et al., 2017). Also, deep learning is used in voice control for consumer products such as phones and hands-free speakers (Zhang et al., 2017). Deep learning can obtain a high state-of- art-accuracy that can exceed the human-level of performance since it is based on artificial neural networks (Nishani & Biba, 2016).

2.8.7. General Applications of Machine Learning

Corporations across the spectrum have recognized the worth of ML technology in businesses. Most of the companies that use ML handle large volumes of data that need to be interpreted for effective decision making. Through the process, companies can realize enhanced performance, efficiency, reduced costs, and heightened security of the collected data. The discussed below are classic cases where ML is applied.

2.8.7.1.Financial Services

The financial industry is one of the critical sectors that are highly influenced by ML algorithms. ML gives banking institutions protection against online or cyber fraud, which can jeopardize service delivery to clients (Mathur, 2019). For instance, companies such as PayPal profoundly depend on ML to detect money laundering practices (Mathur, 2019). In effect, the company can reduce losses and costs that result from common Federal fines (Siau & Yang,

2017). The use of ML in financial services help companies to quickly identify individuals or critical financial institutions that are at high risk for fraud and other sets of financial risks (Mathur, 2019). Secondly, ML utilizes a high level of deep learning that can analyze various sets of customer information and offer insights that can be used to identify growth opportunities or possible areas that reduce performance in the banking industry (Mathur, 2019). ML helps banking institutions to predict the future and develop necessary strategies to counter market contingencies (Mathur, 2019).

2.8.7.2. Marketing and Sales

Companies have employed ML in myriad spheres to strengthen and fortify their market share. Evidenced by vicissitudes in consumer tastes and preferences, companies maintain must keep trends to offer customers services that best satisfy them (Siau & Yang, 2017). In this regard, organizations have quadrupled the use of ML to study and analyze consumer behavior (Swathi & Seshadri, 2017). Some of the vital information gathered for analysis include purchase history, shopping price, purchase time, volume purchased and other sets of data that can be used to understand customer buying behavior (Siau & Yang, 2017). Prediction in marketing and sales allows producers to offer consumers with the right products and services and heighten the profitability of the companies (Rajasekar & Subramani, 2016). ML predicts future consumer behavior based on the past shopping experiences which can be used to generate necessary product promotion and marketing programs to target specific customers with unique goods (Siau & Yang, 2017).

2.8.7.3. Medical Services

The use of ML in medicine has had a significant impact on diagnosis and delivery to patients and has promoted health. With the introduction of wearable sensors and devices which can be used to assess patients' data in real-time, ML has become a fast-growing development in medicine (Rajkomar, Dean & Kohane, 2019). Currently, practitioners can use different ML technologies to obtain real-time data about patients' conditions such as blood pressure, heart rate, and many others. The information gained can be used by doctors to draw patterns and history of the sickness and develop the best drugs to handle them (Rajkomar et al., 2019). ML empowers doctors with enhanced analysis that can be used to improve diagnosis and treatment. Through ML, medical diagnosis has been revamped for different conditions (Rajkomar et al., 2019). Besides, it has become easy for practitioners to detect and monitor the progression of diseases, therapy planning and organize patient management programs (Rajasekar & Subramani, 2016).

2.8.7.4. Transport sector

ML has widely been used in the transport sector (Rusitschka & Curry, 2016). For instance, based on data gathered on specific routes, transport companies can predict probable problems on the roads and suggests ways to navigate them (Sebopelo et al., 2019). Transport companies have doubled the use of ML to conduct data analysis and help customers make informed decisions on how to plan a journey (Rusitschka & Curry, 2016).

2.8.7.5. Speech Recognition and Voice recognition

ML has led to the development of speech recognition (SR) software. SR denotes the translation of spoken words into a text. The term is also referred to as “automatic speech recognition” (ASR), “computer speech recognition,” or “speech to text” (STT) (Chorowski, Bahdanau,

Serdyuk, Cho & Bengio, 2015). The technology can recognize spoken words and converts them into written words. Voice recognition software listens to voice input by a person and does a search that matches it before providing the most appropriate answer or response. Image Recognition/face recognition becomes another critical component that uses ML (Chorowski et al., 2017). The software can recognize a face and sends its information or notification to related individuals.

2.8.7.6.Videos Surveillance

ML helps companies and individuals to detect a crime and react before it occurs. The precision is aided by constant monitoring of all activities within a covered region (Verma, Singh & Dixit, 2019). ML uses past experiences to detect unusual or suspecting behavior and alerts the user to take necessary precautions to mitigate the severity of the crime or abort it (Verma et al., 2019). Modern ML is well designed to automatically send signals and alerts to guards to initiate essential safety precautions (Verma et al., 2019).

2.8.8. Machine Learning Models and their Application in MANETS

2.8.8.1.Introduction

This section reviews the various machine learning models as well as existing empirical studies that aim at addressing prevailing security issues in Mobile Ad hoc Networks (MANETS). There are three fundamental classes of machine learning models as originally classified by Qui et al (2015); unsupervised machine learning, supervised machine learning, and reinforcement machine learning.

2.8.8.2. Supervised Learning

Based on the conceptual definition provided by Barami and Gerami (2013), supervised learning is any machine learning process or algorithm in which inputs are mapped to desired outputs based on a function inferred from a training dataset. The training dataset consists of data in form of input-output pairs. The learning algorithm performs analysis of the input-output pairs in the training dataset and develops an inferred function, which it then uses to map a set of inputs to unknown outputs. As indicated in the literature survey performed by Schrider & Kern (2018), most researchers adopt supervised machine learning not only for its simplicity but also the ability to how exactly the algorithm is learning. In particular, supervised machine learning algorithms provide an inferred function which the user can test and verify its authenticity. However, there are also particular demerits of supervised machine learning that are also mentioned in literature such as high potentiality for bias, function complexity, noise in the output variables, and lack of clear guidelines on the amount of training data required for different specific applications. Bias is the probability that the training data will not contain all the necessary input-output pairs to produce a function that is actually representative of the real scenario. Other than bias, the inferred function from a training dataset may be too simplistic hence subject to high bias and low variance, which are undesirable aspects of any machine learning algorithms (Lafferty & Wasserman, 2006). On the contrary, if the function is too complex, it may be subjected to low bias but extremely high variance, which are also undesirable aspects of any effective machine-learning algorithm. In particular, while low bias and high variance may be desirable aspects, too complex functions normally result into overfitting in which the training model captures noise in the training dataset and includes it in the inferred function (Lafferty & Wasserman, 2006). Function complexity is determined by the amount of training data that should be fed into the

training algorithm. Determining an optimal amount of data that neither produces bias nor overfitting is one of the most critical issues that developers of supervised learning models grapple with.

There are several types of supervised machine learning algorithms that have been used extensively in existing academia. As presented in a scientific paper by Qui et al (2015), the different types of supervised machine learning include Support Vector Machines (SVM), Bayesian Networks, and Hidden Markov Models. In the subsections that follow, a review of literature on these specific ML models will be conducted with a special focus on ML in MANETS.

2.8.8.2.1. Support Vector Machines

Generally, Shawe-Taylor & Cristianini (2000) defines a support vector machine (SVM) as a discriminatory SL classification algorithm that places datasets into groupings by creating separation hyperplanes. The SVM model is given a training dataset that indicates input-output pairs such that each output value is a category in which a certain input data value is placed. The SVM algorithm then assigns new data values (inputs) to their respective classes based on the classification criteria (or inferred function) developed during the training phase. Shao et al (2014) uses the concept of separating hyperplanes to illustrate how a SVM categorizes data represented as points in space. The separation margin must be as wide as possible in order for the classification to be considered effective.

While various SVM algorithms have been developed for application in Mobile Ad hoc networks, a significant portion of such experimental studies focus on development of intrusion detection systems. Lakshmi & Valluvan (2015) developed an SVM based intrusion detection system (IDS)

in order to address particular challenges in existing anomaly detection systems. In particular, the researchers pointed out the lack of a standard training technique in the anomaly detection systems as well as the degradation of network performance due to the isolation of critical routing nodes as the main challenges. In developing their solution, Lakshmi & Valluvan (2015) initially classified nodes into cluster heads (CH) and cluster members (CM). Cluster heads represented the nodes that had the highest stability index in the sample MANET, while cluster members were the nodes that had average to minimum stability index. In a typical intrusion, the affected nodes show a change in behavior (Lakshmi & Valluvan, 2015). As such, the researchers used SVM algorithms to differentiate between misbehaving nodes and well-behaving nodes. Finally, the researchers employed fuzzy logic to isolate misbehaving nodes (which typically represent nodes affected by the intrusion) hence improve network performance.

Whereas the study by (Lakshmi & Valluvan, 2015) focuses on general Intrusion detection, other studies have focused on specific types of intrusions. Shams & Rizaner (2018) developed an SVM-based intrusion detection model for explicitly for the detection of Denial of Service Attacks in MANETS. The model takes in three types of data from the MANET; the amount of data sent by each node, the amount received by each node, and the average amount of time taken for a packet to arrive. The researchers collected training and testing data from the NS-2 Network Simulator under the AODV routing protocol. The data was collected in two scenarios; when the network is under attack, and when the network is functioning normally. According to Shams & Rizaner (2018), in a MANET that functions normally, the nodes play three significant roles; receiving, routing, and sending data packets. However, in a Denial of Service (DoS) attack scenario, the attack nodes normally send huge amounts of data packets, which are received by some of the attacked nodes. In the SVM DoS detection model by Shams & Rizaner (2018), the

main detection parameter is the amount and size of data packets sent by the attack nodes. The SVM model faithfully detects and isolates the attack nodes. Whereas the SVM model by Shams & Rizaner (2018) proves useful in detecting and preventing Denial of Service attacks in MANETS, it is purely based on simulation data. The training and testing data sets were not collected from a real world network but rather, from the NS- 2 network simulation tool. Therefore, it might be quite impossible to establish whether similar accuracy levels can be achieved when the SVM is applied to a real world network.

There are other studies which have attempted to develop MANET intrusion detection systems using SVM techniques such as Devi et al (2016) and Oberle et al (2013). While most of these studies are conducted on real MANETS as opposed to simulations like in the study by Shams & Rizaner (2018), they do not offer high accuracy levels and reduced computation complexities. For instance, a study by Devi et al (2016) developed an SVM based intrusion detection algorithm for MANETS. However, the algorithm filters every data packets sent and received by every node thus increasing computational complexity. In essence, filtering all data packets for both sent and received data introduces redundancy, which is highly undesirable in machine learning as it increases computational complexity.

A study by Barani & Gerami (2013) enhances on the model developed by Devi et al (2016) by including more features for the detection criteria. In particular, the model includes detects network anomalies based on four types of network data; Constant Bit Rate data, route discovery data, route disruption data, and route protocol data. The four features significantly limit the potentiality of the SVM model producing false positives or negatives. In the SVM model developed by Devi et al (2016), the researchers only use data on Constant Bit Rate to isolate

attack nodes from normal nodes. However, in as much as the model by Barani & Gerami (2013) improves the accuracy of detection, it might probably result in greater computational complexity due to the multiple detection criteria included. Nevertheless, security in MANETS is much more important that it cannot be compromised for a slight increase in computational complexity.

2.8.8.2.2. Bayesian Networks

A Bayesian network is a graphical representation of a set of variables and how they relate to each other through conditional probabilities. The variables and their conditional dependencies and probabilities are normally represented on a Directed Acyclic Graph (DAG) (Jonas et al, 2013). The concept of Bayesian networks can be traced to conditional probability theory in which a prior probability is used to compute a posterior probability. In the case of symptoms being used to predict diseases, a prior probability could be looked at as the symptoms that are fed into the networks (Jonas et al, 2013). The Bayesian network then computes the most probable disease(s) based on some causal relationship between symptoms and diseases. Bayesian networks are mainly applied in predicting the possibility that a certain factor (or set of factors) contributed to a particular outcome.

Numerous previous researchers have utilized Bayesian networks to develop Intrusion Detection Algorithms in Mobile Ad hoc Networks. In a study by Wei et al (2014), an IDS was developed based on Bayesian learning that could predict the presence of malicious attackers on the MANET based on causal reasoning. The system typically uses prior probabilities, which are the causes and indicators of an attack (e.g. unreliable network connections) in order to determine whether the network is under attack. In the presence of an attack, unreliable connections must have been caused by packet dropping or modification by the malicious attacker. Elizabeth et al (2011) also

developed a trust model for application in MANETS using Bayesian learning. Essentially, the Bayesian network calculates the trust level of a node using its history of interaction with other nodes in the network. Using this model, it is quite easier to single out malicious nodes based on how they interact with other nodes. Unfortunately, the researchers did not provide the specific parameters that are used in computing the degree of trust for every node. Nevertheless, the main predictors of trust probably include the behavior of a node such as package dropping, modification, and sending out of too many requests like in the case of a DoS attack.

Other several researchers such as Wang & Zeng (2010), Rezaul Karim et al (2006), Li & Wu (2008), Nguyen et al (2007), and Serrat-Olmos et al (2012), among others, have developed models for improving security in MANETS using Bayesian networks. One of the main advantages of Bayesian networks that makes them ideal for applicability securing MANETS is the ability to model causal relationships between variables using conditional probabilities. In MANETS, an attack could be modelled as a prior occurrence that causes a number of posterior occurrences such as dropping of packages, modification of packages, extremely high latency, or an extremely large number of packets being sent from a single node in the case of a flooding attack. However, setting the prior variables in a Bayesian network is quite challenging and if wrongly done, could result in wrong outcomes (Jonas et al, 2013).

2.8.8.2.3. Hidden Markov Models

A hidden Markov Model (HMM) is fundamentally a Markov Process as it satisfies the Markov Property (Eddy, 2004). The Markov Property indicates that the future state in a sequence of events must only be predicted using the current state with as high an accuracy just like the history of the sequence were known. However, the HMM differs from the typical Markov

process in the sense that it (HMM) has its states concealed behind the observations made. It could also be regarded as a simplistic form of Bayesian network in which some unknown variable or set of variables can be predicted from observed variables (Eddy, 2004). However, the main distinction between a Bayesian Network and HMM is that the latter, a causal relationship not necessarily need to exist between the observed variables and the hidden states. A simpler example of a HMM could be the prediction of weather (which is considered unknown to a foreigner but known to the local) based on the kind of clothes worn by the largest number of people (the clothes worn is an observed state that the foreigner knows but does not know about the whether). If the largest number of people put on light and bright clothes, it is almost unlikely that the weather is cold and wet.

The application of HMM in MANETS is quite common among researchers. In a study by Pathak et al (2017), an HMM is used to predict the reputation of every node in a network and hence its probability of being an attacker node. However, this approach is quite complex as the HMM has to calculate the reputation value for every node. This weakness is overcome by the researcher conducted by Ye et al (2010) in which cluster heads instead are used to provide observational sequences that are then used to predict the network state.

Ye et al (2010) developed a HMM for detecting the presence of an attack in MANETS. Ye et al (2010) initially subdivided the network into equal clusters with each clusters containing one cluster head and several cluster members. The cluster head monitors the activities of the other nodes. Therefore, the cluster head was the main target for extracting data on an observed sequence. The observed sequence can be used to predict two possible states; normal network or abnormal network. Initially, the model is trained using observed sequences for a normal network

and also for an abnormal network. Once sufficient levels of accuracy have been achieved, the model is deployed to a simulated network to determine its ability to make accurate predictions of the network state. However, the model by Ye et al (2010) has two main weaknesses. First, the model may not be useful in predicting normal changes in a network. For instance, an increase in the number of nodes could be detected as the presence of an Intrusion. Second, the model does not provide specific details about the type of the Intrusion. These two weaknesses make it quite unreliable.

A similar pattern of weaknesses can be observed in several other machine learning models such as Pathak et al (2017) and Hamza & Vigila (2019). However, the study by Sivanesan & Thangavel (2015) takes quite a different path as far as application of HMM in MANETS is concerned. The researchers develop a HMM algorithm capable of predicting available free bandwidth for efficient resource allocation. The model works perfectly when the network is subdivided into clusters with each cluster contain a cluster head and several cluster members like in the case of Ye et al (2010). The free available bandwidth is normally predicted based on the present traffic load on each node in each cluster. There is a possibility that the model developed by Sivanesan & Thangavel (2015) could be used to overcome one of the challenges in the model by Ye et al (2010) – the challenge associated with introducing normal changes to the network such as addition of new users. In this case, the HMM algorithm should predict the future bandwidth available as well as the addition of a new user (when the predicted bandwidth lowers by a smaller factor). In attack situations, the bandwidth must lower by a significant factor, particularly flooding attacks where there may not even be free bandwidth.

In another study aimed at enhancing the accuracy of anomaly detection in MANETS, Mirzaghali & Faez (2016) developed an HMM algorithm that combines observations from several physical sensors to determine whether the system is under an attack or not. The biosensors on themselves detect whether there is an anomaly. In order to increase the level of detection accuracy, several groups of sensors are deployed such that each group detects anomaly based on a specific parameter such as significant load differences, average jitter, or signal propagation delay.

2.8.8.3. Unsupervised Learning

Unsupervised Learning (UL) is a type of machine learning in which an algorithm analyzes and looks for patterns in a particular dataset (Denny & Spirling, 2018). The data used in this process has no labels like in supervised learning algorithms. Unlike supervised learning, unsupervised learning does not require the use of input-output data pairs to develop an inferred function. Instead, the algorithms in unsupervised learning simply discover the relationship among data values (Ghahramani, 2003). Therefore, unsupervised learning eliminates the need for users to initially define the input-output pairs for developing an inferred function. An example of an unsupervised learning model may be a system that develops a number of market segments from raw unclassified and unlabeled data. There is no need for model training and testing since in unsupervised learning (Ghahramani, 2003). While there are several unsupervised machine learning models, the most common include artificial neural networks and clustering.

2.8.8.3.1. Clustering

Clustering is the placement of data values into categories/groups based on a criteria that the machine learning algorithm discovers. The criteria makes use of relationships existing among the data values. In order to determine whether particular data values belong to the same category or

not, the algorithm evaluates the strength of the relationship existing between them. If the relationship is stronger than a particular threshold parameter, then the data values are placed in the same category. However, if the relationship is weaker than a given threshold parameter, the data values are placed in different groups. There are many applications of clustering such as image & pattern recognition, and data compression. Types of clustering common in existing literature and data science practice include k-means clustering, hierarchical clustering, and mixed clustering models (a mixture of hierarchical and k-means).

Other than the previously mentioned general applications of clustering, this unsupervised ML approach has been widely applied in MANETS.

2.8.8.3.2. Artificial Neural Networks

Mitra et al (2013) developed an Artificial Neural Network model (ANN) for the detection of Denial of Service attacks in mobile ad hoc networks. The system was quite effective as it is dynamic and allows the detection and updating of anomalies to prevent routing disruption. However, the models by (Ghahramani, 2003) are quite simplistic as they utilized only a single criterion to detect an anomaly. As for Mitra et al (2013), the only criterion used was throughput, whereas for (Ghahramani, 2003), packet delivery ratio was the only variable fed into the ANN. In another ANN system developed by Kaur & Kaur (2014) for detecting anomalies or intrusions in MANETS, several variables were used as input in order to enhance the accuracy of the model; packet delivery ratio, throughput, average jitter, and end-to-end delay. Throughput is the total size of data packets sent and received over the MANET. Closely related to throughput is packet delivery ratio, which refers to the ratio between sent packets and packets delivered successfully. Under normal conditions, packet delivery ratio should almost be constant and throughput should

be almost 100% (Kaur & Kaur, 2014). However, under attack situations, both packet delivery ratio and throughput significantly lower. This is because, attack nodes, particular in intrusions and black hole attacks, establish false short-cut routes in the network, which disrupt nodes from accurately sending data packets to their desired destinations (Kaur & Kaur, 2014). In some situations, as indicated by Raju & Setty (2015), an intrusion attack node deliberately drops the packets sent to it. In such instances, the packets are simply lost and not delivered to their final destinations. As such, detection of changes in amount of packets sent vs amount received should be indicative of an intrusion attack. However, this criteria alone, as indicated in the ANN model by (Kaur & Kaur, 2014), cannot be used to authoritatively confirm an attack. Changes in propagation delay (end-to-end delay) and average jitter or period of the sent and received signals should augment the accuracy of any ANN model. One of the major advantages of Artificial Neural Networks for applicability in MANETS is that they can still provide a reliable output (on whether the network is under attack or not) even with incomplete information (Kaur & Kaur, 2014). For instance, if some of the nodes in the network become compromised, the rest may comfortably compute the outcome with reliable accuracy. However, a greatly compromised ANN may produce biased results especially if there are several compromised neurons that determine a particular outcome.

2.8.9. Research Gap in General ABIDS Techniques for MANET

Table 1 compares the various techniques used in machine learning and their respective weaknesses, which disadvantage them from being used effectively for intrusion detection in MANETS.

Table 1. **Research Gap in General ABIDS Techniques for MANET** (Source: Author)

No	ABIDS Technique	Use in MANET	Weakness
1	RSSI Value	Discovers Masquerade	Huge Number of False Positives
2	TDMA	Tracks Schedule	Large Database of schedules
3	SMAC	Tracks Sleep/Wake Schedule	Large Database of schedules
4	ROUTING	Provides Authentication	Numerous ADT Updates
5	RTT	Discovers Masquerade	Huge Number of False Positives
6	Triangulation	Discovers Masquerade	False Alarms

On the other hand, Table 2 shows a comparison of the two machine learning techniques i.e support vector machines and artificial neural networks used in MANETs and their respective weaknesses, which make them deficient to be used as standalone for intrusion detection in MANETs. This explains the need to fuse the two to enhance intrusion detection capability for MANETs.

Table 2: **Research Gap in Machine Learning based Techniques for MANET**(Source: Author)

No	Machine Learning Technique	Use in MANET	Weakness Identified
1	Neural Networks	Yes	Slow Learning Process. Not suitable in Real time scenarios
2	Support Vector Machine	Yes	Slow Learning Process. Unable to deduce more information about an attack.

Support Vector Machines have the advantage of working extremely well with large sets of data, even when the data is unstructured and semi structured data like text, images and trees. It is powerful in classification with robust algorithms for producing classes from huge unclassified data. Artificial Neural Networks are well suited to pattern recognition problems and thus a fusion of the two techniques is proposed noting that SVMs have shown good results when developed in the reverse order so as to propel and support the development of artificial neural networks.

2.8.10. Conceptual Framework

An anomaly-based intrusion detection model that fuses SVM and ANN is thus proposed to address the gap between bottlenecks in the two machine learning techniques. This is achieved by combining a variable matrix of the two machine learning techniques. A fusion of artificial neural networks and support vector machine data classifier was implemented. This enables proper monitoring and profiling of traffic emanating from the WSN. The neural network is also supported through reinforcement learning in order to maximize the cumulative result. The network is then trained by introducing internet packet traces. The technical model will have the following components;

- a) A Data mapping separator
 - i) A Support Vector Machine.
- b) Anomaly Detection Engine
 - ii) An Artificial Neural Network
- c) Alarm/Reporting Arm.

The Network collects all incoming/outgoing data transitioning through the interfaces. Packets are separated depending on interest and mapped accordingly to a higher dimensional feature space. This is fed into a Support Vector Machine that transforms a linearly non-separable problem into a linearly separable one. This is due to its strengths in data classification. Further, the classified data is fed into an artificial neural network that performs pattern recognition tasks. The ANN makes use of modified probabilistic radial basis function. Data packets with anomalous symbols are thereafter passed into the anomaly detection engine. If the data is positive, an alarm is raised and particular anomaly is reported.

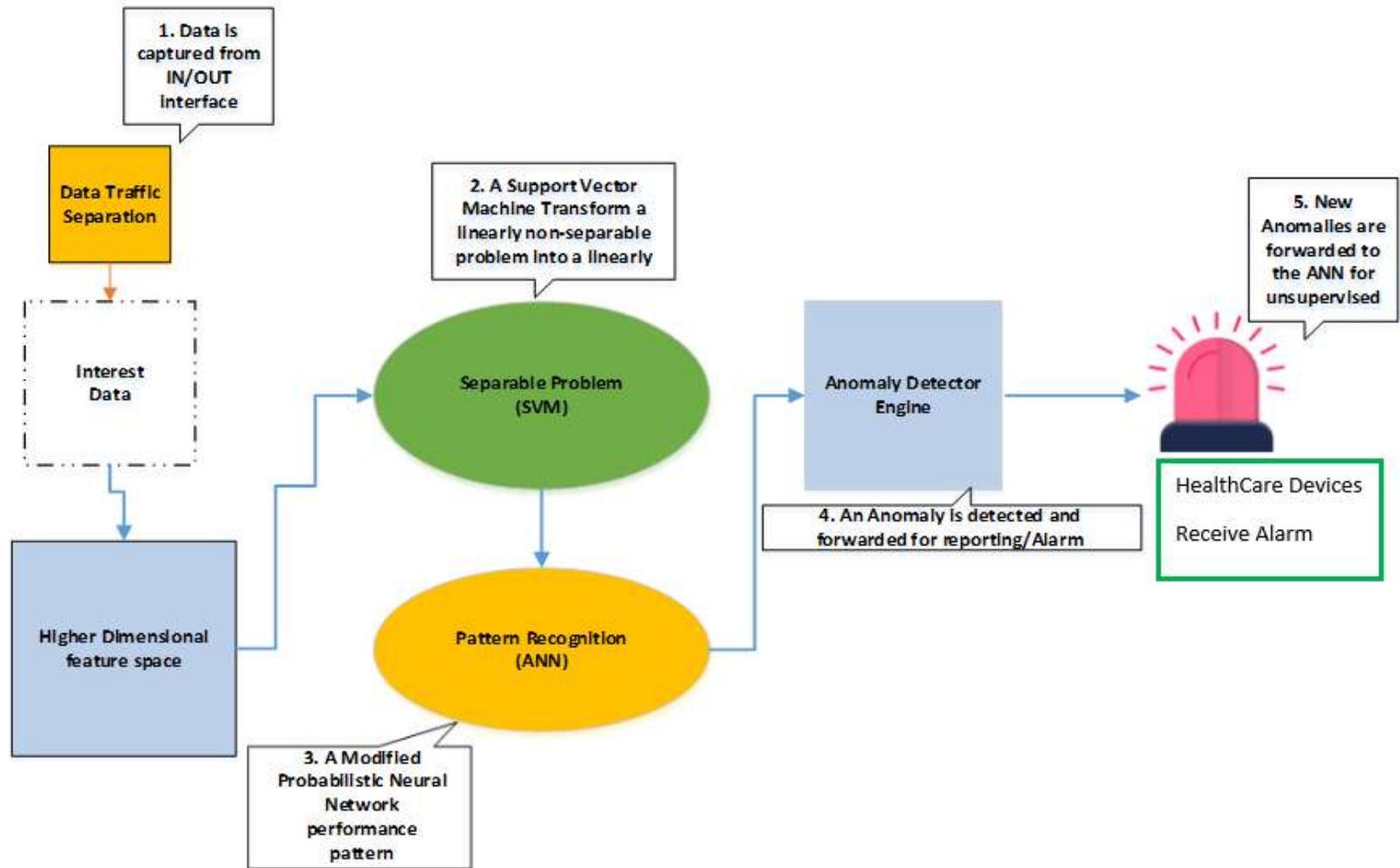


Figure 3: Conceptual Framework (Source: Author)

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This chapter will discuss the methodologies that were used in the collection of primary and secondary data for review, design, development, implementation and testing of the fused machine learning intrusion detection model for the provision of smart health care in MANETS.

3.1.1 General Literature Review Methodology

The purpose of this section was to identify and review secondary data relevant to objective one of this study. A general literature review methodology was used to answer research question one. A literature review was therefore carried out to examine and identify the weaknesses in the MANETs system that hinder their application in the provision of smart health care. This entailed review of relevant and current literature as well as secondary data relevant to the topic of application of machine learning in intrusion detection systems for MANETs. The objective of this literature review to identify, if any, the various weaknesses that hinder adoption of MANETs in healthcare. Thereafter, these would guide in the design and implementation of a model that would circumnavigate these security weaknesses in MANETs that have inhibited rapid adoption in smart devices for provision of healthcare.

3.1.2 Research Philosophy

This research was undertaken using the Ontology Philosophy. This philosophy vibrated with this research since it was concerned with what actually existance of smart healthcare and

security bottlenecks that affect the adoption of smart health care. The ontology philosophy was critical in answering the following questions;

- i) Is there existence of security bottlenecks that hinder the absorption of smart health care?
- ii) What type and categories do these vulnerabilities belong to?
- iii) Are there security measures that can be taken to protect and secure data, systems used in smart health care?

Ontology philosophy guided and enabled this research to investigate through various methodologies, the existence of both problems and solutions towards achieving objectives of this study elucidated earlier.

3.1.3 Proof of Concept Methodology

The purpose of this section was to provide a guidance towards an approach of designing and implementing a model for a fused machine learning intrusion detection model for the provision of smart health care in MANETS. The Proof of concept (PoC) therefore provided the best methodology to achieve this objective. PoC is the actualization of a particular technique or idea so as to demonstrate its feasibility or implementability (Carvalho, 2019). PoC is a demonstration in principle which aims to verify that a proposed concept or model has a practical potential, which can be implemented successfully. In the field of engineering and technology, a prototype of a novel idea can be constructed as a proof of concept (Sanghera, 2019). This was the approach adopted in this research. The proof of concept involved design, implementation and testing of the fused anomaly based intrusion detection system for MANETS in smart health,

3.1.4 Design of a MANET ABID Model for Smart Healthcare.

3.1.5 PPDIOO methodology Research Design for Model Design

This study adopted a mixed methods approach research methodology to design, develop, implement and test the model for a fused anomalous intrusion detection system. The two major methodologies used were the Prepare, Plan, Design, Implement, Operate, and Optimize (PPDIOO) and the Proof of Concept Methodology.

The PPDIOO methodology is essential for wireless sensor networks that are inherently complex and difficult to manage. This methodology defines a continuous top-down life-cycle of services that supports dynamically evolving networks. It follows six phases, each that describes the continuous life-cycle of MANET network services that are essential for the interconnection and propagation of a network (Cisco, 2011). The model experientially simulated a computational immunology status of an anomalous intrusion identification, isolation and detection. This methodology is idyllic since it lowers cost of ownership, increases the MANET network availability and improves business agility. Ultimately, the methodology accelerates MANET devices access and integration with application layer services. The PPDIOO methodology follows the following phases;

3.1.6 The Preparation Stage

MANET Network requirements which are critical to the proposed ABIDS were identified and enumerated in this phase. An optimal intrusion detection strategy was proposed for the smart health network. The devices, specifications, medium and wireless environment are consolidated, harnessed and aligned into the strategy for purposes of proposing a high-level conceptual

framework. This phase also evaluated formulating a cost-effective strategy that meets the needs of, and makes a business case for the proposed intrusion detection model.

3.1.7 The Planning Stage

This stage involved identifying preliminary smart health MANET network requirements based on the objectives, services and user requirements. The planning stage identified and illustrated the ABIDS location, evaluated current protection systems and identified the loopholes within the existing system and appraised the ability to integrate the proposed ABIDS into the existing smart health network.

The following requirements were identified;

- Basic Hardware Requirements: PC/Laptop, Raspberry Boards, Smart Watch.
- Basic Software Requirements: NS-2, Python, Linux.
- Specifications: MANET on IEEE 802.15.4 specification.
- Rules and Functional Modules, namely;
- Attribute Formation Module: Observed forms of the target MANET smart health ecosystem which are portrayed in a pre-established format.
- Observation Stage Module: A model which describes behavioral characteristics of the system.
- Espial Stage Module: This module was involved matching experiential traffic to the designed model.

3.1.8 The Design Stage

This section elucidates the methodology used to achieve objective two of the research, which was to design the various logical and physical tenets of the network. This phase enumerated tasks and responsibilities of the MANET network design specialists. The design for the MANET eco-system incorporated and enhanced confidentiality, availability, and integrity at the core of its design. As a result reliability, scalability and performance were as well enhanced. The design phase prepared the platform for the implementation of proposed MANET ABIDS in smart health. The ABIDS incorporated a fusion of computational immunology by integrating machine learning for neural networking and use of support vector machines.

3.1.8.1 Functional decomposition methodology

This methodology was utilized to design the logical topology. The methodology entails separating a complex entity so as to describe the separate contributing components. It enables understanding and organization of complex entities which are used to help solve problems and thus help to develop business operations, computer systems and machine learning among various other use cases (Harris and Davies, 2019). The methodology was deemed best since the research comprised of fusion of two discrete individual processes and techniques. This enabled a flowing description of the functional relationship between the various constituent components and integration of separable logical processes herein compressed into a representation of the global IDS.

3.1.8.2 Finite State Automaton

This design phase included the incorporation of finite state automata to describe threat isolation and identification. Each automaton illustrated a state transition diagram of events that an attacker can successfully perform so as to attack a network. The FSM was very crucial in producing an abstract representation of the model.

3.1.8.3 Machine Learning Design

Machine learning techniques were merged and integrated to enable the ABIDS absorb the best attribute and strengths of each, and to continuously improve on past performance.

3.1.8.4 Neural Networks Methodology Design

Neural MANET networking will take computational output of one wireless sensor device as input to another. Data consolidated as a result will then train the MANET neural network to observe and recognize the various activities.

3.1.8.5 Support Vector Machine Methodology

The SVM transforms a linearly non-separable problem into a linearly separable one. This is due to its strengths in data classification. Further, the classified data is fed into an artificial neural network that performs pattern recognition tasks methodology will collect and evaluate the frequency of activities within the MANET network, CPU activity and MANET device session connection durations. As a result generate an “if-then” criterion for every functional module.

3.1.8.6 The Implementation Stage

This phase includes actual building of the smart health MANET network. It is during this phase that built or additional components are incorporated in accordance to the design specifications. The ABIDS incorporated will have the essentials of machine learning, fuzzy logic and neural networks therein.

3.1.8.7 The Operation Stage

This stage tested the appropriateness of the resultant MANET network designed. The operational phase will involve monitoring of intrusion detection, correction, and performance monitoring of the ABIDS.

3.1.8.8 The Optimize Stage

This stage involved the proactive improvement, recording, tuning and troubleshooting of the MANET smart healthcare network. If performance of the ABID Model does not meet the security expectations, then the strategy was redesigned and improvements made so as to meet technical goals (Cisco, 2011).

3.1.9 Implementation of the Fused Machine Learning Intrusion Detection Model.

3.1.10 Simulation on NS 2 on Linux

This section presents the methodology used to implement the model on NS 2 on Linux. The purpose of implementing this study through a linux simulation was to as to provide a quick, cost effective virtual experimentation of the fused machine learning intrusion detection model. Linux is a free and open source operating system that lowers the total cost of achieving this

objective. Linux is also a network operating system thus supports network abilities. This virtual implementation of the real experiment enabled predictions about imminent future behaviors of the IDS possible quickly and cost-effectively. This enabled assumptions and approximations on how the real experiment would perform accurately by drawing inferences concerning the operation characteristics of the real IDS system.

The implementation of the ABIDS Model on a MANET environment was performed as follows:

- The MANET network was simulated in NS 2 on Linux operating system.
- Descriptive script for the fused intrusion detection system was thereby introduced within the MANET network.
- The simulation was analyzed and simulated to determine if the MANET network while vulnerable to attacks like replay, relay, and man-in-the-middle attacks is actually experiencing the same.
- The Simulation was limited to a period of 5 hours since the MANET network can propagate data to infinity if no attack is discovered; or at least until the computer runs out of memory.

3.1.11 Implementation Smart Healthcare Network on Raspberry Board Microcontroller

For purposes of implementing a live experiment of a MANET network, a test bed was set up to fashion a health monitoring device i.e – a smartwatch with capability to collect blood pressure data. However, since the smartwatches are proprietary, there was need to reconfigure certain aspects of the MANET thus Raspberry Pi microcontrollers were introduced into the

network. They were critical in the ability to upload enhanced software, thus to be able to propagate, this research created a test bed for the testing and propagation of patient body pulse levels. A Raspberry Pi Board was set up as a MANET network. The microcontrollers were then connected to smart watch which would propagate data to and from the board.

A Raspberry Board microcontroller were used in collection of patient health data that was propagated over the MANET. The Raspberry Pi is a low cost, high processing and with easy integration to a pulse sensor. This microprocessor uses an Atmel AVR processor. The Raspberry Pi is embedded with standard programming language compiler and firmware which will execute the software. The Raspberry microcontroller has 14 digital input/output pins which can accommodate analogue input. The following items are required for this prototype;

- Raspberry Pi 3 Model B+
- Smart Watches
- MANET Radio Modules
- Character Display Module
- 5Volts Voltage Regulator
- 3 Micro Switches
- Prototype Circuit Board
- SD Memory card

Below is the Raspberry Pi Model B+ specification for this prototype;

- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Dimensions: 82mm x 56mm x 19.5mm, 50g

3.1.12 Evaluation of the Smart Health Care MANET Anomaly-Based Intrusion Detection Model.

3.1.13 Evaluation of the Finite State Automata

The evaluation of finite state automata, enabled a study of the logic used to describe normal and anomalous activity within the MANET. Evaluation of finite state automata involved evaluating acceptable and unacceptable inputs and unreachable transition states. If there existed a combination of inputs that are not accepted by the various states in the automaton, or various inputs enabled a deadlocked state to be exited, this would affirm the validity of the automaton.

The evaluation also checked on the ability of the automaton to accommodate diverse asynchronous inputs. It followed the schedule below;

- Listed all functional modules
- Design the Logic
- Draw the Logic Table and Diagram
- Test the Automaton

3.1.14 Evaluation of the Network Simulator 2 Environment

Evaluation of the intrusion detection model machine learning system was performed on variety of functions that were used to evaluate the system. The implemented ABIDS model was evaluated by propagating data and testing the performance of the ABID model on the following functionalities;

- Monitoring Component - Ability to monitor sensor nodes and capture activities, traffic and resource utilization.
- Analysis Component - Ability to segregate normal and abnormal activities within the network.
- Detection Component – This is the main task of the ABID Model. This is the ability to declare a malicious activity correctly without having false alarms.
- Logging and Alarming Component – Ability to log these events and continuously learn as well as trigger the alarm so as to alert of an anomalous activity.

The above was performed in the stages below;

- Created a set of test scripts;
- Established the desired conditions such as permitted TCP packets flow in the computing environment;
- Started the IDS;
- Ran the test scripts;
- Analyzed the IDS's output.

3.1.15 Evaluation of the Smart Health Care Network on Raspberry Board Microcontroller

Evaluation of DDOS attack on the MANET using Raspberry Boards was administered by using the LOIC application for network stress test. The resulting case scenarios implemented using the IPERF application will display the MANET under DDOS an attack and another in normal circumstances.

Four test scenarios were evaluated;

- Hardware MANET module without a DDOS attack.
- Hardware MANET module under a DDOS attack.
- Software MANET module without any DDOS attack.
- Software MANET module with a DDOS attack.

Two use cases were evaluated;

- i) Performance test without any DDOS attack.

Actual test involved measurement of the network performance for 60 seconds without a DDOS network attack.

- ii) Performance test with a DDOS attack

Actual test involved measurement of the network performance for an initial 30 seconds without a DDOS network attack and final 30 seconds with DDOS network attack.

3.1.16 Equipment to be used in the Experiment

- i. **Processor Intel(R) Core (TM) i7 CPU 2.0 GHz :** although a lower version could have been used but using a higher version is recommended for best results
- ii. **Memory 4 MB RAM:** for optimal environment to run the network simulator
- iii. **Raspberry Pi 3 Model B Microcontroller Board:** require to inter-connect the smart watch and host the IDS. This is recommended since the IDS cannot be hosted in a proprietary device such as the smart watch.

CHAPTER FOUR

DATA ANALYSIS, PRESENTATION AND DISCUSSION

4.1 Introduction

This chapter discusses a summary of the literature review on bottlenecks experienced in the application of MANETs for smart health. The design, implementation and evaluation of a fused machine learning intrusion detection model for the provision of smart health care in MANETS. The prototype development process, design process and tenets informing the finite state machines, various attributes that greatly contribute to anomalous activities, observed data patterns and results, model, system design, limitations, challenges, results of the evaluation of the IDS for the MANET network.

4.1.1 Weaknesses and Security Obstacles in The Application Of MANETs For Provision of Smart Health Care.

Objective one of the study required a review of weaknesses and security obstacles in the application of MANETs for provision of smart health care. This was achieved through desk research of literature review and a validation process through a MANET device setup on Linux, to interrogate and verify results of the literature review. This methodology provides a comprehensive, critical and objective analysis of current knowledge on Security bottlenecks in MANET. Secondary data on various weaknesses and security obstacles in the application of MANET for provision of smart health care was collected, which is presented herein.

4.1.2 Methodology for the Identification of Existing Weaknesses and Security Obstacles.

A literature review study was carried out to examine and identify the weaknesses and security obstacles in the application of MANETs in the provision of Smart Health. The following were the objectives of the literature review;

- To identify weaknesses if any, within the MANET ecosystem as a consequence of the device(s) physiognomy.
- To identify the various risks and attacks that can happen or be experienced within the MANET ecosystem as a result of the weaknesses identified.

The literature review was premised on the following empirical research questions.

- To what degree do the weaknesses within a MANET ecosystem contribute to vulnerability?
- To what extent do the weaknesses of MANET devices and the ecosystem, contribute to risks and expose the devices, data and network to attacks.
- To what magnitude do the various weaknesses contribute to loss of confidentiality and availability of the devices and/or data?

4.1.3 The Weaknesses and Security Obstacles In The Application Of MANETs For Provision of Smart Health Care.

The literature review study exposed numerous weaknesses that hinder adoption of MANETs in healthcare due to the nature of these devices and their physiognomies. The results

identified the security weaknesses in MANETs that have inhibited rapid adoption in smart devices for provision of healthcare, as follows.

4.1.4 Weakness 1: Distributed Operation

One of the characteristics of MANETs is that they have no centralized control of network operations. This lack of coordination can bring addressing conflicts, routing and data loops. This results from the lack of a well-coordinated defense mechanism as shown (Inzillo, Serianni and Quintana, 2019)

4.1.5 Weakness 2: Multi-Hop routing

Devices in MANETs forward packets via an intermediate node thus bringing up possibility of eavesdropping and man in the middle attacks. Due to their mobile nature, a device that requires to remotely forwards packets to a neighboring hopping device, which can turn out to be a malicious device, or one that is not authorized to handle the traffic (Zhang et al., 2018).

4.1.6 Weakness 3: Light Weight Terminals

These devices in MANETs are considered Light Weight Terminals with Low CPU capability, low power storage and small memory size. Thus they do not have the ability to provide robust security and protection. Low CPU capability translates to their inability to run high key security algorithms. Low power storage is a weakness that can cause the device to deplete its power resource once overworked by malicious attacks. In addition, its small memory size incapacitates it from running robust security systems (Kamakshi and Kumar, 2018).

4.1.7 Weakness 4: Shared Physical Medium

MANETs ecosystem is by nature a wireless shared medium propagated by CSMA/CA for purposes of collision avoidance due to the shared nature of the physical medium. These devices are thus visible to other devices on the same channel and or any devices with sniffing capability. Further, attacks like MAC addressing snooping are easily propagated in such an environment (Kamakshi and Kumar, 2018).

4.1.8 Weakness 5: Limited bandwidth

Devices in MANET ecosystems mainly exhibit a small packet data size, which propagates bandwidth with a data rate of upto 250kbit/s. This is quite limiting as compared with other devices as those on Wireless Fidelity. This also means that a large case of DDOS on such an ecosystem can easily clog the network (Delkesh and Jamali, 2019).

4.1.9 Weakness 6: Dynamic topology

There is a rapid and dynamic topology change, due to the mobility of the devices in MANETs. This brings upon disturbed trust among nodes, due to their reconfiguration and reorientation to new networks and unfamiliar intermediary devices (Chaudhary and Shrimal, 2019).

4.1.10 Weakness 7: Routing Overhead

Intermediary devices within the MANET ecosystem experience a lot of routing information overhead due to the dynamic networks and mostly stale routes. There are also numerous and unnecessary routing overhead as a result which clog and slow up route resolution functionalities (Garikipati and Rao, 2019).

4.1.11 Weakness 8: Hidden terminal problem

The hidden terminal problem is a common phenomenon which multiplies transmissions thus resulting to collision of packets in some cases. This hidden terminal can also lead to packet losses due to transmission errors. Collisions and packet losses especially in UDP communication are considered expensive since UDP is an un-reliable protocol without strategies for recovery in data loss (Tomar et al., 2019).

4.1.12 Weakness 9: Wireless Radio

Devices in MANETs communicate over wireless links thus suffer from electro-magnetic interference, uni-directional links and frequent path breaks due to mobility of nodes which can lead to loss of data or duplicate frames. (Das and Pal, 2019).

4.1.13 Weakness 10: Mobility

By the fact that they have dynamic mobility, this nature brings about induced route changes and frequent route changes which can cause data loss. This would have dire consequences especially when this technology is applied to monitor healthcare for users whom their lives depend on monitoring devices (Fatima et al., 2019).

4.1.14 Weakness 11: Battery constraints

Most MANET devices rely on batteries to provide power. If the device experiences DDOS attacks, it can lead to draining of battery resources and thus result to broken links or dead links which lead to data loss (Singh et al., 2018).

4.1.15 Auxilliary Security Weakness

The nature of these devices in MANETs is auxilliary node cooperation which can lead to exposure to numerous security attacks. Devices are required to first corporate with similar devices within the ecosystem, while cautious connectivity is discouraged. As a result, a device looking to gather reconnaissance data, finds cooperative devices (Aldaej, 2019).

Table 3: Summary of MANET Vulnerabilities that propagate security threats.

No	Weakness	Vulnerability	Attack / Risk /	Source
1.	Distributed Operation	No centralized control of the network operations	Each node is a relay	(Inzillo, Serianni and Quintana, 2019)
2.	Multi-Hop routing	Packets forwarded via an intermediate node	Eavesdropping	(Zhang et al., 2018)
3.	Light Weight Terminals	Low CPU capability, low power storage and small memory size	Non Robust systems	(Kamakshi and Kumar, 2018)
4.	Shared Physical Medium	Wireless communication	Medium accessible to other entities	(Kamakshi and Kumar, 2018)
5.	Limited bandwidth	Lower capacity	Lower throughput	(Delkesh and Jamali, 2019)
6.	Dynamic topology	Rapid Topology change	Disturbed trust among nodes	(Chaudhary and Shrimal, 2019)
7.	Routing Overhead	Dynamic networks	Stale routes and unnecessary routing overhead.	(Garikipati and Rao, 2019)

8.	Hidden terminal problem	Multiple transmissions	Collision of packets Packet losses due to transmission errors High packet loss	(Tomar et al., 2019)
9.	Wireless Radio	EMI interference	Uni-directional links, frequent path breaks due to mobility of nodes.	(Das and Pal, 2019)
10.	Mobility	induced route changes	Frequent route changes which can cause data loss.	(Fatima et al., 2019)
11.	Battery constraints	Restricted power source	Lack keep alive	(Singh et al., 2018)
12.	Auxilliary Security threats	node cooperation	Exposure to numerous security attacks.	(Aldaej, 2019).

(Source: Author)

4.1.17. Validation of the identified Weaknesses and Security Obstacles in the Application of MANETs for Provision of Smart Health Care.

Following the general literature review on security weaknesses in the application of MANETs for the provision of Smart Health care, there was need to perform validation of the results recorded above, to ascertain that malicious attacks like DDOS do happen on MANETS

easily. This section was achieved through using Proof of Concept methodology. The main objective of this section was to interrogate whether, by taking advantage of weaknesses in MANETs, the following activities can be achieved;

- To verify if a DDOS can be easily propagated within a MANET.
- To verify if a Blackhole attack can be carried out within a MANET.

A MANET network was implemented on Linux, to review the performance of devices to validate and ascertain the results of the desk research. This network was setup without any intrusion detection scheme implemented to note and review the weaknesses that can cause loss of confidentiality, availability and integrity thus inhibit the application of MANET for provision of smart health care. The Figure 4 below shows the MANET set up without any security IDS and the various malicious attacks that were experienced

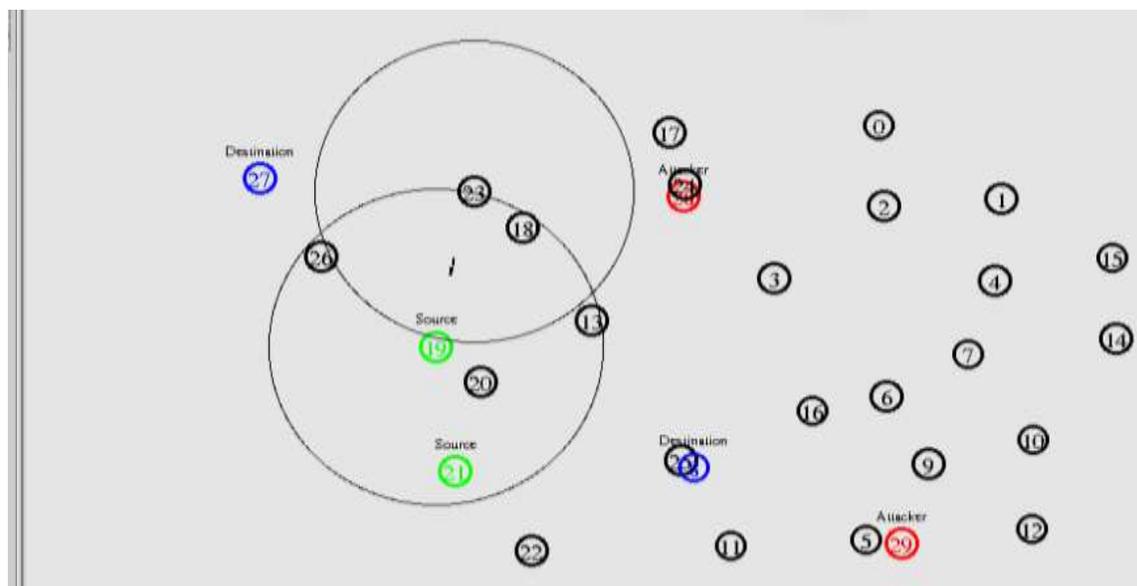


Figure 4: An open vulnerable MANET

The various devices in the above network propagated data using Bluetooth, with two source (19 and 20) and two destination devices (26 and 27). Two other devices were configured as malicious nodes which are required to propagate DDOS and blackhole attacks on the MANET. Running the network above, the experiment enabled the deployment of two types of malicious data in DDOS and blackhole attacks, which were feasible and that run successfully. As a result, there was a high propagation of duplicate packets on the network. This is extremely dangerous, since an attacker can take advantage of this gap and launch data interception, man in the middle attacks, packet replay as well as packet delay.

Following the ability to propagate these attacks, the experiment above confirms that ,if replayed in a real-world environment, it can have serious ramifications to healthcare users. These includes and especially those ailing patients with heart beat monitors, pacemakers, blood pressure monitors can bring life threatening consequences. Table 4 below shows the various malicious packets that were successfully permeated and their anomalous characteristic.

Table 4: Malicious Packets permeated into the MANET successfully.

No	Data Type	Disposition	Vulnerability	Attack
1.	ICMP	High Rates	DDOS	Availability
2.	TCP	Black hole	Reconnaissance	Confidentiality

(Source: Author)

Table 4 above, shows that MANETs are vulnerable and thus propagate two types of attacks against confidentiality and availability as shown above by Blackhole and DDOS attacks.

Blackhole attacks on a network, can affect the data in two ways;

- A malicious node transmits an erroneous RREP (Route Replay) message to the source device; masquerading as the shortest path to the destination thus packets are forwarded to the malicious node.
- In another scenario, incoming or outgoing traffic can be redirected to a blackhole (or a dev/null) without the source device knowing that the traffic did not reach its intended destination

In both cases above, a third party can receive unauthorized or unsolicited packets leading to both loss of confidentiality and/or availability of data. DDOS can be disastrous since systems, data or routes can be completely unavailable to devices, people or systems that direly need them.

4.1.18. Design Recommendations to Address the identified weaknesses and Security

Obstacles In The Application Of MANETs For Provision Of Smart Health Care.

Subsequent to the validation of general literature review on security weaknesses and obstacles in the application of MANETs for the provision of Smart Health care, there emerged observable and feasible recommendations which are enumerated below. The resulting design would take cognizance of the weaknesses in IDS systems for MANET and ensure that MANETs design should not introduce a window for added vulnerabilities to the system and should be self-managed to monitor and identify both hardware and software abnormalities and modifications

spontaneously. The following are key areas to apply when designing a secure IDS system for MANETs.

- The Smart MANET IDS should have ability to identify intrusions by taking cognizance of unfamiliar device addresses;
- The Smart MANET IDS should not permit TCP sessions that are initiated by devices outside its network to get into fruition;
- The Smart MANET IDS should detect the intrusions with low processing and communication overhead;
- The Smart MANET IDS should identify scenarios that cause high resource usage in COU, Ram and bandwidth within the ecosystem;
- Dynamic network topology and mobility of MANET devices should not affect the detection accuracy of the Smart IDS MANET system;
- The resultant IDS system should be scalable so as to accommodate with negligible errors, the increasing density of devices joining or within the MANET increase.

4.1.19. Conclusion.

This section discussed the literature review on weaknesses that inhibit application of MANETs for smart health. The literature provided vital knowledge on various other researchers' experiences that will guide the design, implementation and evaluation of a fused machine learning intrusion detection model for the provision of smart health care in MANETS.

The results showed that existing intrusion detection methods are mainly built for compute-intensive systems and mainly and most commonly, for networks with a rigid architecture and topology. MANET are mobile and deployed in a scattered fashion, with a frequent change in network topology which translates from their continuously changing addresses schemes, depending on the hosting network that they plug into. As a consequence, therefore, these devices must consistently reconfigure their routes. As such, MANETs and devices in MANETs lack a central controlling system thus must perform these tasks on their own. As a result of both the literature review and validation of the same, the following research questions were answered;

- The various weaknesses within a MANET ecosystem do contribute to vulnerabilities within the MANET ecosystem.
- These weaknesses of MANET devices and the ecosystem, exposes the devices, data and network to attacks.
- Attacks are experienced within the MANET ecosystem do contribute to loss of confidentiality and availability of the devices and/or data.

The section 4.2 next presents the design of a fused machine learning intrusion detection model for the provision of smart health care in MANETS taking into cognizance recommendations i,ii, iv and vi which are within the scope of this study

4.2. Design of A MANET Anomaly-Based Intrusion Detection Model for Smart Health care.

This section presents the results of objective two of the study, which set out to design a fused machine learning intrusion detection model for the provision of smart health care in MANETS.

The design was based on the recommendations provided in section 4.1 (c). Recommendations i, ii, iv and xi were taken into account to attain the design requirements. The Design aspect of the Model was implemented in various ways as shown in the following steps, below. The design of the fused MANET anomaly-based intrusion detection model for the provision of Smart Healthcare undertook this design approach so as to accomplish three critical achievements; the logical topology, the model logical events, and the network connectivity.

Logical design is implemented as a model using the functional decomposition methodology. Achieve a Logical Topology to describe the organization and connectivity of smart devices to the smart IDS system within the MANET. How these devices receive/send data and how the IDS audits the data being propagated in the MANET ecosystem. This does not describe how the devices physically interconnect however; this logical topology is considered isomorphic to the physical topology. On the contrary, the Model logical events design implemented as a Finite State Machine (FSM) using PPDIIO methodology. Model the various Logical Events that describe normal and anomalous activity within a MANET. Lastly, the Network connectivity design is implemented as a model using PPDIIO methodology and implemented on Linux NS2 and as a live experiment as well. Achieve a physical network topology describing how the devices physically interconnect however, this physical topology is considered isomorphic to the logical topology.

4.2.1. Design of Logical Topology.

A logical topology refers to the arrangement of components on a computer network and describes how they communicate with each other on the network. Logical topologies show how signals are transferred and how they act on the network (Ganichev et al., 2018).

The design of this logical topology was achieved using functional decomposition. Functional decomposition is a methodology of systems analysis that separates a complex entity so as to describe the separate contributing components. Functional decomposition enables understanding and organization of complex entities which are used to help solve problems and thus help to develop business operations, computer systems and machine learning among various other use cases (Harris and Davies, 2019). The methodology was deemed best since the research comprised of fusion of two discrete individual processes and techniques. This enabled a flowing description of the functional relationship between the various constituent components and integration of separable logical processes herein compressed into a representation of the global IDS.

The steps followed in Functional Decomposition are;

- Find the basic function – This describes the basic task a device or process must succeed at achieving.
- List the essential sub-functions – This describes the various sub-functions which are key to the success of the basic function.
- List the next tier of sub-functions - This describes the sub-functions which serve the upper level sub-functions.
- Inspect the diagram – Enables the researcher to inspect and identify functions that might have been omitted and add them to the diagram.

The two constituent components were;

- A Data mapping separator using a Support Vector Machine.
- Anomaly Detection Engine using an Artificial Neural Network

4.2.1.1. Design of a Data Mapping Separator using Support Vector Machines.

The design of the Data Mapping Separator was achieved by delinking this component from the overall conceptual framework presented in Figure 5 below. A review of Support Vector Machine algorithms for data classifiers was done. This involved identifying the right and relevant interest data which is consequently captured, segregated and analyzed. In the proposed model, as shown in Figure 4 below, a node in the mobile ad-hoc network participates in MANET by sending or receiving packets via Bluetooth.

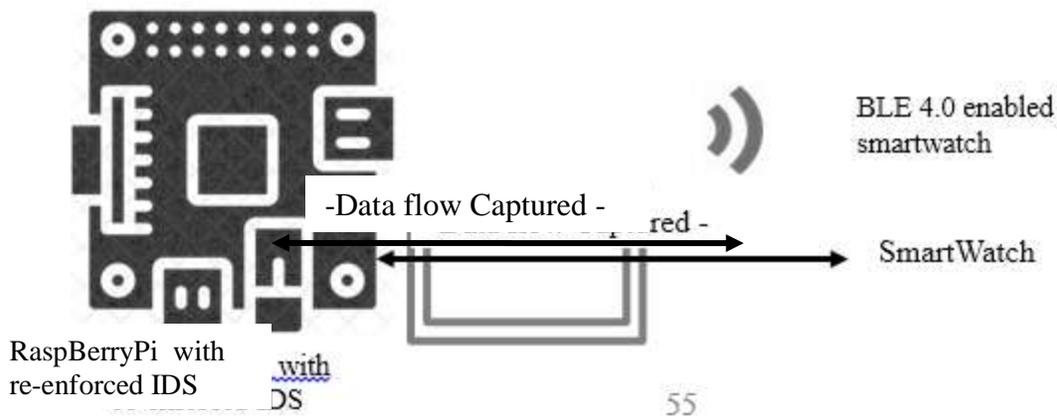


Figure 5 : The Logical Topology for Data Capture from the smart watch Bluetooth device
(Source: Author)

Once the devices are connected, an instance of a Linear SVM classifier is initiated as shown in Figure 5 below. Interest data is separated from all the data available within the

MANET. This is classified into no-interest, interest and high interest data as shown in Table 4 below;

Table 5: Categorization of packet flow to interest data

No.	No-Interest	Interest	High Interest
1.	DSR	BTHCI_ACL	TCP
2.	AODV	BTHCI_CMD	UDP
3.	TORA / LMR	BTHCI_EVT	

(Source: Author)

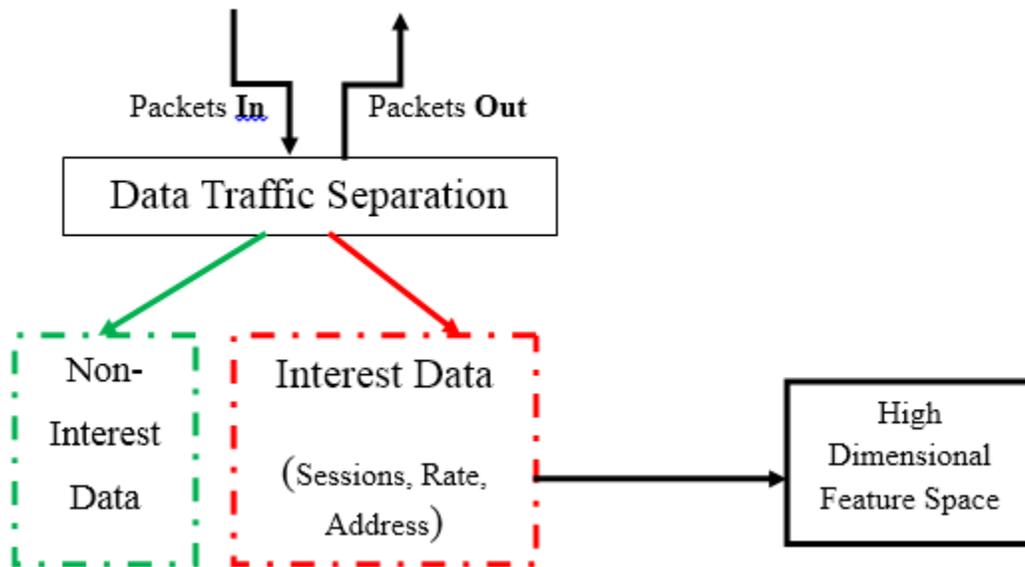


Figure 6 : Data Classifier Model (Source: Author)

Figure 6 shows the Data Separator Sub-Model which was designed using the functional decomposition methodology and implemented as a running experiment using Support vector Machine data classifier algorithms. The model in figure 5 monitors packets flowing into and out of the network and separates them accordingly. Packets which are of interest to the IDS are forwarded for analysis. The SVM algorithm generates a hyper-plane which segregates the interest data according to behavior, rate and type separating the two classes correctly. The trained a Linear SVM classifier compares the two vectors separated by the decision boundary or hyper-plane with the two nearest neighbor packets data points (D+ and D-). The results of SVM are then fed into an Artificial Neural Network algorithm whose design is described in the next section.

4.2.1.2.Design of the Anomaly Detection Engine using Artificial Neural Networks

The design of the Anomaly Detection Engine using Artificial Neural Networks was also achieved through functional decomposition methodology. This was a result of delinking the various components from the overall conceptual framework, so as to identify the sub-model designs. Figure 7 below shows the interaction between SVM and ANN engines. This is the fusion stage of the fused Anomaly detection model

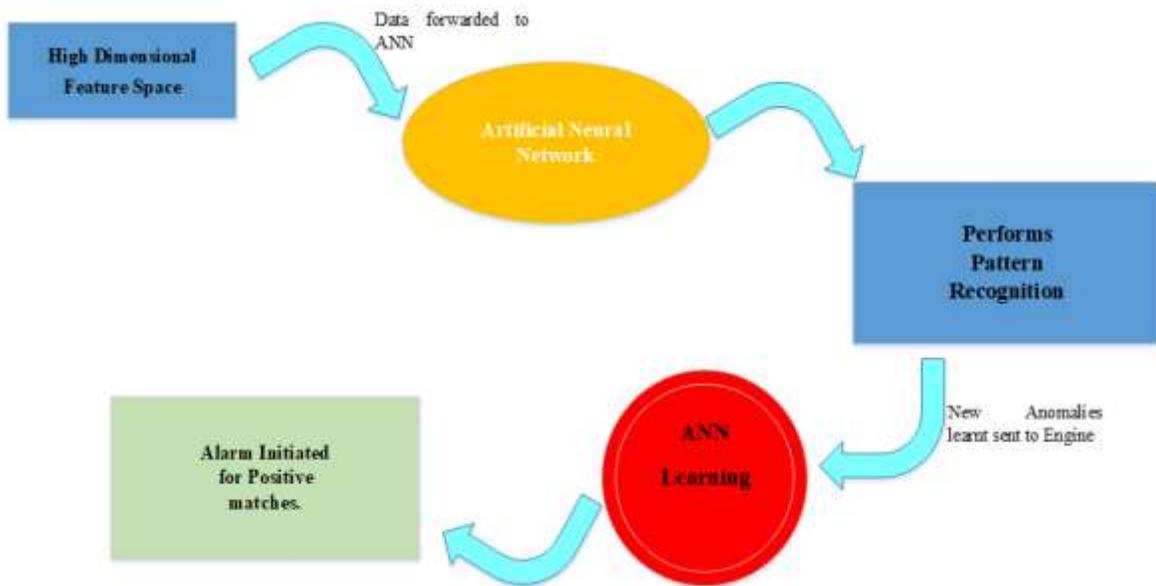


Figure 7: SVM into ANN integration (Source: Author)

Data classified by the SVM algorithm is fed into the Artificial Neural Network that performs pattern recognition tasks. The ANN applies modified probabilistic radial basis function (PRBF) to perform pattern recognition tasks. This approach is normally applied on various test problems which has multiple uncertain parameters, so as to improve and harness some insights. Combinations of different radial basis functions and sampling techniques are used to study the performance of different combinations. Data packets with anomalous symbol are thereafter passed into the anomaly detection engine. If the data is positive, an alarm is raised and particular anomaly is reported. The Alarm function can also feed forward into the artificial neural network for continuous learning.

4.2.2. Design of a Finite State Machine to describe Normal and Anomalous Events within a MANETs

This design section was implemented as a Finite State Machine (FSM) using PPDIIO methodology. FSM is an artificial intelligence technique that allows predictability with a given set of inputs and a known current state; therefore, state transitions can be easily predicted and thus enable for easy testing. FSM also enable determination of reachability of a state thus when these states are represented in an abstract form, it is insentiently clear whether one state can be arrived at from another state, and what is required to arrive the state (Groz et al., 2018). The design of this FSM was based on evaluation of MANET traffic data that enables identification of the anomalous data and or intruder node and type of attack. Normally FSM design follows five (5) steps, as shown in figure 8, below;

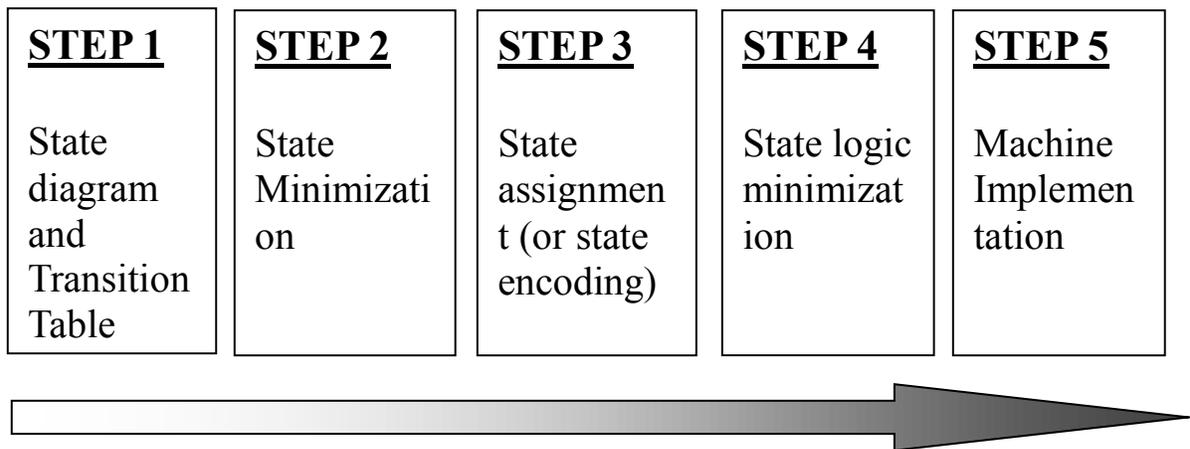


Figure 8: Finite State Machine Design and Construction Process (Source: Author)

4.2.2.1.State Transition Table

The State transition table describes what state a finite semi-automaton or finite state machine will move to, based on the current state and other inputs. These states were achieved

through functional decomposition. Functional decomposition in FSM enables discrete states to be treated as such. Consequently, the following scenarios would provide the logic behind identifying anomalous activity within the MANET. They were isolated for observation against the following TCP and UDP actions; table 4 below illustrates the scenarios involved.

Table 6: Logical and Data Propagation Scenarios in a FSM

No	Data Propagated (Σ)	Scenario (q0-F)	Transition Function (δ)
1.	Number of packets sent/received	Normal MANET Listen disposition including protocol overhead.	LEARN
2.	Number and Type of packets sent/received/forwarded	Normal MANET traffic flow (period/duration)	LEARN/FORWARD
3.	Number of packets drop	TCP 3-Way Handshake	LEARN/FORWARD
4.	Data Rate against source/destination IP address/MAC	Data Transfer (after Handshake)	LEARN/FORWARD/CLOSE
5.	Percentage of number of packet received/sent	Percentage and density of packets sent by Source IP	LEARN/FORWARD
6.	Percentage of number of packet received/sent	Percentage and density of packets sent to Destination IP	LEARN/FORWARD
7.	Percentage of number of packet sent	Cascading bit rate and interval in bps	LEARN/FORWARD
8.	Percentage of number of packet sent	Observed maximum frame size	LEARN/FORWARD

The formal notations for the above are defined by $\{Q, \Sigma, \delta, q_0, F\}$ whereby;

Q = set of all MANET states

Σ = set of all packet propagation sent, received or forwarded

δ = transition function

q_0 = idling state (start state)

F = good activity / acceptable normal activity(s)

4.2.2.2. Design of a State Diagrams for MANET propagation activities

State diagrams are visual representation of the various states, inputs and outputs of a finite state machine. The state diagrams design was produced as a result of state transition tables which describe the outright logic for every state and resulting state once input is received to the automaton. On the basis of the transition table in (a) above that presents the various data propagation scenarios, an abstracted visual connectivity design for the prototype MANET network was designed so as to exhibit connectivity between two nodes. The Transition table 6 above abstracts the role of nodes and routers. MANET nodes are mobile and are most commonly connected dynamically in an arbitrary fashion. These nodes within MANETS behave as both nodes and also as routers which are responsible for route discovery and route maintenance of routes.

4.2.2.2.1. Finite State Diagram Logic for Normal Activity within MANETs

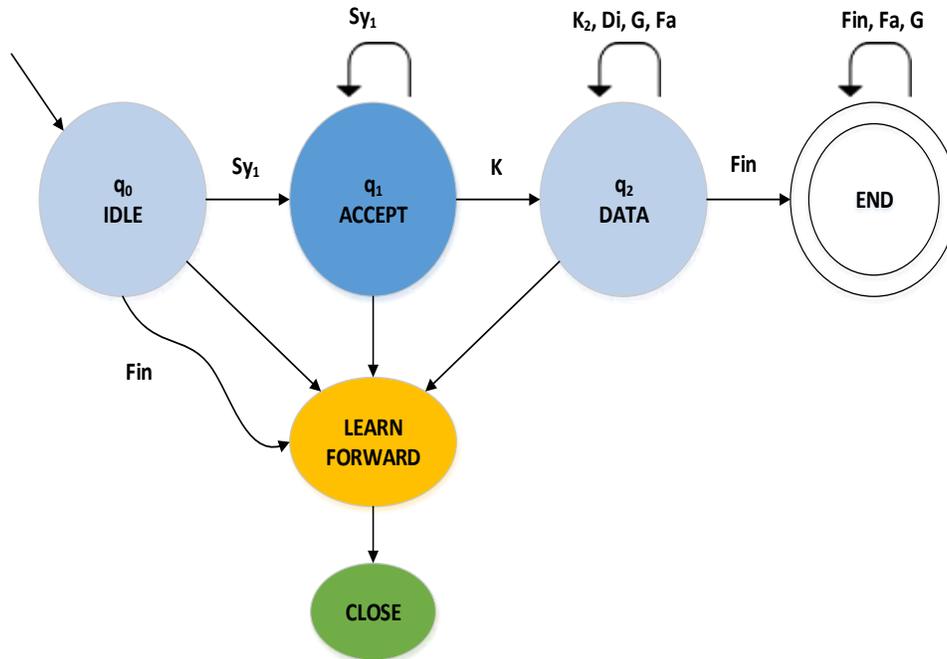


Figure 9: FSM for Normal Transition Activity in MANETs (Source: Author)

Figure 9 above is an abstract MANET machine where nodes transition from a sleep or idle mode and initiate data transfer in various ways depending on type of data – control data, normal data encapsulated either as TCP or UDP type. The transitions regarded as normal transitions with any deviation marked as anomalous (X) as in the state assignment step in (d) below.

4.2.2.2.2. Finite State Diagram Logic for TCP activity

A spread algorithm was used to create a Finite State Machine for abstracting TCP sessions in MANETs. The machine includes formal notations in TCP connections where MANETS are only established by satisfying the following;

- Packets are exchanged over the same TCP port;
- The source and destination IP addresses are commensurately matched;
- The sequence numbers are in sync;
- And, acknowledgement numbers match correctly. The FSM design below thus takes the above into consideration and the following case scenarios were set as shown in the Transition Figure 10 below.

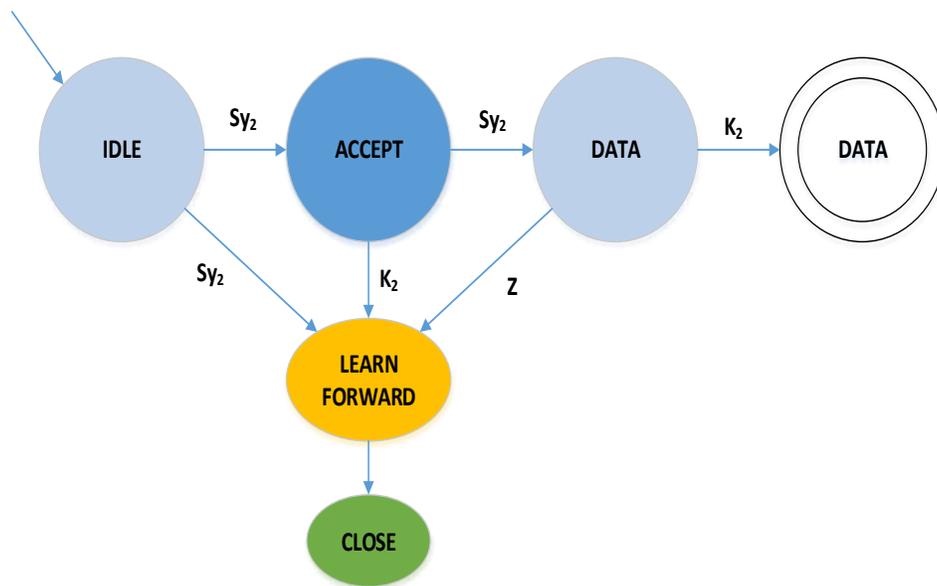


Figure 10: Shows the Normal and Anomalous FSM for TCP sessions in MANETS (Source: Author)

TCP in MANET follows the same transition to achieve the TCP 3-Way Handshake. MANET devices have a sleep or idle state from where all sessions originate. The anomalous transitions are indicated with an (X) as in the state assignment step in (d) below.

4.2.2.2.3. Finite State Diagram for UDP activity

In the formal notation below, for every set of packet request (PR) sent, the abstract machine evenly matches that request to the number of packets forwarded (PF). This automata design is built on the theory of minimizing machines. For protocol data ie Hello packets data, these packets sent have to be not only received within a certain time frame but also have to match the sender ID. Data packets have to also meet the send/receive protocol otherwise, the device and data is declared a victim or attacker. The base abstract machine therefore follows the following notation;

- PR – Packet Request
- PF – Packet Forward, thus;
- Algorithm – $PR^i PF^i \mid i > 0$

Figure 11 below shows the transitions in a voluminous amplification.

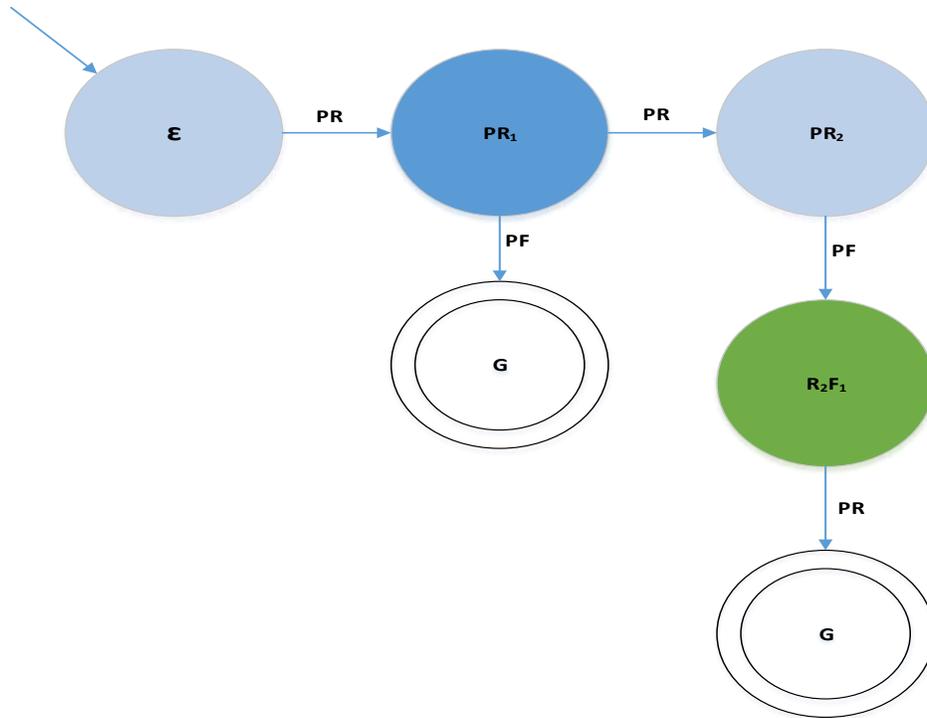


Figure 11: Design of a Formal Notation in Voluminous Amplification in UDP and Denial of Sleep

The devices transaction begins with a q_0 state with no data requested/received represented by the epsilon (ϵ). As a result, the algorithm is minimized by iteration from every PR and closing data transfer at PF as shown in the state assignment step in (d) below.

4.2.2.3.State Minimization of Finite State Machines

State minimization is the process of reducing or transformation of a given FSM into an equivalent but smaller machine with n redundant states. This step can be optional for FSM that do not have redundant or indistinguishable states. The goal of this step is to identify and remove redundant states if any.

The FSM for UDP voluminous amplification in UDP exhibited various indistinguishable states and as well as amplification of Denial of Sleep. Since the devices amplify every iteration PR with a commensurate PF as shown in Figure 12 below, a minimized FSM for this is presented as shown in figure 8 below.

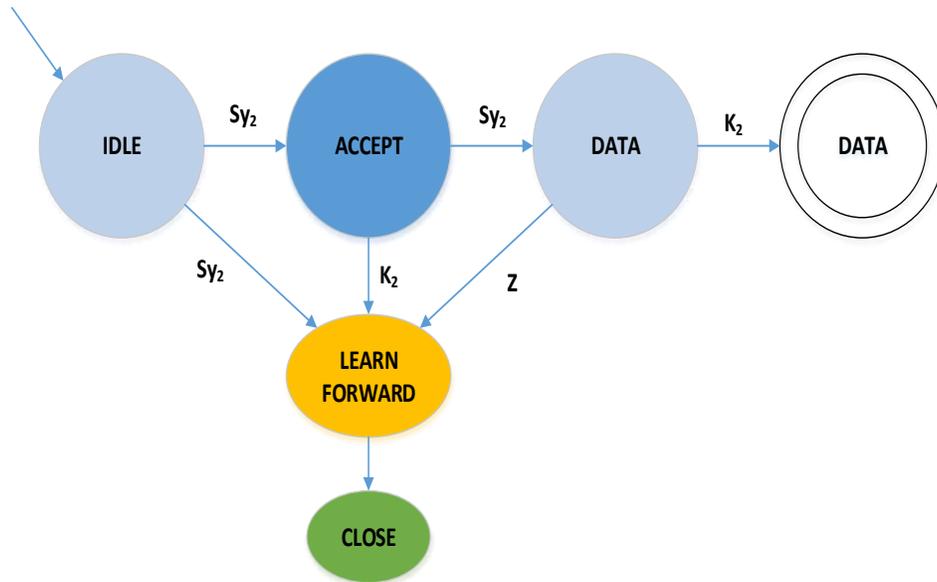


Figure 12: Minimization of Finite State Machines

4.2.2.4.State Assignment for acceptable and anomalous inputs

State assignment is the process of using state transition variables to express specific states and expound how the values of these state variables are arrived at. Table 7 below enumerates the state transition variables that identify invalid inputs within a TCP activity showing during normal and anomalous data exchange.

Table 7: State transition variables that identify invalid inputs within a TCP session

Transition	q0	q1	q2	q3	LF	Close
State	O	Sy1	X	X	X	O
	Sy1	Sy1	K2	Sy1	X	G
	K2	K2	K2	D	X	G
	D	X	X	D	X	G
	Fin	X	X	X	Close	G
	G	Sy1	X	X	X	G
	An	-	-	-	-	G

(Source: Author)

The following transitions are thereby flagged as anomalous by the function notation

- !normal Transition/State = X
- For State O, X = Sy1 for q1,q2,q3
- For State Sy1, X = q3
- For State K2, X = q3
- For State D, X = q0, q1, q3

- For State Fin, X = q0, q1, q2, q3
- For State G q1, q2, q3

The assignment in Table 7 espouses the various machine states (O, Sy1, K2, D, Fin and G) and their respective acceptable inputs when the machine is at that particular state. Any other input to the machine, apart from those listed are considered anomalous. Table 8 below enumerates normal and anomalous activity between two devices exchanging data after having created a TCP session.

Table 8: State logic assignment for TCP activity between TCP devices

	Transition	q0		q1		q2		q3
State	O	Sy1		X		X		X
		Sy1		Sy1		K2		Sy1
		K2		K2		K2		D
		D		X		X		D
		Fin		X		X		X
		G		Sy1		X		X

(Source: Author)

The behavioral model of the machine in TCP connection identifies the anomalous functions as the following;

- During the idle or sleep state, $X = Sy2, K2$ and Z where only $Sy1$ is a normal function.
- During the Open state, $X = K2$ where only TCP Sync traffic is the normal function
- During the Data state, $X = Sy1$ and $Sy2$ which should not be initiated during this transition.
- During Fin state, $X = Sy1, Sy2, K2$ and D since the only acceptable function is the TCP Close.

This section achieved to produce the FSM designs based on the recommendations provided in section 4.1 (c). The FSM were successfully able to model various logical events that are experienced in a MANET. In addition, and most importantly, the automata expressed these events discretely and successfully indicated events/inputs that were acceptable and those events/inputs that were deemed unacceptable thus anomalous. This allowed the expression of anomalous events through decomposition of the various states. Knowledge produced here was represented in the states, and thus it was easy to transfer these abstract representations to a coded implementation as shown the section (4.3) that follows.

4.2.3. Design of the Integrated Network Connectivity within a MANET

This Network Connectivity Model was designed using Prepare, Plan, Design, Implement, Operate, and Optimize (PPDIOO) methodology. This was to later be implemented on Linux NS2 and as a live experiment. This section achieved a physical network topology describing how the

devices physically interconnect however, this physical topology is considered isomorphic to the logical topology.

The objective of this network connectivity design was;

- Show the interrelation between the physical network and logical topology
- Show the interaction between data handlers and the fused IDS engine

The design of the network connectivity is presented in Figure 13 that follows below;

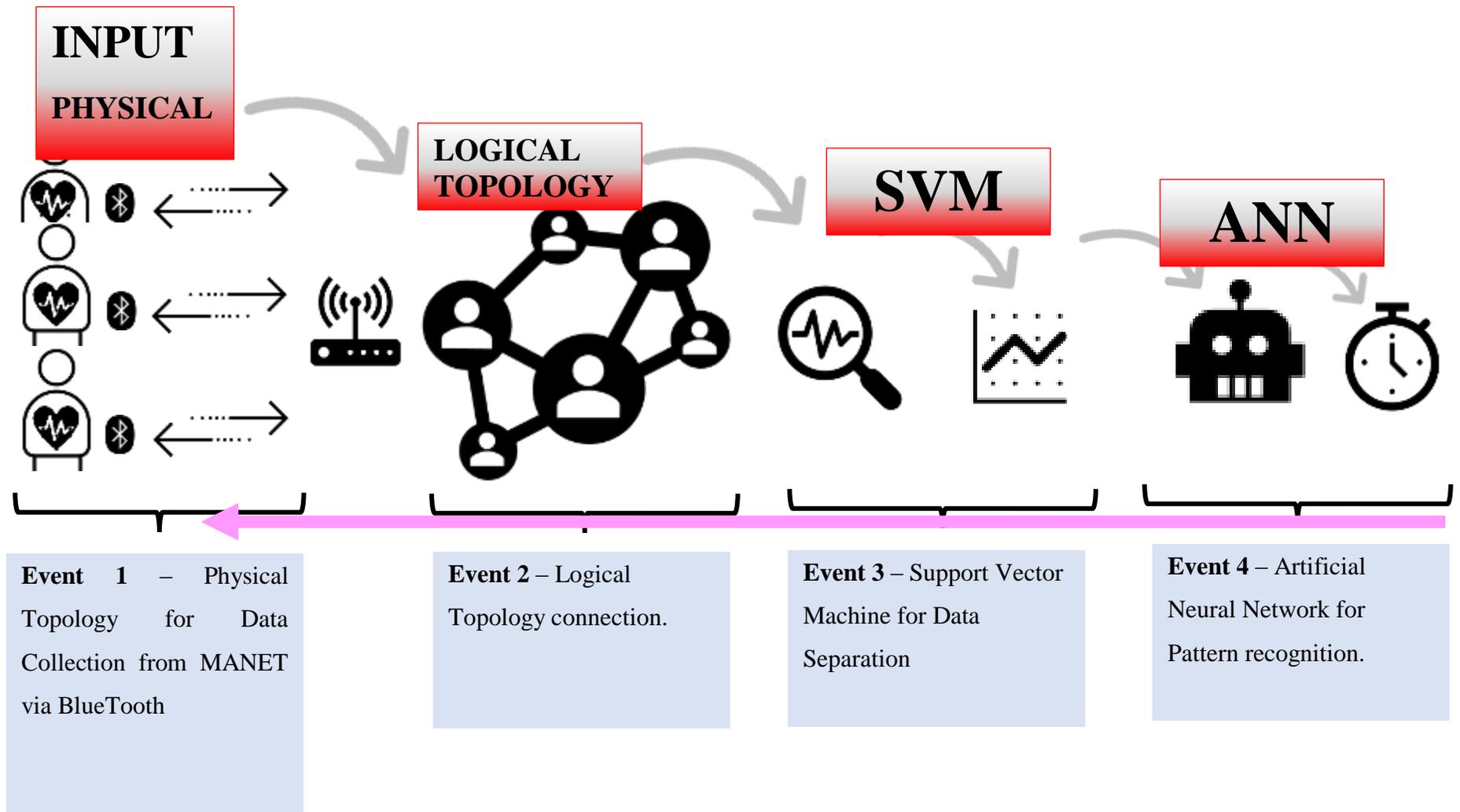


Figure 13: Model Design for the Fused Machine Learning Intrusion Detection Model For The Provision Of Smart Health Care In MANETS.

4.2.4. The Model Design for the Fused Machine Learning Intrusion Detection Model for The Provision Of Smart Health Care in MANETS

This section achieved to produce the model design for the fused machine learning intrusion detection model for the provision of smart health care in MANETS. This model intergrated all the three elements in a physical topology, logical topology and the fused machine learning intrusion detection engine and describes how these entities are intertwined and intergrated.

The first event 1 presents the physical topology of the MANET whereby Bluetooth enabled smartwatch captures raw user data and forwards the same to the MANET ecosystem. For this particular research, the envisaged data to be collected is the patients blood pressure, and the hardware devices are suited for the same. The physical connection is deemed to be Blue Tooth 4.0 and above. These devices are deemed to be mobile and exhibit the characteristics discussed in section (4.1) above. The results of this section flow into the Event 2 on logical topology.

Event 2 presents the logical topology provided through ad-hoc connectivity service set. MANET Devices connect randomly and dynamically since they are free to roam dynamically changing the network topology frequently. Each node is expected to forward traffic unrelated to its own use, and thus must behave as a router. The MANET is required to consistently keep network routing information so as to properly forward packets correctly and efficiently. This MANETs topology can either be a standalone or can also be connected to the Internet. The results of this section flow into the Event 3 for SVM data capture.

Event 3 above presents the support vector machine engine that is responsible for capturing all the data propagated in the MANET. The SVM thereafter identifies interest data and segregates the data into various interest groupings. The data is segregated into interest and non-interest data. Interest data is regarded as that whose input is likely to lead to anomalous state. This captured interest data is flow into Event 4 which is the Artificial Nueral Network.

The ANN function is to perform pattern recognition. Interest data received from the SVM is matched against rules described by the FSM to check whether its inputs or events are considered unacceptable, thus anomalous. If the ANN finds events or input which match those patterns considered anomalous, the alarm is raised and these events are recorded and also fed forward into the smart devices.

4.3. Implementation of The Fused Machine Learning Intrusion Detection Model for The Provision of Smart Health Care in MANETS.

This section presents the results of objective three of the study which set out to implement the fused machine learning intrusion detection model for the provision of smart health care in MANETS designed in objective 2. To assure the validity of the results, research triangulation was done using a simulated experiment and a live experiment. According to National Academies of Sciences (2018), triangulation refers to the use of more than one method to collect data on the same topic to assure validity of research.

4.3.1. Implementation of the MANET IDS on Linux using NS 2

In the simulation step, a MANET network was implemented on Linux, with the following set of objectives;

- Set up a typical MANET network to depict mundane MANET disposition.

- Put the MANET network under various malicious attacks.
- Introduce the fused intrusion detection system to protect the MANET from these attacks.

In the proposed model, as shown in Figure 14 below, every node in the mobile ad-hoc network participates in intrusion detection. Each node is responsible for detecting signs of intrusion however, neighboring nodes can collaboratively investigate in a broader range. The MANET network was comprised of the following devices, as shown;

- A total of 25 devices were configured
- Two (2) data source devices where data originated
- Two (2) data destination devices where data was sent to
- Two (2) attacker/malicious devices which propagated various attacks

The MANET network is designed and implemented on NS 2 as shown in figure 14 below;

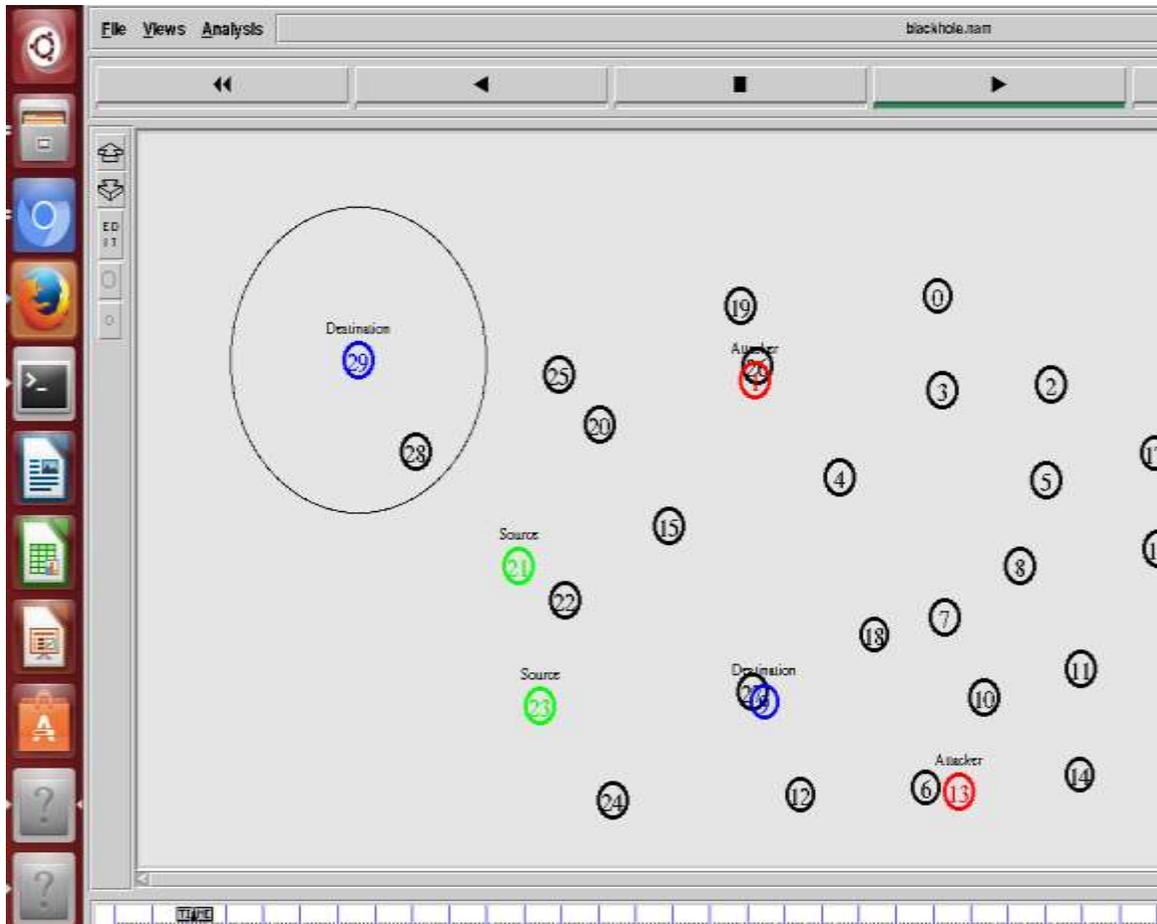


Figure 14: MANET with RFDs, FFDs and Malicious radios.

There are 25 participating devices in the network, of which 19 are set to promiscuous mode but can go live when needed to undertake the MANET under stress tests. 2 devices (marked green) are set to send/originate data which is received by two devices (marked blue). Within this ecosystem, there are two malicious nodes (marked red) which are the sources of various attacks – blackhole and DDOS. This IDS, analyzes the packets going into or out of the MANET, in search of undesirable and suspicious activities. To effectively monitor and protect against threats, a machine learning module is added by creating a description of newly discovered abnormalities. The device(s) generally has to find a match between current activities

and anomalies, only when a positive match is found, does the alarm is generated. This anomaly-based detection machine learning technique creates normal profiles of system states or user behaviors and compares them with current activities within the MANET. If a significant deviation is observed, the IDS raises an alarm and most importantly adds this to its learning gene. The Linux IDS Script for the MANET Topology is presented in appendix B, code listing 1. Line 16 of of appendix B, code listing 1 below specifies the number of devices;

```
16. set val(nn) 25 ;# number of mobilenodes
```

While line 44 of appendix B, code listing 1 enables ad-hoc routing for MANETS.

```
44. $ns node-config -adhocRouting $val(rp) \
```

As a result, a MANET environment is created and data ispropagated on the network simulating a real world MANET environment.

4.3.1.1.Simulation of Machine Learning Phase 1- Setting Algorithm learning parameters for normal and abnormal modes

Figure 15 below show the MANET in normal propagation mode.

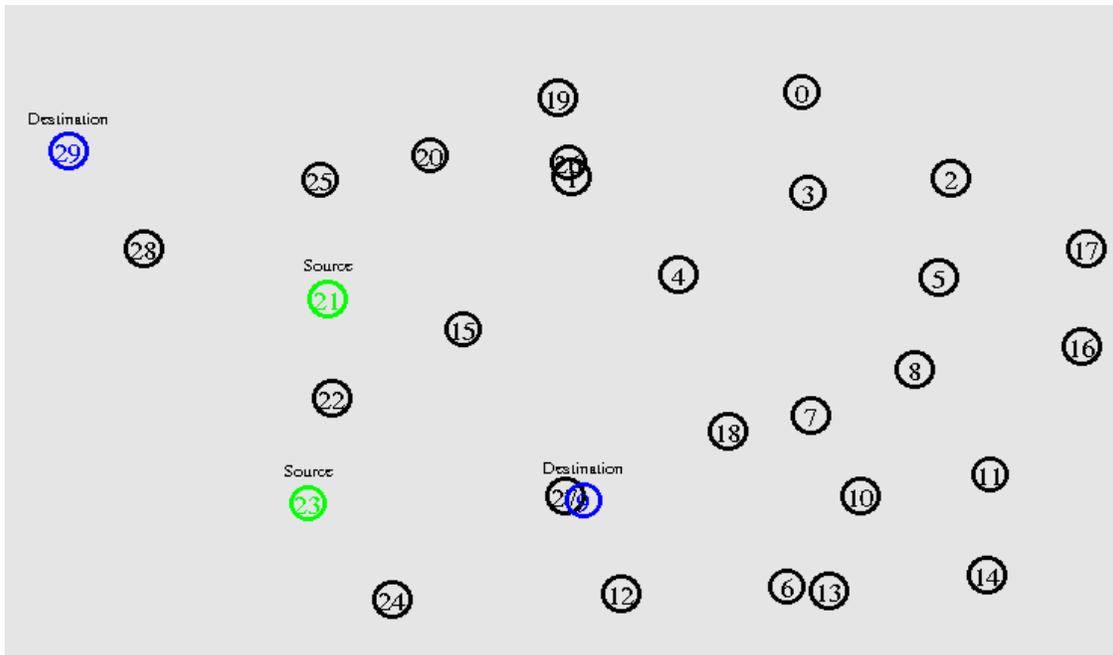


Figure 15: Normal Data Propagation Mode

Figure 16 below show the MANET propagation when malicious packets are sent;

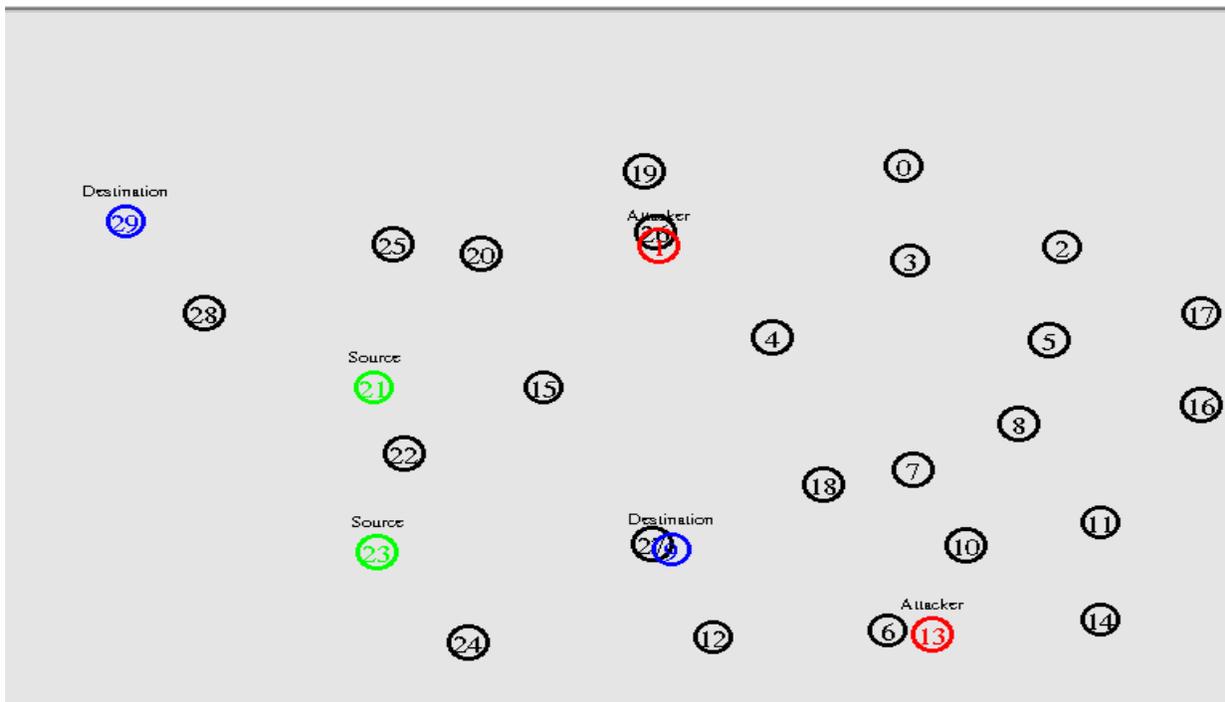


Figure 16: MANET With Malicious Data is Propagating

Since in a MANET, mobility-induced dynamics make it challenging to distinguish between normalcy and anomaly, there is need to have a good representation of normal devices (in promiscuous mode) that can be used to comparatively give what is considered as the following

- Normal Idle
- Normal Propagating (sending)
- Normal Propagating (receiving)

As well as;

- Anomalous Idle
- Anomalous Propagating

This proposed IDS provides a promising alternative and specification based on detection techniques which combine the advantages of misuse detection and anomaly detection by using machine studied specifications to characterize legitimate system behaviors. The support vector machine code for collecting interesting data into the engine is listed under appendix B, code listing 2. Line 65 of code listing 2 shows the importation of data for purposes of classification, as shown below.

```
set list {}
```

```
foreach dir $import_dirs_{
```

```
lappend list [$self file join $dir \
```

```
[$self class_to_file \
```

This enables interest data to be loaded for malicious detection. Attacks are identified as deviations from a normal profile, and is improved by continuously comparing propagation of the 25 nodes within the MANET. However, the downside is that the development of detailed specifications can be time consuming. The Linux IDS Script for Learning Normal and Anomalous activity for machine learning phase 1 is presented in appendix B, source code 3. Line 29 of code listing 3 creates a tracefile that records all anomalous activity relating to blackhole attacks. This enables the IDS to learn how balckhole attacks are propagated

29. *set tracefile [open blackhole.tr w]*

The resultant file is blackhole.tr. A Trace file written by an application to store overall network information. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Line 32 of coe listing 3, initated visual reports to enable viewing of the propagation of packets during a blackhole attack.

32. *set namfile [open blackhole.nam w]*

To differentiate between the various devices on the network, color coding was impemented. Red color would indicate a malicious attacker, green would indicate the source of data and blue the destination as show on lines 201 to 211 on code listing 3, appendix B.

201. *\$ns at 0.0 "\$n13 color red"*

202. *\$ns at 0.0 "\$n13 label Attacker"*

204. *\$ns at 0.0 "\$n23 color green"*

205. \$ns at 0.0 "\$n23 label Source"

207. \$ns at 0.0 "\$n21 color green"

208. \$ns at 0.0 "\$n21 label Source"

210. \$ns at 0.0 "\$n29 color blue"

211. \$ns at 0.0 "\$n29 label Destination"

Nam is a TCL based animation tool that enables viewing of network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various other data inspection tools.

4.3.1.2. Machine Learning Phase 2 - Propagating Blackhole Attacks on a MANET using NS2

During a BlackHole MANET Attack, it is critical to ensure RREP is set with a Destination address and organization more noteworthy than the destination arrangement of the receiver node. This makes the sender node trust the black hole node and addition, interconnects with the malicious blackhole node in its place of the real trusted destination node. This mischievous setting, frequently harms the victim node interfacing with the attacker, and thus consuming all network resources rendering assets not only unusable but also causes packet loss. Figure 17 below illustrates a blackhole attack topology.

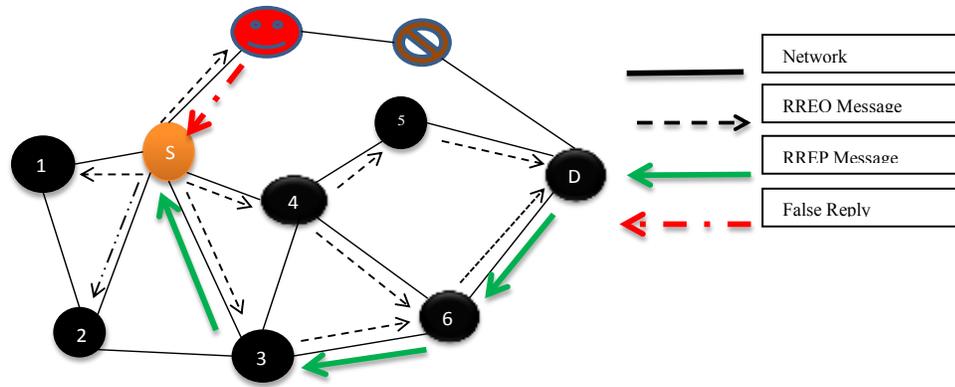


Figure 17: Topology of a blackhole attack.

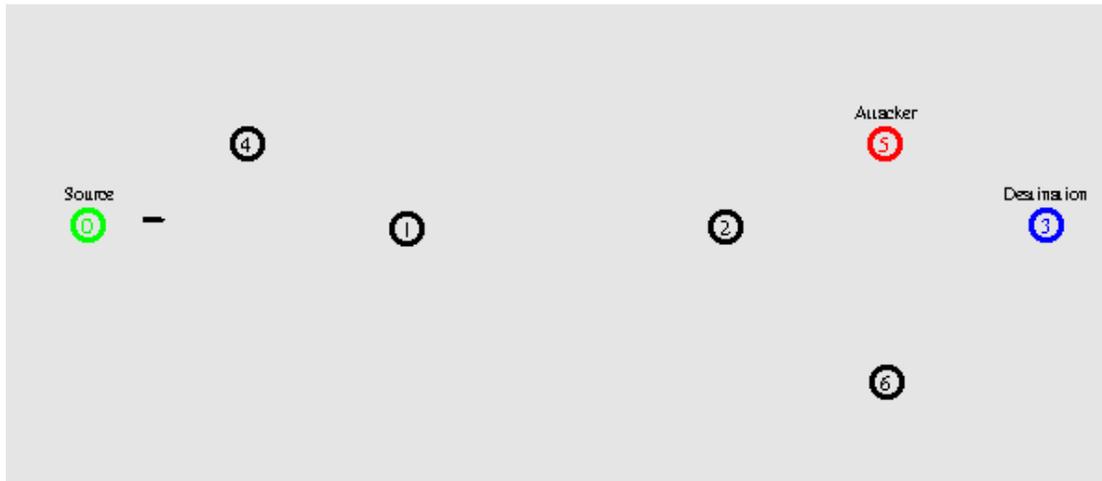


Figure 18: Implementation of IDS script for blackhole learning environment

In this study a blackhole attacks were propagated by malicious node 5 within the MANET. To simplify the attack, 7 nodes were used for purposes of initial assisted machine learning with the following functions;

- Node 1, 2, 4 and 6 – Normal promiscuous mode.

- Node 0 – Source of Data
- Node 6 – Destination of the Data

This design and implementation is as indicated in the figure 18 as shown above;

In this learning phase, Node 5 propagates malicious blackhole scripts that forces Source Node (0), to redirect traffic meant for destination Node (3) to instead route traffic to malicious Node (5). During this machine learning phase behavioral genetics of an idle node, source node, destination node and malicious node is reviewed and logged. The Linux IDS Script for Propagating a Blackhole attack on the MANET Topology in figure 18 is presented in appendix B, code listing 3. Line 29 of code listing 3 creates a tracefile that records all anomalous activity relating to blackhole attacks. This enables the IDS to learn how balckhole attacks are propagated

29. *set tracefile [open blackhole.tr w]*

The resultant file is blackhole.tr. A Trace file written by an application to store overall network information. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Line 32 of coe listing 3, initated visual reports to view the propagation of packets during a blackhole attack, by color coding.

32. *set namfile [open blackhole.nam w]*

Nam is a TCL based animation tool that enables viewing of network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various other data inspection tools.

4.3.2. Implementation and Testing of the MANET IDS on Linux and NS 2 against TCP SYN Flood, Blackhole and malicious traffic

After integration of the algorithms simulated, it was critical to evaluate the performance of the integrated machine learnt IDS against TCP SYN Flood, Blackhole and malicious traffic on the MANET. The MANET network was set-up with the same exact Nodes, namely;

- Two (2) source nodes to send data.
- Two (2) destination node to receive data.
- Two (2) attacker/malicious devices (blackhole, TCP SYN).
- In total, 30 devices converged within the network.

The MANET topology for this network was designed and implemented as shown below by figure 19.

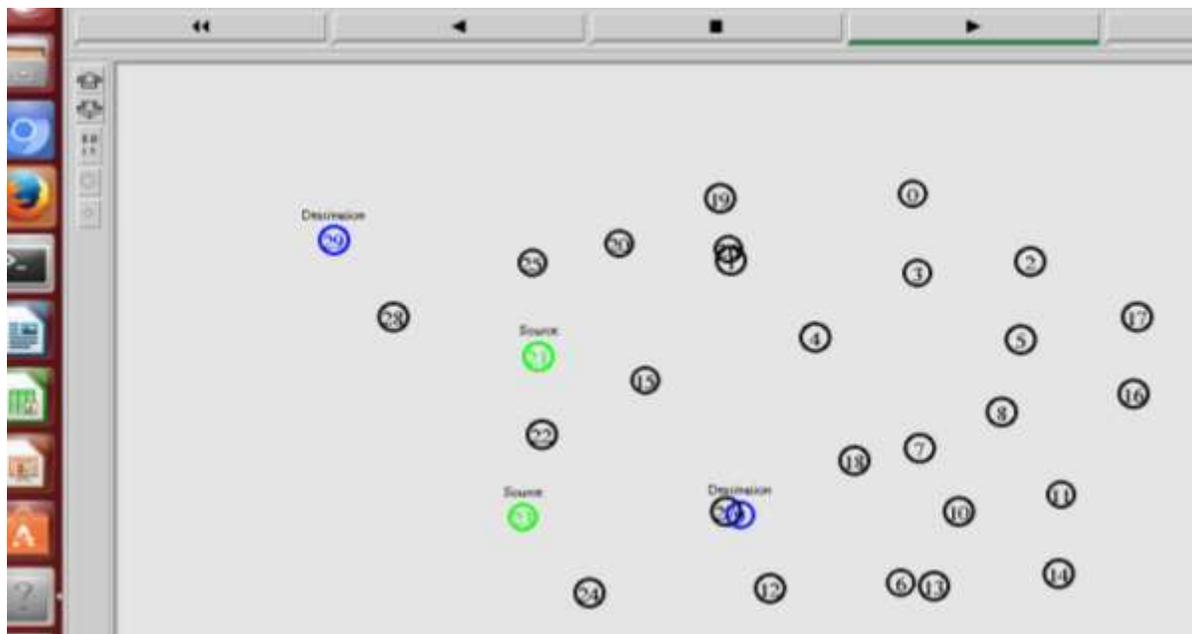


Figure 19: MANET topology with IDS algorithm injected into the FFD

To achieve this, the following are important member functions that participate in the project for the following key objectives;

- Member function to specify and initiate the packet format.
- Member function to create a scheduler of events and actions within the MANET.
- Member function to enable selection of the default packet addressing scheme. This can be IP or IPv6.
- Member function to create MANET devices such as nodes and links.
- Member function to inter-connect network component objects created eg via Bluetooth, AODV etc.
- Member function to create connections between agents. These can be either TCP or UDP connections.

All the above member functions are intergrated in a file ns-lib.tcl which performs the intergration. The code for this file is found under appendix B, code listing 4. This file is however truncated for editing purposes. Key highlights are as follows;

- `$ns trace-all file-des` – traces and records all simulation events
- `proc finish { }` – terminates the simulation
- `set n[0] [$ns node]` – sets up the nodes (in this case 30 nodes)
- `$ns duplex-link nodex nodey bandwidth` – kicks off a duplex communication
- `set tcp [new Agent/TCP]` – enables TCP communication on object instances
- `$ns attach-agent` – is a member function that matches traffic to objects

- `$ns connect` – establishes a virtual logical connection between two or more objects.

The Script for the fused IDS Model for MANET Topology in Figure 19 is presented in appendix B, code listing 3. There are important additions to the file `ns-lib.tcl` as indicated below; Line 128 to 134 enables data logged by the member function `$ns trace-all file-des` to be fed into the SVM machine for purposes of classification. The populates and pipes the results into the `port_file`, if it does not exist, one is created.

128.set flag 1

129.} elseif {[file exists \$PORT_FILE_] && [file readable \$PORT_FILE_]} {

130.for kernel in kernels:

131. svc = svm.SVC(kernel=kernel).fit(X, y)

132. plotSVC('kernel=' + str(kernel))

133.} else {

134.set flag 1

Further, an addition to the IDS script enabled pattern recognition on the interest data. This was done by intergrating an artificial nueral network technique for purposes of performing pattern recognition as shown in line 184 to 189 of code snippet 4, appendix B

```

184. INT n,i,j;

185. for (n=0; n<NUM_DATA; n++) {

186.   for (i=0; i<Y; i++) {

187.     for (j=0; j<X; j++) {

188.       Input[n][i*X+j] = (Pattern[n][i][j] == 'O') ? HI : LO;

189.     }

```

In addition, any resultant data reviewed for pattern recognition that was a true positive was forwarded to the alarm as show in line 635 to 638 of code snippet 4, appendix b

```

635.$self notifyObservers $now

636.$self instvar netAddress

637.if ![info exists netAddress] {

638.set netAddress [new Address]

```

4.3.3. Implementation of MANET IDS prototype on Raspberry Pi and Generic Smart-watch

This section presents the findings of implementation of the MANET anomaly-based intrusion detection model for smart health care prototype using a live experiment via a Proof of Concept methodology. The project prototype implementation involved setting up of a live MANET with a smart-watch with blood pressure monitoring capability, which forwarded this data to a fully function device - Bluetooth Router configured on a Raspberry Pi.

The prototype was implemented to measure and monitor blood pressure and forward the same readings through the MANET ecosystem. Blood pressure measurements were carried out through smart-watches wrapped on the wrist. The generic smart-watch was used to collect data in the form of blood pressure and propagated this data to the fully functional device (or Bluetooth router).

The fused IDS was installed on the Raspberry Pi and dynamically monitored the system and users' actions in the system so as to detect intrusions. As the results show, the model successfully protected against blackhole attacks, brute force and TCP SYN based denial of service among other DDOS attacks. The image below shows the Bluetooth smart-watch on Figure 20 and 21 that collected and forwarded client data.



Figure 20: Smart Watch Physical Address



Figure 21: Smart Watch Collecting Readings

Figure 20 above shows the Bluetooth smart-watch with a physical MAC address of A4:C1:3C:EB:07:DB as the address which propagated data to the router. Figure 21 above shows the smart-watch initiating reading of blood pressure on a user.

The generic smart-watch acted as the reduced function device on the edge of the network. The smart-watch collected data in the form of blood pressure and propagated this data to the fully functional device (or Bluetooth router). The smart-watch has the following specifications;

- Processor - Nordic-Nrf51822
- Operating System - Proprietary OS
- Display - 0.86 inch OLED

- RAM –
- Battery - 60 mAh Polymer lithium battery Normal use:7days;Standby -15days
- Sensors - Pedometer, Heart Rate Monitor, 6 axis acceleration sensor, Blood Pressure Measurement
- Input voltage: 5V
- Weight: 6.9 grams net weight
- Heart rate: S7000
- BT: BT 4.0 /compatible (android and IOS)
- Touch: touch mode button
- Bluetooth- 4.0BLE forward compatible (Low power consumption)

The client propagates data via Bluetooth 4.0 but is forward compatible with Bluetooth 4.5 as well.

Figure 22 below shows the Raspberry Pi which acted as the Bluetooth MANET fully functional device that collected and forwarded client data from the MANET clients for purposes of evaluation on the IDS system. The IDS system was uploaded on the single board computing device since the smart-watches run a proprietary operating system. This is the newest device as at the research date, which runs on a 1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT). Figure 22 below shows the Raspberry Pi device used in the setup.



Figure 22: Raspberry Pi 3 Model B+ (Source: Buy a Raspberry Pi 3 Model B – Raspberry Pi. (n.d.). Retrieved from <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>)

The Raspberry Pi 3 Model B+ has the following specifications;

- SOC: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC.
- CPU: 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU.
- RAM: 1GB LPDDR2 SDRAM.
- WIFI: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 4.2.
- Ethernet: Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB mass-storage booting.
- Thermal management

- Video: Yes – VideoCore IV 3D. Full-size HDMI.
- Audio: Yes.
- USB 2.0: 4 ports
- GPIO: 40-pin
- Power: 5V/2.5A DC power input
- Operating system support: Linux and Unix

The figure below shows the 16Gb micro sd card on which the applications, scripts and IDS was pre-installed so as to run on the Raspberry that doesn't have enough storage.

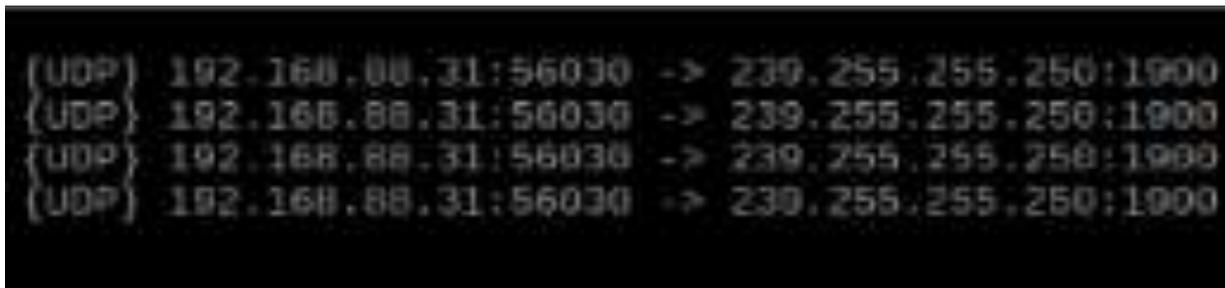
4.3.3.1. Equipment Configuration, Setup and Data Propagation

The python code in appendix C code listing 1, connects to the smart watches which were used in this experiment to form a MANET network. The python modules needed to run the IDS are first imported in line 1 to line 4 of the appendix C, code listing 1. The imported modules include: the subprocess module to run the hciconfig tool and the gat tool as well as the threading module which allows multiple connections. The code scans for bluetooth devices for ten seconds, then connects to the found devices and turns the blood pressure rate notifications on to get notified when the blood pressure changes. The IDS mainly uses linux 'hciconfig' to search for bluetooth devices and gat tool to connect and interact with them. This is shown in the code snippet below.

- *import subprocess*

- *from subprocess import **
- *import time*
- *import threading*

As a result of this setup, the data being propagated on the MANET is visible when scanned, as shown in Figure 23 below. The captured Raspberry Pi interface depicts a normal UDP communication between source address 192.168.88.31 forwarding packets to 239.255.255.250, the MANET network is propagating data in a normal case scenario as depicted in the Finite State Machine (1) without any attack emanating within the network.



```
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
```

Figure 23: Snort analyzer sniffing packets on the MANET

Figure 23 above shows normal packet flow within the MANET using Snort. Snort is free and open source network analysis module that has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching. In this case the home address of 192.168.88.31 was captured with a special analysis on port 56030 for both reliable TCP and unreliable UDP packets on the MANET.

4.3.3.2. Python Code to Capture Usage, RAM and Network Bandwidth on the MANET Topology

This section presents the results of monitoring device activity as a contributor to anomalous activity. A MANET node undergoing heavy attack, can receive tonnes of requests which as a consequence causes the device to overuse its resources – CPU, RAM and bandwidth. These activities on the MANET device itself can be monitored to establish where a device is busy or is overusing its resources in response to an attack. The code lines 13 to 33 on code listing 2, appendix C was written in Python to capture amount of data propagated between the devices, RAM, CPU usage as well as bandwidth and throughput on the network. Special attention is brought to line 21 and 22 specifically monitor the amounts of packets uploaded and downloaded respectively. This is shown in the code snippet below.

```
21.    upload=psutil.net_io_counters(pernic=True)[network_interface][0]
```

```
22.    download=psutil.net_io_counters(pernic=True)[network_interface][1]
```

This activity is thereafter evaluated against what is considered normal and/or anomalous so as ascertain whether the activity is potentially malicious or outright malicious.

The outcome of this section was data recordings on system and process utilities usage by the MANET device. The logged file contained data showing CPU, RAM and Bandwidth usage which can be monitored to identify a pattern of high resource usage that is tantamount to a node undergoing heavy attack.

4.3.3.3. Machine Learning using Support Vector Machines: Data importation capture and Separation.

This data importation capture and separation module captures general packets and segregates interest data by mapping each type and address accordingly to a higher dimensional feature space. This output is consequently taken as source input for the Support Vector Machine for separation.

To achieve data capture, snort – which has capabilities for sniffing and packet logging was used to capture packets from a manipulated syn flood attack. Figure 24 below shows the MANET under ddos attack, to which the snort was able to detect the anomaly as being a possible TCP DoS attack. This was effected by running LOIC (Low Orbit Ion Canon). LOIC is a free network analysis tool, that is popular for initiating DOS attacks. LOIC tool is freely available on the Internet. LOIC performed the DDoS attack by sending successive SYN requests to addresses on the MANET in an attempt to rid the devices of resources, thus make it unresponsive to legitimate requests. Figure 24 shows DoS recognition on the MANET.

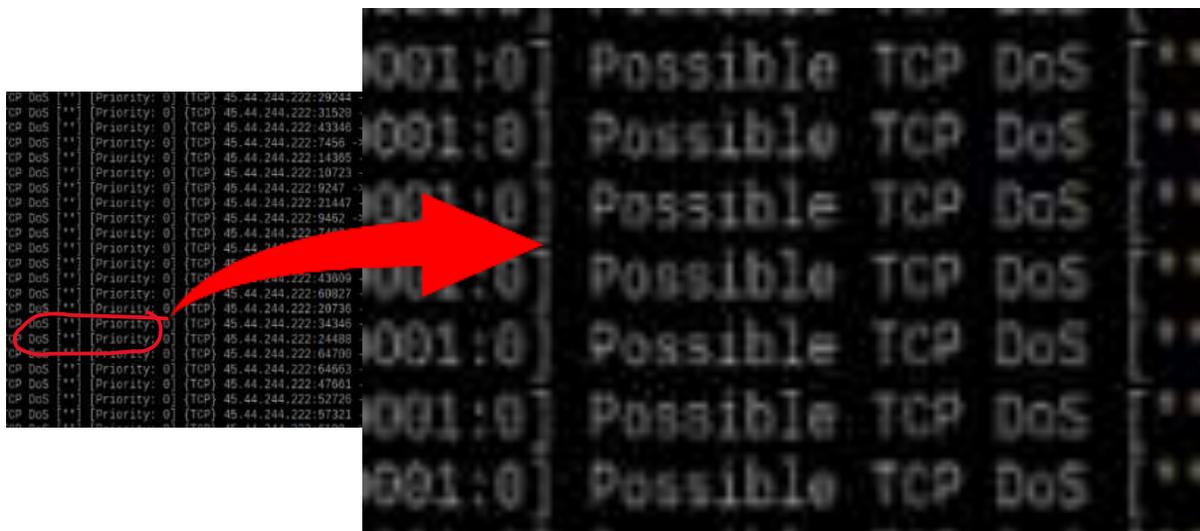


Figure 24: MANET under manipulated DDOS attack

Figure 24 above receives numerous fictitious requests from a public IP 45.44.244.222 attacking various open and accessible ports in the range of 49152-65535 towards 192.168.88.31 on its open http port 80. Most ddos tools listen on the dynamic range of 49152-65535 ports to find available and / or unprotected ports.

Based on combining misuse with anomaly detection, our IDS for MANETs accurately and efficiently detects attacks such as DoS, replay attack and compromised nodes. Our results have shown a great promise for the future, which would focus on making the scheme more robust by taking a broader range of attacks into consideration and making use of audit data to accurately adjust the threshold.

This algorithm provides the ability to perform unsupervised neighbors-based learning techniques. Unsupervised nearest neighbors provides a foundation for various other learning methods. Two techniques are used in particular - manifold learning and spectral clustering. In this particular module data captured from the MANET module is classified by the KNeighborsClassifier and uses the following methods;

- `Fit (X, y)` Uses X as training data and y as target values.
- `get_params ([deep])` Sets the various constraints.
- `Kneighbors ([X, n_neighbors, return_distance])` this calculates the K-neighbors of a point.
- `predict_proba (X)` Computes all the various possibilities for the test data X.

- `score (X, y[, sample_weight])` Computes possible variation from mean accuracy to avoid false alarm

The data logged from the capturing activity can be seen as Data Listing 1 and 2, appendix C. The python code in appendix C code listing 3, presents the intergration of SVM algorithm for data importation capture and separation. This is shown in the code snippet below.

```
10.  #loads the data set

11.  data=[]

12.  fp=open('all_logs','r')

13.  for line in fp.readlines():

    vals = line.strip().split(' ')

    try:

    l_elem = [float(i) if '.' in i else str(i) for i in vals ]

    data.append(l_elem)

    except Exception as e:

    pass

14.  import pandas as pd

15.  import numpy as np

16.  _dataset=np.asarray(data)
```

The python code in appendix C code listing 3, presents the scatter plot generation from the captured data. This is shown in the code snippet below.

```
27. plt.scatter(_dataset[:2591, 0], _dataset[:2591, 2], c='r', label='default')

28. plt.scatter(_dataset[2591+1:2591+2201, 0], _dataset[2591+1:2591+2201, 2],
c='g',label='ddos')

29. plt.scatter(_dataset[2591+2201+1:2591+2201+2155, 0],
_dataset[2591+2201+1:2591+2201+2155, 2], c='b',label='no_IDS')

30. plt.scatter(_dataset[2591+2201+2155+1+1012:, 0],
_dataset[2591+2201+2155+1+1012:, 1], c='yellow',label='norm')
```

4.3.3.4.Machine Learning Implementation: Support Vector Machine for separable problems.

Support Vector Machine algorithms were preferred in this study due to their ability to give very high accuracy in comparison to other classifiers. While logistic regression and decision trees have been implemented before, SVMs were deemed perfect for intrusion detection on the basis of accuracy. The SVM data classifier segregates packets using a hyperplane with the largest amount of margin.

The model performs the following steps to achieve learning from the propagated data;

- Prepare data:

This is performed prior by SVM so as the right and relevant interest data is captured, segregated and analyzed.

- Create an instance of a Linear SVM classifier:

Generate hyperplanes which segregates the interest data according to behavior, rate and type separating the two classes correctly.

- Train a Linear SVM classifier:

This compares the two vectors separated by the decision boundary or hyperplane with the two nearest neighbour packets data points (D+ and D-).

Thereafter, the interest data is input into the training stage so as to enhance the IDS engine's accuracy. This is shown on the code line 25 to 27 from code listing 4, appendix C, where the engine is trained on the behaviors.

25. `train_data,test_data,train_label,test_label = train_test_split(dataset.iloc[:,1], dataset.iloc[:,2], test_size=0.2, random_state=1)`

26. `#the k (n_neighbors) parameter is often an odd number to avoid ties in the voting scores. eg 1-9 neighbors = np.arange(1,9)--has 9 neighbors`

27. `#2 numpy zero matrices namely train_accuracy and test_accuracy each for training and testing accuracy`

SVM achieves training and testing set split by importing sklearn library which has an in-built splitting function called `train_test_split`. This uses the `random_state` as a seed that takes a `random_state` as input. Changing the number of seeds will also affect and change the split of the data. Maintaining the same `random_state` while running the cell multiple times, will ensure the

data splitting remains unchanged. Further, the performance of the training and testing are presented by running the code lines 36 to 39 from code listing 4, appendix C

```
36. knn = KNeighborsClassifier(n_neighbors=11)  
37. knn.fit(train_data, train_label)  
38. train_accuracy[i] = knn.score(train_data, train_label)  
39. test_accuracy[i] = knn.score(test_data, test_label)
```

The output of this section is displayed in the following Figure 26 below which plots the results and correlation between TCP and packet propagation in the MANET. The results indicated here shows that the Fused Model successfully separated interest data as required. The results in Figure 26 also shows how the SVM separted the data packets propagated on the network using color codes where green shows data from source device, blue shows data from the destination device and red data from a malicious device.

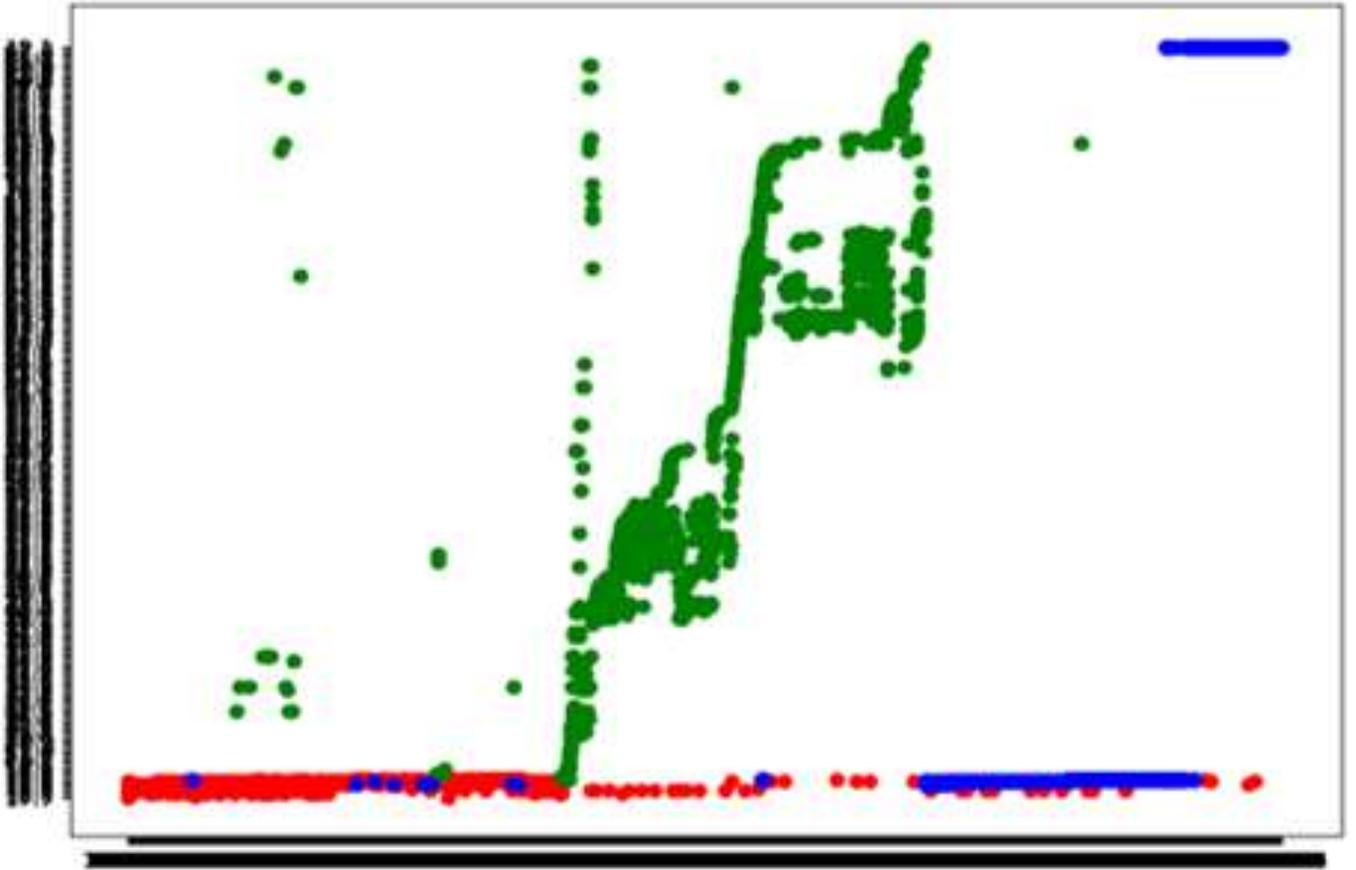


Figure 25: Output

The accuracy of the interest data was evaluated further by feeding it into an artificial neural network and MLP Classifier initiated to perform train/test split and results compared. The results of this test are presented in Figure 27 below that shows the Artificial Neural Network test split. It plots the training and testing accuracy, with accuracy against varying number of neighbor's graph. The results below indicate that the data inherited from the support vector machine (SVM) had negligible deviation. The difference in results between training data and test data is negligible.

The k value from this graph indicates that the fused anomaly model accuracy results between D+ of 0.3 to D- of 0.4. This is impressive accuracy inverse to the the data. The difference between test data and training data is presented in Figure 27. K value refers to the kernel value. The function of kernel is to accept and collect data which is input and thereafter calculate and convert it into the various required format. Different support vector machine algorithms use different types of kernel functions. These kernel functions can ether be linear, nonlinear, polynomial and or radial basis functions.

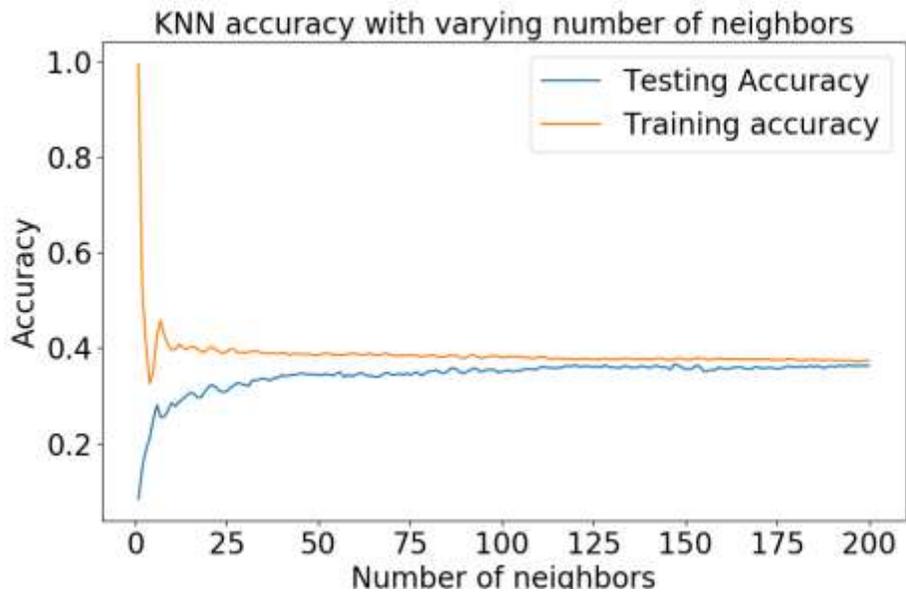


Figure 26: KNN accuracy of Testing and Training Data

KNN refers to k-nearest neighbors (KNN) which is a simple algorithm and easy to run and implement through supervised machine learning. KNN is mainly used to solve both classification and regression problems. In this study data collected from the MANET was fed into KNN through supervised machine learning for purposes classifying the data and segregating interest data from non-interest data. The advantage of KNN is that it responds faster to changes in the data input while in real-time learning. KNN is also quite easy to implement for multiple classified problems or single problems by transitioning to multi class effortlessly. KNN is also very robust to noisy training data and is also very effective in training large amounts of data.

4.3.3.5. Machine Learning Fusion: Support Vector Machine into Artificial Neural Networks

This section involved integrating two machine learning concepts each contributing to the strength of the model.

- The Support Vector Machine, due to its strengths in data classification, identifies interest data and separates the data accordingly.
- This classified data is fed into an artificial neural network that performs pattern recognition tasks.
- The pattern recognition is performed on interest data on both sides of the hyperplane, in accordance with the anomaly data pattern.
- Data packets with anomalous symbol are thereafter passed into the anomaly detection engine.

The resulting positive vector is imported into the Artificial Neural Network algorithms MLPClassifier and hidden_layer_sizes so as to quickly perform function approximation. The ANN module essentially performs three major tasks which include pre-processing already performed by SVM;

- Train Test Split – This is achieved by dividing the achieved vector data to both training and test splits. Training will be achieved by using the training data, and performance of the system tested through the test data.
- Feature Scaling – This will aid to review how the ANN makes predictions when large number of packets, which are anticipated if and when the MANET is propagating huge amounts of packets or when the MANET is experiencing high data rates while maintaining accuracy.
- Alarm - The resulting positive match to anomaly patterns is identified and forwarded for reporting.
- Learning Loop - New found anomalies that have not been experienced but do not satisfy norms are filtered, clustered and forwarded to the ANN for learning and future reference.

The code listing 4, appendix C shows implementation of train test, feature scaling, alarm and learning for ANN in python. This is achieved by importing the class perceptron from scikit and creating a new perceptron to handle the data x and y. This is shown in the truncated code listing 5, appendix C. A snippet is displayed below.

2. *Perceptron(alpha=0.005, class_weight=None, early_stopping=False, eta0=0.1,*
3. *fit_intercept=True, max_iter=30, n_iter=None, n_iter_no_change=2,*
4. *n_jobs=None, penalty=None, random_state=42, shuffle=True, tol=0.001,*
5. *validation_fraction=0.02, warm_start=False*

Thereafter, the an MPL Classifier is imported into the IDS. A multilayer perceptron (MLP) is a feedforward artificial neural network inbuilt function, that maps various sets of input data onto a set of appropriate outputs. This is presented in code listing 5, appendix C and also in the lines 7 to 12 below.

7. *from sklearn.neural_network import MLPClassifier*
8. *X = [[0., 0.], [0., 1.], [1., 0.], [1., 1.]]*
9. *y = [0, 0, 0, 1]*
10. *clf = MLPClassifier(solver='lbfgs', alpha=1e-5,*
11. *hidden_layer_sizes=(5, 2), random_state=1)*
12. *print(clf.fit(X, y))*

To check on the accuracy of the module results, a function to calculate the number of neurons is activated. The purpose is to count the number of neurons between the inputs and the outputs of the ANN. For best accuracy, they should be equal to the two-thirds of the sum.

16. *for i in range(len(clf.coefs_)):*
17. *number_neurons_in_layer = clf.coefs_[i].shape[1]*
18. *for j in range(number_neurons_in_layer):*

The resulting IDS is introduced in the MANET and a DoS is executed to test its ability. Figure 25 below shows introduction of the smart IDS to counter DDOS attacks within the MANET, the results presented here show that is able to detect anomalous events within the MANET.

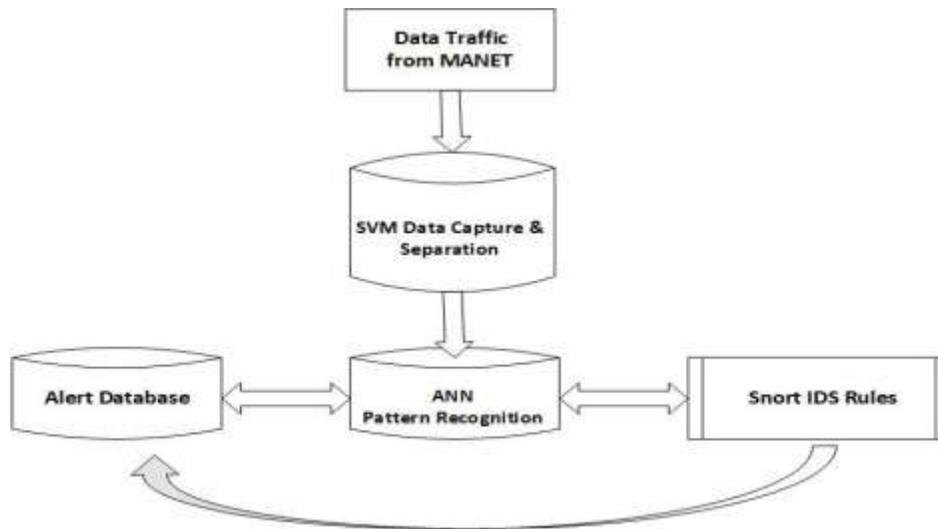


Figure 28: IDS Fusion Framework

The following code files are added to original IDS module:

- src/preprocessors/MLP_ANN.c
- src/preprocessors/MLP_ANN.h

This are MultiLayer Perceptron (MLP) Perceptron neural network source files.

- src/preprocessors/svm_data.c
- src/preprocessors/svm_log.h

This is a preprocessor source file that intergrates the Support Vector Machine SVM data capture and separation techniques.

The following files are edited on the original IDS:

- etc/snort.conf

- src/plugbase.c
- src/preprocessors/flow/flow_callback.c

This integration in detail within the files is indicated below

a. etc/snort.conf

The following values on snort.conf were changed so as to enable portscan use the machine learning SVM techniques for data collection and separation. The source code is found under Appendix D Data Listing 1 from line 100 to 105

```
100 preprocessor portscansvm: ignorebc 1 \  
  
101 analyze_thr_lower 100 \  
  
102 analyze_thr_upper 1600 \  
  
103 sense_level 0.05 \  
  
104 net_topology 0 \  
  
105 log_method 1
```

b. src/plugbase.c

The following values on plugbase.c were added so as to include the machine learning source files added – svm_log.h and MLP_ANN.h. These files invoke a multilayer perceptron which is a class of feedforward artificial neural network that calculates and produces values from a set of

given inputs. The source code is found under Appendix D Data Listing 2 at various lines 54, 66, 154 and 155.

```
54 #include "preprocessors/svm_log.h"  
  
66 #include "preprocessors/MLP_ANN.h"  
  
154 extern PreprocConfigFuncNode *preproc_svm_log;  
  
155 extern PreprocConfigFuncNode *preproc_MLP_ANN;
```

c. src/preprocessors/flow/flow_callback.c

The following values on flow_callback.c were added so as to include the svm and ANN source files added – svm_data.c and MLP_ANN.c. The source code is found under Appendix D Data Listing 3 at lines 265 and 267.

```
265 src/preprocessors/Stream6/svm_data.c,  
  
267 src/preprocessors/Stream6/MLP_ANN.c,
```

The result of this section is a Fused IDS engine which contains the following innovative attributes;

- A preprocessor which separates data using support vector machines as opposed to the original data preprocessor
- An Artificial Neural Network that performs pattern recognition from received portscans captured and separated having common characteristics of anomalous nature.
- Captured and stored dataset of normal and anomalous traffic
- Data sets that have been learned by the ANN during the training time
- Weights that are defined by the simulator using ANN learning function
- A fused IDS that drop packets that meet the anomalous criteria described by snort rules but earned and identified by machine learning techniques.

4.4. Performance of The Fused Anomaly-Based Intrusion Detection Model For MANET In Smart Healthcare.

This section presents the results of objective four of the study which set out to evaluate the performance of the fused machine learning intrusion detection model for the provision of smart health care in MANETS implemented in objective 3.

The Fused Model developed in this study was tested comparatively against three scenarios;

- Performance of the MANET without any protection
- Performance of the MANET while undergoing malicious attacks.
- Performance of the MANET while undergoing malicious attacks and a Fused IDS system introduced into the MANET.

In addition, comparative tests on the MANET network so as to evaluate and review effectiveness and efficiency of the proposed model against MANET which was studied and recorded with the help of network simulator 2 (NS-2.35) on Linux operating system. The movement of presented networks nodes was generated with MANET network. For purposes of successful evaluation, the script on NS2 presented in appendix B listing 1 considered the AODV protocol which is commonly used in MANETS. Three case scenarios were evaluated against which the results were evaluated. For purposes of giving an accurate measure of performance, various metrics were tested. The metrics used to test this performance connotation are presented in the sections that follow.

4.4.1. Network Performance Test Results

This section presents the network performance results. Various network performance metrics were used to assess the network performance. The test involved comparing the network performance for anomalous data without the fused IDS protection and with the Fused IDS protection. The connotation of this test is that at the learning phase of the Fused IDS with DDOS attacks the packets moves slow but after learning the speed improves since the IDS has identified the malicious packets hence the learning curved reduces and as a result the malicious requests which are detected are ignored and the delay decreases. The metrics used to test this performance connotation included: average delay, average Jitter, average network throughput and packet delivery ratio. These are standard network metric tests Mehta and Gupta (2013).

4.4.1.1.The Test Case 1: Average Delay

Average delay between each packet propagated was measured by noting the average normal time taken by a packet in transmission, from the source device to the destination device.

However it is critical to take note of normal causes of delay in networks which include buffering, lining and spread postponements. Other factors depend on the routing protocol chosen, the size of the packet in kilobytes and Direct Sequence Spread Spectrum (DSSS) rate. Thus, average delay would be calculated on purely packet transmission. It is important to note that when the distance increased between transmission and reception, probability of a drop packet is also increased. Quantitatively, the calculation of average delay (D) and total number of packets delivery successfully (n) in this scenario is shown in the equation below.

The results of this test case 1 are presented next.

Expected output: It was expected that as the DDoS packets are introduced in the network the delay in propagation of the normal packets is higher. With introduction of the fused IDS the performance should improve with time as the IDS learns and dispenses the anomolus traffic from flowing through the devices.

Actual observation: The actual output of this test case are shown in Figure 28 below.

The results show the base average delay in end to end transmission. The scale is as follows

Blue - Perfomance of the MANET propagation without Fused IDS

Red - Perfomance of the MANET with Fused IDS

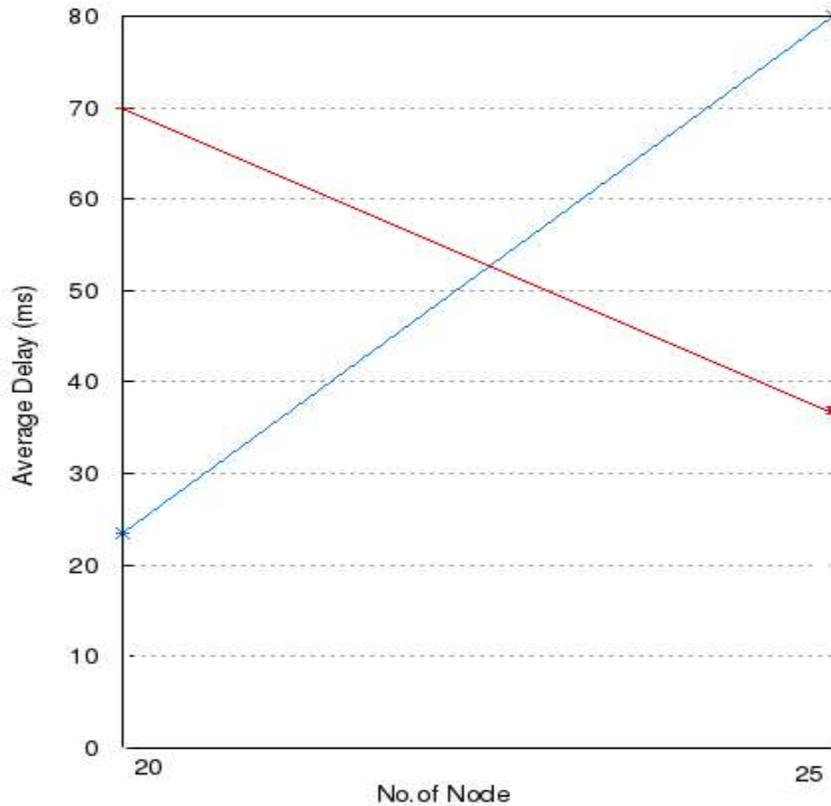


Figure 29: Average delay comparison with Fused IDS and malicious traffic

As observed in the Figure 29, initially the propagation delay for normal packets on the MANET is high. This is because the Fused IDS adds an extra packet handling step that contributes to packet delay. After introducing the Fused IDS, the packet delay improves from 70 milliseconds and reduces to below 40 milliseconds as shown by the red line graph. The delay however without the fused IDS increases as node density increases from 24 to a maximum of 80 milliseconds. After identifying anomolus traffic, the Fused IDS drops these packets from propagating on the network hence improving performance of the MANET.

Non Fused IDS	Fused IDS
Delay increases with increase in Node density from 20 to 80	Delay decreases with increase in Node density from 70 to 38

4.4.1.2. Test Case 2: Average Jitter

Jitter refers to the variation in data flow between two communicating devices. This is normally caused by congestion in a network. Congestion can be as a result of too many devices within a single broadcast domain, broadcast and flooding storms, low bandwidth, old equipment as hubs and repiters. Incorrect configuration or malicious traffic on the network.

In this experiment, it was thus critical to measure jitter. This jitter or variation in delivery of signals can be calculated in terms of varying amplitude, varying phase or simply on the size of the signal pulse. Other sources of jitter include electromagnetic interference and/ or crosstalk with other electromagnetic signals.

Expected output: It was expected that presence of both malicious traffic and an IDS for protection would tremendously increase the jitter within the MANET. This is because, blackhole attacks and DDoS attacks serve to redirect traffic, cause packet loss and generally affect availability of the network. For this experiment, jitter was expected to be below 30 ms. Packet loss was expected to be an less than 1%, while network latency not over 150 ms.

Actual observation: The actual observed results showed average loss of packets to stand at 1.1% while average jitter was 28 % and average latency stood at 98 ms. Normally, jitter below 25ms or at less than 1% of the overall throughput is considered very acceptable especially in

wireless networks. The lost packets could be translated to show that the Fused IDS was actually dropping packets deemed to be anomalous. According to the International Telecommunication Union (ITU), acceptable packet loss of between 1% to 2.5% is acceptable RFC 2436 ISOC/IETF

Packet Loss Data results

Average Loss = 1.1274629%

Average Jitter = 28.3279%

Average Latency = 98 ms

Non Fused IDS	Fused IDS
Packet loss at less than 1%	Packet loss at more than 1.1%

4.4.1.3. Test Case 3: Average network throughput

Average network throughput is a common measurement metric used to gauge the performance of a network. Throughput refers to the actual amount of data that is transferred from one point to another over a given period of time. Throughput is normally measured in bits per second (bits/s) or multiples of it (Kbs, Mbs or GigaBits per second.) gives the results of the actual data that was transferred over a network. Figure 30 below, shows the average network throughput when the attacks were propagated and the IDS was running.

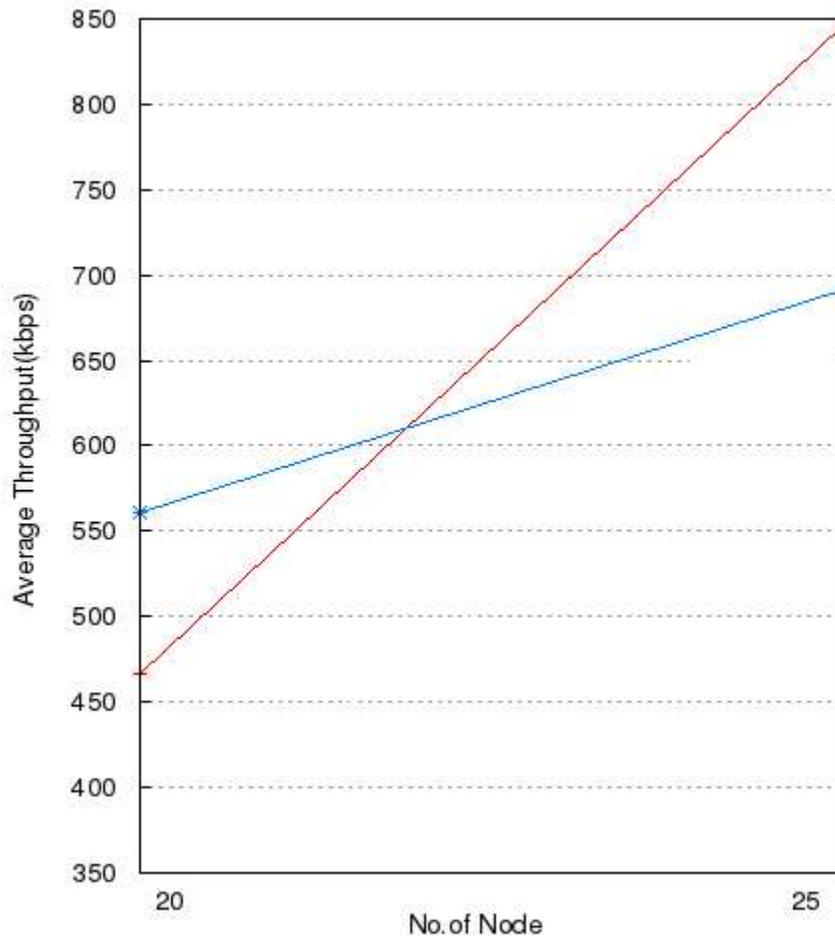


Figure 30: Average throughput on the MANET while under attack with Fused IDS

Expected Output: The expectation in this case is that the average throughput will be affected by both the malicious traffic and IDS. Both malicious traffic and IDS is expected to suppress the amount of data that is transmitted across the MANET from one source to another destination. However, as the Fused IDS learns of the anomalies, the throughput should steadily rise with an increase in throughput for good traffic measured in bits per second.

Actual Observation: There was suppressed throughput as expected. This is due to the degree of congestion within the MANET. From a low of 450 kbits per second, the MANET is able to steadily progress and reach a high of 850 kbits per second. This improved performance is

attributed to the IDS ability to reduce cases of packet loss emanating earlier from the malicious attacks.

Non Fused IDS	Fused IDS
Throughput increase slower from 550 to 700	Throughput increases from 400 to 850

From the average throughput table above, the MANET showed good throughput even when the device density increased.

4.4.1.4. Test Case 4: Packet Delivery Ratio (PDR)

The packet delivery ratio refers to the ratio of total number of packets successfully received at the destination to the total number of packets sent from the source. This defines the ratio of data packets received by the destinations to those generated by the sources. Arithmetically, it can be defined as: $PDR = D1 \div S1$ Where, D1 is the sum of data packets received by each destination device and S1 is the sum of data packets generated by the each source.

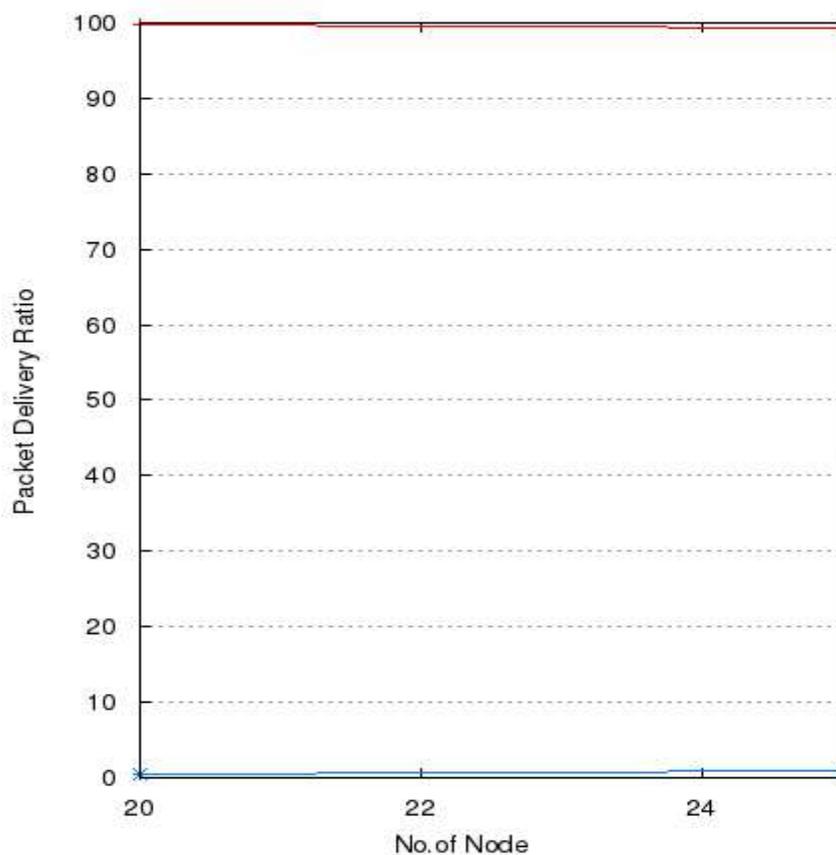


Figure 31: DDOS Packet Delivery Ratio with the Fused IDS

Expected output: DDOS should adversely affect packet delivery as its major attacks looks at limiting availability or response to request(s). In this case scenario, it is expected that the DDOS will affect the delivery ratio by causing non-responses from the nodes, while IDS protected MANETs should have as little DDOS delivery of packets as possible.

Actual observation: The actual observed results showed the Fused IDS affected DDOS packets by dropping all affected traffic with only a delivery ratio of 1.46%. A packet delivery ratio by the attack at less than 2% of the overall total propagation under various heavy attacks is considered good ratio.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1.Introduction

This chapter presents the conclusions and recommendations on the research on the design, implementation and evaluation of a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS. It also provides further areas of study.

5.2.Summary

Intrusion detection is conceptually based in principle either on anomaly detection or misuse detection. Our research involved intrusion detection in wireless mobile ad hoc networks based on integrating misuse detection with anomaly detection. There is a small difference between normalcy and anomaly, in general, and more so, in the case of ad hoc networks, which leads to a high degree of false alarms if only anomaly detection is used. Compound detection brings together the accuracy of misuse detection and the normal profiling of anomaly detection, leading to a powerful audit data analysis for intrusion detection. The focus of this research was on some significant deterministic attacks that can have a damaging effect on the working of a MANET especially in militant or attack scenarios. A compromised node, for instance, can, not only lead to passive eavesdropping but also to active interfering. The routing protocol's correct functioning breaks down if a malicious node launches a replay attack on the system. The bandwidth constraints of a MANET can be exploited by a malicious attacker by launching a Denial-of-Service attack.

5.2.1. Review security weaknesses in the application of MANETs for provision of smart health care.

Research Question 1: What are the security weaknesses encountered in the implementation of secure MANET systems used in provision of smart health care?

This study reviewed the various weaknesses of smart devices in MANETS that are important in providing mission critical support in healthcare. The results showed that their physiognomy exposes various security weaknesses which make these MANETs lack the adequate capability to devise robust systems to shield themselves against eavesdropping, malicious attacks, packet sniffing and other security threats, especially DDOS. While there has been an attempt to implement technological solutions for provision of quality healthcare, there are growing concerns within the facts that, cases of DDOS are on the rapid rise. That smart healthcare experienced an average of 32,000 intrusion attacks per day in 2017 (Adefala, 2018), signals a glaring security hole that should be addressed at the earliest opportunity. The study revealed that Machine Learning based IDS systems that can be efficiently applied, are yet to be implemented in the light of their glaring advantages over competing technologies. The study also exposed possible areas which the IDS system can be improved to provide better security and reduce on consumption of resources such as bandwidth, processor and memory thereby elongating the lifetime of smart healthcare devices.

5.2.2. Design a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

Research Question 2: How can a fused anomaly-based intrusion detection model for MANET in smart healthcare be designed?

This study implemented the model design by taking recommendations from the literature review. As a result, the design was premised on the ability of the MANET to identify intrusions by taking cognizance of unfamiliar device addresses, not permitting TCP sessions that are initiated by devices outside its network to get into fruition and also identify scenarios that cause high resource usage in COU, Ram and bandwidth within the ecosystem as well. To achieve this, three designs were resultant. A logical topology design which was a refined conceptual framework using functional decomposition methodology, to present the logical interconnection. A model of logical events which described the various MANET activities considered as normal and those considered anomalous. This was achieved using PPDIIO methodology. In addition, a network connectivity model, which was designed using PPDIIO methodology which merged all the three arms of the design.

5.2.3. Implementation of a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

Research Question 3: How can a fused anomaly-based intrusion detection model for MANET in smart healthcare be implemented?

This study implemented the model through project research triangulation. This involved setting up two experiment scenarios, in using a simulated experiment and a live prototype

experiment. The virtual simulation was achieved using NS2 in Linux and the live experiment was achieved by using a generic smartwatch and raspberry pi for purposes of setting up the dummy MANET for smart healthcare. Both setups experienced an induced DDoS attack that further worked to prove the MANETs weaknesses. In the experiments above, a fused IDS was also implemented so as to review the ability to protect the MANET successfully. The actual practical model should also look as implementing more advanced and powerful smart devices ie smartwatches, with a view of increasing data redundancy and reducing data handling capability.

The implementation involved editing open source intrusion detection system based on snort rules and infusing two attributes of machine learning. This was achieved by introducing following code files are added to original IDS module; namely the `src/preprocessors/MLP_ANN.c` and `src/preprocessors/svm_data.c`. These are MultiLayer Perceptron (MLP) Perceptron neural network source files as well as the `src/preprocessors/svm_log.h` and `src/preprocessors/MLP_ANN.h` header files. This is a preprocessor source file that intergrates the Support Vector Machine SVM data capture and separation techniques. By configuring the `snort.conf` file and `flow_callback.c` file, these extra files were intergrated into the Fused IDS.

5.2.4. Evaluation of the Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

Research Question 4: What is the performance of a fused anomaly-based intrusion detection model for MANET in smart healthcare?

Three case scenarios were reviewed against which the results were evaluated. The metrics used to test this performance connotation were average delay, average Jitter, average network throughput and packet delivery ratio. The MANETs proved to suffer from congestion especially on the onset, due to high propagation of malicious packets and also due to the fused IDS when implemented inline, adds a time-consuming packet handling functionality that contributes to packet delay.

5.3 Recommendations

The dynamism of MANETs is such that quite a wide and varying amount of research needs to be taken especially in security of MANETs. Various weaknesses in the model require to be further interrogated in future research, primarily; scaled number of devices and electromagnetic interference.

5.4 Policy Recommendations

There is a great need to look at improving packet delay when inline IDS systems are deployed especially in MANETs with low data capacity. Inline IDS systems add an extra packet handling function that causes packet delay and can lead to congestion and loss of data. Instances where the MANET device is required for realtime health monitoring, such delays can inhibit device, system and data availability with possibility of the resulting system being totally unacceptable for health use cases.

5.5 Recommendations for Future Research

The integration of an IDS systems and other significant software applications onto healthcare MANET device firmware is critical. This repository is critical mostly because available MANET devices in healthcare run on proprietary software thus advancements in security are not shared

with the developer community and vendor communities as well. This sharing of applications will accelerate improvements in MANET application development and as well accelerate its adoption in healthcare.

What could be other technological factors affecting the adoption of MANETs in Healthcare? Is there need for user and vendor education towards the usage and applications of these devices that are so critical in Healthcare monitoring?

- i) Is the user experience (UX) design for these devices' undesirable?
- ii) Are users apprehensive about Confidentiality of their Data?
- iii) Does rapid power consumption in MANETs limit user adoption?
- iv) Does multiplicity of other smart devices in phones, laptops, ipods owned by users make MANET usage cumbersome?

There is need to investigate on introduction of key assignment and key revocation during device network hopping. This should reduce burden of trusted node identification as well as curb rouge nodes intergration into vulnerable networks. There is also need to investigate on renewal and expiry of certificates to avoid re-use and abuse.

There is need to perform further research on the suitability of this intrusion detection approach for source-initiated on-demand routing protocols. In addition, it is important to analyze the impact of non-deterministic attacks, which require statistical analysis. Further, collection of mobile nodes working in tandem to launch an attack, which essentially involves study of Byzantine failure is a still an open threat in MANETS that needs further research.

REFERENCES

- Abdelhaq, M., Alsaqour, R., & Abdelhaq, S. (2015). Securing mobile ad hoc networks using danger theory-based artificial immune algorithm. *PloS one*, *10*(5), e0120715.
- Aburukba, R. O., Al-Ali, A. R., Landolsi, T., Rashid, M., & Hassan, R. (2016, May). MANET based energy management for residential area. In Consumer Electronics-Taiwan (ICCE-TW), 2016 *IEEE International Conference on* (pp. 1-2). IEEE.
- Accorsi, R., Lehmann, A., & Lohmann, N. (2015). *Information leak detection in business process models: Theory, application, and tool support. Information Systems*, *47*, 244-257.
- Adefala., L (2018, March 06). *Healthcare Experiences Twice the Number of Cyber Attacks As Other Industries*. Fortinet Labs. Retrieved from <https://www.fortinet.com/blog/business-and-technology/healthcare-experiences-twice-the-number-of-cyber-attacks-as-othe.html>
- Agrawal, S., & Vieira, D. (2013). A survey on Internet of Things-DOI 10.5752/P. 2316-9451.2013 v1n2p78. *Abakós*, *1*(2), 78-95.
- Aidoo, R., & Hess, S. (2015). Non-interference 2.0: China's evolving foreign policy towards a changing Africa. *Journal of Current Chinese Affairs*, *44*(1), 107-139.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347-2376.
- Aldaej, A. (2019). Enhancing Cyber Security in Modern Internet of things (IoT) Using Intrusion Prevention Algorithm for IoT (IPAI). *IEEE Access*.
- Al-Khaldi, M. (2017). A highly compact multiband antenna for Bluetooth/WLAN, WiMAX, and Wi-Fi applications. *Microwave and Optical Technology Letters*, *59*(1), 77-80.

- Almusallam, N. Y., Tari, Z., Bertok, P., & Zomaya, A. Y. (2017). *Dimensionality Reduction for Intrusion Detection Systems in Multi-data Streams—A Review and Proposal of Unsupervised Feature Selection Scheme*. In *Emergent Computation* (pp. 467-487). Springer International Publishing.
- Alam, T. (2019). *Middleware Implementation in Cloud-MANET Mobility Model for Internet of Smart Devices*. *arXiv preprint arXiv:1902.09744*
- Amalfitano, D., Fasolino, A. R., Tramontana, P., Ta, B. D., & Memon, A. M. (2014). *MobiGUITAR: Automated model-based testing of mobile apps*. *IEEE software*, 32(5), 53-59.
- Amiri, E., Keshavarz, H., Heidari, H., Mohamadi, E., & Moradzadeh, H. (2013). *Intrusion detection systems in MANET: a review*. *Procedia-Social and Behavioral Sciences*, 129, 453-459
- Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., & Adeli, H. (2017). *Current efforts for prediction and assessment of natural disasters: Earthquakes, tsunamis, volcanic eruptions, hurricanes, tornados, and floods*. *Scientia Iranica*, 24(6), 2645-2664.
- Amouri, A., Alaparthi, V. T., & Morgera, S. D. (2018, April). *Cross layer-based intrusion detection based on network behavior for IoT*. In *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)* (pp. 1-4). IEEE.
- Arapi, K. (2018). *The Healthcare Industry: Evolving Cyber Threats and Risks* (Doctoral dissertation, Utica College).
- Atre, A., & Singh, R. (2016). *A Concept on Intrusion Detection System Genetic Algorithm, Fuzzy Logic and Challenges—A Review*. *International Journal of Scientific Research in Science, Engineering and Technology*, 2(1), 287-89.

- Azmoodeh, A., Dehghantanha, A., Conti, M., & Choo, K. K. R. (2018). Detecting cryptoransomware in IoT networks based on energy consumption footprint. *Journal of Ambient Intelligence and Humanized Computing*, 9(4), 1141-1152.
- Balakrishna, T., & Swetha, R. N. (2016). development of arm7 based sensor interface for industrial wireless Sensor network (wsn) in iot environment. *Int. J. EMINENT Eng. Technol. IJOEET*, 4, 59-67.
- Barricelli, B. R., & Valtolina, S. (2015, May). *Designing for end-user development in the internet of things*. In *International Symposium on End User Development* (pp. 9-24). Springer, Cham.
- Barani, F., & Gerami, S. (2013, August). ManetSVM: Dynamic anomaly detection using one-class support vector machine in MANETs. In *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)* (pp. 1-6). IEEE.
- Bann, L. L., Singh, M. M., & Samsudin, A. (2015). Trusted Security Policies for Tackling Advanced Persistent Threat via Spear Phishing in BYOD Environment. *Procedia Computer Science*, 72, 129-136.
- Baek, J., Vu, Q. H., Liu, J. K., Huang, X., & Xiang, Y. (2014). A secure cloud computing based framework for big data information management of smart grid. *IEEE transactions on cloud computing*, 3(2), 233-244.
- Bakour, K., Daş, G. S., & Ünver, H. M. (2017, October). An intrusion detection system based on a hybrid Tabu-genetic algorithm. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 215-220). IEEE.

- Baron, L., Klacza, R., Rahman, M. Y., Scognamiglio, C., Friedman, T., Fdida, S., & Saint-Marcel, F. (2016, April). OneLab: On-demand deployment of MANET over IPv6 Infrastructure as a service for IEEE INFOCOM community. In IEEE Infocom 2016.
- Bartoli, A., Dohler, M., Hernández-Serrano, J., Kountouris, A., & Barthel, D. (2011, May). Low-power low-rate goes long-range: The case for secure and cooperative machine-to-machine communications. In *International Conference on Research in Networking* (pp. 219-230). Springer, Berlin, Heidelberg.
- Batalla, J. M., Mastorakis, G., Mavromoustakis, C. X., & Zurek, J. (2016). On cohabitating networking technologies with common wireless access for home automation system purposes. *IEEE Wireless Communications*, 23(5), 76-83.
- Bhuvaneswari, A. (2017). A Survey on Internet of Things [MANET]. *International Journal of Advanced Research in Computer Science*, 8(1).
- Bedi, G., Venayagamoorthy, G. K., & Singh, R. (2016). Internet of Things (MANET) sensors for smart home electric energy usage management. In *Information and Automation for Sustainability (ICIAfS), 2016 IEEE International Conference on* (pp. 1-6). IEEE
- Berger, J. L., PiccMANETto, J., Woodward, J. P. L., & Cummings, P. T. (1990). Compartmented mode workstation: Prototype highlights. *IEEE Transactions on Software Engineering*, 16(6), 608-618.
- Berthier, R., & Sanders, W. H. (2011). Specification-based intrusion detection for advanced metering infrastructures. In *2011 IEEE 17th Pacific Rim International Symposium on Dependable Computing* (pp. 184-193). IEEE.

- Bhattacharjee, P. S., Fujail, A. K. M., & Begum, S. A. (2017). Intrusion Detection System for NSL-KDD Data Set using Vectorised Fitness Function in Genetic Algorithm. *Advances in Computational Sciences and Technology*, 10(2), 235-246.
- Benkhelifa, E., Welsh, T., & Hamouda, W. (2018). A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. *IEEE Communications Surveys & Tutorials*, 20(4), 3496-3509.
- Benz, H. L., Yao, J., Rose, L., Olgac, O., Kreutz, K., Saha, A., & Civillico, E. F. (2016, August). Upper extremity prosthesis user perspectives on unmet needs and innovative technology. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 287-290). IEEE.
- Borgohain, T., Kumar, U., & Sanyal, S. (2015). Survey of security and privacy issues of internet of things. *arXiv preprint arXiv:1501.02211*.
- Bertino, E. (2014). Data trustworthiness—approaches and research challenges. In *Data privacy management, autonomous spontaneous security, and security assurance* (pp. 17-25). Springer, Cham.
- Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176.
- Butavicius, M., Parsons, K., Pattinson, M., & McCormac, A. (2016). Breaching the human firewall: Social engineering in phishing and spear-phishing emails. *arXiv preprint arXiv:1606.00887*.

- Carvalho, J. Á. (2019). Research and practice in IS: insights from medicine that might contribute to overcoming the relevance deficit in the IS domain. *SciKA-Association for Promotion and Dissemination of Scientific Knowledge*.
- Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., & Tarricone, L. (2015). An MANET-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2(6), 515-526.
- Cao, Z., Chung, Y. W., Xiong, Y., Chu, C. C., & Gadh, R. (2016, November). MANET based manufacturing system with a focus on energy efficiency. In *Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, 2016 IEEE (pp. 545-552). IEEE.
- Cao, T. D., Hoang, H. H., Huynh, H. X., Nguyen, B. M., Pham, T. V., Tran-Minh, Q., ... & Truong, H. L. (2016). MANET Services for Solving Critical Problems in Vietnam: A Research Landscape and Directions. *IEEE Internet Computing*, 20(5), 76-81.
- Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., & Tarricone, L. (2015). An MANET-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2(6), 515-526.
- Capraro, G. T. (2016, January). Artificial Intelligence (AI), Big Data, and Healthcare. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (p. 425). *The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*.
- Calic, D., Pattinson, M. R., Parsons, K., Butavicius, M. A., & McCormac, A. (2016). Naïve and Accidental Behaviours that Compromise Information Security: What the Experts Think. *In HAISA* (pp. 12-21).

- Chan, C. H., & Fu, K. W. (2018). The “mutual ignoring” mechanism of cyberbalkanization: triangulating observational data analysis and agent-based modeling. *Journal of Information Technology & Politics*, 15(4), 378-387.
- Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, 80, 340-355.
- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems* (pp. 577-585).
- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A recurrent latent variable model for sequential data. In *Advances in neural information processing systems* (pp. 2980-2988).
- Celesti, A., Fazio, M., Longo, F., Merlino, G., & Puliafito, A. (2017). Secure Registration and Remote Attestation of MANET Devices Joining the Cloud: The Stack4Things Case of Study. *Security and Privacy in Cyber-Physical Systems: Foundations, Principles, and Applications*.
- Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the MANET and smart city scenarios. *IEEE Wireless Communications*, 23(5), 60-67.
- Christopoulos, K., Spournias, A., Orfanoudakis, T., Antonopoulos, C., & Voros, N. (2016, November). Designing the Next Generation of Home Automation Combining MANET

- and Robotic Technologies. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics* (p. 10). ACM.
- Chae, Y., Katenka, N., & DiPippo, L. (2016, December). Adaptive Threshold Selection for Trust-based Detection Systems. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on* (pp. 281-287). IEEE.
- Chen, M., Qian, Y., Mao, S., Tang, W., & Yang, X. (2016). Software-defined mobile networks security. *Mobile Networks and Applications*, 21(5), 729-743.
- Chan, G. Y., Chua, F. F., & Lee, C. S. (2016). Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns. *Journal of Intelligent & Fuzzy Systems*, 31(2), 749-764.
- Chaudhary, A., & Shrimal, G. (2019). Intrusion Detection System Based on Genetic Algorithm for Detection of Distribution Denial of Service Attacks in MANETs. *Available at SSRN 3351807*.
- Ciocarlie, G. F., Stavrou, A., Stolfo, S. J., & Keromytis, A. D. (2019). *U.S. Patent Application No. 10/178,113*.
- CISCO. (2011). CISCO's PPDIOO Network Cycle.
<http://www.ciscopress.com/articles/article.asp?p=1697888&seqNum=2>
- Dalli, A., & Bri, S. (2016, November). Design of Electronic Ticket System for Smart Tourism. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2016 12th International Conference on* (pp. 490-492). IEEE.

- Das, R., Tuna, A., Demirel, S., NETAS AS, K., & Yurdakul, M. K. (2017). A Survey on the Internet of Things Solutions for the Elderly and Disabled: Applications, Prospects, and Challenges. *International Journal of Computer Networks and Applications (IJCNA)*.
- Das, S., & Pal, S. (2019). *Analysis of Energy-Efficient Routing Protocols in Mobile Ad Hoc Network*. In *Advances in Computer, Communication and Control* (pp. 285-295). Springer, Singapore.
- Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis*, 26(2), 168-189.
- Delkesh, T., & Jamali, M. A. J. (2019). EAODV: Detection and removal of multiple black hole attacks through sending forged packets in MANETs. *Journal of Ambient Intelligence and Humanized Computing*, 10(5), 1897-1914.
- Dhindsa, K. S., & Bhushan, B. (2019). Flow-based Attack Detection and Defense Scheme against DDoS Attacks in Cluster based Ad Hoc Networks. *International Journal of Advanced Networking and Applications*, 10(4), 3905-3910.
- Domingo, A., & Wietgreffe, H. (2015, October). An applied model for secure information release between federated military and non-military networks. In *Military Communications Conference, MILCOM 2015-2015 IEEE* (pp. 465-470). IEEE.
- Dorri, A., Kamel, S. R., & Kheirkhah, E. (2015). Security challenges in mobile ad hoc networks: A survey. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 6(1), 15-29

- Doss, S., Nayyar, A., Suseendran, G., Tanwar, S., Khanna, A., & Thong, P. H. (2018). APD-JFAD: Accurate prevention and detection of Jelly Fish attack in MANET. *Ieee Access*, 6, 56954-56965.
- Duan, R., Chen, X., & Xing, T. (2011, October). A QoS architecture for MANET. In *Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing* (pp. 717-720). IEEE.
- El Hassani, A. A., El Kalam, A. A., Bouhoula, A., Abassi, R., & Ouahman, A. A. (2015). Integrity-OrBAC: a new model to preserve Critical Infrastructures integrity. *International Journal of Information Security*, 14(4), 367-385.
- Elias, A. R., Golubovic, N., Krintz, C., & Wolski, R. (2017, April). Where's the Bear?-Automating Wildlife Image Processing Using MANET and Edge Cloud Systems. In *Internet-of-Things Design and Implementation (MANETDI), 2017 IEEE/ACM Second International Conference on* (pp. 247-258). IEEE.
- Elhag, S., Fernández, A., Alshomrani, S., & Herrera, F. (2019). *Evolutionary Fuzzy Systems: A Case Study for Intrusion Detection Systems. In Evolutionary and Swarm Intelligence Algorithms* (pp. 169-190). Springer, Cham.
- El Naqa, I., & Murphy, M. J. (2015). What is machine learning?. In *Machine Learning in Radiation Oncology* (pp. 3-11). Springer, Cham.
- Elsokah, M. M., & Zerek, A. R. (2019, March). Next Generation of Medical Care Bed with Internet of Things Solutions. In *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* (pp. 84-89). IEEE.
- Eddy, S. R. (2004). What is a hidden Markov model?. *Nature biotechnology*, 22(10), 1315.

- Elwahsh, H., Gamal, M., Salama, A. A., & El-Henawy, I. M. (2018). A Novel Approach for Classifying MANETs Attacks with a Neutrosophic Intelligent System based on Genetic Algorithm. *Security and Communication Networks*, 2018.
- Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008, June). The pothole patrol: using a mobile sensor network for road surface monitoring. *In Proceedings of the 6th international conference on Mobile systems, applications, and services* (pp. 29-39). ACM.
- Esfandeh, T., Kwon, C., & Batta, R. (2016). Regulating hazardous materials transportation by dual toll pricing. *Transportation Research Part B: Methodological*, 83, 20-35.
- Elizabeth, B. L., Aaishwarya, R., Kiruthika, P., Shrada, M. N., Prakash, A. J., & Uthariaraj, V. R. (2011, June). Bayesian based confidence model for trust inference in MANETs. *In 2011 International Conference on Recent Trends in Information Technology (ICRTIT)* (pp. 402-406). IEEE
- Fatima, M., Bandopadhyay, T. K., & Gupta, R. (2019). *Unconventional Prediction Algorithm for Quick Route Convergence and Stability in MANET. In Computing, Communication and Signal Processing* (pp. 409-418). Springer, Singapore.
- Fernandez, F., & Pallis, G. C. (2014, November). Opportunities and challenges of the Internet of Things for healthcare: Systems engineering perspective. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on* (pp. 263-266). IEEE.
- Fidalcastro, A., & Baburaj, E. (2017). Sequential Pattern Mining for Intrusion Detection System with Feature Selection on Big Data. *KSII Transactions on Internet & Information Systems*, 11(10).

- Fiedler, J., Kupka, T., Ehlert, S., Magedanz, T., & Sisalem, D. (2007, July). VoIP defender: highly scalable SIP-based security architecture. *In Proceedings of the 1st international conference on Principles, systems and applications of IP telecommunications* (pp. 11-17). ACM.
- Flammini, F., Gaglione, A., Ottello, F., Pappalardo, A., Pragliola, C., & Tedesco, A. (2010, October). Towards wireless sensor networks for railway infrastructure monitoring. *In Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS), 2010* (pp. 1-6). IEEE.
- Folorunso, O., Ayo, F. E., & Babalola, Y. E. (2016). Ca-NIDS: A network intrusion detection system using combinatorial algorithm approach. *Journal of Information Privacy and Security*, 12(4), 181-196.
- Ford, M., Mallery, C., Palmasani, F., Rabb, M., Turner, R., Soles, L., & Snider, D. (2016, March). A process to transfer Fail2ban data to an adaptive enterprise intrusion detection and prevention system. *In SoutheastCon, 2016* (pp. 1-4). IEEE.
- Foradis, T., & Thramboulidis, K. (2017). From Mechatronic Components to Industrial Automation Things: An MANET Model for Cyber-Physical Manufacturing Systems. *Journal of Software Engineering and Applications*, 10(08), 734.
- Frederix, I. (2009, May). Internet of things and radio frequency identification in care taking, facts and privacy challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on* (pp. 319-323). IEEE.
- Freitas, J., Teixeira, A., Dias, M. S., & Silva, S. (2017). *Introduction. In An Introduction to Silent Speech Interfaces* (pp. 1-14). Springer International Publishing.

- Fu, K., Kohno, T., Lopresti, D., Mynatt, E., Nahrstedt, K., Patel, S., ...& Zorn, B. (2017). Safety, Security, and Privacy Threats Posed by Accelerating Trends in the Internet of Things. Technical Report. *Computing Community Consortium*. <http://cra.org/ccc/wp-content/uploads/sites/2/2017/02/Safety-Security-and-Privacy-Threats-in-MANET.pdf>.
- Gade, N. R., Gade, N. R., & Reddy, G. U. (2016). Internet of Things (MANET) for Smart Cities-The Future Technology Revolution. *Global Journal of Computer Science and Technology*, 16(1).
- Gai, K., Qiu, M., Tao, L., & Zhu, Y. (2016). Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. *Security and Communication Networks*, 9(16), 3049-3058.
- Gammar, S. M., Amine, E., & Kamoun, F. (2010). Distributed address auto configuration protocol for Manet networks. *Telecommunication Systems*, 44(1-2), 39-48.
- Ganichev, I., Zhang, R., Koponen, T., Dubovik, G., & THAKKAR, P. (2018). *U.S. Patent Application No. 10/110,431*.
- Garikipati, V., & Rao, N. N. M. (2019). *Secured Cluster-Based Distributed Fault Diagnosis Routing for MANET*. In *Soft Computing and Signal Processing* (pp. 35-51). Springer, Singapore.
- Gartner News (2019, February 15). Gartner Says a Typical Family Home Could Contain More Than 500 Smart Devices by 2022.. Retrieved November 6, 2019, from <https://www.gartner.com/en/newsroom/press-releases/2014-09-08-gartner-says-a-typical-family-home-could-contain-more-than-500-smart-devices-by-2022>.

- Garcia-de-Prado, A., Ortiz, G., & Boubeta-Puig, J. (2017). COLLECT: COLLaborative ConText-aware service oriented architecture for intelligent decision-making in the Internet of Things. *Expert Systems with Applications*, 85, 231-248.
- Ghanem, M. C., & Ratnayake, D. N. (2016, June). Enhancing WPA2-PSK four-way handshaking after re-authentication to deal with de-authentication followed by brute-force attack a novel re-authentication protocol. In *Cyber Situational Awareness, Data Analytics And Assessment (CyberSA), 2016 International Conference On* (pp. 1-7). IEEE.
- Ghahramani, Z. (2003, February). Unsupervised learning. In *Summer School on Machine Learning* (pp. 72-112). Springer, Berlin, Heidelberg.
- Ghazvinei, P. T., Shamsirband, S., Motamedi, S., Darvishi, H. H., & Salwana, E. (2017). Performance investigation of the dam intake physical hydraulic model using Support Vector Machine with a discrete wavelet transform algorithm. *Computers and Electronics in Agriculture*, 140, 48-57.
- Gonzalez-Vicente, R. (2015). The limits to China's non-interference foreign policy: pro-state interventionism and the rescaling of economic governance. *Australian Journal of International Affairs*, 69(2), 205-223.
- Gope, P., & Hwang, T. (2016). A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 63(11), 7124-7132.
- Groz, R., Simao, A., Bremond, N., & Oriat, C. (2018, May). Revisiting AI and testing methods to infer FSM models of black-box systems. In *2018 IEEE/ACM 13th International Workshop on Automation of Software Test (AST)* (pp. 16-19). IEEE.

- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (MANET): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (MANET): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.
- Harris, G., & Davies, H. (2019). Simulating Information Retrieval Systems and Functional Decomposition. *International Journal of Software Systems Research and Methodology*, 4(1).
- Hassan, M. H., Mostafa, S. A., Budiyo, A., Mustapha, A., & Gunasekaran, S. S. (2018). A Hybrid Algorithm for Improving the Quality of Service in MANET. *International Journal on Advanced Science, Engineering and Information Technology*, 8(4), 1218-1225.
- Hamza, F., & Vigila, S. M. C. (2019). Review of Machine Learning-Based Intrusion Detection Techniques for MANETs. In *Computing and Network Sustainability* (pp. 367-374). Springer, Singapore.
- Hoffmann, H., & Holland, G. D. (2019). U.S. Patent Application No. 10/187,404.
- Hongach Jr, W. J. (2018). Mitigating Security Flaws in the TCP/IP Protocol Suite (Doctoral dissertation, Utica College).
- Hsiao, S. J., Lian, K. Y., & Sung, W. T. (2016). Employing Cross-Platform Smart Home Control System with MANET Technology Based. In *Computer, Consumer and Control (IS3C), 2016 International Symposium on* (pp. 264-267). IEEE.

- Hu, H., Ahn, G. J., & Kulkarni, K. (2012). Detecting and resolving firewall policy anomalies. *IEEE Transactions on dependable and secure computing*, 9(3), 318-331.
- Huber, S., Seiger, R., Kuehnert, A., & Schlegel, T. (2016, February). Using semantic queries to enable dynamic service invocation for processes in the internet of things. *In Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on* (pp. 214-221). IEEE.
- Hui, K. L., Kim, S. H., & Wang, Q. H. (2017). Cybercrime deterrence and international legislation: Evidence from distributed denial of service attacks. *MIS Quarterly*, 41(2), 497.
- Hwang, G., Park, J., Lee, J., Park, J., Chang, T. W., & Won, J. (2017). Analysis of MANET Usage in Korean Key Manufacturing Industries. *Journal of Society for e-Business Studies*, 21(4).
- Ishaq, I., Carels, D., Teklemariam, G. K., Hoebeke, J., Abeele, F. V. D., Poorter, E. D., ... & Demeester, P. (2013). IETF standardization in the field of the internet of things (MANET): a survey. *Journal of Sensor and Actuator Networks*, 2(2), 235-287.
- Iyengar, A., Kundu, A., & Pallis, G. (2018). Healthcare Informatics and Privacy. *IEEE Internet Computing*, 22(2), 29-31.
- Iqbal, I. M., & Calix, R. A. (2016, October). Analysis of a Payload-based Network Intrusion Detection System Using Pattern Recognition Processors. *In Collaboration Technologies and Systems (CTS), 2016 International Conference on* (pp. 398-403). IEEE.
- Honig, A., Howard, A., Eskin, E., & Stolfo, S. J. (2016). U.S. Patent No. 9,497,203. Washington, DC: *U.S. Patent and Trademark Office*.

- Iqbal, S., Kiah, M. L. M., Dhaghghi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*, 74, 98-120.
- Inzillo, V., Serianni, A., & Quintana, A. A. (2019). A secure adaptive beamforming mechanism exploiting deafness in directional beamforming MANET. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII (Vol. 11018, p. 110181F)*. International Society for Optics and Photonics.
- Jaiswal, M., Liu, Y., & Ling, N. (2016). Design and Implementation of a Greener Home Automation System. In PROCEEDINGS OF THE 9TH IEEE INTERNATIONAL CONFERENCE ON UBI-MEDIA COMPUTING" UMEDIA-2016" (pp. 182-187).
- Janis, P., Chia-Hao, Y. U., Doppler, K., Ribeiro, C., Wijting, C., Klaus, H. U. G. L., ...& Koivunen, V. (2009). Device-to-device communication underlaying cellular communications systems. *International Journal of Communications, Network and System Sciences*, 2(03), 169.
- Jang, J., & Kim, E. J. (2016). Survey on Industrial Wireless Network Technologies for Smart Factory. *JOURNAL OF PLATFORM TECHNOLOGY*, 4(1), 3-10.
- Jeyanthi, N., Abraham, A., & Mcheick, H. (2019). *Studies in Big Data 47 Ubiquitous Computing and Computing Security of IoT*. Springer.
- Jha, J., & Ragha, L. (2013). Intrusion detection system using support vector machine. *International Journal of Applied Information Systems (IJ AIS)*, (3), 25-30.
- Jonas, D. E., Wilkins, T. M., Bangdiwala, S., Bann, C. M., Morgan, L. C., Thaler, K. J., ... & Gartlehner, G. (2013). Findings of bayesian mixed treatment comparison meta-analyses.

- Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018, April). A survey on anomaly based host intrusion detection system. In *Journal of Physics: Conference Series* (Vol. 1000, No. 1, p. 012049). IOP Publishing.
- Jyothsna, V. V. R. P. V., Prasad, V. R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7), 26-35.
- Kapoor, A., Wiebe, N., & Svore, K. (2016). Quantum perceptron models. In *Advances in Neural Information Processing Systems* (pp. 3999-4007).
- Kaur, R., & Kaur, A. (2014). Blackhole Detection In Manets Using Artificial Neural Networks. *International Journal For Technological Research In Engineering*, 1(9), 959-962.
- Kaur, M., & Saini, K. S. (2017). *A Framework for Recyclable Household Waste Management System in Smart Home Using MANET*. In *Computing and Network Sustainability* (pp. 213-223). Springer, Singapore.
- Karlsson, J., Dooley, L. S., & Pulkkis, G. (2018, August). Secure routing for MANET connected Internet of Things systems. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 114-119). IEEE.
- Kabir, M. R., Onik, A. R., & Samad, T. (2017). A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach. *International Journal of Computer Applications*, 166(4).
- Kamakshi, Y. L., & Kumar, M. M. (2018). A Novel Approach to Secure Route Discovery for Dynamic Source Routing in MANETs.

- Kang, J. J., Adibi, S., Larkin, H., & Luan, T. (2015, November). Predictive data mining for converged internet of things: A mobile health perspective. *In Telecommunication Networks and Applications Conference (ITNAC), 2015 International* (pp. 5-10). IEEE.
- Kao, D. Y., & Hsiao, S. C. (2018, February). The dynamic analysis of WannaCry ransomware. *In Advanced Communication Technology (ICACT), 2018 20th International Conference on* (pp. 159-166). IEEE.
- Kevric, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications*, 28(1), 1051-1058.
- Kenkre, P. S., Pai, A., & Colaco, L. (2015). Real time intrusion detection and prevention system. *In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014* (pp. 405-411). Springer, Cham.
- Komninos, N., & Procopiou, A. (2019). Bio/Nature-inspired algorithms in AI for malicious activity detection.
- Kondaiah, R., & Sathyanarayana, B. (2018). TRUST FACTOR AND FUZZY-FIREFLY INTEGRATED PARTICLE SWARM OPTIMIZATION BASED INTRUSION DETECTION AND PREVENTION SYSTEM FOR SECURE ROUTING OF MANET. *International Journal of Computer Sciences and Engineering*, 10(1).
- Kovarasan, R. K., & Rajkumar, M. (2019). An Effective Intrusion Detection System Using Flawless Feature Selection, Outlier Detection and Classification. *In Progress in Advanced Computing and Intelligent Engineering* (pp. 203-213). Springer, Singapore.

- Khan, J. Y., Chen, D., & Hulin, O. (2014). Enabling Technologies for Effective Deployment of Internet of Things (MANET) Systems. *Australian Journal of Telecommunications and the Digital Economy*, 2(4).
- Kim, H., Lee, E., Kwon, D., & Ju, H. (2017, June). Chemical laboratory safety management service using MANET sensors and open APIs. *In Information and Communications (ICIC), 2017 International Conference on* (pp. 262-263). IEEE.
- Kumar, G. R., Mangathayaru, N., & Narsimha, G. (2016). Intrusion Detection-A Text Mining Based Approach. *International Journal of Computer Science and Information Security*, 14, 76.
- Khan, Z. & Herrmann, P. (2019). Recent Advancements in intrusion detection systems for the internet of things. *Security and Communication Networks*, DOI: [org/10.1155/2019/4301409](https://doi.org/10.1155/2019/4301409)
- Kumar S., Raju, C., Ratnakar, M., Baba, D., & Ratnakar, N. (2013). Intrusion detection system: types and prevention. *International Journal of Computer Science and Information Technologies*, 4(1), 77-82
- Kumar, S., & Dutta, K. (2016). Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges. *Security and Communication Networks*, 9(14), 2484-2556.
- Kušen, E., & Strembeck, M. (2017). Security-related Research in Ubiquitous Computing-- Results of a Systematic Literature Review. *arXiv preprint arXiv:1701.00773*.

- Koshti, M., Ganorkar, S., & Chiari, L. (2016). MANET Based Health Monitoring System by Using Raspberry Pi and ECG Signaly. *International Journal of Innovative Research in Science, Engineering and Technology*, 5(5).
- Konstantinidis, E. I., Billis, A. S., Dupre, R., Montenegro, J. M. F., Conti, G., Argyriou, V., & Bamidis, P. D. (2017). MANET of active and healthy ageing: cases from indoor location analytics in the wild. *Health and Technology*, 7(1), 41-49.
- Kolhe, P., Bhosale, S., Lathe, S., Mane, S., & Bhattad, R. (2016). Network Intrusion Detection by Finding Correlation between Multiple Features using K-means Algorithm & Multivariate Correlation Analysis. *Networking and Communication Engineering*, 8(3), 61-66.
- Lafferty, J. D., & Wasserman, L. (2006). Challenges in statistical machine learning. *Statistica Sinica*, 16, 307.
- Lakshmi, A. A., & Valluvan, K. R. (2015). Support vector machine and fuzzy-based intrusion detection and prevention for attacks in MANETs. *International Journal of Mobile Network Design and Innovation*, 6(2), 63-72.
- Li, F., & Wu, J. (2008, June). Hit and run: A bayesian game between malicious and regular nodes in manets. In *2008 5th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks* (pp. 432-440). IEEE.
- Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6), 321-332.

- Lee, H. R., Lin, C. H., & Kim, W. J. (2016, October). Development of an IoT-based visitor detection system. In *2016 International SoC Design Conference (ISOCC)* (pp. 281-282). IEEE.
- Liu, Z., Jin, H., Hu, Y. C., & Bailey, M. (2018). Practical Proactive DDoS-Attack Mitigation via Endpoint-Driven In-Network Traffic Control. *IEEE/ACM Transactions on Networking*, (99), 1-14.
- Lee, I., & Lee, K. (2015). The Internet of Things (MANET): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431-440.
- Lee, J., & Cho, S. H. (2015). Study of Relation Between Consumers Advertisement Attitude and Need for Cognition for MANET-Implemented Advertisement. *Journal of Digital Contents Society*, 16(1), 165-172.
- Li, W., & Kara, S. (2017). Methodology for Monitoring Manufacturing Environment by Using Wireless Sensor Networks (WSN) and the Internet of Things (MANET). *Procedia CIRP*, 61, 323-328.
- Li, Z., Jing, T., Ma, L., Huo, Y., & Qian, J. (2016). Worst-case cooperative jamming for secure communications in CIoT networks. *Sensors*, 16(3), 339.
- Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-Based Intrusion Detection System with System Calls: Review and Future Trends. *ACM Computing Surveys (CSUR)*, 51(5), 98.
- Liu, Y., Zhang, G., Chen, W., & Wang, X. (2016, October). An efficient privacy protection solution for smart home application platform. In *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on* (pp. 2281-2285). IEEE.

- López-Matencio, P., Vales-Alonso, J., & Costa-Montenegro, E. (2017). ANT: Agent Stigmergy-Based MANET-Network for Enhanced Tourist Mobility. *Mobile Information Systems*, 2017.
- Lin, Z., Mak, P. I., & Martins, R. P. (2016). A sub-GHz multi-ISM-band ZigBee receiver using function-reuse and gain-boosted N-path techniques for IoT applications. In *Ultra-Low-Power and Ultra-Low-Cost Short-Range Wireless Receivers in Nanoscale CMOS* (pp. 81-103). Springer, Cham..
- Li, Y., Chai, K. K., Chen, Y., & Loo, J. (2016). Distributed access control framework For IPv6-based hierarchical internet of things. *IEEE Wireless Communications*, 23(5), 17-23.
- Li, X., Lu, R., Liang, X., Shen, X., Chen, J., & Lin, X. (2011). Smart community: an internet of things application. *IEEE Communications Magazine*, 49(11).
- Lutz, R. (2016). Requirements for molecular programmed nano systems. In *24th IEEE REQUIREMENTS ENGINEERING CONFERENCE* (p. 2).
- Le Dang, N., Le, D. N., & Le, V. T. (2016). A new multiple-pattern matching algorithm for the network intrusion detection system. *International Journal of Engineering and Technology*, 8(2), 94.
- Luong, N. T., Vo, T. T., & Hoang, D. (2019). FAPRP: A Machine Learning Approach to Flooding Attacks Prevention Routing Protocol in Mobile Ad Hoc Networks. *Wireless Communications and Mobile Computing*, 2019.
- Johnson, C. (2016). Securing the Participation of Safety-Critical SCADA Systems in the Industrial Internet of Things.

- Klassen, G., & Buske, M. (2018). City as a Service and City On-Demand—New concepts for intelligent urban development. *In Digital Marketplaces Unleashed* (pp. 795-807). Springer, Berlin, Heidelberg.
- Kim, K. T., Kim, H., Park, H., & Kim, S. T. (2017, February). An industrial MANET MAC protocol based on IEEE 802.15. 4e TSCH for a large-scale network. In *Advanced Communication Technology (ICACT), 2017 19th International Conference on* (pp. 721-724). IEEE.
- Kor, A. L., Pattinson, C., Yanovsky, M., & Kharchenko, V. (2018). MANET-Enabled Smart Living. *In Technology for Smart Futures* (pp. 3-28). Springer, Cham.
- Lai, Y. L., Chou, Y. H., & Chang, L. C. (2017). An intelligent MANET emergency vehicle warning system using RFID and WiFi technologies for emergency medical services. *Technology and health care*, (Preprint), 1-13.
- Li, B., Li, Y., & Wang, S. (2016, August). Optimal workflows of AllJoyn network with centralized management gateway. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)* (pp. 303-307). IEEE.
- Marteau, P. F. (2018). Sequence covering for efficient host-based intrusion detection. *IEEE Transactions on Information Forensics and Security*, *14*(4), 994-1006.
- Mkuzangwe, N. N. P., & Nelwamondo, F. V. (2017, April). A fuzzy logic based network intrusion detection system for predicting the TCP SYN flooding attack. In *Asian conference on intelligent information and database systems* (pp. 14-22). Springer, Cham.
- Moustafa, N., Creech, G., & Slay, J. (2017). Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In *Data analytics and decision support for cybersecurity* (pp. 127-156). Springer, Cham.

- Ma, D., Wang, L., Lei, C., Xu, Z., Zhang, H., & Li, M. (2016, December). Thwart eavesdropping attacks on network communication based on moving target defense. In *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)* (pp. 1-2). IEEE.
- Manek, A. S., Shenoy, P. D., Mohan, M. C., & Venugopal, K. R. (2016). Detection of fraudulent and malicious websites by analysing user reviews for online shopping websites. *International Journal of Knowledge and Web Intelligence*, 5(3), 171-189.
- Mishra, D., & Naik, B. (2019). Detecting Intrusive Behaviors using Swarm-based Fuzzy Clustering Approach. In *Soft Computing in Data Analytics* (pp. 837-846). Springer, Singapore.
- Mitrokotsa, A., Tsagkaris, M., & Douligeris, C. (2008, June). Intrusion detection in mobile ad hoc networks using classification algorithms. In *IFIP Annual Mediterranean Ad Hoc Networking Workshop* (pp. 133-144). Springer, Boston, MA.
- Mir, N. M., Khan, S., Butt, M. A., & Zaman, M. (2016). An experimental evaluation of bayesian classifiers applied to intrusion detection. *Indian Journal of Science and Technology*, 9(12), 1-7.
- Malhotra, R. (2015). A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing*, 27, 504-518.
- Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10, 94.

- Mathur, P. (2019). Overview of Machine Learning in Finance. In *Machine Learning Applications Using Python* (pp. 259-270). Apress, Berkeley, CA.
- Mehta, V., & Gupta, D. N. (2012). Performance analysis of qos parameters for wimax networks. *International Journal of Engineering and Innovative Technology (IJEIT)*, 1(5), 105-110.
- Mirzagholi, S., & Faez, K. (2016). MHIDCA: Multi Level Hybrid Intrusion Detection and Continuous Authentication for MANET Security. *Journal of Computer & Robotics*, 9(1), 1-11.
- Misra, G., Kumar, V., Agarwal, A., & Agarwal, K. (2016). Internet of things (iot)—a technological analysis and survey on vision, concepts, challenges, innovation directions, technologies, and applications (an upcoming or future generation computer communication system technology). *American Journal of Electrical and Electronic Engineering*, 4(1), 23-32.
- Mishra, S., Li, X., Pan, T., Kuhnle, A., Thai, M. T., & Seo, J. (2017). Price modification attack and protection scheme in smart grid. *IEEE Transactions on Smart Grid*.
- Medina, B. E., & Manera, L. T. (2017, May). Retrofit of air conditioning systems through an Wireless Sensor and Actuator Network: An MANET-based application for smart buildings. In *Networking, Sensing and Control (ICNSC), 2017 IEEE 14th International Conference on* (pp. 49-53). IEEE.
- Minoli, D., Sohraby, K., & Occhiogrosso, B. (2017). MANET Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet of Things Journal*, 4(1), 269-283.

- Munns, C., & Basu, S. (2017). *Privacy and Healthcare Data: 'choice of Control' to 'choice' and 'control'*. Routledge.
- Muralidharan, S., Roy, A., & Saxena, N. (2016). An Exhaustive Review on Internet of Things from Korea's Perspective. *Wireless Personal Communications*, 90(3), 1463-1486.
- Medvedev, A., Fedchenkov, P., Zaslavsky, A., Anagnostopoulos, T., & Khoruzhnikov, S. (2015, August). Waste management as an MANET-enabled service in smart cities. *In Conference on Smart Spaces* (pp. 104-115). Springer International Publishing.
- Michael, K. (2017). Go? Get Chipped?: A Brief Overview of Non-Medical Implants between 1997-2013 (Part 1). *IEEE Technology and Society Magazine*, 36(3), 6-9.
- National Academies of Sciences, Engineering, and Medicine. (2018). *Changing Sociocultural Dynamics and Implications for National Security: Proceedings of a Workshop*. Washington, DC: The National Academies Press. doi: <https://doi.org/10.17226/25056>.
- Nguyen, C. T., Camp, O., & Loiseau, S. (2007, March). A Bayesian network based trust model for improving collaboration in mobile ad hoc networks. In *2007 IEEE International Conference on Research, Innovation and Vision for the Future* (pp. 144-151). IEEE.
- Nishani, L., & Biba, M. (2016). Machine learning for intrusion detection in MANET: a state-of-the-art survey. *Journal of Intelligent Information Systems*, 46(2), 391-407.
- Notario, N., Crespo, A., Martín, Y. S., Del Alamo, J. M., Le Métayer, D., Antignac, T., ... & Wright, D. (2015, May). PRIPARE: integrating privacy best practices into a privacy engineering methodology. *In Security and Privacy Workshops (SPW)*, 2015 IEEE (pp. 151-158). IEEE.

- Narvekar, A. N., & Joshi, K. K. (2017, January). Security sandbox model for modern web environment. In *Nascent Technologies in Engineering (ICNTE), 2017 International Conference on* (pp. 1-6). IEEE.
- NG, K. K. R., & Rajeshwari, K. (2017, January). Interactive clothes based on MANET using NFC and Mobile Application. In *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual* (pp. 1-4). IEEE.
- Ngomane, I., Velempini, M., & Dlamini, S. V. (2018). The detection of the spectrum sensing data falsification attack in cognitive radio ad hoc networks. In *2018 Conference on Information Communications Technology and Society (ICTAS)* (pp. 1-5). IEEE.
- Nigam, S., Asthana, S., & Gupta, P. (2016, February). MANET based intelligent billboard using data mining. In *Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on* (pp. 107-110). IEEE.
- Norman, D. (2017). Design, Business Models, and Human-Technology Teamwork: As automation and artificial intelligence technologies develop, we need to think less about human-machine interfaces and more about human-machine teamwork. *Research-Technology Management*, 60(1), 26-30.
- Oberle, A., Rein, A., Kuntze, N., Rudolph, C., Paatero, J., Lunn, A., & Racz, P. (2013, April). Integrating trust establishment into routing protocols of today's MANETs. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 2369-2374). IEEE.
- Oksanen, T., Linkolehto, R., & Seilonen, I. (2016). Adapting an industrial automation protocol to remote monitoring of mobile agricultural machinery: a combine harvester with MANET. *IFAC-PapersOnLine*, 49(16), 127-131.

- O'hern, W. A., Amoroso, E. G., Barry, M., Ramos, A., Solero, D., Sparrell, D. K., & Dilts, R. (2016). *U.S. Patent No. 9,456,003*. Washington, DC: U.S. Patent and Trademark Office.
- Olufisoye, A. C. (2016). Design of a Voice Based Intelligent Prototype Model for Automatic Control of Multiple Home Appliances. *Transactions on Machine Learning and Artificial Intelligence*, 4(2), 23.
- Pandey, R. C., Verma, M., & Sahu, L. K. (2017). Internet of Things (MANET) Based Gas Leakage Monitoring and Alerting System with MQ-2 Sensor.
- Patel, C., & Diwanji, H. (2016). A Research on Web Content Extraction and Noise Reduction through Text Density Using Malicious URL Pattern Detection.
- Patel, N. J. K., & Tripathi, K. (2018). Trust Value based Algorithm to Identify and Defense Gray-Hole and Black-Hole attack present in MANET using Clustering Method.
- Patel, S., Hughes, R., Hester, T., Stein, J., Akay, M., Dy, J. G., & Bonato, P. (2010). A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology. *Proceedings of the IEEE*, 98(3), 450-461.
- Patel, W. D., Patel, C., & Valderrama, C. IoMT based Efficient Vital Signs Monitoring System for Elderly Healthcare Using Neural Network.
- Poularakis, K., Iosifidis, G., & Tassiulas, L. (2018). SDN-enabled Tactical Ad Hoc Networks: Extending Programmable Control to the Edge. arXiv preprint arXiv:1801.02909.
- Pathak, P., Chauhan, E., Rathi, S., & Kosti, S. (2017). *HMM-Based IDS for Attack Detection and Prevention in MANET. Lecture Notes in Networks and Systems*, 413–421. doi:10.1007/978-981-10-3920-1_42

- Pullin, A., Pattinson, C., & Kor, A. L. (2018, April). Building Realistic Mobility Models for Mobile Ad Hoc Networks. In *Informatics* (Vol. 5, No. 2, p. 22). *Multidisciplinary Digital Publishing Institute*.
- Peters, G. W., & Panayi, E. (2016). Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. In *Banking Beyond Banks and Money* (pp. 239-278). Springer International Publishing.
- Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1), 67.
- Rafsanjani, M. K. (2009). Evaluating Intrusion Detection Systems and Comparison of Intrusion Detection Techniques in Detecting Misbehaving Nodes for MANET. In *Advanced Technologies*. IntechOpen.
- Raju, K. N., & Setty, S. P. (2015). Artificial Neural Network Based Decision on Parameter Values in AODV to Enhance the Performance of Mobile Ad Hoc Networks. *International Journal of Computer Science and Information Technologies*, 6(5), 4375-4377.
- Rajasekar, S., & Subramani, A. (2016). A review on routing protocols for mobile Adhoc networks. *i-manager's Journal on Mobile Applications and Technologies*, 3(1), 39.
- Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14), 1347-1358.

- Rath, M., Swain, J., Pati, B., & Pattanayak, B. K. (2018). Network Security: Attacks and Control in MANET. *In Handbook of Research on Network Forensics and Analysis Techniques* (pp. 19-37). IGI Global.
- Reddy, G. D., Chutke, S., Reddy, M. S. V. R., & Rao, D. N. (2018). Wireless Sensor Network Application for IoT based HealthCare System. *In International Journal of Emerging Technologies and Innovative Research JETIR (Vol. 5, No. 2 (February-2018))*. JETIR.
- Rezaul Karim, A. H. M., Rajatheva, R. M. A. P., & Ahmed, K. M. (2006, October). An efficient collaborative intrusion detection system for MANET using Bayesian Approach. *In Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems* (pp. 187-190). ACM.
- Rusitschka, S., & Curry, E. (2016). Big data in the energy and transport sectors. *In New Horizons for a Data-Driven Economy* (pp. 225-244). Springer, Cham.
- Roux, J., Alata, E., Auriol, G., Nicomette, V., & Kaâniche, M. (2017, September). Toward an intrusion detection approach for IoT based on radio communications profiling. *In 2017 13th European Dependable Computing Conference (EDCC)* (pp. 147-150). IEEE.
- Saha, H. N., Mandal, A., & Sinha, A. (2017, January). Recent trends in the Internet of Things. *In Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual* (pp. 1-4). IEEE.
- Sanghera, P. (2019). *Project Scope Management. In CAPM® in Depth* (pp. 135-171). Apress, Berkeley, CA.

- Shinde, A. M., Gresham, G. K., Hendifar, A. E., Li, Q., Spiegel, B., Rimel, B., ... & Figlin, R. A. (2017). *Correlating wearable activity monitor data with PROMIS detected distress and physical functioning in advanced cancer patients*.
- Sartran, L., Gay, S., Savalle, P. A., Mermoud, G., & Vasseur, J. P. (2018). *U.S. Patent Application No. 15/263,487*. The Washington Trademark Office.
- Samuel, A. L. (1988). Some studies in machine learning using the game of checkers. II—recent progress. In *Computer Games I* (pp. 366-400). Springer, New York, NY.
- Schrider, D. R., & Kern, A. D. (2018). Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, *34*(4), 301-312.
- Sarika, S., Pravin, A., Vijayakumar, A., & Selvamani, K. (2016). Security issues in mobile ad hoc networks. *Procedia Computer Science*, *92*, 329-335.
- Singh, D., & Bedi, S. S. (2015). Novel Intrusion Detection in MANETs Based on Trust. *International Journal of Computer Science and Information Technology* *6*(4), 3556-3560.
- Sheth, A., Jaimini, U., & Yip, H. Y. (2018). How will the Internet of Things enable augmented personalized health?. *IEEE intelligent systems*, *33*(1), 89-97.
- Subba, B., Biswas, S., & Karmakar, S. (2016). Intrusion detection in Mobile Ad-hoc Networks: Bayesian game formulation. *Engineering Science and Technology, an International Journal*, *19*(2), 782-799.
- Sun, B., Wu, K., & Pooch, U. W. (2006). Zone-Based Intrusion Detection for Mobile Ad Hoc Networks. *Ad hoc & sensor wireless networks*, *2*(3), 297-324.

- Serrat-Olmos, M. D., Hernández-Orallo, E., Cano, J. C., Calafate, C. T., & Manzoni, P. (2012, November). Accurate detection of black holes in MANETs using collaborative bayesian watchdogs. In *2012 IFIP Wireless Days* (pp. 1-6). IEEE.
- Sebopelo, R., Isong, B., & Gasela, N. (2019). Identification of Compromised Nodes in MANETs using Machine Learning Technique. *International Journal of Computer Network and Information Security*, *11*(1), 1.
- Siau, K., & Yang, Y. (2017, May). Impact of artificial intelligence, robotics, and machine learning on sales and marketing. In *Twelve Annual Midwest Association for Information Systems Conference (MWAIS 2017)* (pp. 18-19).
- Shams, E. A., & Rizaner, A. (2018). A novel support vector machine based intrusion detection system for mobile ad hoc networks. *Wireless Networks*, *24*(5), 1821-1829.
- Shao, Y. H., Chen, W. J., & Deng, N. Y. (2014). Nonparallel hyperplane support vector machine for binary classification problems. *Information Sciences*, *263*, 22-35.
- Shawe-Taylor, J., & Cristianini, N. (2000). Support vector machines. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 93-112.
- Sivanesan, P., & Thangavel, S. (2015). HMM based resource allocation and fuzzy based rate adaptation technique for MANET. *Optik*, *126*(3), 331-336.
- Swathi, R., & Seshadri, R. (2017, June). Systematic survey on evolution of machine learning for big data. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 204-209). IEEE.

- Singh, P., Gupta, S., Sejwal, L., & Mohan, A. (2018). Power Issues of MANET. *In Information and Communication Technology* (pp. 123-128). Springer, Singapore.
- Shrouf, F. (2016). Utilizing the Internet of Things to promote energy awareness and efficiency at discrete production processes: Practices and methodology.
- Singh, S., & Singh, N. (2015, October). Internet of Things (MANET): Security challenges, business opportunities & reference architecture for E-commerce. In *Green Computing and Internet of Things (ICGCMANET), 2015 International Conference on* (pp. 1577-1581). IEEE.
- Siddesh, G. M., Srinivasa, K. G., Kaushik, S., Varun, S. V., Subramanyam, V., & Patil, V. M. (2017). Internet of Things (MANET) Solution for Increasing the Quality of Life of Physically Challenged People. *Journal of Organizational and End User Computing (JOEUC)*, 29(4), 72-83.
- Sivanathan, A., Sherratt, D., Gharakheili, H. H., Sivaraman, V., & Vishwanath, A. (2016, November). Low-cost flow-based security solutions for smart-home MANET devices. In *Advanced Networks and Telecommunications Systems (ANTS), 2016 IEEE International Conference on* (pp. 1-6). IEEE.
- Soheily-Khah, S., Marteau, P. F., & Béchet, N. (2018, April). Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: a case study on the ISCX dataset. In *2018 1st International Conference on Data Intelligence and Security (ICDIS)* (pp. 219-226). IEEE.
- Solanas, A., Patsakis, C., Conti, M., Vlachos, I. S., Ramos, V., Falcone, F., ... & Martinez-Balleste, A. (2014). Smart health: a context-aware health paradigm within smart cities. *IEEE Communications Magazine*, 52(8), 74-81.

- Sowah, R. A., Ofori-Amanfo, K. B., Mills, G. A., & Koumadi, K. M. (2019). Detection and Prevention of Man-in-the-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN). *Journal of Computer Networks and Communications*, 2019.
- Spanos, D. (2018). Intrusion Detection Systems for Mobile Ad Hoc Networks.
- Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges for realising the Internet of Things. *Cluster of European Research Projects on the Internet of Things, European Commission*, 3(3), 34-36.
- Sotres, P., Santana, J. R., Sánchez, L., Lanza, J., & Muñoz, L. (2017). Practical Lessons From the Deployment and Management of a Smart City Internet-of-Things Infrastructure: The SmartSantander Testbed Case. *IEEE Access*, 5, 14309-14322.
- Stankovic, J. A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1), 3-9.
- Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2019). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, 12(2), 493-501.
- Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. (2019). Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection. *In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (pp. 643-646). IEEE.
- Tang, J., Deng, C., & Huang, G. B. (2015). Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems*, 27(4), 809-821.

- Tomar, R. S., Sharma, M. S. P., Jha, S., & Chaurasia, B. K. (2019). Performance Analysis of Hidden Terminal Problem in VANET for Safe Transportation System. *In Harmony Search and Nature Inspired Optimization Algorithms* (pp. 1199-1208). Springer, Singapore.
- Rizwan, P., Suresh, K., & Babu, M. R. (2016, October). Real-time smart traffic management system for smart cities by using Internet of Things and big data. *In Emerging Technological Trends (ICETT), International Conference on* (pp. 1-7). IEEE.
- Roselin, A. G., Nanda, P., & Nepal, S. (2017, August). Lightweight Authentication Protocol (LAUP) for 6LoWPAN Wireless Sensor Networks. *In Trustcom/BigDataSE/ICSS, 2017 IEEE* (pp. 371-378). IEEE.
- Rosli, A., Taib, A. M., & Ali, W. N. A. W. (2017). Utilizing the Enhanced Risk Assessment Equation to Determine the Apparent Risk due to User Datagram Protocol (UDP) Flooding Attack. *Sains Humanika*, 9(1-4).
- Tiburski, R. T., Amaral, L. A., de Matos, E., de Azevedo, D. F., & Hessel, F. (2017, January). Evaluating the use of TLS and DTLS protocols in MANET middleware systems applied to E-health. *In Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual* (pp. 480-485). IEEE.
- Pradilla, J., González, R., Esteve, M., & Palau, C. (2016, April). Sensor Observation Service (SOS)/Constrained Application Protocol (CoAP) proxy design. In *Electrotechnical Conference (MELECON), 2016 18th Mediterranean* (pp. 1-5). IEEE.
- Ray, P. P., Mukherjee, M., & Shu, L. (2017). Internet of Things for Disaster Management: State-of-the-Art and Prospects. *IEEE Access*, 5, 18818-18835.

- Raposo, D., Rodrigues, A., Silva, J. S., Boavida, F., Oliveira, J., Herrera, C., & Egas, C. (2016, June). An autonomous diagnostic tool for the WirelessHART industrial standard. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A* (pp. 1-3). IEEE.
- Roy, S. S., Mallik, A., Gulati, R., Obaidat, M. S., & Krishna, P. V. (2017, January). A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection. In *International Conference on Mathematics and Computing* (pp. 44-53). Springer, Singapore.
- Shi, C., Liu, J., Liu, H., & Chen, Y. (2017, July). Smart User Authentication through Actuation of Daily Activities Leveraging WiFi-enabled MANET. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (p. 5). ACM.
- Silverajan, B., Ocak, M., Jiménez, J., & Kolehmainen, A. (2016, December). Enhancing Lightweight M2M Operations for Managing MANET Gateways. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on* (pp. 187-192). IEEE.
- Singh, D., Tripathi, G., & Jara, A. J. (2014, March). A survey of Internet-of-Things: Future vision, architecture, challenges and services. In *Internet of things (WF-MANET), 2014 IEEE world forum on* (pp. 287-292). IEEE.
- S.M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989.*

- Saikumar, T., SudhaRani, G., Keerthi, K., Sneha, K., & Srikar, B. (2017). Modified Improved Kernel Fuzzy Adaptive Threshold Algorithm on Modified Level set method for Picture Segmentation. *evolution*.
- Singh, M., & Singh, R. K. (2016). Comparative Analysis of IDS Approaches and their Techniques. *IITM Journal of Management and IT*, 7(1), 83-88.
- Singh, G., & Dhir, V. (2018). Performance Analysis of Adhoc On Demand Distance Vector (AODV) and Destination Sequence Routing (DSR) protocols in Mobile Adhoc Networks (MANET). *Global Journal of Computer Science and Technology*.
- Smith, R. E. (1994, October). Constructing a high assurance mail guard. *In Proceedings of the 17th National Computer Security Conference* (pp. 247-253).
- Smith, R. E. (1994, October). Constructing a high assurance mail guard. *In Proceedings of the 17th National Computer Security Conference* (pp. 247-253).
- Seo, D., Jeon, Y. B., Lee, S. H., & Lee, K. H. (2016). Cloud computing for ubiquitous computing on M2M and MANET environment mobile application. *Cluster Computing*, 19(2), 1001-1013.
- Sher, M., & Magedanz, T. (2005). Network access security management (NASM) model for next generation mobile telecommunication networks. *Mobility Aware Technologies and Applications*, 263-272.
- Tachmazidis, I., Davies, J., Batsakis, S., Duke, A., Antoniou, G., & Clarke, S. S. (2017). A Semantically Enriched Hypercat-enabled Internet of Things Data Hub. *In The Semantic Web–ISWC 2017*. Springer.
- Tawbi, N., & Martinelli, F. (2016, February). Fast and Effective Clustering of Spam Emails Based on Structural Similarity. *In Foundations and Practice of Security: 8th International*

- Symposium, FPS 2015, Clermont-Ferrand, France, October 26-28, 2015, Revised Selected Papers (Vol. 9482, p. 195). Springer.*
- Thanuja, R., & Umamakeswari, A. (2016). Effective Intrusion Detection System Design Using Genetic Algorithm For MANETs. *APRN J. Eng. Applied Sci*, 11(7), 4696-4700.
- Terán, M., Aranda, J., Carrillo, H., Mendez, D., & Parra, C. (2017, August). MANET-based system for indoor location using bluetooth low energy. *In Communications and Computing (COLCOM), 2017 IEEE Colombian Conference on* (pp. 1-6). IEEE.
- Thomas, D., Wilkie, E., & Irvine, J. (2016). Comparison of Power Consumption of WiFi Inbuilt Internet of Things Device with Bluetooth Low Energy. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(10), 1775-1778.
- Trivedi, Y. (2016). Innovation & competition: succeeding through global standards: a new massive open online course delivered on IEEE X. org. *IEEE Communications Magazine*, 54(3), 7-9.
- Urquhart, L., & McAuley, D. (2018). Avoiding the internet of insecure industrial things. *Computer Law & Security Review*, 34(3), 450-466.
- Vatcharatiansakul, N., Tuwanut, P., & Pornavalai, C. (2017, July). Experimental performance evaluation of LoRaWAN: A case study in Bangkok. *In Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on* (pp. 1-4). IEEE.
- Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., & Doody, P. (2011). Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, 1, 9-52.

- Vimala, S., Khanaa, V., & Nalini, C. (2018). A study on supervised machine learning algorithm to improvise intrusion detection systems for mobile ad hoc networks. *Cluster Computing*, 1-10.
- Vongpradhip, S., & Rungraungsilp, S. (2012, January). QR code using invisible watermarking in frequency domain. In *ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2011 9th International Conference on* (pp. 47-52). IEEE.
- Wamuyu, P. K. (2017). A Conceptual Framework for Implementing a WSN Based Cattle Recovery System in Case of Cattle Rustling in Kenya. *Technologies*, 5(3), 54.
- Wang, H., Xiong, D., Wang, P., & Liu, Y. (2017). A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained MANET Devices. *IEEE Access*, 5, 16393-16405.
- Wang, S., Chen, Y., & Tian, H. (2016, June). An intrusion detection algorithm based on chaos theory for selecting the detection window size. In *Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on* (pp. 556-560). IEEE.
- Wang, D., Long, Y., Xiao, Z., Xiang, Z., & Chen, W. (2016, July). A temporal self-organizing neural network for adaptive sub-sequence clustering and case studies. In *Computer, Information and Telecommunication Systems (CITS), 2016 International Conference on* (pp. 1-5). IEEE.
- Wang, S., Lei, G., Wan, W., Zhang, Y., & Li, C. (2017, May). Low-temperature thawing refrigerator based on the internet of things. In *Control And Decision Conference (CCDC), 2017 29th Chinese* (pp. 5961-5965). IEEE.
- Walker, S. T. (1985, April). Network security overview. In *Security and Privacy, 1985 IEEE Symposium on* (pp. 62-62). IEEE.

- Walker, S. T. (1985, April). Network security overview. In *Security and Privacy, 1985 IEEE Symposium on* (pp. 62-62). IEEE.
- Wei, Z., Tang, H., Yu, F. R., & Mason, P. (2014, October). Trust establishment based on Bayesian networks for threat mitigation in mobile ad hoc networks. In *2014 IEEE Military Communications Conference* (pp. 171-177). IEEE.
- Wu, Z., Meng, Z., & Gray, J. (2017). MANET-based Techniques for Online M2M-Interactive Itemised Data Registration and Offline Information Traceability in a Digital Manufacturing System. *IEEE Transactions on Industrial Informatics*.
- Wen, Z., Cala, J., Watson, P., & Romanovsky, A. (2016). Cost effective, reliable and secure workflow deployment over federated clouds. *IEEE Transactions on Services Computing*.
- Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2016). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1), 602-622.
- Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., & Guizani, M. (2017). Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE wireless communications*, 24(3), 10-16.
- Yeh, L. Y., Tsaur, W. J., & Juang, T. Y. (2016, January). Cryptanalysis and Efficient Improvement of a Robust and Scalable One-Way Hash Chain Authentication Protocol in Vehicular Communication. In Proceedings of the International Conference on Wireless Networks (ICWN) (p. 17). *The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*.

- Yeo, L. H., Che, X., & Lakkaraju, S. (2017). Understanding Modern Intrusion Detection Systems: A Survey. *arXiv preprint arXiv:1708.07174*.
- Ye, K. (2019). Key Feature Recognition Algorithm of Network Intrusion Signal Based on Neural Network and Support Vector Machine. *Symmetry*, 11(3), 380.
- Yeh, K. H., Su, C., Hsu, C. L., Chiu, W., & Hsueh, Y. F. (2016, October). Transparent authentication scheme with adaptive biometric features for MANET networks. *In Consumer Electronics, 2016 IEEE 5th Global Conference on* (pp. 1-2). IEEE.
- Yim, H. J., Son, Y. H., & Lee, K. C. (2016). A Data Distribution Service Quality of Services Policy Configuration for Data/Events/Services in the Internet of Things. *Advanced Science Letters*, 22(11), 3612-3617.
- Yong, H., Pengcheng, N., & Fei, L. (2013). Advancement and trend of internet of things in agriculture and sensing instrument. *Transactions of the Chinese Society for Agricultural Machinery*, 44(10), 216-226.
- Yu, T., Sekar, V., Seshan, S., Agarwal, Y., & Xu, C. (2015). Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. *In Proceedings of the 14th ACM Workshop on Hot Topics in Networks* (p. 5). ACM.
- Ye, X., Li, J., & Li, Y. (2010). *An Anomaly Detection System Based on Hide Markov Model for MANET*. *2010 International Conference on Computational Intelligence and Software Engineering*. doi:10.1109/wicom.2010.5601345
- Zhang, L., Tan, J., Han, D., & Zhu, H. (2017). From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today*, 22(11), 1680-1685.

- Zaidi, K., Milojevic, M. B., Rakocevic, V., Nallanathan, A., & Rajarajan, M. (2016). Host-based intrusion detection for VANETs: a statistical approach to rogue node detection. *IEEE Transactions on Vehicular Technology*, 65(8), 6703-6714.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1), 22-32.
- Zhang, N., Demetriou, S., Mi, X., Diao, W., Yuan, K., Zong, P., ...& Gunter, C. A. (2017). Understanding MANET Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be. arXiv preprint arXiv:1703.09809.
- Zhang, M., Yang, M., Wu, Q., Zheng, R., & Zhu, J. (2018). Smart perception and autonomic optimization: A novel bio-inspired hybrid routing protocol for MANETs. *Future Generation Computer Systems*, 81, 505-513.
- Zhu, Y. H., Qiu, S., Chi, K., & Fang, Y. (2017). Latency aware IPv6 packet delivery scheme over IEEE 802.15. 4 based battery-free wireless sensor networks. *IEEE Transactions on Mobile Computing*, 16(6), 1691-1704.
- Zhuang, Y., & Song, S. (2013). Use of Internet of Things for Ship Management of Inland Rivers. In *ICTIS 2013: Improving Multimodal Transportation Systems-Information, Safety, and Integration* (pp. 2425-2431).

APENDIX A

LIST OF EQUIPMENT USED IN THE STUDY

- A. Raspbery Pi Model B+
- B. Generic Smart Watch.
- C. Jumper Cables.
- D. Micro SD Card.
- E. PC running on Ubuntu Linux 18.04.1 release
- F. BlueTooth Module expander
- G. HDMI Cable
- H. VGA Monitor

APENDIX B

CODE LISTING 1 – SOURCE CODE FOR MANET SET-UP

1. set val(chan) Channel/WirelessChannel ;# channel type
2. set val(prop) Propagation/TwoRayGround ;# radio-propagation model
3. set val(netif) Phy/WirelessPhy ;# network interface type
4. set val(mac) Mac/802_11 ;# MAC type
5. set val(ifq) Queue/DropTail/PriQueue ;# interface queue type

CODE LISTING 2 – SOURCE CODE FOR SVM DATA SETS AND TABLE LIST

1. # For the objects provided as <i>args</i>, source the files in
2. # which their class & methods are defined.
3. # After attempting to import all supplied items, if any were unimportable,
4. # an error will be flagged with a detailed errmsg.
5. #
6. Import public import { args } {
 - a. \$self instvar import_dirs_ table_
 - b. # initialize the import table only on demand
 - c. if { ![info exists import_dirs_] } {
 - i. \$self init_table
 - d. }
 - e. # ensure that the TCLCL_IMPORT_DIRS env var hasn't changed since we
 - f. # initialized the table
 - g. \$self consistency_check
 - h. foreach item \$args {
 - i. if [info exists table_(\$item)] {

```

1. set file_list $table_($item)
2. # although it's poor programming practice,
3. # an object can be defined in multiple files
4. foreach file $table_($item) {
    a. if { [set msg [$self source_file $file]]!=""} {
        i. error "could not source $file for\
7. # As long as the import procedure has not yet been invoked, the user is
8. # free to override mappings that may be read from importTables.
9. #
10. Import public override_importTable_mapping { object file_list } {
    a. $self instvar overrideTable_import_dirs_
    b. if { [info exists import_dirs_] } {
        i. puts stderr "warning: ignoring \"override_importTable_mapping\
            a. $object $file_list\" \n\
            b. It is illegal to modify the internal table \
            c. after the first call to import."
        ii. return
    c. }
    d. if { [info exists overrideTable_($object)] } {
        i. unset overrideTable_($object)
    e. }
    f. foreach file $file_list {
        i. set fname [$self condense_into_absolute_filename \
            a. [$self file join [pwd] $file]]
        ii. lappend overrideTable_($object) $fname
    g. }
11. }

```

```

12. # tcl-object.tcl code is processed.
13. # is actually defined when Tcl_AppInit calls Tcl_Init to set up the
14. # script library facility.)
15. #
16. Import proc.private redefine_unknown { } {
    a. #
    b. # If a proc/instproc is called on an unknown class, you'll wind
    c. # up here. If auto-importing is enabled, attempt to import the class.
    d. rename unknown unknown.orig
    e. # Rather than redefining this procedure in tcl/library/init.tcl,
    f. # we can rename & augment it here for the mash interpreter.
    g. proc unknown { args } {
        i. # first try tcl's original unknown proc and return if
        ii. # successful
        iii. if { ![catch "eval {unknown.orig} $args" m] } {
            1. return
        iv. # otherwise, if autoimporting is enabled,
        v. # if able to import an item by this name, do so and return
        vi. # btw, if the stuff in catch quotes causes "unknown" to
        vii. # get called, error = "too many nested calls to
        viii. # Tcl_EvalObj (infinite loop?)"
        ix. $self instvar autoimport_
        x. if { [info exists autoimport_] && $autoimport_ } {
        xi. really_import [lindex $args 0]
        xii. } else {
            1. # if not trying to import, puts original error msg
            2. error "$m"

```

- h. # prevent this method from being called again
- i. Import proc.private redefine_unknown {} {}

17. # As new objects are needed, the unknown proc will catch them and

18. # import the files that define them and their methods.

19. # As an intended side-effect, explicit imports will be ignored

20. # until auto-import is disabled.

21. Import proc.public enable_autoimport {} {

- a. # XXX this should be done somewhere else
- b. import Class Object mashutils
- c. Import set autoimport_1
- d. \$self redefine_unknown
- e. return

22. Import proc.public disable_autoimport {} {

- a. Import set autoimport_0
- b. return

23. # Auto-importing is disabled by default.

24. # (And because dynamic loading using unknown yet, due to the fact the unknown

25. # isn't called when the -auperclass attribute is used in a Class defn.)

26. #

27. Import disable_autoimport

28. # Read the environment variable, TCLCL_IMPORT_DIRS, and store the directories in a list instvar.

29. # Afterwards, makes a call to the instproc that generates the table.

30. Import private init_table {} {

- a. \$self instvar import_dirs_
- b. global env

```

c. # If TCLCL_IMPORT_DIRS is not set before first time import proc
d. # is called, it is set to '.'
e. # Note that otherwise, '.' is not appended to TCLCL_IMPORT_DIRS.
f. if { ![info exists env(TCLCL_IMPORT_DIRS)] } {
    i. set env(TCLCL_IMPORT_DIRS) .
g. set import_dirs_ ""
h. foreach dir [$self smart_parse_env_var $env(TCLCL_IMPORT_DIRS)] {
    i. # If dir is relative, it is expanded to absolute.
    ii. # Relative pathnames in TCLCL_IMPORT_DIRS will be considered
    iii. # relative to '.'
    iv. # which is the directory the mash interpreter was launched from
    v. # (unless the cd proc has been called since then)
    vi. # So if mashlets are being run from a browser, '.' starts out as
    vii. # the directory from which the browser was launched.
    viii. lappend import_dirs_ [$self condense_to_absolute_filename $dir]
i. # locate the actual import directories
j. set dirs [$self find_import_dirs $import_dirs_]
k. # the first time import is called, build a table of mappings from
l. # objects to the file(s) they are defined in
m. $self make_table $dirs
31. # Build an internal table of mappings from objects to the file(s) they
32. # are defined in.
33. Import private make_table { dirs } {
    a. foreach d $dirs {
        i. $self read_dir $d
    b. $self incorporate_table_overrides
34. }

```

```

35. Import private incorporate_table_overrides { } {
    a. $self instvar overrideTable_ table_

    b. foreach object [array names overrideTable_] {
        i. set table_($object) $overrideTable_($object)
36. # Return a list of directories in which a readable importTable can be found.
37. # For every path in TCLCL_IMPORT_DIRS, importLocation will be read, if
38. # it exists, and the absolute pathnames of the importTables that it points
39. # to will be appended to import_table_list_ .
40. # For dirs in which a readable importLocation file is not found,
41. # the absolute pathname of the importTable in that dir, if one exists,
42. # will be appended instead.
43. # Directories may be named using complete pathnames or pathnames relative
44. # to CURRENTDIR.
45. Import private find_import_dirs { dirs } {
    a. # Generate a list of potential directories in which an importTable
    b. # may be found
    c. set list { }
    d. foreach dir $dirs {
        i. set importLocation [$self file join $dir importLocation]
        ii. set r [$self file readable $importLocation]
        iii. if [lindex $r 0] {
            1. set lines [$self read_file_into_list $importLocation]
            2. foreach line $lines {
                a. # append absolute filename on this line to
                b. # the list
                c. lappend list [$self \

```

```

1. condense_to_absolute_filename \
2. [$self file join $dir $line]]
3. }
4. if { [lindex $r 1] != {} } {
    a. # destroy the http token
    b. unset [lindex $r 1]
5. lappend list $dir
e. # prune the list down to the directories in which an importTable
f. # is actually readable
g. $self instvar last_modified_
h. set dirs ""
i. foreach d $list {
    i. set import_table [$self file join $d importTable]
    ii. set last_modified_($import_table) -1
    iii. set r [$self file readable $import_table]
    iv. if [lindex $r 0] {
        1. lappend dirs $d
        2. if { [lindex $r 1] != {} } {
            a. # destroy the http token
            b. unset [lindex $r 1]
        }
    }
j. #if { [llength $dirs] > 0 } {
k. # puts stderr "readable importTables found in: $dirs"
l. #} else {
m. # puts stderr "no readable importTables found"
46. # By reading the importTable in the provided directory <i>dir</i>,
47. # continue to define the elements of the table_ array. Each element of

```

```

48. # this array is indexed using an object name and consists of a list of
49. # files (using absolute pathnames) in which the the object and its
50. # methods are defined. Returns the table_ in list form.
51. #
52. Import private read_dir { dir } {
    a. $self instvar table_ classes_mapped_ last_modified_

    b. set importTableFile [$self condense_to_absolute_filename \
        1. [$self file join $dir importTable]]
    c. set last_modified_($importTableFile) -1
    d. # fetch the importTable and break it into a list of lines
    e. set lines [$self read_file_into_list $importTableFile]
    f. # for every line in the importTable, parse out the object and
    g. # the filename, adding the file to the appropriate list element
    h. # of the table_ array if it is not there already
    i. foreach line $lines {
        i. set index [lindex $line 0]
        ii. # use the absolute file name
        iii. # (relative filenames are considered to be relative to
        iv. # the directory containing the importTable)
        v. set fname [$self condense_to_absolute_filename \
            a. [$self file join $dir [lindex $line 1]]]
        vi. set last_modified [string trim [lindex $line 2]]

        vii. # if a mapping for this object was already read from
        viii. # another importTable, ignore this one
        ix. if [info exists classes_mapped_($index)] {

```

- x. # if this object already has a mapping to this filename,
- xi. # skip it
- xii. if {[info exists table_(\$index)]} {
 - 1. if {-1!=[search -exact \$table_(\$index) \$fname]} {
- xiii. lappend table_(\$index) \$fname
- xiv. if { \$last_modified!={ } } {

CODE LISTING 3 – SOURCE CODE FOR PROPAGATING BLACKHOLE

1. #=====
2. # Blackhole activity for 30 nodes
3. #=====
4. set val(chan) Channel/WirelessChannel ;# channel type
5. set val(prop) Propagation/TwoRayGround ;# radio-propagation model
6. set val(netif) Phy/WirelessPhy ;# network interface type
7. set val(mac) Mac/802_11 ;# MAC type
8. set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
9. set val(ll) LL ;# link layer type
10. set val(ant) Antenna/OmniAntenna ;# antenna model
11. set val(ifqlen) 50 ;# max packet in ifq
12. set val(nn) 30 ;# number of mobilenodes
13. set val(rp) AODV ;# routing protocol
14. set val(x) 1186 ;# X dimension of topography
15. set val(y) 600 ;# Y dimension of topography
16. set val(stop) 40 ;# time of simulation end

CODE LISTING 4 – THE ns-lib.tcl SOURCE FILE

1. proc time2real v {
2. global uscale

3. option add \$name. foundry adobe startupFile
4. set ff [option get . foundry \$name]
5. if {\$tcl_platform(platform)!="windows"} {
6. option add *font \$helv12b startupFile

APPENDIX C

CODE LISTING 1 - SOURCE CODE FOR PROTOTYPE IMPLEMENTATION

Code from ble_bt.py

5. import subprocess
6. from subprocess import *
7. import time
8. import threading

9. All ble devices found are stored here on a set to prevent duplication of ble devices. Set only contains different
10. items, if the item is already on a set, its passed and not added to the list
11. ble_devices=set()

12. The three subprocess commands initialize Bluetooth and Bluetooth BLE connections
13. print '>>>setting ble bt'
14. subprocess.call(['hciconfig'])
15. subprocess.call(['sudo','hciconfig','hci0','down'])
16. subprocess.call(['sudo','hciconfig','hci0','up'])
17. print '>>>done setting ble bt'

18. Bluetooth Ble scanning starts here by executing lscan of the hcitool for only 10 seconds

```

19. cmd=[&quot;sudo&quot;,&quot;timeout&quot;,&quot;-
s&quot;,&quot;SIGINT&quot;,&quot;10s&quot;,&quot;hcitool&quot;,&quot;-
i&quot;,&quot;hci0&quot;,&quot;lescan&quot;]
20. print &#39;&gt;&gt;searching for ble bt devices&#39;
21. avdf=subprocess.Popen(cmd,stdin=PIPE,stdout=PIPE)
31. print &#39;&gt;&gt;search complete&#39;
32. print &#39;&gt;&gt;&#39;,len(ble_devices),&#39; device(s) found&#39;
33. On execution reaching here, the devices are already found and stored on our set list. The
function below –
34. recv_until is used to check if a string exists in a buffer and while not, to wait and keep
polling data until the
35. check is in the buffer.
36. def recv_until(handle,check):
37. b=&#39;&#39;
38. while check not in b:
39. b+=handle.stdout.readline()
40. return 1
41. The class blebtconn supports connection of multiple ble connections via threading.
42. class blebtconn (threading.Thread):
43. def __init__(self, name, mac_address):
44. threading.Thread.__init__(self)
45. self.mac_address = mac_address
46. self.name = name
47. def run(self):
48. read_heart_rate_yoho_sports(self.name,self.mac_address)
49. The read_heart_rate_yoho_sports function is used to connect to connect to the ble device
found as well as turn

```

```

50.     the ble device notification on. On our case the yoho smart watch band. It also prints out
the notifications for the
51.     smart watch hearh sensor when used.
52.     def read_heart_rate_yoho_sports(name,mac_address):
53.     print '&#39;&gt;&gt;connecting to: &#39;,name
54.     cmd2=['&quot;sudo&quot; , &quot;gatttool&quot;, &quot;-I&quot; ,&quot;-b&quot;,]
55.     cmd2.append(mac_address)
56.     p = subprocess.Popen(cmd2,
57.     stdout=subprocess.PIPE,
58.     stderr=subprocess.STDOUT,
59.     stdin=PIPE,
60.     bufsize=0)

```

CODE LISTING 2 - SOURCE CODE MEASURING DEVICE RESOURCE ACTIVITY

capture_ram_cpu_net_traffic.py

1. This script is used to capture network traffic, that is both download and upload network bandwidth as cpu usage
2. and ram usage and logs it to net_trafic_recorder.log. Please make sure you rename or move the log file after
3. usage or it will spoil your next results since it starts from zero. You may have two different graphs. To change to
4. other interfaces for testing, replace wlan0 to any type of interface you using.
15. upload=psutil.net_io_counters(pernic=True)[network_interface][0]
16. download=psutil.net_io_counters(pernic=True)[network_interface][1]
17. up_down=(upload,download)

```

18. with open('net_traffic_recorder.log', 'a') as fp:
19.     while True:
20.         last_up_down = up_down
21.         upload=psutil.net_io_counters(pernic=True)[network_interface][0]
22.         download=psutil.net_io_counters(pernic=True)[network_interface][1]
23.         t1 = time.time()
24.         up_down = (upload,download)
25.         try:
26.             ul, dl = [(now - last) / (t1 - t0) / 1024.0 for now,last in zip(up_down, last_up_down)]
32.             fp.write(
                '{:0.2f}'.format(time.time()-t0ld)+'
                '+str(psutil.cpu_percent())+'
33.             '+str(psutil.virtual_memory()[2])+'  {:0.2f}'.format(ul)+'  {:0.2f}
                '+str(dl)+'\n')

```

CODE LISTING 3 - SOURCE CODE FOR RESULTS ANALYSIS

1. The code below is used to analyze log files generated by capture_ram_cpu_net_traffic.py
2. It takes the file name and what aspect you interested to analyze. The aspects include ram , cpu as well as the
3. network bandwidth.
4. Then a graph is plotted against time and one can analyze and make conclusions.

```

21. import matplotlib.lines as mlines

22. blue_line = mlines.Line2D([],[], color='blue', label='RAM %')
23. yellow_line = mlines.Line2D([],[], color='yellow', label='upload
speed')
24. orange_line = mlines.Line2D([],[], color='red', label='download
speed')
25. cyan_line = mlines.Line2D([],[], color='cyan', label='CPU %')

```

CODE LISTING 4 – SOURCE CODE FOR SVM CLASSIFIER

```
1.    from sklearn.model_selection import train_test_split
2.    from sklearn.neighbors import KNeighborsClassifier
17.   plt.scatter(_dataset[:2591, 0], _dataset[:2591, 2], c='r', label='snort_default')
18.   plt.scatter(_dataset[2591+1:2591+2201, 0], _dataset[2591+1:2591+2201, 2],
c='g',label='snort_ddos')
19.   plt.scatter(_dataset[2591+2201+1:2591+2201+2155, 0],
_ dataset[2591+2201+1:2591+2201+2155, 2], c='b',label='no_snort')
20.   plt.scatter(_dataset[2591+2201+2155+1+1012:, 0], _dataset[2591+2201+2155+1+1012:,
1], c='yellow',label='norm')
21.   #for training and testing set split, sklearn library which has an in-built splitting function
called train_test_split is used
22.   #random_state is a seed that takes a random_state as input
23.   #if you change the number the split of the data will also change.
24.   #if you keep the random_state same and run the cell multiple times the data splitting will
remain unchanged.
25.   train_data,test_data,train_label,test_label = train_test_split(dataset.iloc[:,1],
dataset.iloc[:,2], test_size=0.2, random_state=1)
26.   #the k (n_neighbors) parameter is often an odd number to avoid ties in the voting scores.
eg 1-9 neighbors = np.arange(1,9)--has 9 neighbors
27.   #2 numpy zero matrices namely train_accuracy and test_accuracy each for training and
testing accuracy
28.   #later needed to plot a graph to choose the best neighbor value.
29.   neighbors = np.arange(1,11)
30.   train_accuracy =np.zeros(len(neighbors))
31.   test_accuracy = np.zeros(len(neighbors))
32.   #enumerate over all the range of neighbor(1,201) - 200 neighbours values
33.   #and for each neighbor predict both on training and testing data
34.   #store the accuracy in the train_accuracy and test_accuracy numpy arrays
```

```
35. #for i,k in enumerate(neighbors):
36. knn = KNeighborsClassifier(n_neighbors=11)
37. knn.fit(train_data, train_label)
38. train_accuracy[i] = knn.score(train_data, train_label)
39. test_accuracy[i] = knn.score(test_data, test_label)
40. #plot the training and testing accuracy using matplotlib with accuracy vs. varying number
of neighbors
41. print dataset.describe()
42. plt.figure(figsize=(10,6))
43. plt.title('KNN accuracy with varying number of neighbors',fontsize=20)
44. plt.plot(neighbors, test_accuracy, label='Testing Accuracy')
45. plt.plot(neighbors, train_accuracy, label='Training accuracy')
46. plt.legend(prop={'size': 20})
47. plt.xlabel('Number of neighbors',fontsize=20)
48. plt.ylabel('Accuracy',fontsize=20)
49. plt.xticks(fontsize=20)
50. plt.yticks(fontsize=20)
51. plt.show()
```

**CODE LISTING 5 – LOG INJECTION INTO ANN FOR PATTERN RECOGNITION
(TRUNCATED)**

```
1.
2.     Perceptron(alpha=0.0001, class_weight=None, early_stopping=False, eta0=1.0,
3.         fit_intercept=True, max_iter=10, n_iter=None, n_iter_no_change=5,
4.         n_jobs=None, penalty=None, random_state=42, shuffle=True, tol=0.001,
5.         validation_fraction=0.1, verbose=0, warm_start=False
6.
7.     from sklearn.neural_network import MLPClassifier
8.     X = [[0., 0.], [0., 1.], [1., 0.], [1., 1.]]
9.     y = [0, 0, 0, 1]
10.    clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
11.        hidden_layer_sizes=(5, 2), random_state=1)
12.    print(clf.fit(X, y))
16.    for i in range(len(clf.coefs_)):
17.        number_neurons_in_layer = clf.coefs_[i].shape[1]
18.        for j in range(number_neurons_in_layer):
19.            weights = clf.coefs_[i][:,j]
20.            print(i, j, weights, end=" ")
21.            print()
22.    print()
```

CODE LISTING 1 SNORT.CONF

```
26 # This file contains a sample snort configuration.
27 # You should take the following steps to create your own custom configuration:
86 # List of ports you run ftp servers on
99                                     ipvar                                     AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.
188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]
100 preprocessor portscansvm: ignorebc 1 \
101 analyze_thr_lower 100 \
102 analyze_thr_upper 1600 \
103 sense_level 0.05 \
104 net_topology 0 \
105 log_method 1
1061 # Path to your rules files (this can be a relative path)
107 # Note for Windows users: You are advised to make this an absolute path,
108 # such as: c:\snort\rules
var RULE_PATH ../rules
var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH ../preproc_rules

# If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
```

CODE LISTING 3 – A TRUNCATED SOURCE CODE FOR SRC/PLUGBASE.C

```
85 #include "detection-plugins/sp_ip_same_check.h"
86 #include "detection-plugins/sp_clientserver.h"
87 #include "detection-plugins/sp_byte_check.h"
88 #include "detection-plugins/sp_byte_jump.h"
89 #include "detection-plugins/sp_byte_extract.h"
90 #include "detection-plugins/sp_byte_math.h"
91 #include "detection-plugins/sp_isdataat.h"
92 #include "detection-plugins/sp_pcre.h"
93 #include "detection-plugins/sp_flowbits.h"
94 #include "detection-plugins/sp_file_data.h"
95 #include "detection-plugins/sp_base64_decode.h"
96 #include "detection-plugins/sp_base64_data.h"
97 #include "detection-plugins/sp_pkt_data.h"
98 #include "detection-plugins/sp_asn1.h"
```

CODE LISTING 4 – A TRUNCATED SOURCE CODE FOR flow_callback.c

252 * src/encode.c, src/reload.h, src/sfdaq.c,
253 src/dynamic-preprocessors/dcerpc2/dce2_co.c,
254 src/dynamic-preprocessors/dcerpc2/dce2_config.c,
255 src/dynamic-preprocessors/dcerpc2/dce2_smb.c,
256 src/dynamic-preprocessors/dcerpc2/dce2_smb2.c,
257 src/dynamic-preprocessors/dcerpc2/spp_dce2.c,
258 src/dynamic-preprocessors/sdf/spp_sdf.c,
259 src/preprocessors/spp_frag3.c, src/preprocessors/spp_session.c,
260 src/preprocessors/spp_sfportscan.c,
261 src/preprocessors/Stream6/snort_stream_ip.c,
262 src/preprocessors/Stream6/snort_stream_tcp.c, src/sfutil/acsmx.c,
263 src/sfutil/sfksearch.c, src/sfutil/sfportobject.c,
264 tools/u2spewfoo/u2spewfoo.c

DATA LISTING 1 – LOG RESULTS FROM THE MANET - NORMAL

16.57 35.4 19.2 294.63 151.05
17.55 35.6 19.2 270.23 138.63
18.52 31.1 19.2 297.66 152.64
19.49 32.2 19.2 362.37 185.94
20.47 32.7 19.2 367.53 188.47
21.44 32.6 19.2 370.07 189.96
22.41 34.6 19.2 360.60 185.04
23.39 36.5 19.2 366.58 187.79
24.36 36.1 19.2 360.78 185.01
25.33 32.3 19.2 363.96 186.76
26.31 34.1 19.2 362.68 186.30
27.28 36.6 19.2 353.12 180.76

APENDIX D

AUTHOR PROFILE



Kirori Mindo is a community absorbed Lecturer and Researcher at Kabarak University 7 years' experience in planning, organization and mobilization for research. Kirori has previously been involved in Wireless Sensor Networks Research under National Commission for Science and Technology (NACOSTI) and has presented papers at various national and international conferences for the same. He studied his undergraduate degree from JKUAT, and has a Masters in IT from Kabarak University. He is currently undertaking Machine Learning research for security provision in MANETS for his Doctoral Studies.

LIST OF PUBLICATIONS

APENDIX E

RESEARCH PERMIT FROM NACOSTI

THIS IS TO CERTIFY THAT:
MR. KIRORI GATHUO MINDO
of KABARAK UNIVERSITY, 0-20100
Nakuru, has been permitted to conduct
research in Nakuru County
on the topic: A FUSED MACHINE
LEARNING MODEL FOR THE PROVISION
OF SMART HEALTH CARE MONITORING
for the period ending:
17th January, 2020

Permit No : NACOSTI/P/19/39487/27250
Date Of Issue : 17th January, 2019
Fee Received :Ksh' 2000



Applicant's Signature
Director General
National Commission for Science, Technology & Innovation

APENDIX F

NACOSTI LETTER OF INTRODUCTION



NATIONAL COMMISSION FOR SCIENCE, TECHNOLOGY AND INNOVATION

Telephone: +254-20-2213471,
2241349, 3310571, 2219420
Fas: +254-20-318245, 318249
Email: dg@nacosti.go.ke
Website : www.nacosti.go.ke
When replying please quote

NACOSTI, Upper Kabete
Off Wajaki Way
P.O. Box 30623-00100
NAIROBI-KENYA

Ref. No. **NACOSTI/P/19/39487/27250**

Date: **17th January, 2019**

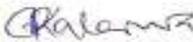
Kirori Gathuo Mindo
Kabarak University
Private Bag - 20157
KABARAK.

RE: RESEARCH AUTHORIZATION

Following your application for authority to carry out research on "*A fused machine learning model for the provision of Smart Health Care Monitoring*" I am pleased to inform you that you have been authorized to undertake research in **Nakuru County** for the period ending **17th January, 2020**.

You are advised to report to **the County Commissioner and the County Director of Education, Nakuru County** before embarking on the research project.

Kindly note that, as an applicant who has been licensed under the Science, Technology and Innovation Act, 2013 to conduct research in Kenya, you shall deposit a **copy** of the final research report to the Commission within **one year** of completion. The soft copy of the same should be submitted through the Online Research Information System.


GODFREY P. KALERWA MSc., MBA, MKIM
FOR: DIRECTOR-GENERAL/CEO

Copy to:

The County Commissioner
Nakuru County.

The County Director of Education
Nakuru County.