University of Ghana http://ugspace.ug.edu.gh

UNIVERSITY OF GHANA

DETECTION AND PREVENTION OF MAN-IN-THE-MIDDLE SPOOFING ATTACKS IN MANETS USING PREDICTIVE TECHNIQUES IN ARTIFICIAL NEURAL NETWORKS (ANN)

BY

KWADWO BOATENG OFORI-AMANFO
(10230319)

THIS THESIS/DISSERTATION IS SUBMITTED TO THE UNIVERSITY OF GHANA, LEGON IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MPHIL COMPUTER ENGINEERING DEGREE

APRIL, 2015

University of Ghana http://ugspace.ug.edu.gh

DECLARATION

I KWADWO BOATENG OFORI-AMANFO, author of this thesis, do hereby declare that the work presented in this thesis/dissertation ; "**Detection and Prevention of Man-In-The Middle spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN)**",is my work, produced from research undertaken under supervision in the Department of Computer Engineering, University of Ghana, Legon, from September 2013 to July, 2014. This work has never been presented either in whole or in part for any other degree of this University or elsewhere.

……..

KWADWO BOATENG OFORI-AMANFO

Date……...

This thesis has been submitted for our approval as supervisors

Dr. Robert Adjetey Sowah

(Principal Supervisor)

Date……

Dr. Koudjo M. Koumadi

(Co-Supervisor)

Date…….…….

University of Ghana http://ugspace.ug.edu.gh

Abstract

Detection and Prevention of Man-In-The-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN)

A Mobile Ad-Hoc Network (MANET) is a convenient wireless infrastructure which presents many advantages in network settings. There are however a lot of challenges faced on such networks; with their relatively much more flexible setups. They are comparatively more susceptible to attacks of various kinds as opposed to their wired cousins. This is due to the non-centralization of their overall architecture. Among these numerous attacks are spoofing, blackhole and man-in-the-middle (MITM) attacks. Each of the identified attack methods, have through numerous researches and discussions been tackled to a great degree. The focus of this particular research was on the man-in-the-middle attacks; presenting a possible way of simulating such attacks on the NS2 platform, and after that, using the results to create an Artificial Neural Network(ANN) based software for attack detection, blacklisting, and node reconfiguration; preventing future attacks in the process. The ANN generated provided, generated an average detected rate of 88.235 for packets. With an average true positive reading of 0.25 as oppose to an average false positive reading of 0.0399 from the confusion matrix of 12 test runs on the final detection model, the research not only offered a productive and less expensive way to perform mitm attacks on simulation platforms, but also identified time as a crucial factor in determining such attacks. It is expected that the findings of this research would present a viable way of porting know MITM attack software to the realized software solution in future; for attack signature creation. This is to train the ANN to offer better attack detection. Systems administrators would find such software useful and easily configurable on their networks -

University of Ghana http://ugspace.ug.edu.gh

making them more secure. The research is also intended to be an opening for future malicious software signature creation to supplement existing Intrusion Detection Systems (IDSs).

University of Ghana http://ugspace.ug.edu.gh

ACKNOWLEDGMENT

The first of all thanks and exhortation goes to my Father and King, the almighty God, for the strength to complete this research. My deepest gratitude goes to my supervisor, Dr. Robert Adjetey Sowah and co-supervisor Dr. Koudjo M. Kuomadi for all their support, ideas and directions and without whose help this project would have been far more difficult. I would also like to thank my parents and grandmother, for their invaluable prayers and support, as well as my siblings; Boadiwaa, Awura-Ama and Kofi for their encouragement. Additionally, my sincerest gratitude goes to my dear friend; Miss Natasha Asamoah for offering her assistance in the proofreading of this work.

Finally, I wish to acknowledge the **Carnegie Corporation of New York** through the University of Ghana, under the **UG-Carnegie Next Generation of Academics in Africa** project for financially supporting this research work. The help has been immeasurable and it is my sincerest desire that other African scholars would benefit from such a noble gesture.

INTEGRI PROCEDAMUS

University of Ghana http://ugspace.ug.edu.gh

TABLE OF CONTENTS

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

LIST OF FIGURES

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

LIST OF TABLES

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

LIST OF ABBREVIATIONS

AI. Artificial Intelligence

ANN. Artificial Neural Network

AODV. Ad-Hoc on-Demand Distance Vectoring

API. Application Programming Interface

DDOS. Distributed Denial of Service

DoS. Denial of Services

DSL. Digital Subsciber Line

DSR. Dynamic Source Routing.

GIG. Global Information Grid

GPS. Global Positioning System

HMAC. Hashed Message Authentication Code

IEEE. Institute of Electrical and Electronics Engineers

iMANET. Internet based Mobile Ad-Hoc Network

IP. Internet Protocol

LAN. Local Area Network

MAC. Message Authentication Code, Media Access Control

MAN. Metropolitan Area Network

MANET. Mobile Ad-Hoc Network

University of Ghana http://ugspace.ug.edu.gh

MITM. Man-In-The-Middle

NS2. Network Simulator 2

PAN. Personal Area Network

PDA. Personal Digital Assistant

PGP. Pretty Good Privacy

RFID. Radio Frequency Identification

RREP. Route Reply

RREQ. Route Request

SMB. Small and Medium Scale Business

TSF. Time Signature Figureprint

VANET. Vehicular Ad Hoc Networks

WiMAX. Worldwide Interoperability for Microware Access

WLAN. Wireless Local Area Network

WMAN. Wireless Metropolitan Area Network

WPAN. Wireless Personal Area Network

University of Ghana http://ugspace.ug.edu.gh

CHAPTER ONE

INTRODUCTION

1.0 Background

Computer Security is one of the areas in computer technology which has received a lot of interest from a lot of professionals and "lay" persons. This field was necessitated as a result of previously known and newly developing techniques, which affords attackers the means to launch sophisticated attacks; giving them access to resources on networks and compromising those networks in the process. Notable among such established techniques are **distributed denial of service (DDOS)** attack, **man-in-the-middle (MITM) spoofing** attacks, and **session hijacking**[1]. With man-in-the-middle spoofing attack; the main focus of this research, a third party-the attacker in this case, basically inserts himself between two parties or devices in stealth mode in such a way that all packets between those two legitimate parties, are routed through him. This is quite malicious because the attacker can then alter the information in the packets - potentially sending falsified data to either party[2].

It terms of categorization, MANETs could be seen as VANETs, internet based mobile ad hoc networks (MANETs), or military based MANETs. These wide categorizations imply that MANETs have very broad operational capabilities. However it is their diversity that

University of Ghana http://ugspace.ug.edu.gh

makes such networks very susceptible to the aforementioned attacks.

It is interesting to note that attacks are not just limited to particular devices. Mobile targeted attacks can be performed against small to very large targets and across multiple platforms. Even the various attacks could be broken down based on which software application they have been tuned to violate. The Man-In-the-Middle attack for instance has a slight variant called the Man-in-the-browser attack which is specific to browser based applications and services [3]. It is aimed at interception communications between several clients on browser platforms. Numerous variants of session hijack attacks and buffer overflow attacks exist; with such attacks, unlike in the past, beingautomatable via software tools. **Wireshark**, **Nmap**, **Tcpdump** are among the variety of tools available to today's hackers; even given rise to a new breed of hackers called click-kiddies who in comparison to the earlier breed of hackers, have relatively no to little programming. These are also capable of initiating sophisticated attacks against prime targets. With the sharp growth in the processing power of hardware, as well as the exponential development of software tools and programming languages, the amount of power available to a single user in a cyber-network, has never been greater than in today's global world. It is therefore no surprise that numerous initiatives and investments are being made to protecting wired and wireless networks from prying eyes. History shows that technological development has for centuries been military driven. It is therefore ironical to see the military turn to civilian experts in the area of cyber defense [4]. It represents possibly the one single area where civilians can compete on par with the

University of Ghana http://ugspace.ug.edu.gh

military; thus it has over the last couple of years offered numerous challenges to both worlds.

Warren McCulloch and Walter Pitts (1943) created a computational model for neural networks based on mathematical algorithms. They referred to it as threshold logic. Their research laid the foundation for several others into ANN and today such networks are used in many areas from banks, through academic institutions to the military.[5] Frank Rosenblatt created the perceptron after McCulloch's ground breaking work. The perceptron was an algorithm for pattern recognition founded on a two-layer learning computer network making use of arithmetic operations. However further research into the field was stalled until the **backpropagation** algorithm was developed; enabled by a jump in the processing ability of computers of its time and in the process, helping to solve existing problems relating to exclusive-or computations[6]. Other machine learning algorithms such as **support vector machines** have since then, traded places for several years with ANN as lead algorithms in machine learning but in recent times, interest has been rekindled in ANN with the possibilities presented by deep learning (a set of machine learning algorithms targeted at modeling high-level abstractions)[7]. It is against this backdrop that this research has been carried out to use ANN techniques to solve the problem of the man-in-the-middle attack.

University of Ghana http://ugspace.ug.edu.gh

1.1 Problem

The detection of man-in-the-middle attacks on MANETs is one of the major problems present on MANETs. Their detection and prevention is a very difficult area of research due to the subtle and stealth nature of such attacks.

1.2 Motivation

In the current world of social networking, Internet security and espionage, one of the areas of extreme concern to security experts is MANET (Mobile Ad-Hoc Network) security. Why Man-In-the Middle Attacks on Mobile Ad-Hoc networks? Mainly because, though that area of computer networks has been well researched, there is still some work to be done; particularly in automating the mode of attack detection. Most existing methods focus on protecting the network with static, non-flexible protocols. Thus, the desire to offer a much more flexible predictive method was what motivated my interest in this research.

University of Ghana http://ugspace.ug.edu.gh

1.3 Scope & Objectives

1.3.1 Scope

The research involved the introduction of neural network techniques and IP/MAC address mappings techniques to detect fraudulent nodes. The research focuses on the man-in-the-middle attack mode with the aim of coming up with a viable software model(s) for detecting, recovering from, and preventing future attacks.

1.3.2 Objectives

The objectives of this research are to:

- ✓ Ensure that the man in the middle cannot detect where packets are from on where they are going.
- ✓ Monitor the network for promiscuous nodes.
- ✓ Detect man-in-the-middle attacks particularly those using spoofing.
- ✓ Learn from the noticed scenario and adapt the network to counteract or curtail future exploits.

1.4 Design Methodology

Bearing in mind the lack of very clearly defined requirements and the fact that time-lines could change with the presentation of new challenges and constraints, the proposed

University of Ghana http://ugspace.ug.edu.gh

software process model chosen, was the evolutionary development model specifically using throw-away prototyping and making use of process iteration.

Throw-away prototyping is an evolutionary model, used mainly in cases where the requirements are not that clearly understood. It aims at understanding the variable requirements and hence developing a better set of requirements for the system. It focuses mainly on experimenting with those aspects of the user's requirement that are poorly understood. The model is shown below. Process iteration was incorporated to ensure that as at when requirements were finally well defined, they were also refined to bring out the best possible solution.

An advantage of this design method is that it often results in the production of systems that meet the immediate needs of customers. A disadvantage is that the process is not visible, thus clients or project supervisors need regular deliverables to measure progress. The model is shown below.

1.5 Process Model -Evolutionary Model

The Evolutionary model of development is based on the idea of developing an initial implementation, exposing it to user comments and refining this through many versions, until an adequate system has been developed; as demonstrated in the diagram below. Rather than have separate specification, development and validation activities, these are carried out concurrently and with rapid feedback across these activities –***Figure 1.1***.

University of Ghana http://ugspace.ug.edu.gh

There are basically two main types but the one employed was the throw-away prototyping method. The objective of this variant of the evolutionary model is to understand the users requirements and hence develop a better requirements definition for the system. The prototype concentrates on experimenting with those parts of the customer requirements which are poorly understood.

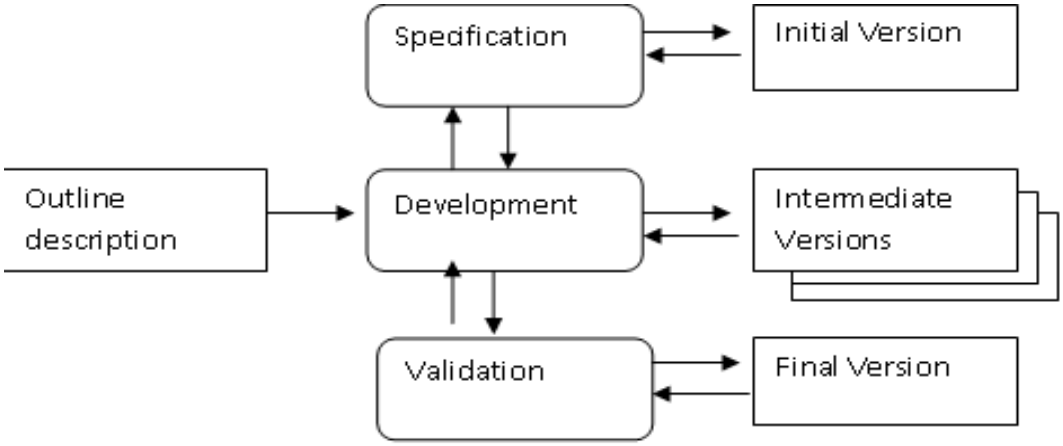


Figure 1.1 - Process Model

1.6 Key Assumption

Thus the underlying assumption of this project is that

Given a network of nodes; $n_1, n_2, \ldots, n_k$ *each with corresponding times of* $t_1, t_2, \ldots, t_k$ *it is possible to model an ANN capable of time signature fingerprinting (TSF) a man-in-the-middle attacking node with a time signature of* δt.

University of Ghana http://ugspace.ug.edu.gh

Where: δt is the latent time delay induced from initiating such an attack.

1.7 Thesis Outline

This research presents a relatively new and effectively new way to simulate MITM attacks. The introduction section of this work, gives a general overview of the research. It presents the problem to be addressed together with the motivation for this research. The scope and objectives of the research have also been highlighted. Design methodology, process model as well as the key assumption was also each discussed. Thus the ground has been well laid for the literature survey section covered in Chapter 2. Chapter 3 covers the system design process and development discussion. Chapter 4 then presents the system implementation and testing phase with Chapter 5 offering results and the discussion of those results. Chapter 6 shows the conclusion and recommendations arrived at. The last section of this document holds the various Appendixes.

1.8 Wireless Networks

From the first generation (1G) through second generation (2G and 2.5G) to the fourth generation (4G) of wireless and other current existing technologies; spanning a period of

University of Ghana http://ugspace.ug.edu.gh

over two decades, there have been relentless developments in wireless research and their implementations. Some examples include; Wifi, WiMAX, Blue-tooth, and Infra-red as, shown in ***Figure 1.2.*** With improvements in transistor technology which has effectively resulted in the realization of Moore's Law across different planes; making the production of electronic components cheaper and faster, there has been a renewed interest of research into mesh technologies for the later part of the twentieth century. To support this level of research - currently ongoing, standards such as IEEE802.15.4 and Zigbee have also been developed; spurring on parallel researches in wireless sensor networks. These relative new fields crave the development of innovative protocols to solve problems relating to power, efficiency, security and scalability. Thus in additions to hardware developments, many papers are published yearly on these topics.

University of Ghana http://ugspace.ug.edu.gh

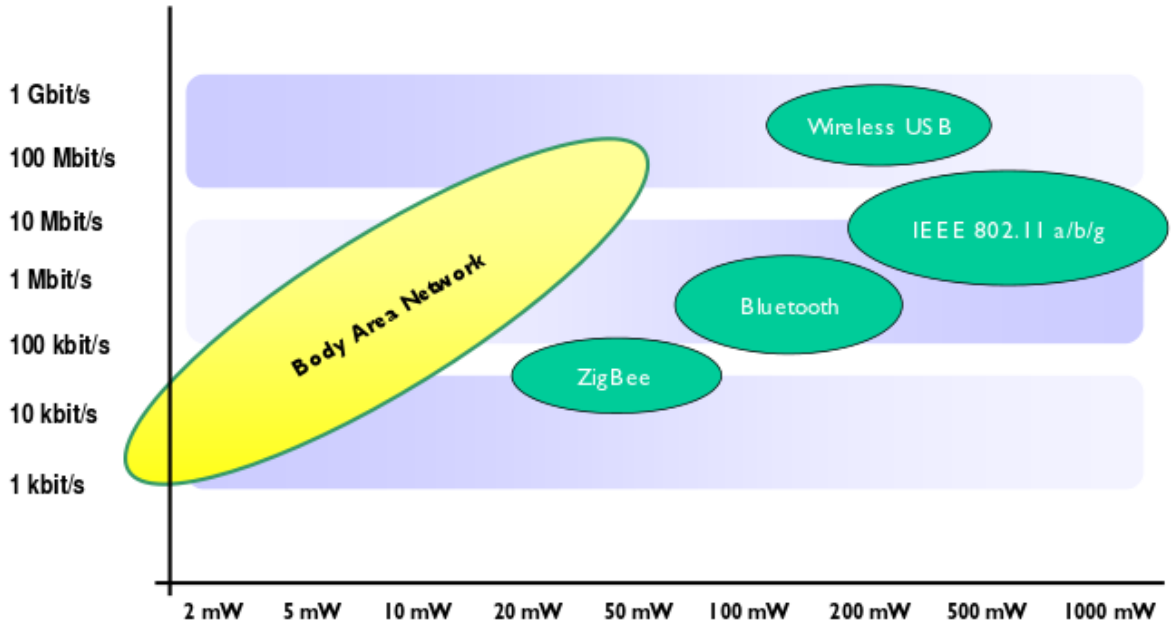


Figure 1.2 - Different developments in wireless technologies

New multilayer protocols have to address each of the aforementioned problems in various degrees and are expected to work very well if they are to gain widespread acceptance. Network complexity and changes have also warranted the production of tools to simulate or emulate and preempt network processes and reactions. The aim of these approaches is to come up with novel and newer ways to configure and maximize the network outputs of the currently existing technologies and newly developed ones. Thus, tools such as **NS2**, **Virtual Machine/VMware** and **Wireshark** have gained widespread acceptance among networking enthusiasts. Such tools offer and provide ways to implement and alter existing protocols. There are approaches to simulation based research; each with its benefits and deficiencies. Small simulation based models, which just make use of

University of Ghana http://ugspace.ug.edu.gh

standards, provide superfluous information which adds relatively no new information to what is already known. Larger simulation models, which to a greater extent, mimic real life scenarios are time consuming and resource intensive. Thus, their appeal is usually not that much. Analytical approaches also exist. These aim at ironing out the problems that exist in the previous two by employing mathematical models to analyze existing protocol from scratch or their parameters, by making predictions as to how they would behave when scaled up or down.

The Types of Wireless Networks

Table 1-1- Types of wired and wireless networks

NETWORK TYPE	WIRED	WIRELESS
LAN	IEEE 802.3 (Ethernet)	IEEE 802.11X
MAN	Broadband (DSL, cable)	IEEE 802.16
PAN	IEEE 1394 USB	IEEE 802.15.1 IEEE 802.15.3

University of Ghana http://ugspace.ug.edu.gh

		IEEE 802.15.4

The Institute of Electrical and Electronic Engineers (IEEE) provides both wireless and wired standards dictated by the 802 series for industry users to adopt. [8]

1.8.1 WPAN

The Wireless Personal Area Network (WPAN) is formed by using technologies that are required to use close range interaction such as Bluetooth, at around 30 feet or 10 meters, making use of the IEEE 108.15 standard. As the name suggests, WPAN is very useful for work around a personal space. Devices such as electronic tables, smart phones, laptops and Personal Digital Assistant (PDA) within the radius of operation, can all be linked to create an ad-hoc network with very low power considerations. The convenience this form of network offers, by cutting out the use of wires and cables, ensures higher productivity and comfort in one's workspace. The Bluetooth protocol supports a 1Mbps data transmission rate. Earphones and headsets, including Modems, all have their various implementations with this standard. Cross platform file and electronic media sharing is supported, and optimization considerations, make the standard indeed very attractive; especially Bluetooth - which has found widespread use. WPAN is sometimes called the Personal Area Network; "PAN" for short. [8]

University of Ghana http://ugspace.ug.edu.gh

1.8.2 WLAN

Educational Institutions, Banks and Large corporate organizations are all now using Wireless Local Area Networks (WLANs); the wireless equivalent of LAN shown in *Error! Reference source not found.*. It uses the IEEE's 802.11 standard. Finding its use in reas quite larger than WPAN, WLAN enables users to roam around working environments, while still connected to the network.

Thus, work does not necessarily have to cease when one changes his or her location; meaning that he or she could continue to work on the move. These networks can also be set up on devices such as laptops, tablets and phones. The applications on such electronic media rely on session cookies and other software tools, which require these networks to be up; in order to maintain sessions or transfer sessions across multiple media. In places like Supermarkets, such technologies can be used in conjunction with Radio-Frequency Identification (RFID) based systems, to ensure the security of products under sale. Coupled with Internet access, a whole large group of users can be granted limited access to Internet services, relevant for work required duties to be carried out. Emails, Research and Video Chat meetings could be supported within a well-defined space; bearing in mind security implications as well as the setting of the right boundaries and user privileges. Both small and medium businesses (SMBs) can utilize this form of network, to cut down the cost of network setup while maintaining high productivity. In some cases such as; highly active factory floors, WLAN represents the only realistic option for

University of Ghana http://ugspace.ug.edu.gh

network infrastructural support. [8]

1.8.3 WMAN

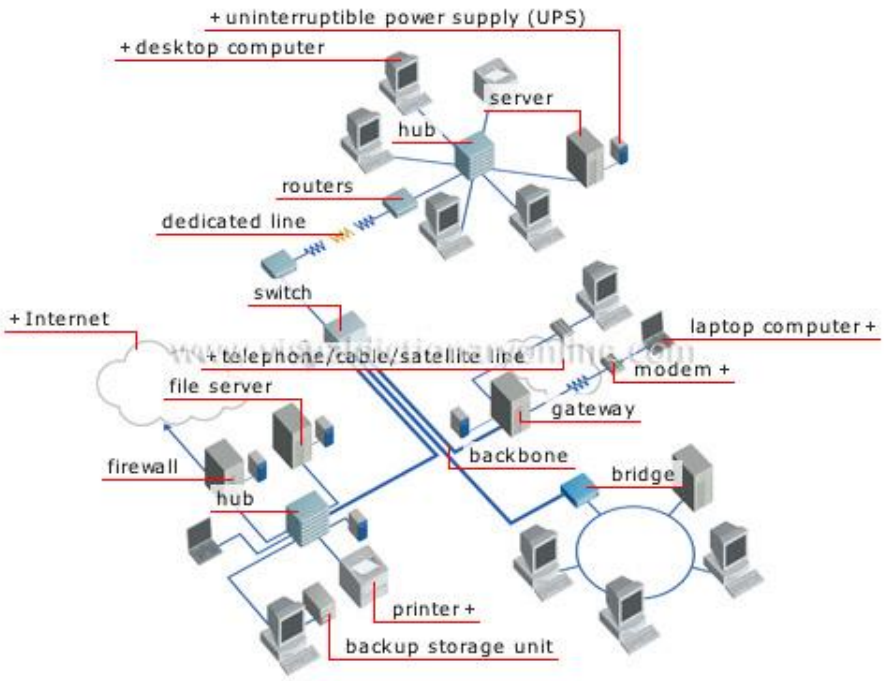


Figure 1.3-WMAN

Also known as Wi-Max or WirelessMAN, Wireless Metropolitan Area Networks (WMAN) as shown in ***Figure 1.3***, is chosen over the previously discussed standards, in scenarios where the distances under consideration, are far greater than of those needed for the prior standards. Longer distances and high speed access are both catered for under the

University of Ghana http://ugspace.ug.edu.gh

IEEE 802.16 standard, which is necessary for WMAN provision. With bandwidth usages in the range of 10-66 GHz, the main focus is on the efficient use of that bandwidth. Obviously, if packets are being transmitted over a very far distance, the last thing one would want to see are packet drops. Further improvements in the standard increased the range from 2-11 GHz under the 802.16a standard. A newer standard -802.20, even provides connectivity speeds of up to 1 Mbps for vehicular traffic of speeds of up to 250 kilometers per hour. When working in an organization that is housed in just one building; usually spanning a few floors, WLAN might be sufficient to cover all networking needs. However, when the distance is extensive and includes multiple buildings - within a fairly large geographical region, WMAN with its antennae and base stations offer a more feasible method for offering connectivity. The use of WLAN guarantees a quality of service that vendors are required to adhere to. It is quite common for business entities to buy space being set up by such vendors with the entire infrastructure already set up. An example of this was such a service offered by VeriLAN Inc. in 2004 around Portland, Oregon.[8]

1.9 Mobile Ad-Hoc Networks

MANETs fall under wireless technologies. It is a configuration of nodes, in which each participating unit acts as both a host and a router. Thus there usually is no dedicated node to manage the network. It can be established between two or more participating nodes.

University of Ghana http://ugspace.ug.edu.gh

Additionally, there are several routing algorithms present to support routing in MANETs as well as several others to maintain security. Algorithms such as the AODV (Ad-Hoc on-Demand Distance Vectoring) and DSR (Dynamic Source Routing) offer methods of establishing and maintaining connectivity. The majority of security infrastructures are reactive in nature and are also partially preemptive (have static configurations to react to attacks). The best of them currently, is the PGP (Pretty Good Privacy) security solution. Major concerns in wireless research seek to address the balance between security and power efficiency. This work does not really towe that line. Rather, it seeks to offer a more dynamic response to the threats presented on such networks.

1.9.1 Definition

MANETS, as explained earlier, fall under wireless networks. In the light of that broader scope, they maintain most of the benefits and deficiencies of that group of networks. For the benefits, there are; flexibility, easy scalability and the convenience of use.

Flexibility: MANETs are easier to set up and take down as opposed to Ethernet based networks, or those with more physically constraining setups. As such, they offer an attractive option in areas where physical infrastructure is bad or insufficient.

Easily Scalable: New nodes can also be easily added to an existing MANET set up.

University of Ghana http://ugspace.ug.edu.gh

Thus, unlike most physical infrastructures that take a lot more time and money to set up, newer units can be added faster; if scale ups are required. Incidentally, this level of convenience is also what presents such a network with the majority of it threats

Self-Healing: Nodes can easily join and leave the network without affecting the normal functioning of the network.

Convenience: The convenience stems from the fact that, most of the wireless sets ups can easily be configured without wires, so, for places where the topology of the working environment is irregular such as cross floor factory set ups, or around places where wired setups could serve as work hazards, MANETs present very attractive alternatives. [9]

1.9.2 Some Existing MANET Applications and Their Uses

Military and Tactical Networks: Military networks use MANETs to maintain communication between their units; providing the military with a well networked and automated system, albeit extremely secured ones for relaying tactics and information and ensuring real-time transmission of critical information to ensure success during military campaigns. During military campaigns, there are very limited scenarios especially on the field of battle in which wired networks can be employed. Humvees, jets and armored vehicles are all linked using wireless technologies as illustrated in ***Figure 1.4.*** Driven by advancements such as; the Global Positioning System (GPS) and other cutting-edge

University of Ghana http://ugspace.ug.edu.gh

satellite based technologies, the push to serve lifesaving information to personnel on the battlefront is of the utmost importance. Under such situations, fixed networks are deficient in many ways. They cannot adapt and support feature-rich and non-confinable data. Fighter jets need ways to stay in touch with ground troops. The navy also needs ways to be plugged in to main-land soldiers. All this cannot be accomplished by wired networks. Wireless networks need to be used is such cases. The standardization of IP networks, with the aim of achieving the objectives outlined in the high-level Global Information Grid (GIG) and Network Operations' (formerly Network Centric Warfare) doctrines, were some of the steps taken to meet the demand for wireless solutions. The goal of Network Operations, is to provide seamless access to timely information to all war-fighters and decision makers at every level in the military hierarchy. This enables soldiers, ground vehicles, and aircraft and command centers to shape collected information into a coherent, accurate view of the battlefield. This enables soldiers, ground vehicles, and aircraft as well as command centers to shape collected information into a coherent and accurate view of the battlefield. In military environments (where confrontation is expected), the realistic option would be set up mobile wireless networks for communication by employing specialized IP-enabled radios and military-grade routers which combine to form several IP-nodes. The radios that make up these kinds of networks are called backhaul radios. They transmit and receive information to and from other such radios and also satellites. The nodes, upon aggregation, provide router capabilities for the network. Military personnel in possession of authorized client devices

University of Ghana http://ugspace.ug.edu.gh

such as laptops, cameras and PDAs could then piggy-back on the network and connect to the same IP-node. This is depicted in ***Figure 1.4***.[10]

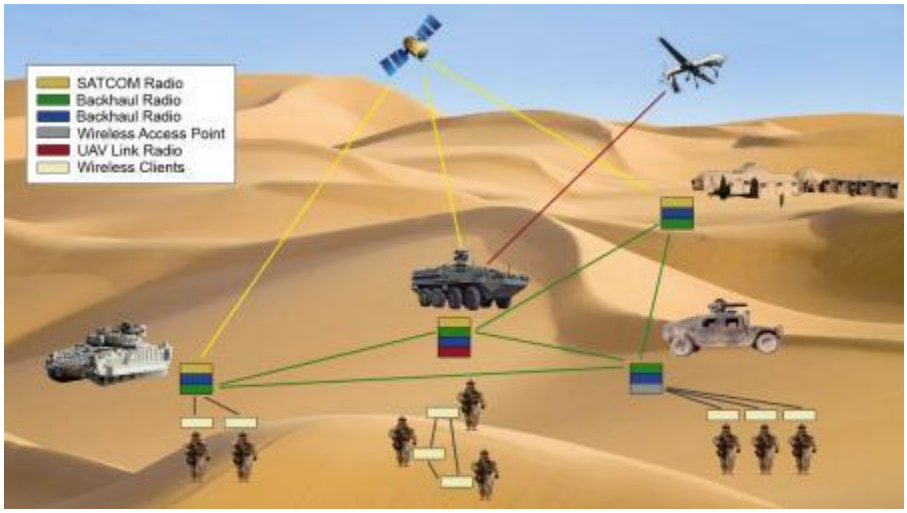


Figure 1.4 - Military scenario of MANET use[10]

Education: Ad-Hoc network is also useful in educational settings; where students and other groups could set up such networks, to shared material, play games and participate in many network related activities, required for smooth academic work to take place.

Context Aware Services: Follow-on services such as call-forwarding, mobile workspace , as well as information services like location specific services, time dependent services ; Provided by companies such as Google and Apple all employ ad-hoc technologies in

University of Ghana http://ugspace.ug.edu.gh

their delivery. They help to personalize and improve user experience and take mobile interactions to a whole new level.

Entertainment: Ad-Hoc networks and wireless networks in general are employed as the backbone to most multi-player game system currently in existence. Here, users can set up peer-to-peer networks and engage each other on a provided game platform, within a defined geographical location. The upsurge in smart phone technology has opened up new possibilities for the use of wireless technologies, particularly those needing mobile ad-hoc capabilities. Consumers' demands; to be able to link their electronic devices to shared music and other electronic content, across multiple hardware platforms. These, aimed at bettering their experience, keeps pushing the envelope of development for what could be done in the field.

Emergency Response: Like the scenarios offered by military cases, emergency cases also need Ad-Hoc networks to respond to unforeseen circumstances and problems relating to human life and safety. Ambulances need current technology for traffic routing information and to save the lives of their patients.

University of Ghana http://ugspace.ug.edu.gh

1.10 Machine Learning

Machine learning is the "***field of study that gives computers the ability to learn without being explicitly programmed" (Samuel A., 1959).***A well posed learning problem under machine learning is usually defined, as by Tom Michell (1999) to be; ***a computer program that learns from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*** [11]. For example, a machine, learning to play the game of chess might improve its performance "P" as measured by its ability to win games under the class of tasks "T" of playing chess games against others, by making use of the experience acquired in preceding games, played against itself. Thus the chess learning problem can be formulated as this;

- ✓ Task-T: playing chess
- ✓ Performance measure - P: percent of games won against opponents
- ✓ Training experience - E: playing practice games against itself

Since the inception of computing systems, scientists and engineers have wondered whether machines can be automated and made to learn on their own. This resulted from studies into the fields of Artificial Intelligence (AI) and machine learning. Machine

University of Ghana http://ugspace.ug.edu.gh

learning, which came out of AI, further spawned researches into other areas; Robotics, Fuzzy Logic, Artificial Neural Networks; just to mention a few. Scientists are of the view that the best way to approach Machine Learning, is to mimic the ways humans learn. Although machines are currently not able to learn nearly as well as humans, there has however been the development of algorithms that make them faster and effective at some specialized tasks. Today, such algorithms are used in face recognition systems, fingerprint authentication systems, car manufacturing systems, real-time computing systems and in robotics. The broad field of data mining, which is used by some Internet giants such as Google and Microsoft in their search engines and even by supermarkets, employs these machine learning algorithms to a very great degree. Email service providers use machine learning to filter out spam. Online advertisements are tailored to suit individuals because of machine learning algorithms based on crowd-source information. ***Table 2.2*** shows some of the few feats achieved by machine learning algorithms which today highlight, even how state-of-the-art that the field was, some years back and still currently is. [11]

Table 1-2 - Achievements of Machine Learning

Machine Learning achievements	**Reference**
Machine learning methods, as used in the	Pomerleau 1989

University of Ghana http://ugspace.ug.edu.gh

ALVINN system (Pomerleau 1989) have been used to train computer-controlled vehicles to steer correctly. ALVINN for instance has used its learned strategies to drive without human intervention; reaching a speed of 70 miles per hour for 90 miles on public highways, with other cars present. Similar techniques have been implemented in Google's driver-less cars used for mapping services.[9]	
Play games such as backgammon reaching performances in the region of world champion-like, and human levels.	Tesauro 1992-1995
Predicting recovery rates of pneumonia patients.	Cooperet al. 1997
Machine learning methods have been applied to a variety of large databases to learn general regularities implicit in the	Fayyad et al. 1995

University of Ghana http://ugspace.ug.edu.gh

data. For example, decision tree learning algorithms have been used by NASA to learn how to classify celestial objects from the second Palomar Observatory SkySurvey (Fayyad et al. 1995). This system is now used to automatically classify all objects in theSky Survey, which consists of three terabytes of image data.	
Program that successfully recognize spoken words.	Waibel 1989; Lee 1989

1.11 Learning Algorithms Types

Supervised

Supervised machine learning methods are probably the most commonly used approaches. It is essentially a learning technique for creating a function based on training data. They are given a set of instances – training data; upon which the algorithm is trained; being fine-tuned through both systematic approaches (cross-validation) as well as try and error approaches. The final model can then be used on totally new data, to classify the data into their predefined classes.

University of Ghana http://ugspace.ug.edu.gh

Unsupervised

With unsupervised learning algorithms such as the **k-Means** algorithm, there is not really any predefined class vector for classification. One of the major methods used by the techniques under this group is **clustering**. The **simple k-means** algorithm offered by the weka toolkit was used to get a fair idea of which features could play a key role in the intended classification algorithms, in this research.

1.12 Supervised Learning Algorithms

Artificial Neural Network (ANN)

Artificial neural networks (ANNs), offers generalized and practical methods for learning real-valued, discrete-valued, vector-valued and matrix valued functions. BACKPROPAGATION, an algorithm under ANN makes use of the gradient descent to fine-tune network parameters (weights and features) to best fit a training set of input-output pairs. ANN learning is robust to errors in the training data; having been successfully applied to problems such as interpreting speech recognition, supervised robotic learning strategies. [11] ANN is just one of the numerous learning algorithms that is used in machine learning, and falls under the supervised class of learning algorithms.

University of Ghana http://ugspace.ug.edu.gh

Neural network techniques offer viable models for the realization of real, discrete and vector valued functions. Problems related to data gathered from sensors have some of their best solutions achieved, when artificial neural networks are applied as learning methods. An example is the **Backpropagation** algorithm - which has found several applications in many areas. These include - handwritten character recognition (LeCun et al. 1989), face recognition (Cottrell 1990) and speech pattern recognition (Lang et al. 1990). [11]

Inspiration for Neural Networks

Interest in Artificial Neural Networks came out of researches to mimic biological neural network systems. A typical example, is the human biological system - specifically the brain[12]. The realization was that, although computing systems were improving in processing and computing power, they couldn't really outperform humans in tasks such as face recognition and color differentiation. Despite the fact that the fundamental unit of computers; the microprocessor was essentially many tenths faster than its corresponding biological equivalent; the neuron, it was far slower in other complex tasks. The reason being that, neurons worked in parallel to form an elaborate system, which are millions of times faster than a single unit or multiple units of microprocessors. Where microprocessors excelled were in routine and dedicated tasks such as mathematical computations.

University of Ghana http://ugspace.ug.edu.gh

The human brain has an estimated 10^{11} neurons each connected to 10^{4} others. Switching time among neurons is quite small, compared to the speeds on computers. That is10^{-3} seconds as opposed to 10^{-10} seconds; but surprisingly, complex decisions take place faster in humans than in computing systems. For example, one takes approximately10^{-1} seconds to make visual recognition of a person he or she is acquitted with. This has led to conclusions by many researchers that, the parallel workings of our neural network, accounts for the higher processing capabilities of humans, when it comes to such complex tasks and it is the desire to achieve such levels of performance in computing systems, which has caused and developed interest in neural network research. Although most artificial neural network software is run on sequential machines, which essentially emulate distributed processes, there exist parallel, customized machines to implement the faster versions of those algorithms. Considering the fact that the motivation for ANN research comes from biological systems, it must be mentioned that not all the characteristics of those biological systems have or can be mimicked as precisely as would have been wanted. Researchers into the field are of two categories. Those with as interest to come up with more efficient machine learning algorithms, which either mimic their bedrock biological systems or not, and the second group, that mainly focuses on realizing fundamental biological system as close as possible in computing environments. [11]

1.12.1 Perceptrons

One type of ANN system has, as its basic unit, the perceptron. This is shown in ***Figure 1.5***. The perceptron, which is a model of the biological neuron as shown in ***Figure 1.6***

University of Ghana http://ugspace.ug.edu.gh

takes in a matrix of vector input, using it to calculate a linear combination of inputs; outputting a value **y1 or y2** obtained by feeding the linear combination to an activation unit. The discrete value given by the system is based on the threshold set by using the activation formula. This can be explained mathematically as; given inputs $x_i$through tox_n, the output$o(x_i, \dots, x_n)$ computed by the perceptron is.

$$o(x_i, \dots, x_n)=\begin{cases}1 & x_0 + x_1 w_{1n} + x_2 w_2 + \cdots + x_n w_n > setthreshold\ (usually\ 0) \\ -1 & otherwise\end{cases}$$

(1.0)

Where$w_n$is a real value constant known as the weight which estimates the contribution of its corresponding$x_n$value to the output of the perceptron. Using the ***perceptron training rule,*** coupled with the ***gradient descent method*** or ***delta rule*** (a slight variant of gradient descent) , the right weights can be achieved through iteration, to correctly classify each instance in the training set, to give a viable model for classification of test sets. The general idea is to get the correct vector of weights that would result in the correct ± 1 value for each example. [11]

University of Ghana http://ugspace.ug.edu.gh

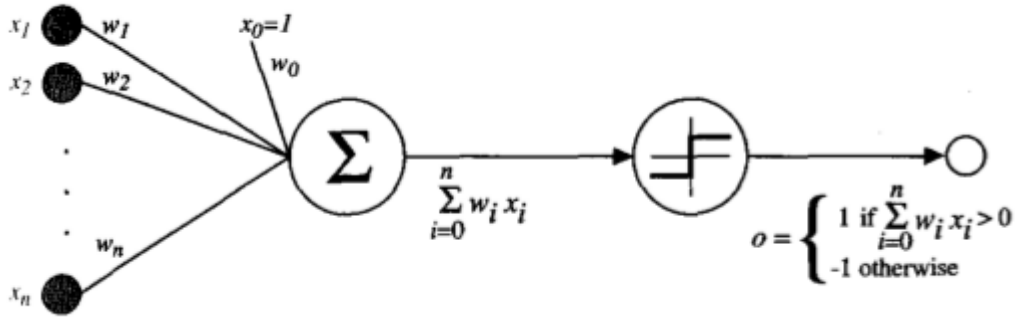


Figure 1.5 – Perceptron[11]

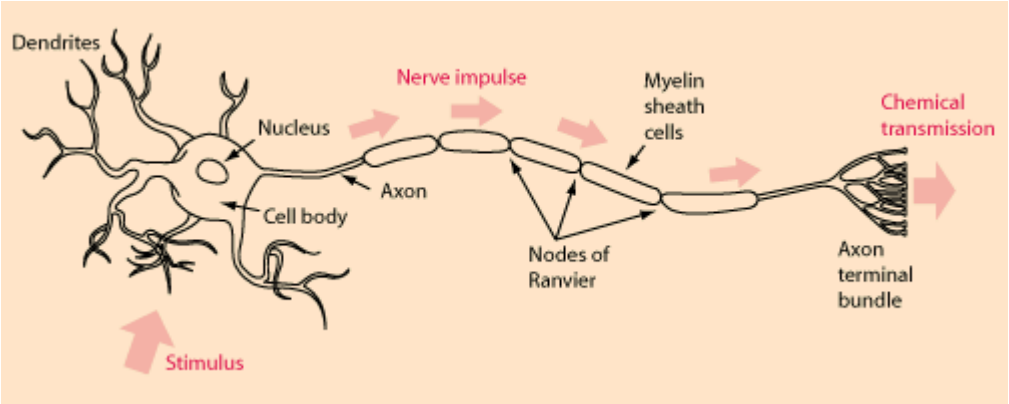


Figure 1.6 – Biological Neural Structure

1.12.2 Perceptron Training Rule

The aim of this rule is that, by means of iteration, and under certain conditions, the learning algorithm is supposed to converge to a certain hypothesis. To achieve a good weight vector, the algorithm first begins with some randomly generated weights; then iteratively applies the perceptron to each example in the training set, with modification of the weights with each instance of misclassification. The process is repeated with iteration taking place through each example until all the given examples are correctly classified

University of Ghana http://ugspace.ug.edu.gh

based on the perceptron training rule - aimed at weight alterations. The rule indicates that for weight$w_i$of input x_i;

$$w_i + \Delta w_i \rightarrow w_i \quad \textbf{(1.1)}$$

Where

$$\Delta w_i = \eta(t - o)x_i \quad \textbf{(1.2)}$$

t: target output

o: output generated by the perceptron and

η: positive constant (learning rate)

The learning rate which is usually a very small value, like 0.1, is used to moderate the degree to which the weights are changed in each step. The perceptron training rule is most useful when the example under consideration, is linearly separable. For non-linearly separable situations, the ***delta rule*** is used. [11]

1.12.3 Delta Rule

This is used when the examples cannot be separated linearly, since in such cases, the perceptron rule could fail to converge. It is used to overcome and to compensate for the problems that arise in such situations. For non-linearly separable examples, the delta rule converges to a best-fit approximation of the target. What the delta rule seeks to investigate is to use the method of gradient descent to navigate the hypothesis space,

University of Ghana http://ugspace.ug.edu.gh

seeking the weights that give the best fit to the examples under consideration.

This rule, also known sometimes as the ***least-mean-square***, which forms the underpinning of the BACKPROPAGATION algorithm, is thus very important; since it is used in networks with several linked units aimed at learning. Additionally, if the hypothesis space has different types of non-discrete parameterized hypotheses; gradient descent offers a fitting solution for learning algorithms that must search through that space. For an unthresholded perceptron; that is a linear unit with output **o** given by;

$$o(\vec{x}) = (\vec{w}).(\vec{x}) \qquad \textbf{(1.3)}$$

; indicating a linear unit at the first stage of a perceptron that is without the threshold, the learning rule for weight alteration is derived by finding an error measure given the sum of the difference between the error from the output offered by the hypothesis, relative to the expected output. As indicated by the training examples shown in the equation below, the output $E(\vec{w})$ is shown as;

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d\epsilon D}(t_d - o_d)^2 \qquad \textbf{(1.4)}$$

Where

University of Ghana http://ugspace.ug.edu.gh

D: set of training examples

t_d: The target output for training example

d, and

o_d: The output of the linear unit for training example d.

Thus $E(\vec{w})$ is half the squared difference between the target output t_d and thelinear unit outputo_d, summed over all training examples. Considering any given instance of the training examples as fixed during training, we express E as a function of $\vec{w}$, thereby defining it to be a function that responds to the changes in the weight vector alone. [11]

1.12.3.1 Derivation of the Gradient Descent Rule

In order to calculate the direction of the steepest descent along the error surface, the derivative of E with respect to each component of the vector $\vec{w}$is computed. It is known as the gradient of E with respect to $\vec{w}$. It is represented as $\Delta E(\vec{w})$ and is itself a vector, formed by the partial derivatives of E with respect to each of the weights w_i.

$$\Delta E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}\right] \quad \textbf{(1.5)}$$

This vector in the weight space indicates the gradient; the direction that indicates path of

University of Ghana http://ugspace.ug.edu.gh

steepest ascent. As such, the negation of this vector gives the path of sharpest decrease; that is $\Delta E(\vec{w})$. The training rule for gradient descent is

$$w + \Delta w \rightarrow w \quad \textbf{(1.6)}$$

Where

$$\Delta w = -\eta \Delta E(\vec{w}) \quad \textbf{(1.7)}$$

Here, as in the case of the delta rule, η is a positive constant called the learning rate, which indicates the step size to use for the gradient descent search. The negative sign is present because we want to move the weight vector in the direction that decreases E. This training rule can also be written in its component form;

$$w_i + \Delta w_i \rightarrow w_i \quad \textbf{(1.8)}$$

Where

$$\Delta w_i = \eta \frac{\partial E}{\partial w_i} \quad \textbf{(1.9)}$$

This makes it clear that, steepest descent is achieved, by changing component-wise each w_i, of $\vec{w}$ in proportion to $\eta \frac{\partial E}{\partial w_i}$. In the generation of a viable algorithm for iteratively

University of Ghana http://ugspace.ug.edu.gh

updating weights accordingto **Equation (1.9)**, an efficient way of calculating the gradient at each step is sought. The vector $\frac{\partial E}{\partial w_i}$, which presents the derivatives that forms the gradient could be computed as follows:

$$\frac{\partial E}{\partial w_i} = \frac{d}{dw_i}\frac{1}{2}\sum_{d\epsilon D}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\epsilon D}\frac{d}{dw_i}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\epsilon D} 2(t_d - o_d)\frac{d}{dw_i}(t_d - o_d)$$

$$= \sum_{d\epsilon D}(t_d - o_d)\frac{d}{dw_i}(t_d - \vec{w}_i.\vec{x}_d) \quad \textbf{(1.10)}$$

$$\frac{\partial E}{\partial w_i} = \sum_{d\epsilon D}(t_d - o_d)(-x_{id}) \quad \textbf{(1.11)}$$

Wherex_{id} represents the single input component x_i, for training example denoted ***d***. The equation presents $\frac{\partial E}{\partial w_i}$ in terms of the linear unit inputs x_{id}, outputs o_d, and target values t_d associated with the training examples. Substituting **Equation (1.11)** into **Equation (1.9)** yields the weight update rule for gradient descent

$$\Delta w_i = \eta \sum_{d\epsilon D}(t_d - o_d)(x_{id}) \quad \textbf{(1.12)}$$

In summary, the gradient descent algorithm for training linear units is applied by picking an initial random weight vector. After applying the linear unit to all training examples, Δw_i is computed for each weight according to **Equation (1.12)**. Thereafter, each weight

University of Ghana http://ugspace.ug.edu.gh

w_i is updated by adding Δw_i, and the process is repeated. Knowing that the error surface contains only a single global minimum, it is understandable that this algorithm will converge to a weight vector with minimum error, regardless of whether the training examples are linearly separable; given that a sufficiently small learning rate η is used. The gradient descent search is at the risk of overstepping the minimum in the error surface rather than settling into it if $\eta$is too large. Forthis reason, one common modification to the algorithm is to gradually reduce the value of η as the number of gradient descent steps grows. [11]

1.13 AODV Routing Protocol Overview

AODV is a reactive protocol by nature. The network is expected to generate routes when communication commences. Each node having been tagged with a specific sequence number, increases upon link changes and makes its judgment as to the current state of a channel based again on the sequence numbers it has available in its routing table. This is shown in ***Figure 1.7***. A source node S1 wishing to communicate with a destination node D1 first looks at its routing table to find out if a recent path to the destination node exists. If that is unsuccessful, it then sends out a broadcast request - RREQ to both A and F with an increment in RREQ ID, each time such a request is sent out from S1. It is up to A and F to decide if any incoming RREQ is a repeated one - upon which it is discarded, or a new once. If the latter is true, they each check their route

University of Ghana http://ugspace.ug.edu.gh

details, to see if either of them or both has a path to the destination. If there is any such positive feedback, either one or both of them send back an RREP response to S1 with the destination sequence number made available to the source. In the case where there are multiple RREP responses, the destination sequence number which is highest is taken and the path to the destination is thus chosen. There is however the possibility that neither A or F would return such RREP response, in which case another chain of RREQ requests might ensue along the different paths until a valid RREP response is obtained. This is illustrated in the diagram below where path S1-F-E-D1 gives the correct path to the destination.

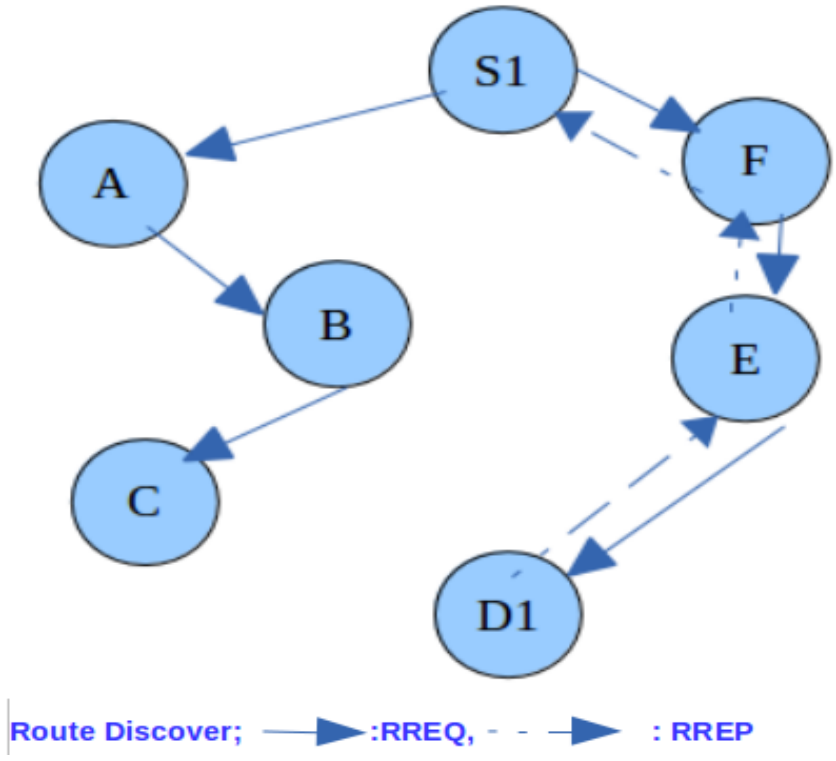


Figure 1.7 - AODV routing protocol

University of Ghana http://ugspace.ug.edu.gh

1.14 Different Kinds of Attacks

Black Hole Attack

This is a classic case of false advertisement for luring packets through a node. The attacking node, by placing the highest sequence number in its routing table and advertising it, presents itself as the shortest path to a desired destination. It therefore takes advantage of the routing protocols specification, to perform its attack. Upon acceptance as the route of choice for the packets, the malicious node proceeds to drop all incoming packets thus preventing them from reaching the desired destination hence the name "black hole"

Gray Hole Attack

Gray hole is initiated by the attacking nodes responding to RREQ messages with a consistent broadcast of RREP messages; a process known as routing misbehavior, until a connection is established. Having initially been inserted between the packet source and destination, packets are however dropped after that, leading to resource consumption. Repetition of that cycle initiates a denial-of-service attack, culminating in excessive resource starvation.

University of Ghana http://ugspace.ug.edu.gh

Denial of Service (DoS) Attack

In a denial of service attack, the attacker basically tries to prevent the legitimate user of the resources and services on the network from having access to them. It usually takes the form of; the flooding of a critical resource distributer or manager with requests. This has the effect of preventing legitimate users of the network from having access to those resources and making the network inactive. Due to the relatively freer structure of Ad-hoc networks to malicious users, this type of attack could be lunched in more diverse ways on such networks.

Distributed Denial of Service (DDoS) Attack

Distributed denial of service attacks are a variant of the previously discussed denial of service attack, having several attack vectors perpetuating the action. Thus the attacker has several machines under his or her command from which he or she lunches such attacks. It has the same effects as a DoS attack, albeit much more serious consequences in this case.

MITM Attack

This is the attack of consideration in this research and is where a malicious node inserts itself in between the source and destination nodes by spoofing and utilizing packet dropping techniques. This could be initiated by changing the routing table of the victim to route all the packets of interest through the attacker where they could either be forwarded

University of Ghana http://ugspace.ug.edu.gh

or read (harvesting user credentials and other information) for possible future attacks. It is among the stealthiest of attacks available to most attackers. The reason for the investigation of this attack time is because MANETs are essentially wireless networks and most wireless networks have considerably lesser restrictive security features or checks in comparison to their wired counterparts. MANETs by nature require even lesser security restrictions to facilitate their flexibility and interaction. The assumption therefore is that they are even more susceptible to the various attacks under discussion. Thus MANETs are not only vulnerable to all the above discuss attack modes but also warrant a different, dynamic and non-restrictive security techniques to combat the most stealth of attack types which in this case is the man-in-the-middle attack. The advantage to be gained from using neural network is that not only would flexibility be available from the minimal use of conventional security infrastructure such as password checking, but also the network would offer an adaptable means of intrusion detection subliminal to traditional security measures.

University of Ghana http://ugspace.ug.edu.gh

CHAPTER TWO

LITERATURE REVIEW

2.0 Literature Review

The central problem is the difficulty in detecting, and counteracting man-in-the-middle spoofing attacks on MANETs; this relates to security on wireless Mobile Ad-Hoc Networks (MANETS). These networks are quite unsecured since they offer opportunities by which hackers can get and exploit resources on them [13]. The above discussed attacks are all possible on these networks. This has spawned several researches into maintaining their security, some of which have been indicated in the previous section. My research, introduced neural network techniques fraudulent nodes. The research focused on the man-in-the-middle attack technique which is one of the most difficult to detect. Owing to the problem unauthorized persons such as hackers, pose on such networks, there have been many works in the field of network security each adopting unique techniques to accomplish their purpose. There have been works such as; "*Secure Data Protocol for Distributed Wireless Sensor*" [14] "*Networks, Secure Program Execution in Wireless Sensor Networks*" [14], and "*Intrusion Detection Model in MANETs using ANNs and ANFIS*" [15]. The above mentioned instances highlight the broad spectrum of researches having been carried out as well as currently ongoing; from

University of Ghana http://ugspace.ug.edu.gh

intrusion detection, tracing and prevention, cryptography, and also network monitoring. It is against the background of such good works, that this research sought alternative and much improved techniques, by adopting neural networks to investigate and solve problems relating to MAN-IN-THE-MIDDLE spoofing attacks on and across MANETs. Such methods offer more flexible advantages which enable detection and manual prevention of such attacks, as well as functionalities, that enable administrators and security services to trace and apprehend offenders. It is worth noting that, one of the most difficult to detect attacks is those with a MAN-IN-THE-MIDDLE component incorporated. This is where specialized neural networking techniques can draw on the progress that has been made hitherto to offer alternatives.

University of Ghana http://ugspace.ug.edu.gh

Table 2-1 – Tabular survey

PAPER	GENERAL OVERVIEW	NETWORK/SECURITY APPROACH	APPROACH TO MITM	PROBLEMS TO BE SOLVE/DISADVANTAGESES NOTICED	POSSIBLE OUTCOMES/ADVANTAGES
1. A Formal Validation Methodology For Manet Routing Protocols Based On Nodes' Self Similarity **By: Stephane Maag,Cyril Grepet,Ana Cavalli**	Indicated that conformance testing for Ad-Hoc networks was necessary for their reliability but it has been given little attention.[16] Majority of research done by experts in this field rely on simulations and/or emulations; with non-formal models provided as input to simulators like; NS2, OpNet, SSF or Glomosim.[16] Formal methods exist in-between real life testing and simulation testing. It is more of **emulation**	Validation	None given	The methodology used could be more costly, computation-wise to set up, as opposed to using a simulation based approach.	It uses formal methods as opposed to computer simulations, and models. [16]

University of Ghana http://ugspace.ug.edu.gh

	testing - this makes it possible to perform almost real life test using real protocol stacks linked to the simulator.[16] The paper identifies that, be it emulator or simulator, they usually yield erroneous results under real life environmental conditions. As such formal description techniques are needed.[16]				
2. A Packet Authentication Scheme For Mobile Ad Hoc Networks **By:Rehan Akbani, Turgay Korkmaz,** **G.V.S. Raju**	The underlying multicast protocol investigated was **ODMRP.** HEAP which is introduced, uses a modified Hashed Message Authentication Code (HMAC)-based algorithm that utilizes two keys. The keys are: **Okey** -a pair-wise secret hash key shared by each node with each of its	-Uses HEAP(a variant of HMAC) for Authentication	-Not really given but it may be possible to use HEAP(a variant of HMAC).	Detection of attacks was only limited to outsider nodes. Insider node attacks were left for the intrusion detection system (IDS)	-HEAP significantly lowers memory requirements -negligible CPU and bandwidth overhead when compared to the absence of any security scheme at all.

University of Ghana http://ugspace.ug.edu.gh

	neighbors. **Ikey** -one common secret hash key by the same node to all its neighbors. [17] The naive way would be to generate a new MAC for every neighbor with its pair-wise key. **New Approach**: Working principles of HMAC reexamined with its computation being divided into two steps to reduce computational cost. - The research used **simulations (on the GloMoSim simulator)** instead of real **formal methods**.[16]				
3. Different Types of Attacks on Integrated MANET-Internet	The paper looks at the different types of attacks across MANETs and the security problems that have not been	A proposal was made for the future use of a framework with	Was not really given.	It was not as specific as possible on how layer specific security could	The proposal offered a broad overview of most of the possible attacks across

University of Ghana http://ugspace.ug.edu.gh

Communication **By:Abhay Kumar Rai, Rajiv Ranjan Tewari, Saurabh Kant Upadhyay**	solved in them up until now. It looks at how such attacks affect the performance of the network.[18]	minimum public key cryptography to avoid network overload with the extensive use of shared key cryptography to offer security.		be implemented although it adequately highlighted some attacks in the layers of concern.	each layer.
4. Detection and Isolation of Packet Dropping Attackers in MANETs **Journal: International Journal for Advanced Computer Science and Applications. (Vol. 4. No 4,2013)** **By: Ahmed Mohammed Abdalla and Ahmad.H. Almazeed**	The paper presents a trio-ID message based approach - Detection and Isolation of Packet Dropped Attackers in MANETs (DIPDAM); as the title implies detecting packet dropping nodes was the main focus of the approach presented.[19]	It makes use of three ID messages; Path Validation Message (PVM), Attacker Finder Message (AFM), and Attacker Isolation Message (AIM) PVM: for end-to-end loop feedback between the source and the destination.	None really given	Smaller attack types take a longer time to warrant the same level of detectability from the new Intrusion Detection System (IDS) as opposed to larger attack types	To save node resources, the algorithm does not monitor all the nodes at all times. The duty of monitoring unlike other solutions that place the responsibility on each node is here shifted only to the source node. This motivated the general architectural solution presented in this paper; that a single node be used in detecting and

University of Ghana http://ugspace.ug.edu.gh

		AFM: to detect the attacker node through the routing path. AIM: to isolate the attacker from routing path and updating the blacklist; informing other nodes of the update in the process.			blacklisting, as oppose to installing the neural network on all nodes - which would increase computational requirements. This is capture in the General System Architecture in ***Figure 3.1***
5. Detecting and Preventing IP-spoofed Distributed DOS attacks. **By: Yoro Chen , Shantann Dors, Pulak Dhar, Abdullowtabel El Saddik, Amuja Najak.**	The paper investigates the detection and prevention of Distributed Denial of Service (DDoS) attacks facilitated by spoofed IPs; a common technique used by most attackers of computing systems. [20]	The technique employed, "colors" or marks with the path of packets coming from a particular source. It "lightens" the path such that spoofed packets can easily be identified whiles those that come from legitimate sources	None specifically given but the paper gave an indication of the heuristic (trial and error) approach, as a possible means for parameter selection during research	It can effectively detect spoofed packets when the routers participation rate is 20%. Lesser percentages might need more research. It works only for DDoS attacks	The scheme has a very low deployment cost unlike the other packet-marking techniques. Additionally, more than 90% of attack packets can be filtered out without affecting legitimate packets to the web server under attack.

University of Ghana http://ugspace.ug.edu.gh

		can be detected. A firewall keeping a memory of genuine markings (learned and stored in a Filter table during normal time – when there is no attack) for legal packets helps to distinguish between legal and illegal packets.			
6. Risk Aware Response for Mitigating MANET Routing Attacks. **By:Ziming Zhao, Hongxin Hu, Gail-Joon Ahn, and Ruoyu Wu**	The paper introduces the Dempster-Shafer mathematical theory of evidence as a risk aware mechanism to cope with the identification of routing attacks; compensating for the unforeseen partitioning that could occur by using binary approaches, as well as the unanticipated uncertainties that come with using fuzzy methods.[21]	An extended form of the Dempster-Shafer mathematical theory of evidence is used and offers an alternative to the traditional probabilistic approach to handling uncertainty. The work was done using the proactive routing protocol OLSR	None specifically given but suggested that Dempster-Shafer with evidence from IDS could be used.[21]	Although an advantage of an increase in **packet delivery ratio** is stated, there could be possible computational stress when the nodes are scaled up.[21]	Instead of using binary approach of isolation which could possibly result in network partitioning, a more flexible and dynamic approach of risk assessment based on the extended form of the Dempster Shafer theory is used. In doing so, two

University of Ghana http://ugspace.ug.edu.gh

		(Optimized Link State Routing)[21]			approaches are used. Namely; *routing table recovery(**both local and global**)* and *node isolation (different from binary node isolation by applying risk assessment methods to give either **temporary isolation** or **permanent isolation** to prevent adverse, isolation based routing problems*[21]
7. Simulation Of Black Hole Attacks in Wireless Ad-Hoc Networks **By: Semih Dokurer**	The work provides a way for simulating and detecting Black Hole attacks on MANETs using the AODV protocol on the NS2 Simulator.[22]	The approach of the work uses the NS2 simulator to investigate Black Hole attacks on MANETs and thus suggest solutions to such attacks	None specifically given but it gave an idea on how MITM attacks could be simulated on the NS2 simulator.	The solutions envisaged by this work were not implemented in the same work. Other routing protocols were not looked at. It was only limited to AODV	Since the work on black hole simulation work was done in NS2, the suggestion by the research for future work was helpful in coming up with further future topics.

University of Ghana http://ugspace.ug.edu.gh

8. Intrusion detection in MANET using classification algorithms: The effects of cost and model selection **By: Aikaterini Mitrokotsa and Christos Dimitrakakis**	The paper looks at five supervised classification algorithms and how to properly use these algorithms in intrusion detection. One of the main focuses of the paper was to look at how performance is affected by the problem cost matrix (that is uniform versus weighted cost matrix). Additionally the paper accessed tuning possibilities for the classifiers when unanticipated spawns of attacks occur. *K-1* part cross validation was used as opposed to the traditional *k* part cross validation with *sequential* cross validation being compared to the normal *randomized* cross validation method. In general, it investigated how hyper-parameter tuning affects performance when new unforeseen attacks are found in the test dataset.[23]	The classifiers are investigated upon four main attack types (Black hole, Forging, Packet Dropping and Flooding attacks).	None specifically given but gave the information that in cost-sensitive classification (a concept that has already been dealt with on numerous occasions in wired networks), decisions are made in order to minimize the expected cost rather than the probability. In other words it is better to err on the part of caution rather that to clean up the mess later. Thus false positives are not always bad	MITM attack was not under the attacks that were investigated. This was one of my motivations for investigating MITM attacks	The method of data collection employed helped to ensure that the research more closely modeled real world attacks since there are some attacks that were not expected to occur in simulations. The results showed that the cost matrix method not only minimized expected cost but also classification error for most classifiers

University of Ghana http://ugspace.ug.edu.gh

9.Detecting and Isolating Malicious Node in AODV Routing Algorithm **By: Priyambada Sahu, Sukant Kishoro Bisoy, Soumya Sahoo**	The paper sought to offer a solution to flooding attacks that take place in Neural networks. The attacks that resulted by taking advantage of the AODV protocols specification for sending out fake control packets (such as RREQ or RREP packets). This is achieved in malicious nodes by disregarding the RREQ_RATELIMIT parameter defined in the protocol.[24]	Security is offered to the network by the cooperation of neighboring nodes to limit the number of RREQ sent out to flood the network.	None specifically given	The paper although clearly showing the performance difference between the presence of a malicious node in a MANET setup using the AODV protocol and its absence, fails to clearly show how this information could be leveraged to prevent such attacks.	The paper was able to indicate that the performance of the AODV routing protocol is degraded by the introduction of malicious nodes. However there is lesser routing overhead.
10. A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks **By: Bing Wu, Jianmin Chen, Jie Wu, Mihaela Cardei**	The paper presents each of the different attacks possible across each of the different protocol stack.[25]	It shows each level, the attacks and their countermeasures. It indicated that because of the resource limitations, current research concentrates mainly on IDS as a second line of defense with focus also on trust	It introduces SSL and TLS as ways to offer protection against MITM attacks. MITM attacks were also shown to be multilayer (they can take place on each layer). Enhance	However it showed that the limitation of nodes computation capabilities limited the ability to implement several complicated key exchange or distribution protocols in MANETs.	The paper gave insight into the different level of attacks.

University of Ghana http://ugspace.ug.edu.gh

		based systems.	Diffie-Hellman (DH) was introduced as a way of countering MITM attacks due to the vulnerabilities of the normal Diffie-Hellman to those same attacks.		
11. SWAT: Small World-based Attacker Traceback in Ad-hoc Networks **By:Yongjin Kim, Ahmed Helmy**	The paper presents the use of the new SWAT architecture to offer an improved IP trace back scheme for being able to locate the source or origin of attacks.[26]	It makes use of an IP trace back scheme.	None really given. The focus was more on the Distributed Denial of Service Attacks but it did offer some indications of what to look out for when one is looking out for attack signatures – one being the excessive propagation of SYN	Swat is not able to detect the real identity of the attacker.	Swat can be used to detect the next hop node to the origin of attack traffic.

University of Ghana http://ugspace.ug.edu.gh

			packets by attack nodes.		
12. Controlling IP Spoofing based DDoS Attacks **Through Inter-Domain Packet Filters** **By: Zhenhai Duan, Xin Yuan, Jaideep Chandrashekar**	The paper aims at giving a method of providing an architecture that limits the level of inter-domain packet spoofing that result in Distributed Denial of Service Attacks that occurs on the Internet. A key feature of the approach is that it does not require global routing information.[27]	It makes use of Inter Domain Routing Filters implemented on border routers.	None really given although the scheme helps to reduce the amount of spoof packets	The addition of extra algorithms would imply more computational costs at boundary routers	Inspired by route-based packet filters, the proposed inter-domain packet filter does not spoof all packets but offers a way to trace the origin of malicious packets.
13. Man-in-the-Middle Attacks in Distributed Hash-Tables **By: Thomas Reidemeister,**	The paper focused on security at the Distributed Hash-Table (DHT) level. That is the transport layer or below. It looked at problems presented by the misuse of DHT - characterized into three forms; free riders (lazy	The paper proposed three replication-based forwarding protocols (multiple realities, multi-path routing, and proximity routing) for	The paper, in its literature review section mentioned the two main groups of approaches aimed at counteracting	The message checking approach fostered the use of a distributed algorithm to observe route path; as proposed by Sit and Morris and	It provided a more secure method for offering replication-based approaches to preventing MITM attacks

University of Ghana http://ugspace.ug.edu.gh

Klemens B¨ohm, Erik Buchmann, Paul A.S. Ward	peers who are not malicious by action but would not help the network unless incentivized), errors (related from inappropriate implementation of DHT specification resulting in malfunctioning peers) of and deliberate attacks (malicious nodes deliberately attacking the network). The paper mainly aimed at showing that the probability of a MITM attack is exponential in relation to peers and linear in relation to attackers.[28]	counteracting MITM attacks	MITM attacks. Namely message checking (an attempt to send a message to a desired destination and checking to verify that the message gets to that destination without any alteration) and replication. The replication approach involves the sending of messages across multiple routes with the belief that it is improbable for MITM to take place across all those multiple channels.	discussed in the literature review, it was found to increase latency and bandwidth overhead. The alternative of using the public key infrastructure defeated the general idea of "peer-to-peer" envisaged in MANETs as well as demands that *a-prior* knowledge of the destinations public key be known, something that is usually not know in advance. Another choice suggested by Aberer and Despotovic involved the concept of trust based reputation where a DHT was used	

University of Ghana http://ugspace.ug.edu.gh

			This was implemented by Ratnasman et al. as multiple independent keys; coining the word "multiple realities" spaces and is extended in the paper. The paper noted that Catro et al. presented a similar replication method which made use of redundant routing with the support of cryptographic puzzle or keys where all peers are required to make payments to obtain such security	to take complaints of from nodes of victims on attackers however the problem with this is that there is no way of verifying that the attacker being reported is not actually the victim of a MITM attack.The approach was targeted mainly at networks with lower constraints and not MANETs - that have relatively severer power constraints and thus the approach may consume a lot of power on them.	

University of Ghana http://ugspace.ug.edu.gh

			tools to reduce the incidences of malicious nodes joining the network.		
14. Detecting Black Hole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method **By: Satoshi Kurosawa, Hidehisa Nakayama, Nei Kato, Abbas Jamalipour, and Yoshiaki Nemoto**	The paper showed the use of dynamic training methods as oppose to existing static approaches to detect Black Hole attack generated anomalies in MANETs. Using the reactive protocol AODV, better and normal behavior characteristics are realized by extracting features, while simulation induced sequence number changes are made.[29]	The security approach aimed at filling in the gap of addressing internally generated Black Hole attacks that external protocols such as Secure Efficient Ad-Hoc Distance vector routing protocol (SEAD) and Authenticated Routing Protocol for Ad-Hoc Networks has failed to adequately address. This is because they concentrated mainly on external attacks. A bypass is also sought	None really given. However the use of NS2 in the simulation informative in the choice of a simulator this research.	Use of a simulation environment could provide results that might deviate from real world scenarios.	Use of simulation offered a cost effective approach to investigating large scale Black Hole attack scenarios and investigating dynamic ways to mitigate them. Also, the use of a dynamic approach made the attack detection scheme more flexible and reliable

University of Ghana http://ugspace.ug.edu.gh

		around the routing overhead created by employing suggested solutions to internal attacks such as those that involve the use of a Route Request (RREQ) switch and another, the introduction of Route Confirmation Request (CREQ) indicator.			
15.Future Directions in Ad hoc Networking Research **By: Anders Lindgren,Kaustubh S. Phanse,Tomas Johansson,Robert Br¨annstr¨om,Christer Ahlund**	The paper addresses some current problems found in existing ad-hoc networks and suggests possible future researches into various areas.[30]	The paper indicates that, current research focuses mainly on connected networks where it is possible to find a contiguous path between source and destination. The ProPHET (Probabilistic Routing in Intermittently	None given	Recognized that current MANET research adopt faulty paradigms based on impressions of wired nodes and links. Identified simulation based research as a problem. Problematic analytical models which	Implementing the suggested approaches, were expected to cater for problems such as routing in periodically partitioned MANETs and also to reduce the number of collisions during routing.

University of Ghana http://ugspace.ug.edu.gh

		Connected Networks) routing protocol was proposed as an alternative. It also offered approaches to reduce interference based on topology control		trade-off accuracy or increase complexity. Solutions have to be developed for situations were network partitioning occurs and there is no contiguous path from source to destination	
16. MANET: Vulnerabilities, Challenges, Attacks, Application **By: Priyanka Goyal, Vinti Parmar, Rahul Rishi**	The paper basically had a look at the problems current MANET research seeks to address, the aim and goals of such researches; security-wise as well as the possible attacks that can take place on such networks. There is also a look at the routing protocols and the different scenarios under which **reactive (source initiated or demand driven) or proactive (table-driven) protocols** would be	Introduces that current research approach is towards mesh architectures and large scale with improvement in bandwidth and capacity being a driving force.	None given		Recognized the absence of a centralized management in MANETS, power supply constraints, scalability issues. Problems related to dynamic topology issues were also mentioned.

University of Ghana http://ugspace.ug.edu.gh

	needed. Hybrid protocols; featuring a combination of intra-zonal proactive approach and an inter-zonal reactive approach are also mentioned.[1]				
17. Security in mobile ad-hoc networks : **Challenges and solutions** **By: Hao Yang, H Aiyun L Uo,Fan Ye , Songwu Lu , And Lixia Z Hang ,**	Focuses on protecting nodes in a multi-hop MANET network by identifying the security problems at the link and network layer in the delivery of data bits from one node to the other. It discussed the issued related to packet routing and forwarding. The observation is made that a truly intrusion-free system is infeasible thus detection and reaction systems and their research particularly in MANETs is paramount.[31]	Presented security approaches used in source routing, distance vector-routing, link state-routing and secure packet forwarding approaches. Both the reactive and proactive approaches to security are discussed and a multi-fenced is suggested.	Not given	The paper indicated that the discussed threats and their solutions such as hashed chain algorithms were reactive in nature against know attacks; first detecting attack possibilities and hardening systems to preempt them. Thus,the necessity of a more intelligent, multi-fenced approach that is embedded in every network component, to add in depth protection. This informed credence	Identified the problems of security related to link-layer and network layer protocols which are mostly found to be susceptible to packet routing and packet forwarding attacks.

University of Ghana http://ugspace.ug.edu.gh

				to the choice of Artificial Neural Network for in the solution covered in this current research.	
18. A Survey on Cryptography Applied to Secure Mobile Ad Hoc Networks and Wireless Sensor Networks **By: Jianmin Chen and Jie Wu**	The paper sought to highlight that security could be achieved on MANETs and Wireless Sensor Networks (WSN) with a broader understanding of cryptographic techniques.[32]	The security approach involved different cryptographic schemes. Cryptographic schemes were broadly classified into four; Symmetric cryptography(e.g. SAODV - secure ad-hoc on demand distance vector) Asymmetric cryptography(e.g. RSA, DSA – digital signature algorithm), Threshold cryptography, and Other cryptographic techniques	None given, though the suggestion of the application of cryptography in MANET/WSN for security purposes was helpful.		The paper helped identify the disadvantage of shared keys was noted as having the computation headache of generating n(n-1)/2 keys Also, under symmetric cryptography and in relation to random nonce which use pseudo - random generators to generate random numbers, it is observed that the weakness of the scheme could at most times be the random generator itself. ID based verification techniques be it single or

University of Ghana http://ugspace.ug.edu.gh

					batched ID are generally more slower than RSA/DSA

University of Ghana http://ugspace.ug.edu.gh

Detection of attacks in MANETs is a growing field in network security. Although there have been several approaches at preventing both proactive (attacks initiated without any prior information on the victims system) and reactive (attacks initiated based on initially response of the system under attack to a previous attack or stimulus such as SQL-injections), attacks as captured in Kurosawa et. al in [29]as well as Reidemeister et. al in [28], none of these papers provides an approach for the possible simulation and remedying of a man-in-the-middle attack on a simulating platform – the NS2 environment. Classification algorithms have been quite widespread in their use in intrusion detection systems; but my approach adopted artificial neural network systems.

In [16]the paper introduces the concept of "*A Formal Validation Methodology for Manet Routing Protocols Based on Nodes' Self Similarity*". The aim was to fill in the gap left by simulation or emulation tests; without having to perform the probable, otherwise costly alternative of actual testing. It sought to apply a conformance testing approach; neglected in the aforementioned approaches, to the Dynamic Source Routing Algorithm (DSR) algorithm. It highlighted the disadvantage of using simulation tests as their inability to completely mimic real world scenarios. However, the paper also highlighted the problem of formal method based approaches as being their inability to take into account the inherent MANET protocol characteristics. Thus in this project, a trade-off was made of the latter; adopting a simulation based approach in the process.

On the authentication schemes that have been looked at, [16] provides insights into various schemes; providing an alternative way of offering authentication called HEAP - an HMAC based algorithm which utilizes two keys. The target of the new schemes was to prevent popular attacks such as DDoS and man-in-the-middle attack.

University of Ghana http://ugspace.ug.edu.gh

Although this approach offered the advantages of lower memory requirement in comparison to other existing schemes such as TESLA and LHAP; with limited CPU and bandwidth overhead, it constrained itself to detecting attacks from outsider nodes.

Insider attack detection was clearly left to the installed IDS. Thus in a scenario where an IDS was not installed, such attacks could go under the radar. The approach taken during this research did not have that constraint; being envisaged to detect attacks whether from both outsiders and insiders.

There are several attacks possible on MANETs. As shown in [18] they are numerous; with equally numerous techniques of defending against them. The issue of performance was specifically touched upon in this work for both MANETs with internet access - the greatest source of most network based attacks in today's world, and also, for standalone MANETs. Suggestions at the end proposed the use of a framework that utilizes minimal public key cryptography interaction to offer security. The conclusion was that, an overelaborate use of such key infrastructure overloaded the network and reduced performance. This also informed my approach of artificial neural network (ANN).

Ahmed Mohammed et. al in [19] showed the possibilities associated with a trio-ID message based approach to perform attack detection and node isolation. The major advantage that this approach offered was the shift in the attack detection responsibility from all the interacting nodes to just the source node. A cumulative effect could be a conservation of network power from the reduction in computation requirements. However, the main disadvantage noted was that with smaller attacks; running for comparatively shorter time spans, these attacks could go undetected. They needed to be

University of Ghana http://ugspace.ug.edu.gh

run for quite a longer time period to trigger the same level of detectability; so obvious in larger attacks. This is where a trained neural network has the envisaged advantage of being able to notice minute changes after training and retraining. Another paper that presented a detection and prevention approach was [20]; "***Marking-based Detection and Filtering (MDADF)"***. It not only differed from [19] in the approach used but also in the attack type which was considered. Whereas [19]focused on packet dropping nodes, [20]was more interested in IP-spoofed initiated DDoS attacks; picking as it's chosen method, a technique that essentially "colored" the path a packet transverses, to enable easier tracking of a promiscuous source for elimination from the network. Presenting an advantage of a lower deployment cost as opposed to other packet marking schemes, it helped eliminate illegitimate nodes by marking legitimate ones; a different approach to the Packet Identification (PI) mechanism it sort to compete against. It offered a 70% acceptance ratio realization when there were only 20% of routers participating in the scheme. This was far better than the 60% acceptance ratio offered by **PI,** achieved with all routers participating. The parameters used in developing the newer technique – **MDADF,** were arrived at by a heuristic approach; informative in presenting the possibility of adopting such an approach during the features extraction and machine learning phase of any research. Thus the **"wrapper"** method (to be discussed in the section 4.1) was adopted during the feature extraction phase.

On attack creation, there were many possibilities; a real attack creation (which would have been resource intensive and quite expensive in its execution), using an emulator and finally performing a simulation. Having read [22], it was realized that a simulation based approach was most appropriate. That paper offered insightful thoughts

University of Ghana http://ugspace.ug.edu.gh

and advice on how to perform black hole attacks on the NS2 simulation software. By making use of the RREQ, RREP packets, it was able to mimic scenarios of a black hole attack. Analysis of the analogies made was helpful in arriving at a similar but considerably different approach towards simulating a man-in-the-middle attack on NS2. Instead of dropping packets as in [22] it was chosen to induce a slight delay, as was hypothesized by this research, as to be expected in a MITM attack. In order to come up with a classification algorithm, several factors had to be considered. Almost every approach used in attack detection has the probable drawback of pulling in false positives as attacks. This was highlighted by [23]. With a comparative analysis of cost-sensitive classification being done; tuning of hyper-parameters as the experimental protocol for the goals of checking the influence of altered cross-validation methods on classification algorithms, as well as investigating how weighed classification contributed to classification accuracy improvement, this paper offered a conclusion that; with several algorithms having security considerations as a focus, it is better to err on the side of caution rather than to "clean up the mess later". The idea propelled forward as done in several researches prior - upon which this builds, was that, the cost incurred by flagging a false positive was far less in comparison to the damage that resulted from an unnoticed attack. Thus, installation of IDSs do not necessarily imply insulation from all attacks, but rather a minimization of the possibility of an attack or at least, a reduction in the probability of an attack not being reported. The aim of my research was to minimize the number of false positives; whiles increasing the number of detected attacks using neural network. Overall, there is a lot of literature on MANET attacks but very few that specifically address the problem of man-in-the-middle attacks across them. This study

University of Ghana http://ugspace.ug.edu.gh

aims at remedying that. Thus, this research specifically focuses on **a)** detecting such attacks and remedying first instance cases(MITM attacks scenarios not noticed before) and **b)** Using the experience gathered in "**a**" above to prevent such an occurrence in future; thereby reducing the cost incurred from unreported attacks or delayed ones. Further literature review is presented in ***Table 2.3*****.**

University of Ghana http://ugspace.ug.edu.gh

CHAPTER THREE

SYSTEM DESIGN PROCESS AND DEVELOPMENT

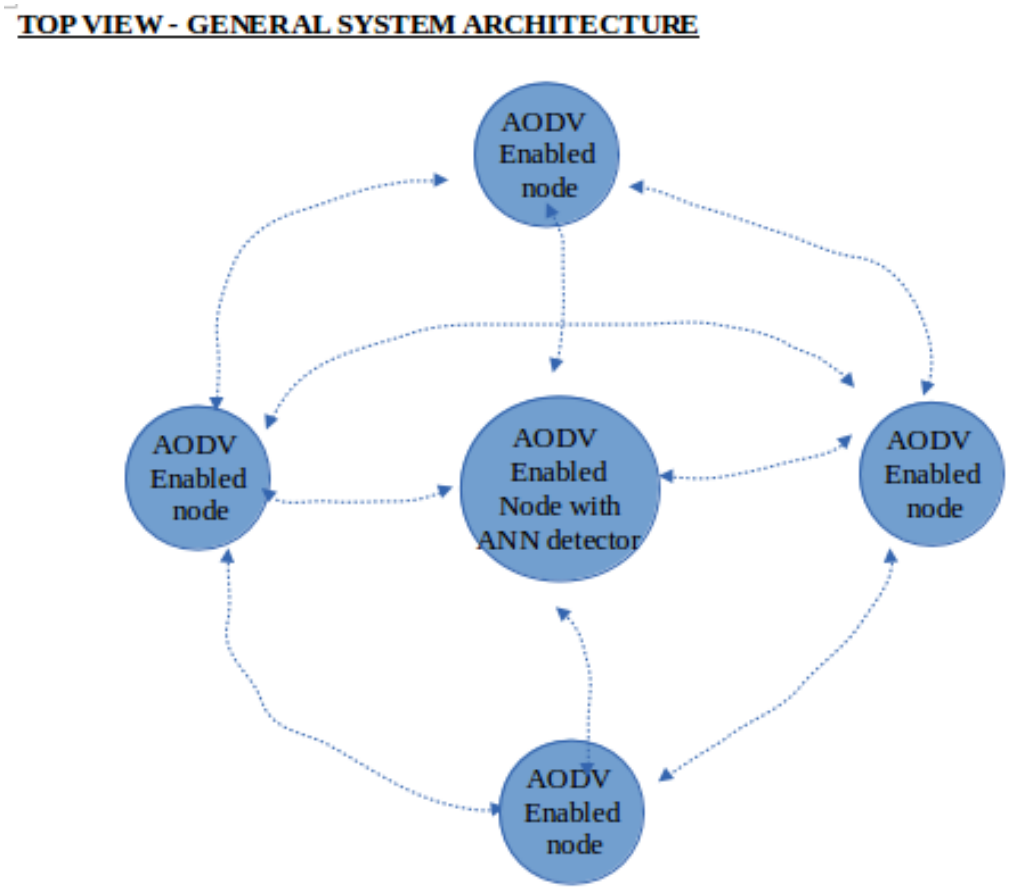


Figure 3.1 – General system architecture of the new working system

3.0 General System Architecture

Figure 3.1shows the General System Architecture of the system as envisage during operation. There is one dedicated node which is triggered as the administrator, to monitor the MANET for malicious nodes, dislodge them and reconfigure the network; as made

University of Ghana http://ugspace.ug.edu.gh

possible by the scripts contained in ***Table B.3***. The central node in the above mentioned figure plays that role in this case with the ANN detector installed on it.

Conceptualization and Preliminary Flowchart Design

Conceptually, the final solution given as a Java application can be made to read logged data from any layer of the TCP-IP protocol stack –***Figure D.2***. The features extracted from the logged details can then be used to train the ANN for attack detection. Software vendors or system administrators wishing to use such a code could; as depicted in ***Figure 3.2***, configure it for single layer logging or multilayer (cross layer) feature extraction for ANN training.

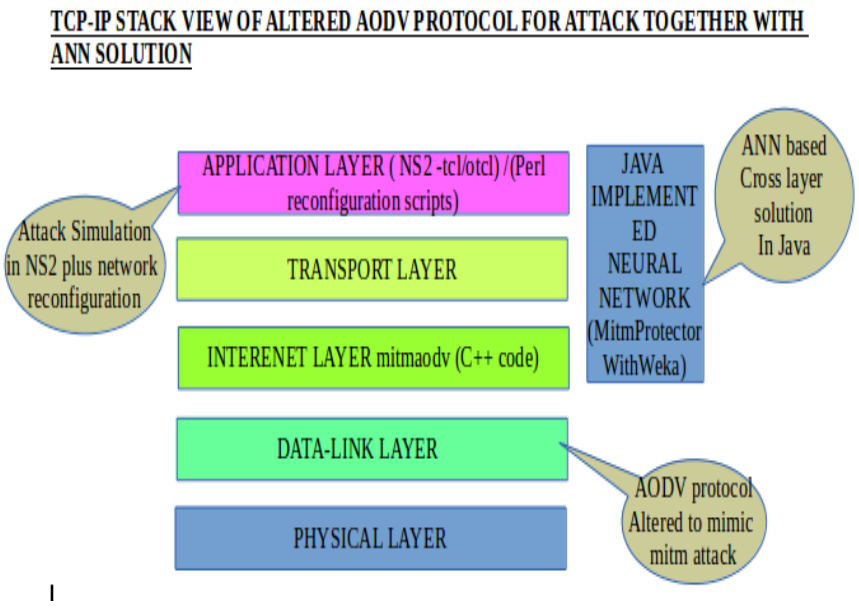


Figure 3.2 - Conceptualization and Design

University of Ghana http://ugspace.ug.edu.gh

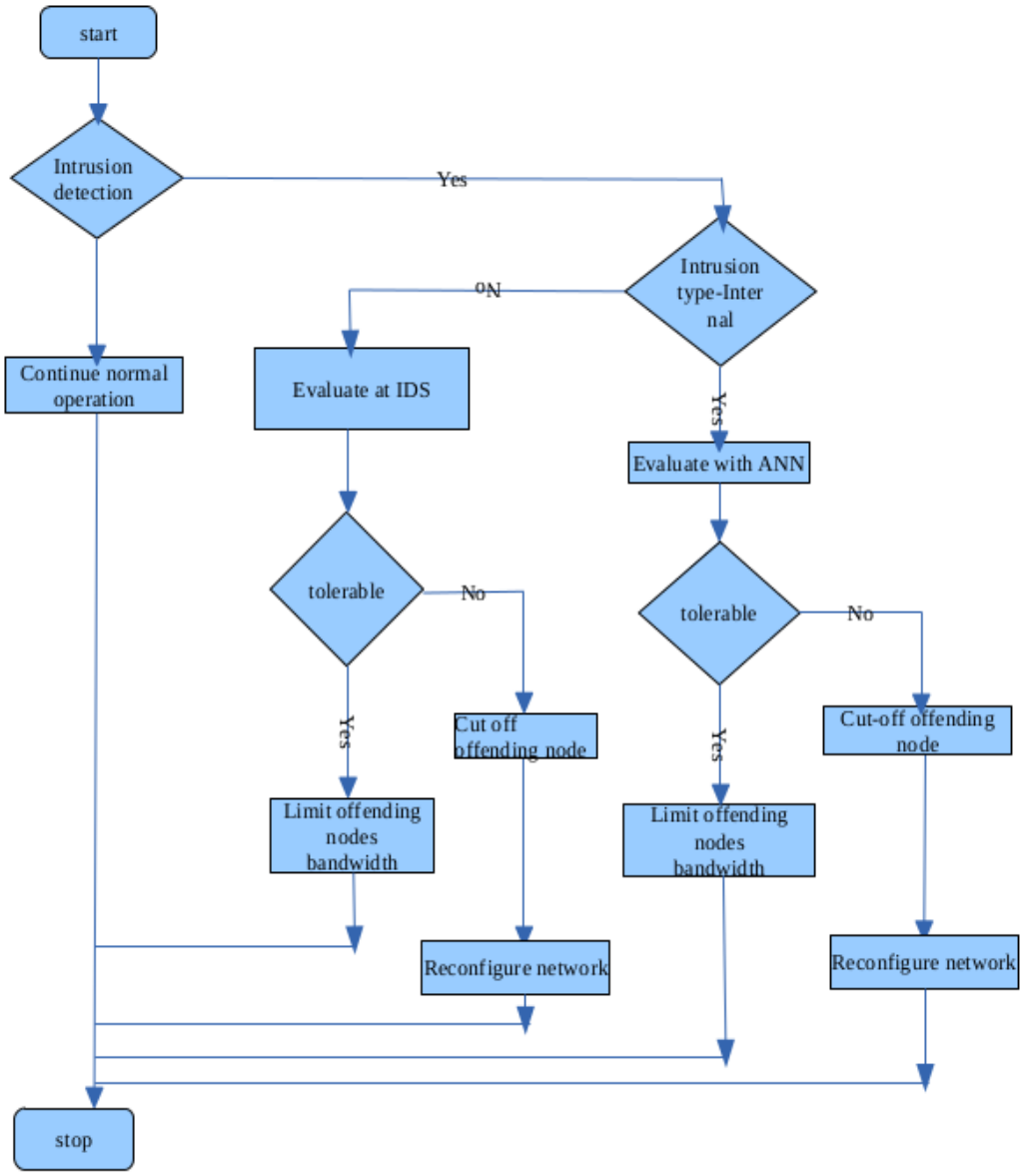


Figure 3.3 - Initial flowchart design

University of Ghana http://ugspace.ug.edu.gh

Final flowchart design

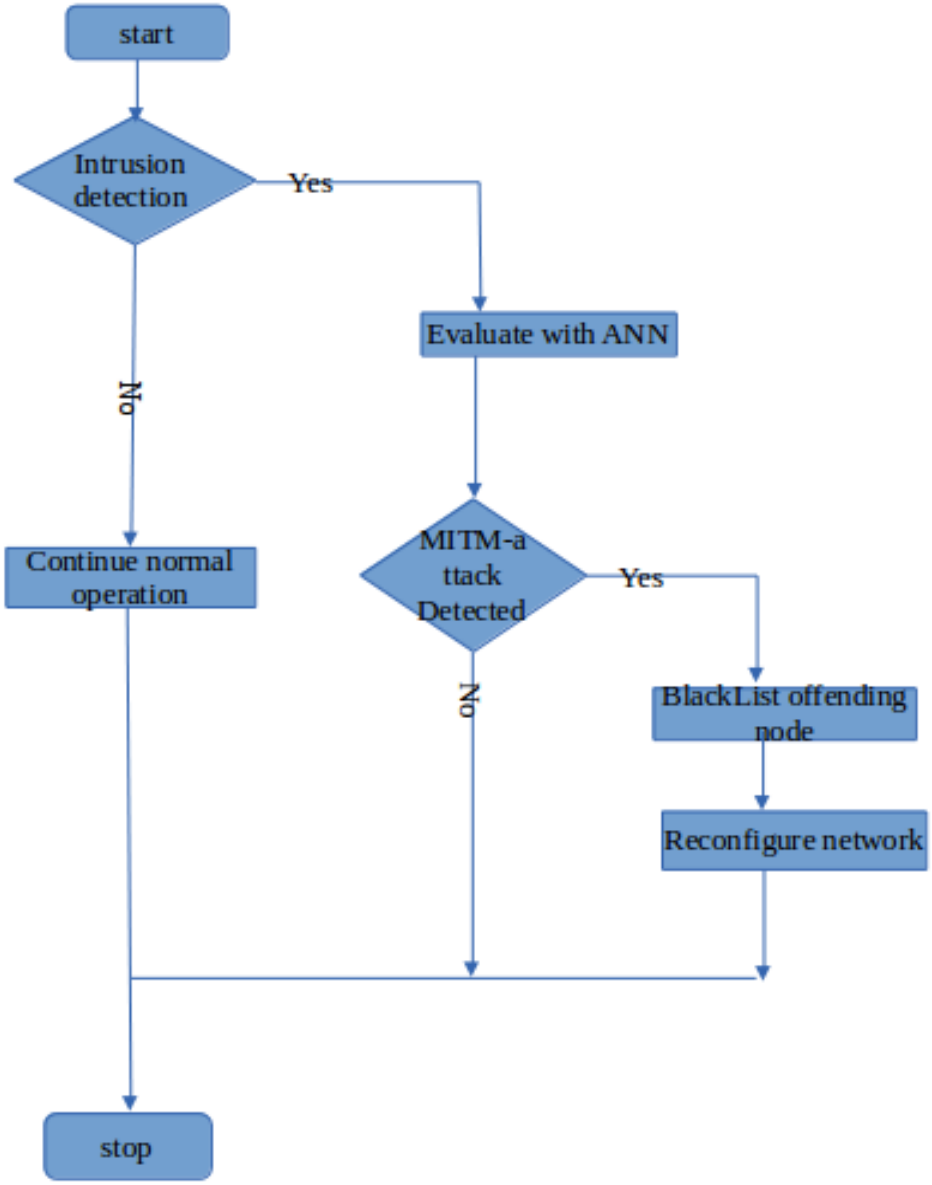


Figure 3.4 - Final flowchart design

University of Ghana http://ugspace.ug.edu.gh

The initial flowchart in ***Figure 3.3***was initially envisaged with an IDS component being attached to the system to generate log files and also help in intrusion detection. The final flowchart was then arrived at after the realization that the installation of an IDS, could result in extra resource (power and processing time) consumption and that the final neural network system shown in ***Figure 3.4*** was adequate in detecting and preventing intruding nodes.

3.1 Methodology

ANN played the major role in the areas of feature acquisition and feature selection. The overall aim was to detect the attack, eliminate the attack vector, while re-establishing a more secure connection taken out the third party in effect. The system was also fashioned to learn from such attack instances and adapt the network to prevent the reoccurrence of such an attack, with similar environmental parameters for future scenarios.

The system was expected to help network administrators and security experts to notice attacks; and automate techniques to counteract their occurrence, as well as learn from past experiences and adapt their networks to make such attacks almost impossible in future.

University of Ghana http://ugspace.ug.edu.gh

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND TESTING

4.0 Implementation

The implementation was done making use of the NS2(from ns-allinone-2.35.tar.gzpackage) simulator together with the NAM animator. The main programming languages used were Perl, C/C++, and Java with the combination of all the different scripts and executable codes being done via shell scripts. The Linux Ubuntu 13.10 Operating system, served as the platform of choice for all the coding and simulations done over the course of the research. For ANN and other machine learning algorithms the weka software package and its API provided the essential resources.

4.0.1 NS2

NS is a UC Berkeley developed event driven network simulator very well tooled to simulate various IP networks. Among the protocols it provides are TCP and UPD. Traffic source behavior is offered by protocols such as FTP, Telnet, and CBR with router queue management mechanism such as Drop Tail, RED and CBQ having been implemented. Its routing algorithms include Dijkstra and also support multicasting. Local Area Network simulations are made possible by the MAC layer protocols that NS2 provides and some of the MAC layer protocols for LAN simulations.[33] The NS project is now a part of the VINT project that develops tools for simulation results display, analysis and converters that convert network topologies generated by well-known generators to NS formats. Currently, NS2 written in C++ and OTcl (The Object oriented flavor - developed at MIT;

University of Ghana http://ugspace.ug.edu.gh

of the Tcl scripting language) is available.[33]

Directory structure and addition of new routing protocol

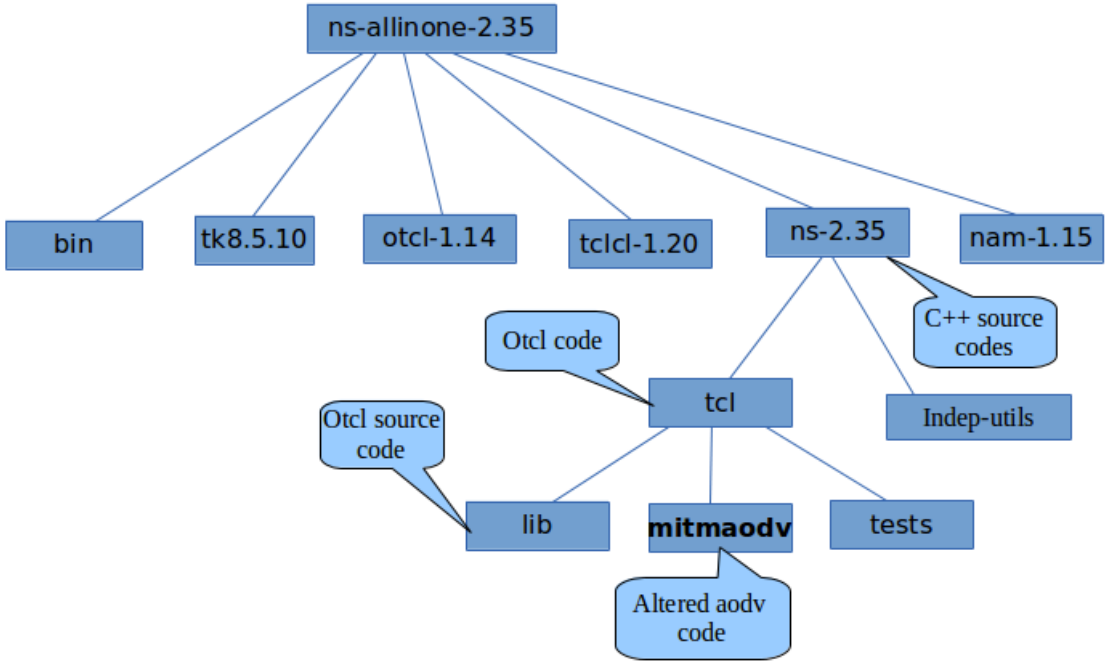


Figure 4.1- Partial directory structure of ns-allinone-2.35 package

Figure 4.1 gives a partial representation of the ns-allione-2.35 directory structure with all the salient portions relevant for development and simulations. All the codes relevant for the simulations to be carried out are placed here. The **tcl** folder which has subdirectories such as **lib,** and **tests,** contains most of the Otcl source code necessary for simulations to be carried out. All additional alterations on all customized C++ codes and projects can be put directly in the ns-2.35 folder. That was where the customized version of the aodv protocol–**mitmadv**, installed as part of the simulator, was placed. It was used in the simulation of the man-in-the-middle attack. The files from the original protocol were renamed. With the exception of the **aodv_packet.h** file, every other file name in that

University of Ghana http://ugspace.ug.edu.gh

directory was appended with the "**mitm**" string. The idea behind this was to ensure that packets could be exchanged between nodes using the native "aodv" protocol and the new customized variant "aodvmitm". All classes and structures were renamed in the new protocol's implementation; except the ones in the aodv_packet.h file.[33]

Table 4-1 - Files present in aodvmitm folder and the changes made to the generic aodv protocol

FILES CREATED	**CHANGES EFFECTED AND USE**
aodv_logsmitm.cc	Renaming of classes from **"AODV"** to **"AODV_MITM"**
aodv_rqueuemitm.cc / aodv_rqueuemitm.h	Renaming of class; changing **aodv_rqueue"** to **"aodv_rqueuemitm"**
aodv_rtablemitm.cc / aodv_rtablemitm.h	Class and function changes such as **"aodv_rt_entry"** to **aodv_rt_entry_mitm"**
aodvmitm.cc / aodvmitm.h	Class changes such as **"AODV"** to **"AODV_MITM"**
aodv_packet.h	Left intact

A panoramic view of the file edits that were made, are shown in ***Table 4.1***. Additionally,

University of Ghana http://ugspace.ug.edu.gh

other changes that occurred are captured in ***Figure 4.2to Figure 4.4*** as class diagrams with ***Figure4.5***showing the component diagram of all the files presented in the new mitmaodv folder. This new extension to the ns2-allinone package was intended to enable a node to send out the newly registered AODV_MITM packet type -***Figure4.6***,while also receiving traditional AODV packets. Inheritance and relationships are presented in these diagrams to show how the aodv_mitm protocol; which was installed as part of the ***ns-allinone.2-35 package*** and is depicted in***Figure4.1,*** was created.

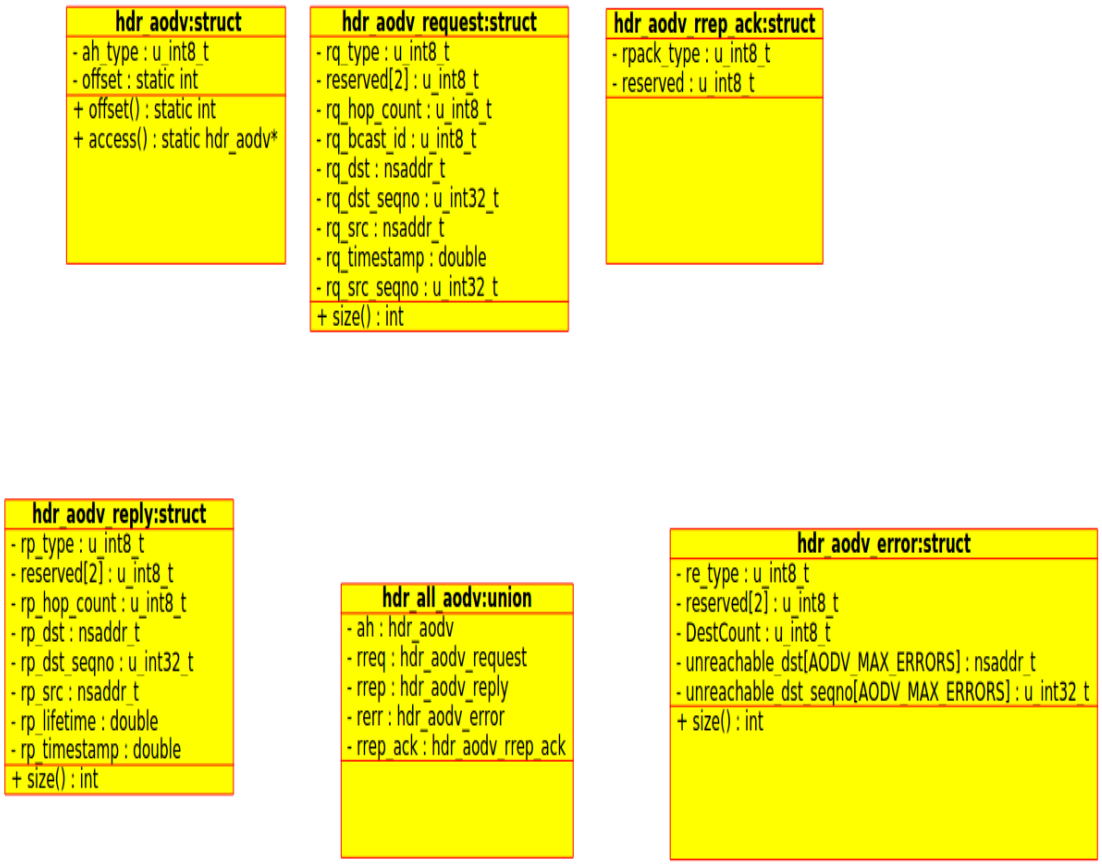


Figure 4.2 - Class diagram for aodv_packetmitm. (cc/h)

University of Ghana http://ugspace.ug.edu.gh

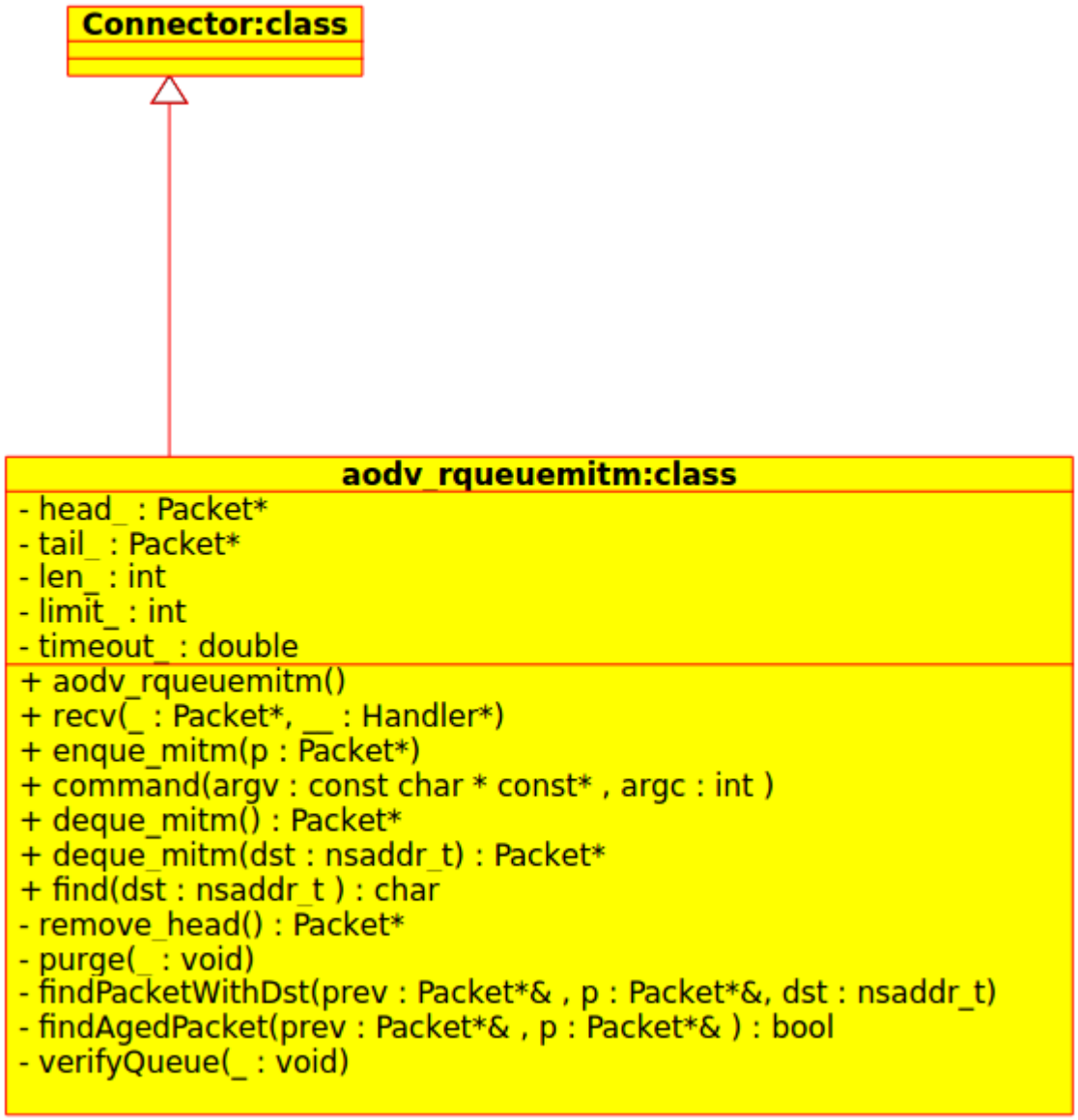


Figure 4.3 - Class diagram for aodv_rqueuemitm.(cc/h)

Figure 4.3 shows the class diagram for the aodv_rqueuemitm class and its relationship with the Connector class; a vital relationship, in maintaining C++/Otcl linkage for the altered version of the aodv protocol, for the simulation.

University of Ghana http://ugspace.ug.edu.gh

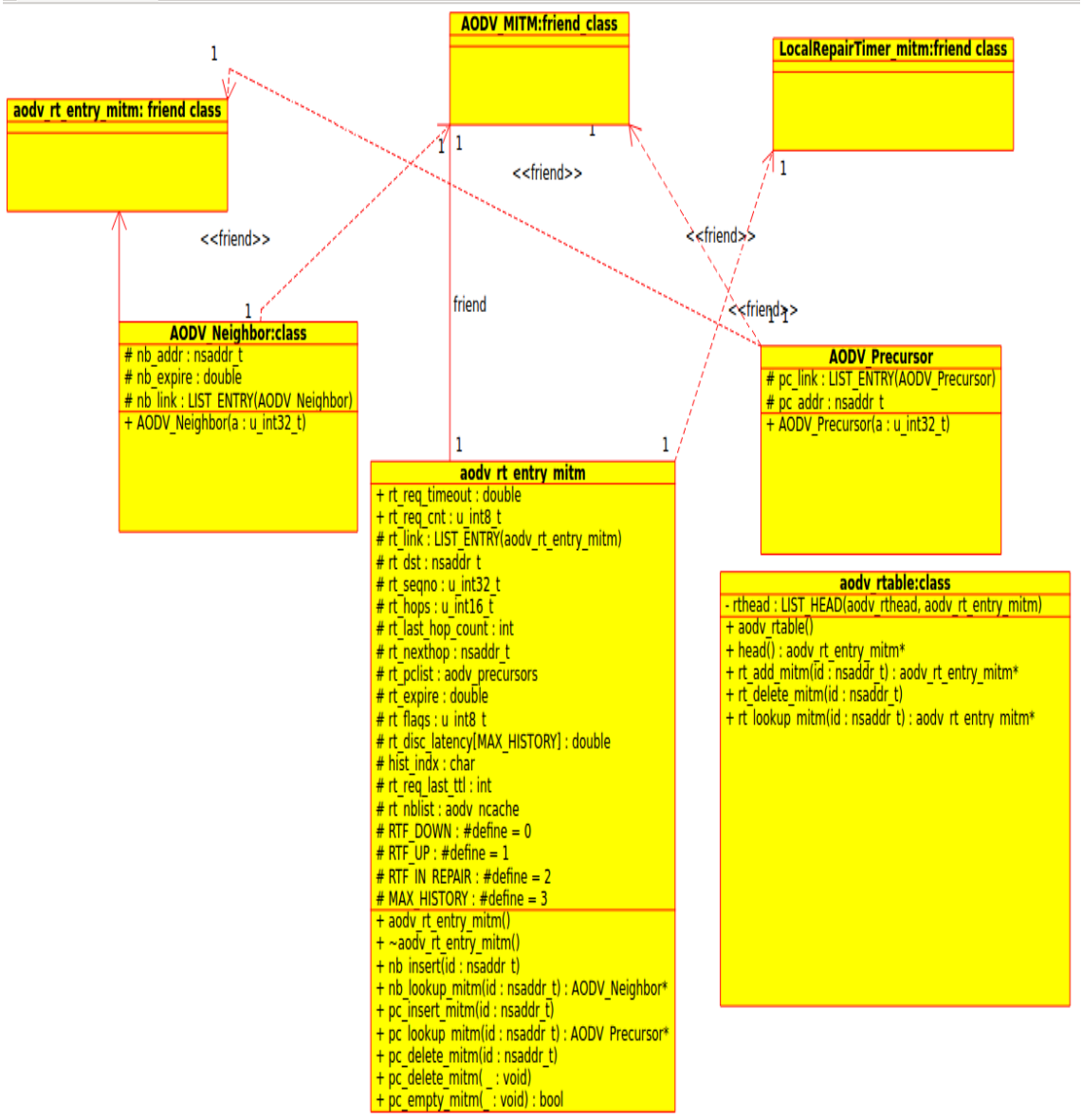


Figure 4.4 - Classdiagram for aodv_rtable_mitm. (cc/h)

University of Ghana http://ugspace.ug.edu.gh

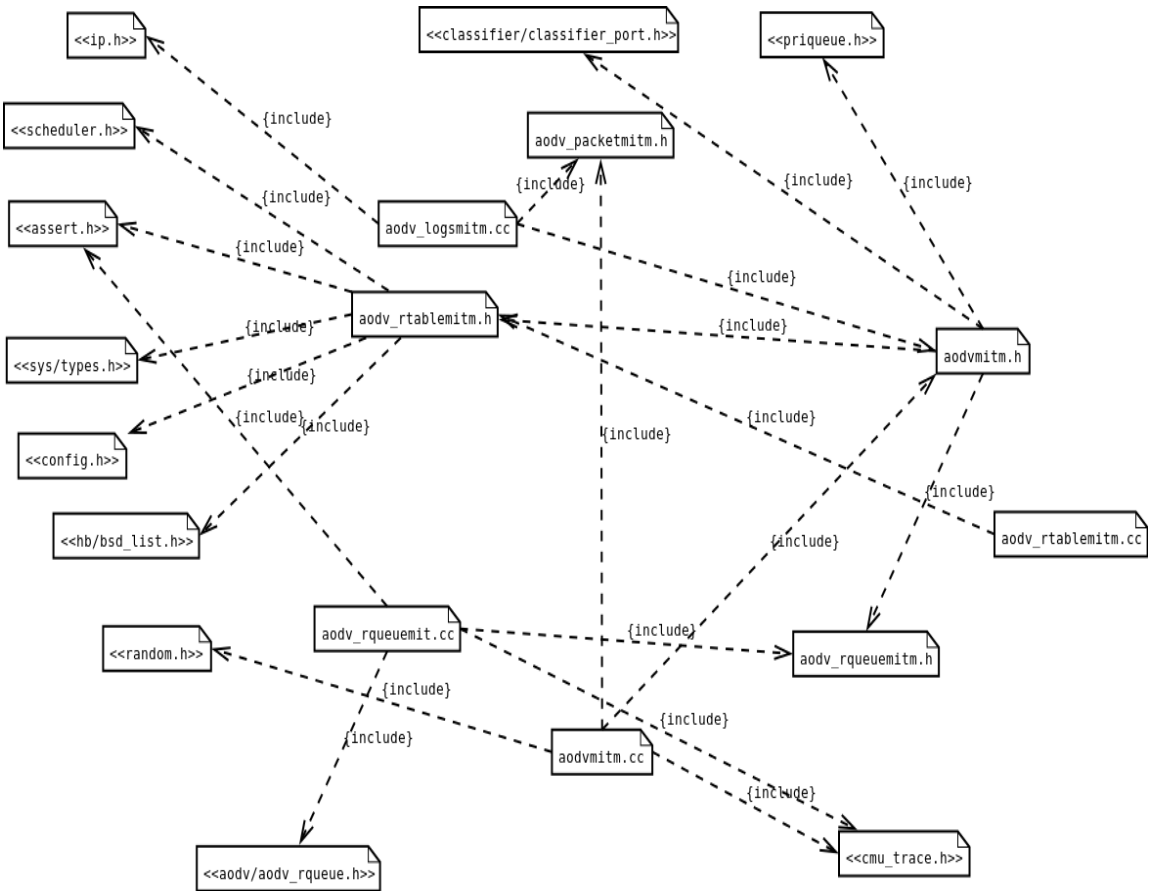


Figure 4.5 - Component diagram for aodvmitm package

4.0.2 Other Changes

Apart from the addition of the new routing protocol as per requirement, other additions were made in the **ns-packet.tcl** file located in **"ns-2.35/tcl/lib"** subdirectory. That particular file is necessary for packet format initialization; anytime a simulation is started. Thus, any new packet created needs to be registered in this file as shown in **line 189** of the ***Figure4.6***.

University of Ghana http://ugspace.ug.edu.gh

```
188  # PNB: added by me for protoname
189      AODV_MITM   # for implementing man in the middle attack using the AODV protocol
190  }
191  set allhdrs [regsub -all {#.*?\n} $protolist \n]; # strip comments from above
192  foreach prot $allhdrs {
193      add-packet-header $prot
```

Figure 4.6 – Packet registration in ns-packet.tcl

Additionally, lines 629-631 of***Figure4.7*** as well as lines 2315-2321of***Figure4.8*** were added to the **ns-lib.tcl** file. This file contains the list of classes and functions that directly mirrors the implementation of classes and function in C++ for NS2's Otcl/C++ linkage.

```
628          #PNB: a hack to satify the procedure node's call to man in the middle attack on using modfied aodv protocol
629          AODV_MITM {
630              set ragent [$self create-aodvmitm-agent $node]
631          }
632          DSDV {
633              set ragent [$self create-dsdv-agent $node]
634          }
635          DSR {
636              $self at 0.0 "$node start-dsr"
637          }
```

Figure 4.7 - Additions to ns-lib.tcl file (lines 629-631)

```
2314     #PNB: function definition for this function's call on line 626
2315   Simulator instproc create-aodvmitm-agent { node } {
2316       #Create Aodvmitm routing agent
2317       set ragent [new Agent/AODV_MITM [$node node-addr]]
2318       $self at 0.0 "$ragent start"
2319       $node set ragent_ $ragent
2320       return $ragent
2321       }
```

Figure 4.8 – Additions to ns-lib.tcl file (lines 2315-2321)

University of Ghana http://ugspace.ug.edu.gh

4.0.3 OTCL

Parameters for configuration

Tcl and Otcl codes were then written, to carry out the simulations; with the node movement being set using the "setdest" script - having as initial parameters the values presented in the first column of ***Table 4.2***. These are captured in the "mitm_attack.tcl" and "mitm_attack_reduced.tcl" files of ***Table B.1*** of APPENDIX B. The characteristics used to set all the genuine nodes are indicated in table ***Table 4-4***being wholly presented by either of the tcl scripts presented by ***Table B-1*** in **AppendixB**. Additionally, traffic patterns are generated with the "cbrgen" file having received the parameters shown in ***Table 4.2***.

Table 4-2 - Parameters Used in the both "setdest" generated files and "cbrgen" generated files

"Setdest" generated Parameters for simulation	**"Cbrgen" generated Parameters for simulation**
Number of nodes (-n) : 7 to 20	Type (-type): cbr
Pause time(-p): 1	Number of Nodes (-nn): 7 to 20
Maximum speed(-M): 20	Seed(-seed): 2.0
Simulation time(-t): 500	Maximum Connection(-mc): 9
Max X(-x): 750	Rate (-rate): 10.0

University of Ghana http://ugspace.ug.edu.gh

Max Y(-y): 750	

4.0.4 Simulation of Attack

Each simulation done is seen as in ***Figure4.9***; with the attacking node being seen as the red node and the genuine nodes as those in black. The nodes were initially simulated between two extremes of 7 nodes and 20 nodes and progressively increased in between that range during the stress testing phase of the project.

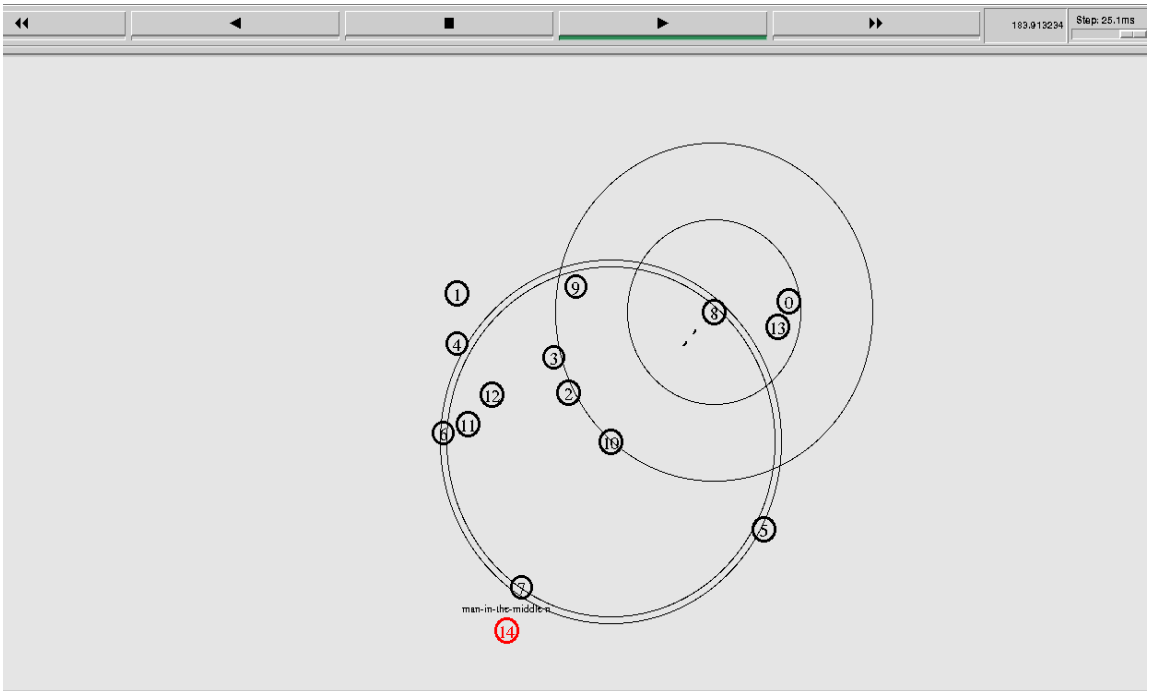


Figure 4.9 - MITM simulation attack with 14 genuine nodes and 1 malicious node

4.1 Machine Learning - "*Wrapper Method*"

There are two popular methods used in feature selection, namely: the **filter method**

University of Ghana http://ugspace.ug.edu.gh

(more suited for data mining)and the **wrapper method** ***(more suited for machine learning)***. Using the weka software tool, it was observed that the wrapper method was the best, due to the fact that the starting number of features - twenty six (26) in total; which were extracted by the Perl script **"dataCleanerBetter.pl"**- shown in ***Table B.2*****,**was relatively small. Essentially, the problem at that stage was a machine learning problem as opposed to a data mining problem. So, the wrapper method helped to identify the features that could offer the most contribution, in terms of machine learning algorithm. Conceptually, the wrapper method creates all possible subsets from the feature vector and then uses a classification algorithm to induce classifiers from each feature in each subset. It would give the set of features in which the classification algorithm (**MultilayerPerceptron in this case**) performs the best. The search technique adopted by the evaluator (**ClassifierSubsetEval as chosen)** in its quest to find the best classifier could be depth first search, breadth first search, a random search or a hybrid search. The**BestFirst** search method was used in this case. However, prior to using the MultilayerPerceptron, features that by inspection added no new information (remained constant in the values they presented) were eliminated. These features were **Pn, Po, Nz, Nw, Ne, Nl, Ma, Md, Ms, and If**. The pruned vector of features contained **Hs, Hd, Id, Ii, Il, ls, lt, lv, Mt, Ni, Nx, Ny, Pf, Pi, t, and PM** with **It** as the classification feature. These features can be seen in **APPENDIX C**, showing files containing both unfiltered results – ***Table C.1***and resulting Perl-filtered ".arff" files as show in ***Table C.2***.

University of Ghana http://ugspace.ug.edu.gh

4.2 Filtering – Perl

To facilitate the necessary stages of clustering and subsequent classification, the Perl scripts presented in ***Table B.2*** are used to extract the necessary features, as well as preprocess the files gotten from the feature extraction phase for possible input to weka or into a ".arff" file, presentable to the***MitmProtectorWithWeka.jar*** software solution meant for attack detection. A template of such a file is given in ***Table C3***.

4.2.1 Clustering

Even before the selection of the classification algorithm, clustering was done using the simple **weka's** implementation of the simple k-means algorithm. This was done to note the features that contributed the most to the clustering.

4.2.2 Classification

Classification was achieved by first training the system to distinguish between three main classes. The packets that were represented in the **It** classification vector were found to be of three types. They were of types AODV, AODV_MITM or CBR. The ANN classification algorithm used was the **MultilayerPerceptron** at 10-fold cross Validation using one hidden layer with ten nodes arrived atby using the weka generated algorithm; ***a = (attribs + classes) /2***;and sby try and error; having been limited by the computational power of the machine being used. This is shown in ***Figure 4.10*** which depicts the various classes, together with the learning rate.Additionally,Table 4-3 show the parameters used in the creation of the**MultilayeredPerceptron** adopted for weka and used in the generation of the various models. After several trial and error approaches, over several

University of Ghana http://ugspace.ug.edu.gh

models; examples of which are ***Figure 4.11 to 4.14***, one of the final solutions achieved is shown in ***Figure4.14*** having 89.7148 % of instances correctly classified out of 20651 instances. The generated and saved models (in .model format) offered viable neural networks by which new data from newly generated attack scenarios could be tested. The models were successively loaded back into weka to be tested again any newly generated attack data (gotten from the .arff files)and those models were sequentially used to classify the packets to see which could best detect the greatest number of attacking nodes

.

Table 4-3 - Multilayered Perceptron Parameters

Parameter for Multilayered Perceptron in Weka	**Value**
Learning rate	0.3
Number of Epochs	Number of Nodes (-nn): 7 to 20
Seed	0
Training Time	500
Momentum	0.2

University of Ghana http://ugspace.ug.edu.gh

Validation Threshold	20
ValidationSetSize	0

Table 4-4 - Characteristics of nodes (set using the *node-config*tcl command)

Parameter of Nodes and Network	**Value**
-llType (Link layer type)	LL
-propType (Radio Propagation)	Propagation/TwoRayGround
-phyType (Network Interface)	Phy/WirelessPhy
-macType (Mac Type)	Mac/802_11
-ifqTyp (Interface queue type)	Queue/DropTail/PriQueue
-antType (Antenna Model)	Antenna/OmniAntenna
-ifqLen (Maximum packet in interface queue – ifq)	150
-adhocRouting (Routing protocol)	AODV

University of Ghana http://ugspace.ug.edu.gh

Packet Size (gotten from the **setdest** generated files	512
-agentTrace	On
-routerTrace	On
-macTrace	On
-movementTrace	On

4.2.3 Model Generation

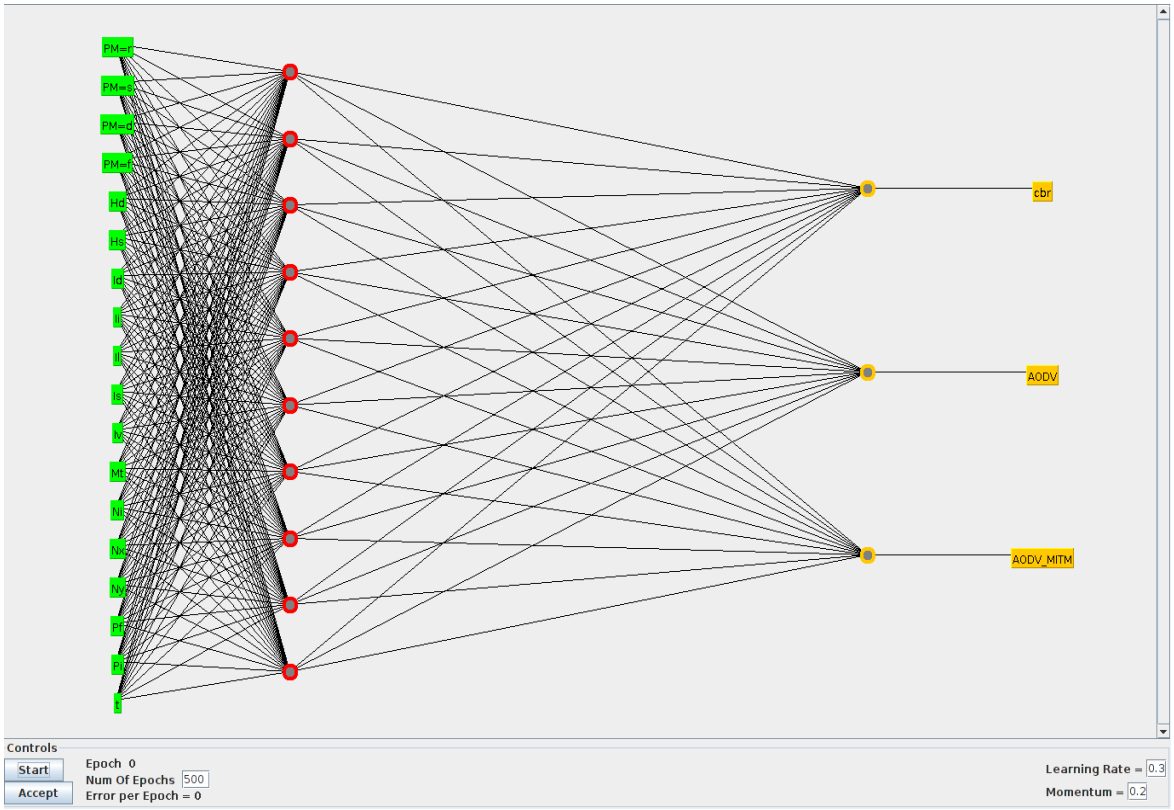


Figure 4.10 -Initial ANN Model Generation

University of Ghana http://ugspace.ug.edu.gh

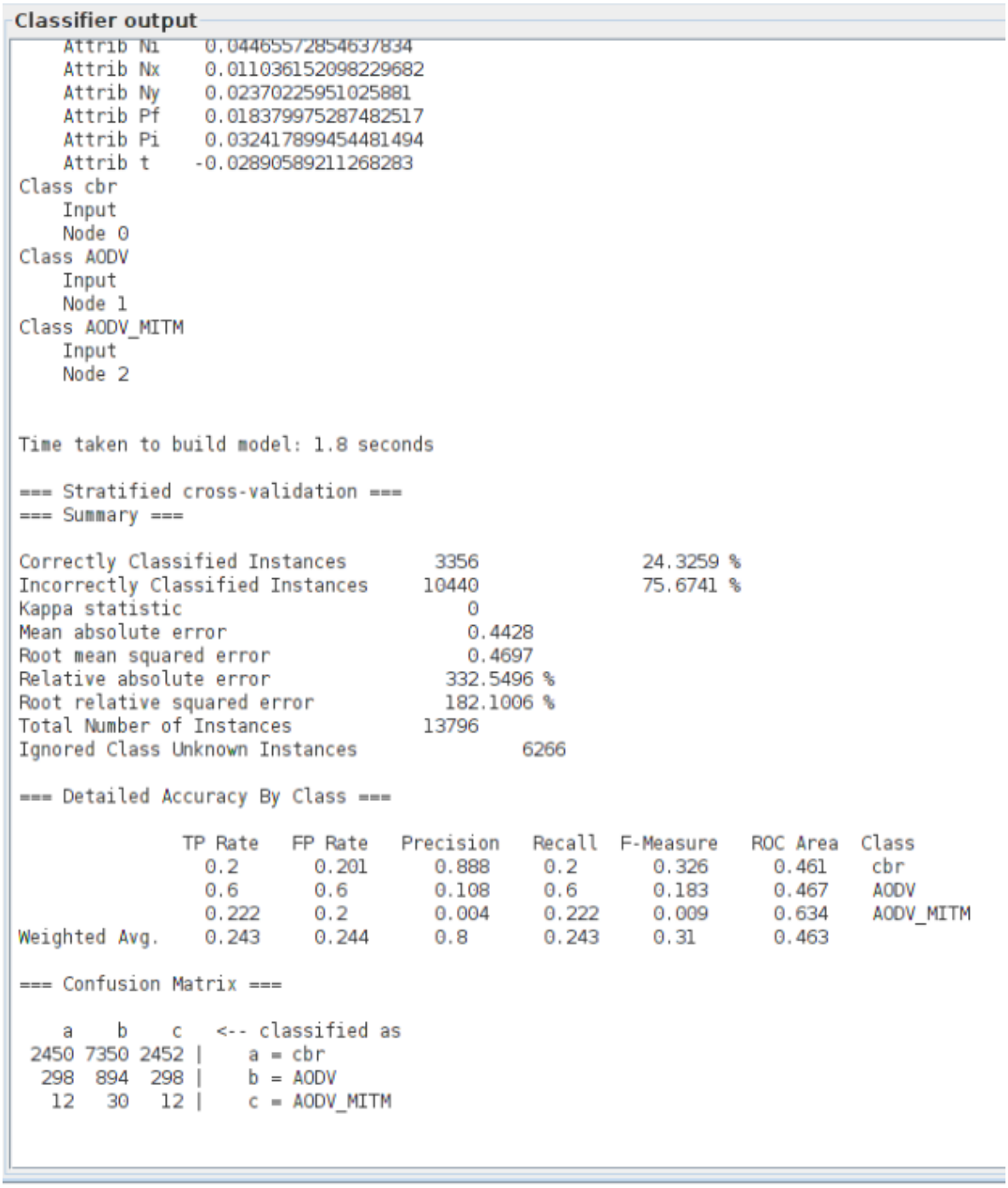


Figure 4.11 - Model 1

University of Ghana http://ugspace.ug.edu.gh

Classifier output

```
    Inputs    Weights
    Threshold    -0.01304586277190635
    Attrib Ii    0.04186672918394452
    Attrib Il    0.007303138466759694
    Attrib Is    0.003943144876576897
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


Time taken to build model: 3.78 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3356               24.3259 %
Incorrectly Classified Instances     10440               75.6741 %
Kappa statistic                          0
Mean absolute error                      0.4447
Root mean squared error                  0.4717
Relative absolute error                333.9885 %
Root relative squared error            182.8726 %
Total Number of Instances            13796
Ignored Class Unknown Instances              6266

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.2       0.201      0.888     0.2       0.326      0.675    cbr
                 0.6       0.6        0.108     0.6       0.183      0.393    AODV
                 0.222     0.2        0.004     0.222     0.009      0.648    AODV_MITM
Weighted Avg.    0.243     0.244      0.8       0.243     0.31       0.644

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2450 7350 2452 |    a = cbr
  298  894  298 |    b = AODV
   12   30   12 |    c = AODV_MITM
```

Figure 4.12 - Model 2

University of Ghana http://ugspace.ug.edu.gh

Classifier output

```
Sigmoid Node 5
    Inputs    Weights
    Threshold    0.0017826907728076408
    Attrib Ii    -0.04738771611408561
    Attrib Il    0.02463487054209977
    Attrib Is    0.04047385972453969
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


Time taken to build model: 1.64 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       13188               63.8613 %
Incorrectly Classified Instances      7463               36.1387 %
Kappa statistic                          0.0003
Mean absolute error                      0.4433
Root mean squared error                  0.4702
Relative absolute error                359.1554 %
Root relative squared error            189.3158 %
Total Number of Instances            20651
Ignored Class Unknown Instances              9411

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.7       0.7       0.897      0.7       0.786      0.717    cbr
                 0.1       0.1       0.1        0.1       0.1        0.735    AODV
                 0.222     0.2       0.003      0.222     0.006      0.666    AODV_MITM
Weighted Avg.    0.639     0.638     0.815      0.639     0.716      0.719

=== Confusion Matrix ===

     a     b     c   <-- classified as
 12969  1852  3706 |     a = cbr
  1449   207   414 |     b = AODV
    37     5    12 |     c = AODV_MITM
```

Figure 4.13 - Model 3

University of Ghana http://ugspace.ug.edu.gh

Classifier output

```
Sigmoid Node 5
    Inputs    Weights
    Threshold    0.0017826907728076408
    Attrib Ii    -0.04738771611408561
    Attrib Il    0.02463487054209977
    Attrib Is    0.04047385972453969
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


Time taken to build model: 1.85 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       18527               89.7148 %
Incorrectly Classified Instances      2124               10.2852 %
Kappa statistic                          0
Mean absolute error                      0.4417
Root mean squared error                  0.4685
Relative absolute error                357.688  %
Root relative squared error            188.6065 %
Total Number of Instances            20651
Ignored Class Unknown Instances               9411

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.897     1         0.946      0.533    cbr
                 0         0          0         0         0          0.75     AODV
                 0         0          0         0         0          0.452    AODV_MITM
Weighted Avg.    0.897     0.897      0.805     0.897     0.849      0.555

=== Confusion Matrix ===

     a     b     c   <-- classified as
 18527     0     0 |     a = cbr
  2070     0     0 |     b = AODV
    54     0     0 |     c = AODV_MITM
```

Figure 4.14 - Model 4

University of Ghana http://ugspace.ug.edu.gh

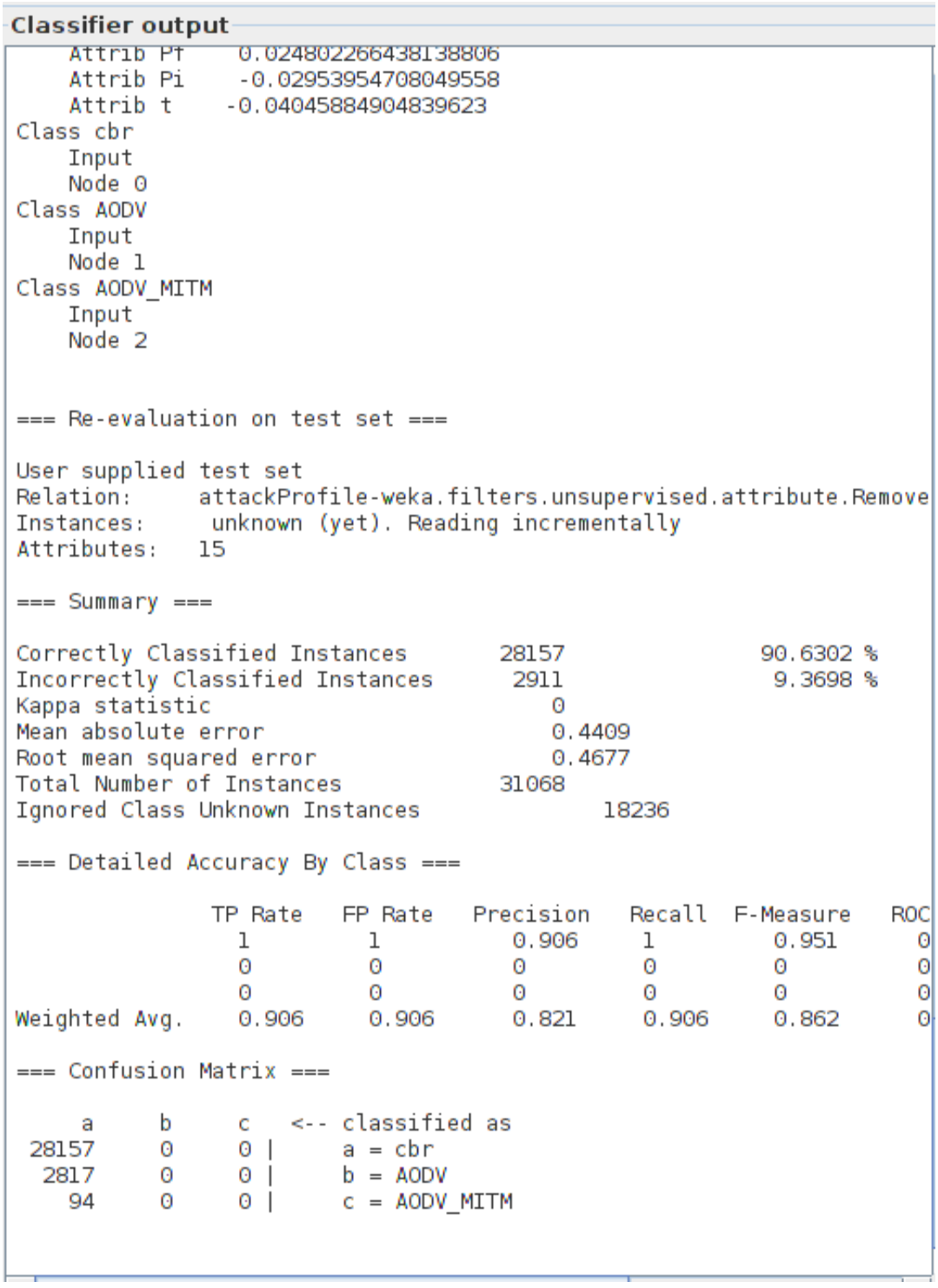
4.3 Software Solution – Java

The Java implementation of the final software solution - ***MitmProtectorWithWeka.jar***, as presented in ***Table B.5*** of **APPENDIX B,** makes use of the **weka** API (Application Programming Interface). It automatically selects the correct features and runs the trained software against any user configured attack detection log file. It is from this that it generates a possible blacklist, for automatic attack detection and system reconfiguration.

4.4 Testing and results

Stress Testing done afterwards, using the weka software tool and later on the final software, for several nodes are shown in ***Table 4-5 and Table 4-6***. Those tests captured the performance of the developed model showing it to be very efficient at attack detection for any number of nodes - ***Figure4.18***.

University of Ghana http://ugspace.ug.edu.gh

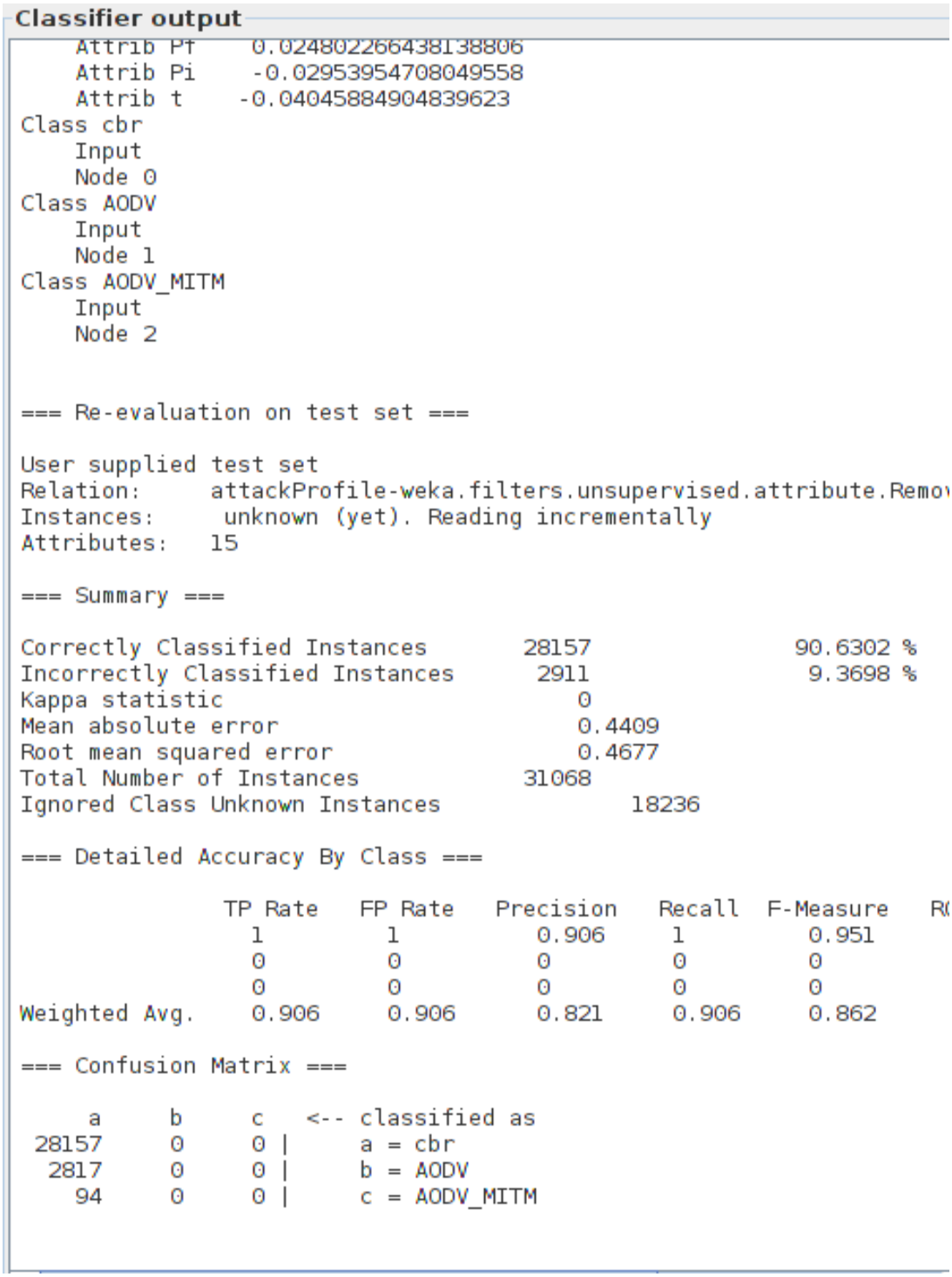


Classifier output

```
    Attrib Pf     0.024802266438138806
    Attrib Pi     -0.02953954708049558
    Attrib t     -0.04045884904839623
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


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

User supplied test set
Relation:     attackProfile-weka.filters.unsupervised.attribute.Remove
Instances:     unknown (yet). Reading incrementally
Attributes:   15

=== Summary ===

Correctly Classified Instances       28157               90.6302 %
Incorrectly Classified Instances      2911                9.3698 %
Kappa statistic                          0
Mean absolute error                      0.4409
Root mean squared error                  0.4677
Total Number of Instances            31068
Ignored Class Unknown Instances              18236

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC
                 1         1          0.906     1         0.951      0
                 0         0          0         0         0          0
                 0         0          0         0         0          0
Weighted Avg.    0.906     0.906      0.821     0.906     0.862      0

=== Confusion Matrix ===

     a     b     c   <-- classified as
 28157     0     0 |     a = cbr
  2817     0     0 |     b = AODV
    94     0     0 |     c = AODV_MITM
```

Figure 4.15 - Testing 1 with ten-fold cross validation given selected features over 60062 instances

University of Ghana http://ugspace.ug.edu.gh

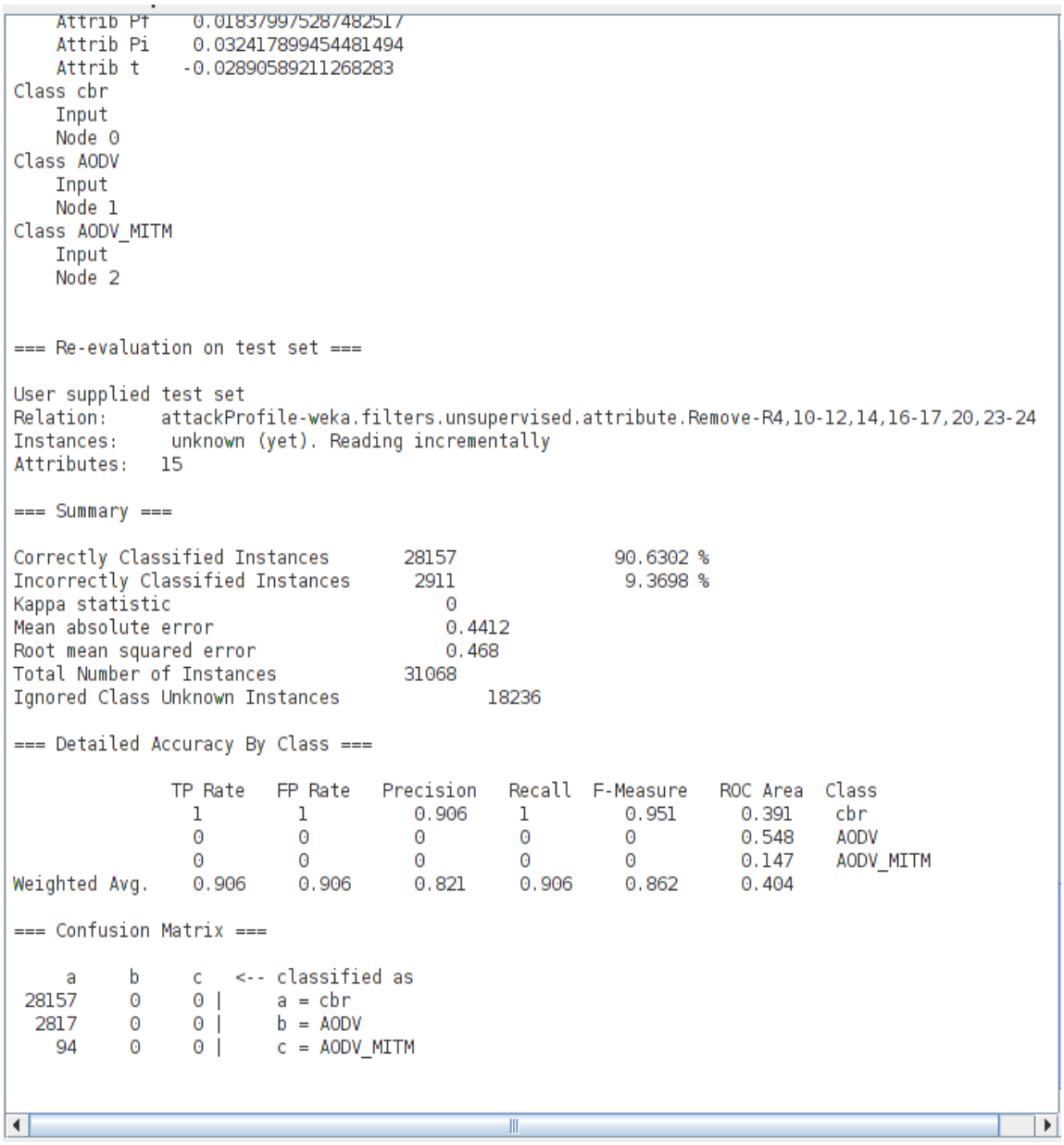


Classifier output

```
    Attrib Pf     0.024802266438138806
    Attrib Pi     -0.02953954708049558
    Attrib t     -0.04045884904839623
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


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

User supplied test set
Relation:     attackProfile-weka.filters.unsupervised.attribute.Remov
Instances:     unknown (yet). Reading incrementally
Attributes:   15

=== Summary ===

Correctly Classified Instances       28157               90.6302 %
Incorrectly Classified Instances      2911                9.3698 %
Kappa statistic                          0
Mean absolute error                      0.4409
Root mean squared error                  0.4677
Total Number of Instances            31068
Ignored Class Unknown Instances              18236

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   RO
                 1         1          0.906     1         0.951
                 0         0          0         0         0
                 0         0          0         0         0
Weighted Avg.    0.906     0.906      0.821     0.906     0.862

=== Confusion Matrix ===

     a     b     c   <-- classified as
 28157     0     0 |     a = cbr
  2817     0     0 |     b = AODV
    94     0     0 |     c = AODV_MITM
```

Figure 4.16 -testing 2, a 66 percentage split model with 15 features selected over 60072 instances

University of Ghana http://ugspace.ug.edu.gh



```
    Attrib Pf    0.018379975287482517
    Attrib Pi    0.032417899454481494
    Attrib t    -0.02890589211268283
Class cbr
    Input
    Node 0
Class AODV
    Input
    Node 1
Class AODV_MITM
    Input
    Node 2


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

User supplied test set
Relation:     attackProfile-weka.filters.unsupervised.attribute.Remove-R4,10-12,14,16-17,20,23-24
Instances:     unknown (yet). Reading incrementally
Attributes:   15

=== Summary ===

Correctly Classified Instances       28157               90.6302 %
Incorrectly Classified Instances      2911                9.3698 %
Kappa statistic                          0
Mean absolute error                      0.4412
Root mean squared error                  0.468
Total Number of Instances            31068
Ignored Class Unknown Instances              18236

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.906     1         0.951      0.391    cbr
                 0         0          0         0         0          0.548    AODV
                 0         0          0         0         0          0.147    AODV_MITM
Weighted Avg.    0.906     0.906      0.821     0.906     0.862      0.404

=== Confusion Matrix ===

     a     b     c   <-- classified as
 28157     0     0 |     a = cbr
  2817     0     0 |     b = AODV
    94     0     0 |     c = AODV_MITM
```

Figure 4.17 -testing 3; a tenfold cross validation model with 15 selected attributes over 20062 instances

4.5 Discussion of Results

It can be seen from the testing figures; ***Figure4.15***to ***Figure 4.17***,as well as ***Table 4-6*** results presented per node for the multilayered perceptron based classification algorithm

University of Ghana http://ugspace.ug.edu.gh

for the classes under consideration; **CBR, AODV, and AODV_MITM** in the various instances have very high **true positive rate (TP)** and very low **false positive rate (FP)**. The *true positive* rate is the proportion of instances which were classified as a specific class x(CBR, AODV or AODV_MITM in this case), among all the examples which truly have that class x. It is equivalent to Recall. The *false positive* rate on the other hand, presents the proportion of instances, classified as x but actually belong to a different class. The *Precision* is the proportion of the examples which truly have class x taken out of all those which were classified as class x. The F-Measure which presents a combined measure for precision and recall is computed using the formula 2*Precision*Recall/(Precision+Recall). All these values were measured using weka after running the generated models against the test data and are presented in the confusion matrix (better known as the contingency table) shown in the aforementioned figures. This shows that the system (software) performs well when used in time signature fingerprinting of packets from attacking nodes. It is able to viably distinguish between genuine and malicious sourced packets. **Table 4-5**shows appreciably high percentages per node for correctly classified instances with an average detection rate of 88.235% over eleven tests runs using different number of nodes; with relatively lower incorrectly classified instances. The **mean absolute error** as well as **root mean square (RMS)** errors of between 0.4409 to 0.4412 and 0.4677 to 0.468 respectively are quite low; each increasing gradually as node number. These error values are not really meaningful for classification purposes. For regression related tasks however, the **RMS** could be very helpful. As such since the work done mainly relates to classification, this value was not factored into the analysis. This means that the system performs better for lower node

University of Ghana http://ugspace.ug.edu.gh

numbers and its performance declines ever slightly with increase in number. That provided a great insight for future research.

4.5.1 Stress Testing

Table 4-5- Stress testing

No of Nodes	Correctly classified instances	Incorrectly classified instances	Mean absolute error	Root mean square error	Total Number of instances	Ignored Class Unknown Instances -before classification
8	15458(99.8643%)	21(0.1357 %)	0.0011	0.0246	15479	14495
9	20190 (99.8912 %)	22(0.1088 %)	0.0012	0.0261	20212	9759
10	15861 (99.5169 %)	77(0.4831 %)	0.004	0.0532	15938	14033

University of Ghana http://ugspace.ug.edu.gh

11	14071(99.4768 %)	74(0.5232 %)	0.004	0.0525	14145	15826
12	15435(99.69%)	48(0.31%)	0.0026	0.041	15483	14488
13	13388(95.7722 %)	591(4.2278 %)	0.0293	0.1555	13979	15992
14	13725(97.368 %)	371(2.632 %)	0.0178	0.1253	14096	15875
15	13541(95.8112 %)	592(4.1888 %)	0.0282	0.1597	14133	15838
17	13804(96.1147%)	558(3.8853 %)	0.8113	0.0267	0.1559	15609
18	13417(92.1244 %)	1147(7.8756 %)	0.0534	0.2279	14564	15407
19	14542(85.1804%)	2530(14.8196%)	0.0995	0.3101	17072	12899

4.5.2 Detailed Accuracy by Class

Table 4-6 - Detailed accuracy by class

8 nodes	TP Rate	FP Rate	Precision	Recall	F-	ROC	Class

University of Ghana http://ugspace.ug.edu.gh

					Measure	Area	
	1	0	1	1	1	1	cbr
	0.658	0	1	0.658	0.794	0.794	AODV
	0	0.039	0	0	0	-	AODV_ MITM
Weighted Average	0.961	0	1	0.961	0.977	0.98	
9 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.991	0	1	0.991	0.996	0.997	AODV
	1	0.001	0.864	1	0.927	1	AODV_ MITM
Weighted Average	0.999	0	0.999	0.999	0.999	1	
10 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.96	0	1	0.96	0.979	0.798	AODV

University of Ghana http://ugspace.ug.edu.gh

	0	0.005	0	0	0	-	AODV_MITM
Weighted Average	0.995	0	1	0.995	0.998	0.997	
11 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	0.999	cbr
	0.941	0	1	0.941	0.97	0.97	AODV
	0	0.005	0	0	0	-	AODV_MITM
Weighted Average	0.995	0	1	0.995	0.997	0.996	
12 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.953	0	1	0.953	0.976	0.976	AODV
	0	0.003	0	0	0	-	AODV_MITM
Weighted Average	0.997	0	1	0.997	0.998	0.998	

University of Ghana http://ugspace.ug.edu.gh

13 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.74	0	1	0.74	0.85	0.844	AODV
	1	0.042	0.039	1	0.075	0.992	AODV_MITM
Weighted Average	0.958	0	0.998	0.958	0.974	0.974	
14 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.751	0	1	0.751	0.858	0.871	AODV
	0	0.026	0	0	0	-	AODV_MITM
Weighted Average	0.974	0	1	0.974	0.985	0.986	
15 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr

University of Ghana http://ugspace.ug.edu.gh

	0.632	0	1	0.632	0.775	0.793	AODV
	1	0.042	0.041	1	0.078	0.984	AODV_MITM
Weighted Average	0.958	0	0.998	0.958	0.973	0.976	
17 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.62	0	1	0.62	0.77	0.78	AODV
	0	0.05	0	0	0	-	AODV_MITM
Weighted Average	0.95	0	1	0.95	0.97	0.971	
18 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.62	0	1	0.615	0.76	0.781	AODV
	0	0.079	0	0	0	-	AODV_MITM

University of Ghana http://ugspace.ug.edu.gh

Weighted Average	0.921	0	1	0.921	0.951	0.955	
19 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	0.999	cbr
	0.547	0	1	0.547	0.707	0.748	AODV
	0	0.148	0	0	0	-	AODV_MITM
Weighted Average	0.852	0	1	0.852	0.904	0.916	
20 nodes	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	cbr
	0.658	0	1	0.658	0.794	0.794	AODV
	0	0.039	0	0	0	-	AODV_MITM
Weighted Average	0.961	0	1	0.961	0.977	0.98	

University of Ghana http://ugspace.ug.edu.gh

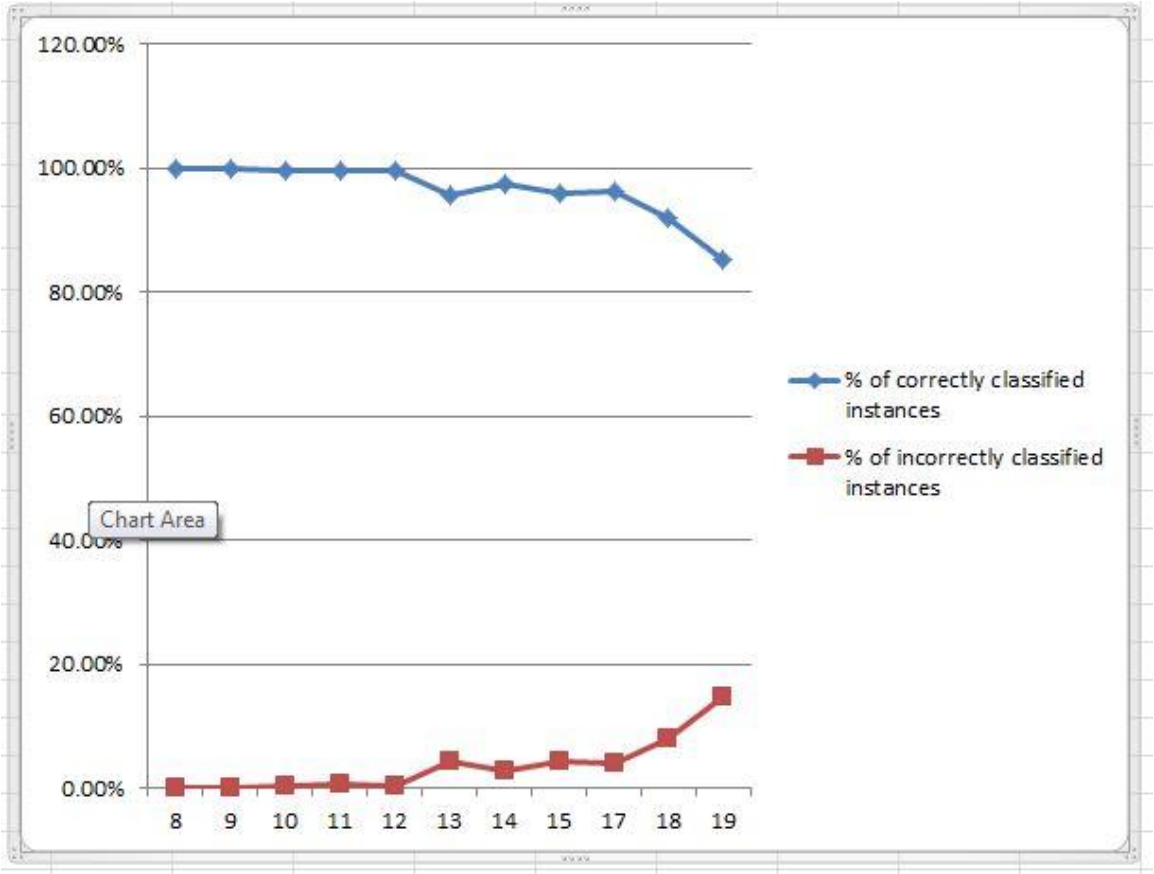


Figure 4.18 - Stress testing trend

4.6 Pre-Research Expected Outcomes

The expected outcome was a software system which offered intelligent and dynamic packet routing. The idea was to; if not completely take out the attacker, really complicate his work since he or she would not only have to be in the middle of just one route but would in addition have to guess or find out how many paths he or she would have to place himself or herself in-between. The system was also to be very easy for anyone to configure and easier for add-ons to be made to make it applicable to other network devices and for future research extensions.

University of Ghana http://ugspace.ug.edu.gh

4.7 Accomplished (Met Objectives)

Table 4-7– Objectives met

Objective	**How it was met**
Ensure that the man in the middle cannot detect where packets are from	After detection and reconfiguration, the illicit node is taken out of the network
Monitor the network for promiscuous nodes.	Obtained by periodically running the ANN section of the **MitmProtectorWithWeka** solution, and generating the blacklist for reconfiguration.
Detect man-in-the-middle attacks particularly those using spoofing	Using the complete set of **Perl based filter scripts, MitmProtectorWithWeka, as well as Perl based reconfiguration scripts**
Learn from the noticed scenario and adapt the network to counteract certain future exploits.	By periodically retraining the ANN

Table 4-7– Objectives met, shows how the various objectives that were set for the

University of Ghana http://ugspace.ug.edu.gh

research were met. Each distinct target was well addressed. It is worth mentioning that, the various solutions somewhat overlap, in the sense that, they are each needed to some degree for the realization of the other solutions and are not independent of each other.

University of Ghana http://ugspace.ug.edu.gh

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.0 Conclusion

From the results as presented in ***Table 4.3***, it can be seen that the model generated for the classification performs admirably well, having been tested for different node numbers – each recording very high percentages, for correctly classified instances, at reasonable Root Mean Square Error (RMS) values. The average detection rate of 88.235 presents a model who's ability to distinguish between AODV packets and cbr packets is quite high. The average true positive reading of 0.25 gotten from the stress tests run from eight (8) to twenty (20) nodes juxtaposed to an average false positive reading of 0.0399 gives indication of a neural network model that can adequately detect attacks when trained on significant time details. That said future research could focus on increasing the time delay window to see how any resulting model will fair with an increase in time delay. This research used a time delay of two (2) time units to come up with its results. An increase in time thus could be envisaged to increase the performance of the model. The implication is that such an approach could be used by system administrators and security experts to time signature fingerprint popular software used by attackers by using these fingerprints to train the ANN and then detect attacks in the process. ***Thus applying ANN in time signature fingerprinting of MITM attacks for attack detection and prevention is a very viable approach.***

University of Ghana http://ugspace.ug.edu.gh

5.1 Recommendations

Future research could include the involving of more of the **unknown instances –** which had to be discarded from the model recorded in attack decision making. One of the difficulties that were noticed during the training phase of the ANN algorithm was the number of instances that were not used because they were not really distinguishable. To improve the detection rate of any future model, any future work should focus on getting higher true positive rates out of the confusion matrix for AODV_MITM packets since this would suggest a better detection rate for attacks. One of the possible ways could be to increase the time lapse for packet time transmission between promiscuous nodes which send out AODV_MITM packets.

The original work could also be extended to include popular commercial or open-source spyware for performing MITM attacks. Time signature fingerprinting them and adding their learnt time signature fingerprints could be vital for security researchers in the detection and prevention of future attack.

University of Ghana http://ugspace.ug.edu.gh

References

[1] P. Goyal, V. Parmar, and R. Rishi, "MANET: Vulnerabilities, Challenges, Attacks, Application," *IJCEM International Journal of Computational Engineering & Management*, vol. 11, pp. 2230–7893, Jan. 2011.

[2] J. Pearlman, "Visualizing network security events using compound glyphs from a service-oriented perspective," Apr. 2007.

[3] RSA Security LLC, "RSA White Paper, Making Sense of Man-In-the-browser Attacks: Threat Analysis and Mitigation for Financial Institutions." 2010.

[4] "Cyberwarface in the United States," *Wikipedia*, 14-Jun-2014. .

[5] W. S. McCulloch and W. H. Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity"," *Bulletin of Mathematical Biophysics*, pp. 115–133.

[6] F. Rosenblatt, "The Perceptron: A Probabilistic Model For Information Storage And Organization in The Brain," pp. 386–408, 1958.

[7] "Deep learning," *Wikipedia*, 16-Jul-2014. .

[8] B. Lewis and P. T. Davis, "Wireless Network for Dummies," 2004, pp. 12–20.

[9] S. Kuchinskas, "Crash Course: Training the Brain of a Driverless Car," *Scientific America*, Apr. 2013.

[10] C. O'Rourke and S. B. Johnson, "Mobile Ad Hoc Networking Revamps Military Communications COTS JOURNAL," *The Journal of Military Electronics and Computing*, 2011.

[11] T. M. Michell, "Machine Learning," pp. 1–10, Mar. 1997.

University of Ghana http://ugspace.ug.edu.gh

[12] S. A. amro, K. Aldrawiesh, and A. Al-Ajlan, "A Comparative study of Computational Intelligence in Computer Security and Forensics."

[13] K. E. Defrawy and G. Tsudik, "ALARM: Anonymous Location-Aided Routing in Suspicious MANETs," *Proc IEEE TRANSACTIONS ON MOBILE COMPUTING*, 2013.

[14] "Catalogue of B.Tech. Project Reports, Batch – 2005-09 Abstracts, Resource Centre DA-IICT, Gandhinager – 382007," pp. 16–36.

[15] Z. Moradi and M. Teshnehlab, "Intrusion Detection Model in MANETs using ANNs and ANFIS," *2011 International Conference on Telecommunication Technology and Applications*, 2011.

[16] S. Maag et al, "A FORMAL VALIDATION METHODOLOGY FOR MANET ROUTING PROTOCOLS BASED ON NODES' SELF SIMILARITY," *Elsevier – ScienceDirect, Computer Communications 31*, pp. 827–841, 2008.

[17] R. Akbani, T. Korkmaz, and G. V. S. Raju, "HEAP: A PACKET AUTHENTICATION SCHEME FOR MOBILE AD HOC NETWORKS," *Elsevier – ScienceDirect, Ad Hoc Networks 6*, pp. 1134–1150, 2008.

[18] A. K. Rai, R. R. Tewari, and S. K. Upadhyay, "Different Types of Attacks on Integrated MANET-InternetCommunication."

[19] M. A. Abdalla, A. H. Almazeed, I. A. Saroit, A. Kotb, and A. Zewail, "Detection and Isolation of Packet Dropping Attackers in MANETs," *International Journal for Advanced Computer Science and Applications*, vol. Vol 4. No 4, 2013.

University of Ghana http://ugspace.ug.edu.gh

[20] Y. Chen, S. Das, P. Dhar, A. E. Saddik, and A. Nayak, "Detecting and Preventing IP-spoofed Distributed DOS attacks," *International Journal of Network Security, Vol.7, No.1*, pp. 70–81, Jul. 2008.

[21] Z. Zhao, H. Hu, G.-J. Ahn, and R. Wu, "Risk Aware Response for Mitigating MANET Routing Attacks," *IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings*, 2010.

[22] S. Dokurer, "SIMULATION OF BLACK HOLE ATTACK IN WIRELESS AD-HOC NETWORKS," 2006.

[23] A. Mitrokotsa and C. Dimitrakakis, "Intrusion detection in MANET using classification algorithms: The effects of cost and model selection," *Elsevier*, May 2012.

[24] P. Sahu, S. K. Bisoy, and S. Sahoo, "Detecting and Isolating Malicious Node in AODV Routing Algorithm," *International Journal of Computer Applications*, vol. 66– No.16, pp. 0975 – 8887, Mar. 2013.

[25] B. Wu, J. Chen, J. Wu, and M. Cardei, "A Survey on Attacks and Countermeasures inMobile Ad Hoc Networks," in *WIRELESS/MOBILE NETWORK SECURITY*, Springer, p. 2006.

[26] Y. Kim and A. Helmy, "SWAT: Small World-based Attacker Traceback in Ad-hoc Networks," 2005.

[27] Z. Duan, X. Yuan, and J. Chandrashekar, "Controlling IP Spoofing based DDoS AttacksThrough Inter-Domain Packet Filters," 2006.

University of Ghana http://ugspace.ug.edu.gh

[28] T. Reidemeister, K. B¨ohm, E. Buchmann, and P. A. S. Ward, “Man-in-the-Middle Attacks in Distributed Hash-Tables.”

[29] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, “Detecting Black Hole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic LearningMethod,” *International Journal of Network Security*, vol. Vol.5, No.3, pp. 338–346, Nov. 2007.

[30] A. Lindgren, K. S. Phanse, T. Johansson, R. Br¨annstr¨om, and C. Ahlund, “Future Directions in Ad hoc Networking Research,” Apr. 2005.

[31] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in mobile ad-hoc networks : Challenges and solutions,” *IEEE Wireless Communications*, Feb. 2004.

[32] J. Chen and J. Wu, “A Survey on Cryptography Applied to Secure Mobile Ad Hoc Networks and Wireless Sensor Networks.”

[33] J. Chung and M. Claypool, “NS by Example.” 01-Aug-2012.

University of Ghana http://ugspace.ug.edu.gh

APPENDICES

A. Appendix A – Architecture

Class diagrams-AODV_MITM

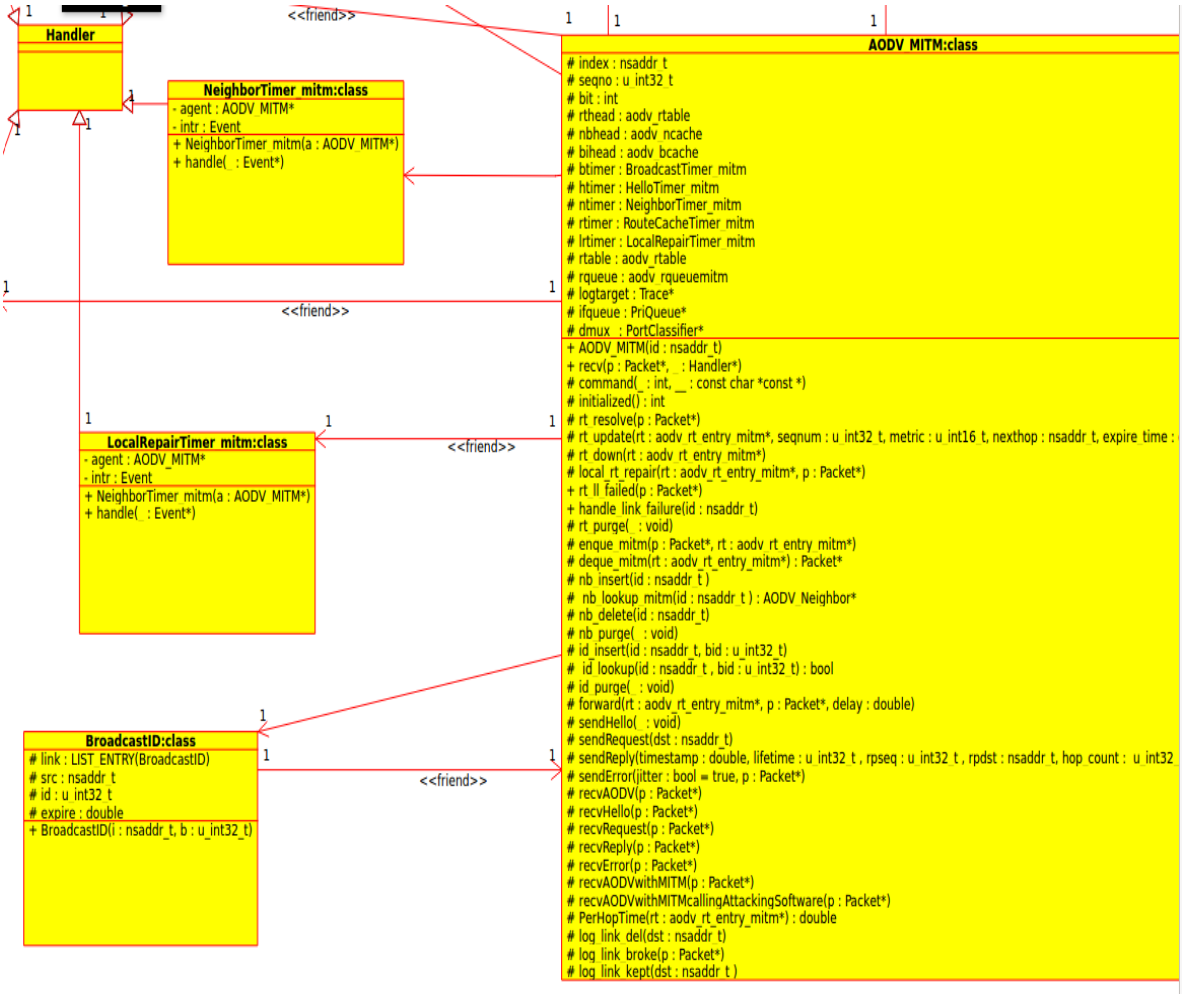


Figure A.1 - Partial class diagram of aodvmitm.(cc/h)

University of Ghana http://ugspace.ug.edu.gh

B. Appendix B – Codes

Tcl codes

Table B-1 - Tcl scripts for reduced and large attacks

***"mitm_attack_reduced.tcl"*file**	***"mitm_attack.tcl"*file**
#Define options ;#Adapted from [22] set val(chan) Channel/WirelessChannel ;#Channel Type set val(prop) Propagation/TwoRayGround ;#radio-propagation type set val(netif) Phy/WirelessPhy ;#network interface type set val(mac) Mac/802_11 ;#MAC type set val(ifq) Queue/DropTail/PriQueue ;#interface queue type set val(ll) LL ;#link layer type set val(ant) Antenna/OmniAntenna ;#antenna model set val(ifqlen) 150 ;#max packet in ifq set val(nn) 7 ;#total number of mobilenodes set val(nnaodv) 6 ;#number of AODV mobilenodes set val(rp) AODV ;#routing protocol set val(x) 750 ;#X dimension of topography set val(y) 750 ;#Y dimension of topography set val(cstop) 451 ;#time of connections end set val(stop) 500 ;#time of simulation end #FOR REDUCED NODES set val(cp) "scenarios/scen1forAODV-n_reduced-t500-x750-y750" ;#Connection Pattern set val(cc) "scenarios/cbr_reduced_nodes" ;#CBR Connections #Initialization Global Variables set ns_ [new Simulator] ##FOR RUNNING REDUCED ATTACKS $ns_ use-newtrace set tracefd [open sim1forNormalNetwork_reduced.tr w]	#Define options ;#Adapted from [22] set val(chan) Channel/WirelessChannel ;#Channel Type set val(prop) Propagation/TwoRayGround ;#radio-propagation type set val(netif) Phy/WirelessPhy ;#network interface type set val(mac) Mac/802_11 ;#MAC type set val(ifq) Queue/DropTail/PriQueue ;#interface queue type set val(ll) LL ;#link layer type set val(ant) Antenna/OmniAntenna ;#antenna model set val(ifqlen) 150 ;#max packet in ifq set val(nn) 20 ;#total number of mobilenodes set val(nnaodv) 19 ;#number of AODV mobilenodes set val(rp) AODV ;#routing protocol set val(x) 750 ;#X dimension of topography set val(y) 750 ;#Y dimension of topography set val(cstop) 451 ;#time of connections end set val(stop) 500 ;#time of simulation end #For SCALED UP set val(cp) "scenarios/scen1forAODV-n20-t500-x750-y750" ;#Connection Pattern set val(cc) "scenarios/cbr_twenty_nodes" ;#CBR Connections #Initialization Global Variables set ns_ [new Simulator] ##FOR RUNNING SCALED UP ATTACKS

University of Ghana http://ugspace.ug.edu.gh

```
$ns_ trace-all $tracefd

set namtrace [open sim1forNormalNetwork_reduced.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


#set up topography object
set topo  [new Topography]
$topo load_flatgrid $val(x) $val(y)


#Create god
create-god $val(nn)

#Create channel #1 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]

#configure node, please note the change below,
$ns_ node-config   -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON \
-channel $chan_1_ \

#Creating mobile AODV nodes for simulation

puts "Creating node..."
for { set i 0 } { $i < $val(nnaodv) } { incr i } {
```

```
$ns_ use-newtrace
set tracefd  [open sim1forNormalNetwork.tr w]
$ns_ trace-all $tracefd
set namtrace [open sim1forNormalNetwork.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

#set up topography object
set topo  [new Topography]
$topo load_flatgrid $val(x) $val(y)


#Create god
create-god $val(nn)

#Create channel #1 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]

#configure node, please note the change below,
$ns_ node-config   -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON \
-channel $chan_1_ \

#Creating mobile AODV nodes for simulation

puts "Creating node..."
for { set i 0 } { $i < $val(nnaodv) } { incr i } {
```

University of Ghana http://ugspace.ug.edu.gh

```
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;#disable random motion
}

#Creating man-in-the-middle nodes for simulation
$ns_ node-config  -adhocRouting AODV_MITM
for { set i $val(nnaodv) } { $i < $val(nn) } { incr i } {
set node_($i) [ $ns_ node ]
$node_($i) random-motion 0 ;#disable random motion
$ns_ at 0.01 "$node_($i) label \"man-in-the-middle node\""
}

#Adding connection pattern which is created using setdest, parameters shown below
#./setdest -n 20 -p 1.0 -M 20.0 -t 500 -x 750 -y 750 > scen1forAODV-n20-t500-x750-y750

puts "Loading random connection pattern..."
set god_ [ God instance ]
source $val(cp)

#################### CBRGEN GENERATE SAME CODE ##############################
#set j 0
#for {set i 0} {$i < 18} {incr i} {
#Create a UDP and NULL agents, then attach them to the appropriate nodes
#    set udp_($j) [new Agent/UDP]
#$ns_ attach-agent $node_($i) $udp_($j)
#    set null_($j) [new Agent/Null]
#    $ns_ attach-agent $node_([expr $i + 1]) $null_($j)
#    #Attach CBR application;
#    set cbr_($j) [new Application/Traffic/CBR]
#puts "cbr_($j) has been created over udp_($j)"
#$cbr_($j) set packet_size_ 512
#$cbr_($j) set interval_ 1
#$cbr_($j) set rate_ 10kb
#$cbr_($j) set random_ false
#$cbr_($j) attach-agent $udp_($j)
#$ns_ connect $udp_($j) $null_($j)
```

```
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;#disable random motion
}

#Creating man-in-the-middle nodes for simulation
$ns_ node-config  -adhocRouting AODV
for { set i $val(nnaodv) } { $i < $val(nn) } { incr i } {
set node_($i) [ $ns_ node ]
$node_($i) random-motion 0 ;#disable random motion
$ns_ at 0.01 "$node_($i) label \"man-in-the-middle node\""
}

#Adding connection pattern which is created using setdest, parameters shown below
#./setdest -n 20 -p 1.0 -M 20.0 -t 500 -x 750 -y 750 > scen1forAODV-n20-t500-x750-y750

puts "Loading random connection pattern..."
set god_ [ God instance ]
source $val(cp)

#################### CBRGEN GENERATE SAME CODE ##############################
#set j 0
#for {set i 0} {$i < 18} {incr i} {
#Create a UDP and NULL agents, then attach them to the appropriate nodes
#    set udp_($j) [new Agent/UDP]
#$ns_ attach-agent $node_($i) $udp_($j)
#    set null_($j) [new Agent/Null]
#    $ns_ attach-agent $node_([expr $i + 1]) $null_($j)
#    #Attach CBR application;
#    set cbr_($j) [new Application/Traffic/CBR]
#puts "cbr_($j) has been created over udp_($j)"
#$cbr_($j) set packet_size_ 512
#$cbr_($j) set interval_ 1
#$cbr_($j) set rate_ 10kb
#$cbr_($j) set random_ false
#$cbr_($j) attach-agent $udp_($j)
#$ns_ connect $udp_($j) $null_($j)
```

University of Ghana http://ugspace.ug.edu.gh

```
#puts "udp_($j) and null_($j) agents has been connected each other"
#$ns_ at 1.0 "$cbr_($j) start"
#    set j [expr $j + 1]
#    set i [expr $i + 1]
#}
#############################################################################
# CBR Connections generated by cbrgen
source $val(cc)



# Define initial node position
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ initial_node_pos $node_($i) 30
}

# CBR connections stops
for {set i 0} {$i < 3 } {incr i} {
$ns_ at $val(cstop) "$cbr_($i) stop"
}

# Tell all nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at $val(stop).000000001 "$node_($i) reset";
}

# Ending nam and simulation
$ns_ at $val(stop) "finish"
$ns_ at $val(stop).0 "$ns_ trace-annotate \"Simulation has ended\""
$ns_ at $val(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"

proc finish {} {
global ns_ tracefd namtrace
$ns_ flush-trace
close $tracefd
close $namtrace
#
exec nam sim1forNormalNetwork_reduced.nam &
exit 0
```

```
#puts "udp_($j) and null_($j) agents has been connected each other"
#$ns_ at 1.0 "$cbr_($j) start"
#    set j [expr $j + 1]
#    set i [expr $i + 1]
#}
#############################################################################
# CBR Connections generated by cbrgen
source $val(cc)



# Define initial node position
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ initial_node_pos $node_($i) 30
}

# CBR connections stops
for {set i 0} {$i < 9 } {incr i} {
$ns_ at $val(cstop) "$cbr_($i) stop"
}

# Tell all nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at $val(stop).000000001 "$node_($i) reset";
}

# Ending nam and simulation
$ns_ at $val(stop) "finish"
$ns_ at $val(stop).0 "$ns_ trace-annotate \"Simulation has ended\""
$ns_ at $val(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"

proc finish {} {
global ns_ tracefd namtrace
$ns_ flush-trace
close $tracefd
close $namtrace
#
exec nam sim1forNormalNetwork.nam &
exit 0
```

University of Ghana http://ugspace.ug.edu.gh

}	}
puts "Starting Simulation..."	puts "Starting Simulation..."
$ns_ run	$ns_ run

Perl scripts

Table B-2 - Perl scripts for feature extration and data filtering

"dataCleanerBetter.pl" file	***"wekaPreprocessorEquivalent.pl" file***
#!/usr/bin/Perl	#!/usr/bin/Perl
	#Author: Kwadwo Boateng Ofori-Amanfo
use warnings;	#Purpose: For doing the preprocessing weka would have been done
use strict;	
use FileHandle;	use warnings;
	use strict;
#GLOBAL VARIABLES	use FileHandle;
my $maxArraySize = 0;	
my @columnHeaders = ('-Ma');#for building up column headers	#GLOBAL VARIABLES
my @features=();#for taking in the extracted features from the headers printed into the file	my $maxArraySize = 0;
my $packetMode = 'NULL';	my $F_IN = FileHandle->new("filteredResults.arff");
my $F_IN = FileHandle->new("resultsWithoutMovement.txt");	my $F_OUT = FileHandle->new(">filteredResults1.arff");#this first open cleans the file and writes afresh to it
my $F_OUT = FileHandle->new(">filteredResults.arff");#this first open cleans the file and writes afresh to it	
	##MAIN PROGRAM
##MAIN PROGRAM	
	my $attribSelect = "\@relation 'attackProfile-weka.filters.unsupervised.attribute.Remove-R11-13,15,17-18,21,24-25-weka.filters.unsupervised.attribute.Remove-R5-weka.filters.unsupervised.attribute.Reorder-R1,2,3,4,5,6,7,9,10,11,12,13,14,15,16,8'";
while (my $line = $F_IN->getline) {	
	while (my $line = $F_IN->getline) {
#for getting the highest number of headers	
chomp $line;	$line =~ s/\@relation attackProfile/$attribSelect/;
my @array = split(' ',$line);	
$packetMode = shift @array;	#attributes selected after heuristics
	$line =~ s/\@attribute PM/\@attribute PM {r,s,d,f}/;

University of Ghana http://ugspace.ug.edu.gh

```
my @nullpadder = ('-NULL','NULL');#padding the array with NULLS if it would result is an odd hash below
if(((scalar @array)/2)%2 != 0) { push(@array,@nullpadder) ;}
my %hashFromArray = @array;
my @newArray = keys %hashFromArray;
my $newArray = join(":",@newArray);#my (@newHeaderList) = @{header_processor($newArray)};
if(header_processor($newArray)){
                @features = header_processor($newArray);
                #$F_OUT->print("NEW HEADERS:",@features,"\n");
                #$F_OUT->print("NEW HEADERS:",header_processor($newArray),"\n");
                }



#for getting the fields with values
my @keysArray;
my @valuesArray;
my $arrayLength = scalar (keys %hashFromArray);
$maxArraySize = $arrayLength if ($arrayLength >= $maxArraySize); #for checking and setting the maximum number of possible columns discovered per line
while ((my $key,my $value) = each(%hashFromArray)){
        push @keysArray,$key;
        push @valuesArray,$value;
#$F_OUT->print("$key:","$value\n");
}


}
        #$F_OUT->print("NEW HEADERS:",@features,"\n");#for checking if the features are have actually been extracted


        $F_OUT->print("\@relation attackProfile\n\n");
        my @featuresEdit = @features;#given to @featuresEdit instead of using @features directly because that would cause the values of @features to be changed during the substitution below
        $F_OUT->print("\@attribute PM\n"); #PM: for printing out the $packetMode attribute header. The only header generated as constant. The others are generated in the loop below
        foreach my $heads(sort @featuresEdit){#this loop is for listing the attributes at the beginning of the file
$line =~ s/\@attribute Hd/\@attribute Hd numeric/;
$line =~ s/\@attribute Hs/\@attribute Hs numeric/;
$line =~ s/\@attribute Id/\@attribute Id numeric/;
$line =~ s/\@attribute Ii/\@attribute Ii numeric/;
$line =~ s/\@attribute Il/\@attribute Il numeric/;
$line =~ s/\@attribute Is/\@attribute Is numeric/;
$line =~ s/\@attribute Iv/\@attribute Iv numeric/;
$line =~ s/\@attribute Mt/\@attribute Mt numeric/;
$line =~ s/\@attribute Ni/\@attribute Ni numeric/;
$line =~ s/\@attribute Nx/\@attribute Nx numeric/;
$line =~ s/\@attribute Ny/\@attribute Ny numeric/;
$line =~ s/\@attribute Pf/\@attribute Pf numeric/;
$line =~ s/\@attribute Pi/\@attribute Pi numeric/;
$line =~ s/\@attribute t/\@attribute t numeric/;
$line =~ s/\@attribute It/\@attribute It {cbr, AODV, AODV_MITM}/;


#attributes not needed after heuristics
$line =~ s/\@attribute If/\@attribute If real/;
$line =~ s/\@attribute Ma/\@attribute Ma string/;
$line =~ s/\@attribute Md/\@attribute Md string/;
$line =~ s/\@attribute Ms/\@attribute Ms string/;
$line =~ s/\@attribute Ne/\@attribute Ne real/;
$line =~ s/\@attribute Nl/\@attribute Nl string/;
$line =~ s/\@attribute Nw/\@attribute Nw string/;
$line =~ s/\@attribute Nz/\@attribute Nz real/;
$line =~ s/\@attribute Pn/\@attribute Pn string/;
$line =~ s/\@attribute Po/\@attribute Po string/;



$F_OUT->print($line);
                }


$F_IN->close();
$F_OUT->close();
```

University of Ghana http://ugspace.ug.edu.gh

```
                    if ($heads eq "-NULL"){next();}#getting rid of the initial placed NULL padder to ensure the correct working of the script
                    else{
                    $heads =~ s/-/\@attribute /;
                    $F_OUT->print($heads,"\n");
            }

#$F_OUT->print("\n\nMAX:","$maxArraySize");#PNB: for printing out the maximum number of possible columns or fields

#$F_IN->close();
#$F_OUT->close();
}

constract_file_body();
$F_IN->close();
$F_OUT->close();

            ##NOW TO PRINT OUT THE DATA



##SUBROUTINES
sub header_processor { #for creating all possible columns
chomp $_[0];
my (@newArray) = split(':',$_[0]);
#           $F_OUT->print("WORKED:",@newArray,"\n");
my $bool_present=0;
            my $incomingArraySize = scalar @newArray;
            #$F_OUT->print("ARRAY SIZE:",$incomingArraySize,"\n");
            my @finalColumns=();
if(@newArray){
foreach my $existingField(@columnHeaders) {

                    foreach my $comparisonField(@newArray){
                                if(@columnHeaders && ($existingField eq $comparisonField)){
```

University of Ghana http://ugspace.ug.edu.gh

```
            $bool_present=1;


                                    }if($bool_present==0){

            push(@columnHeaders,$comparisonField);

                                                @features = @newArray;
                                                #$bool_present=0;
                                                if(scalar
@columnHeaders >=           scalar @newArray){#VIP: very critical
for generating the new headers

                                                @finalColumns =    sort
@newArray;
                                                }
                                    }
                        }


}push(@newArray,shift  @columnHeaders);#building  the  column
headers


}else {
            ;
            #$F_OUT->print("empty array ");
            }
return @features;
}




sub constract_file_body {
            $F_IN                   =                   FileHandle-
>new("resultsWithoutMovement.txt");
            $F_OUT   =   FileHandle->new(">>filteredResults.arff");
#this second open appends to the just written information entered
into the file
            my %hashFromArray = ();
            $F_OUT->print("\n\@data");
            while (my $line = $F_IN->getline) {


            #for getting the highest number of headers
```

University of Ghana http://ugspace.ug.edu.gh

```
        chomp $line;
my @array = split(' ',$line);
$packetMode = shift @array;
my @nullpadder = ('-NULL','NULL');#padding the array with NULLS if it would result is an odd hash below
if(((scalar @array)/2)%2 != 0) { push(@array,@nullpadder) ;}
%hashFromArray = @array;
my @newArray = keys %hashFromArray;
my $newArray = join(":",@newArray);#my (@newHeaderList) = @{header_processor($newArray)};


#for getting the fields with values
my @keysArray;
my @valuesArray;
#$F_OUT->print("\nPacketMode:","$packetMode\n") if defined($packetMode);
#$F_OUT->print("\n");
$F_OUT->print("\n$packetMode,") if defined($packetMode);#for determinining if the packet is a received or sent packet
while ((my $key,my $value) = each(%hashFromArray)){
        push @keysArray,$key;
        push @valuesArray,$value;
#$F_OUT->print("$key:","$value\n");
}


##HAVING GOTTEN THE KEYS AND VALUES, ITERATE THROUGH TO FIND ABSCENT KEYS AND THEIR VALUES EXPECTED TO BE IN THE STANDARD FEATURES AND PAD
#$F_OUT->print(%hashFromArray,"n");
foreach my $feature(@features){
                while ((my $key,my $value) = each(%hashFromArray)){
                        if(!exists $hashFromArray{$feature}){

        $hashFromArray{$feature}="?";
                                }


}
                }


##PRINTING VALUES FROM THE NEW HASH TO FILL
```

University of Ghana http://ugspace.ug.edu.gh

```
THE FILE BODY
my $keycounter = 0;


#$F_OUT->print(%hashFromArray,"\n\n");
#foreach my $key1 (sort keys %hashFromArray){
                        #$keycounter++;
                        #if($key1 eq "-NULL"){next();}#getting rid of the -NULL padding field/column added on line 27 to make sure hashes are always even
                        #else{
                        #if((scalar keys %hashFromArray) == $keycounter){$F_OUT->print($hashFromArray{$key1});}
                        #{$F_OUT->print($hashFromArray{$key1},",");}
                        #}
            #}
            #$F_OUT->print(keys %hashFromArray);$F_OUT->print("|",@features);
            foreach my $key1 (sort @features){
                        $keycounter++;
                        if($key1 eq "-NULL"){next();}#getting rid of the -NULL padding field/column added on line 27 to make sure hashes are always even
                        else{
                        if((scalar @features) == $keycounter){$F_OUT->print($hashFromArray{$key1});}
                        else{$F_OUT->print($hashFromArray{$key1},",");}
                        }
            }
}
            $F_OUT->print("\n");
            $F_IN->close();
            $F_OUT->close();
            }



#$F_IN->close();
#$F_OUT->close();
```

University of Ghana http://ugspace.ug.edu.gh

Table B-3 - Perl scripts for reconfiguring setdest and cbrgen generated files for new secure files

***"setdestMutator.pl"* file**	***"cbrMutator.pl"* file**
#!/usr/bin/Perl #Purpose:For editing the setdest file generated to eliminated attacking nodes for reconfiguration use warnings; use strict; use FileHandle; #GLOBAL VARIABLES my @blacklist = ('none',6.0); my $F_IN = FileHandle->new("MUTABLE_CODE/scen1forAODV-n_mutable_reduced-t500-x750-y750"); my $F_OUT = FileHandle->new(">MUTABLE_CODE/finalScene");#this first open cleans the file and writes afresh to it my $nodeId='first'; my $line=' '; while ($line = $F_IN->getline) { foreach $nodeId(@blacklist){ my $badNodeReplacement = '$node_('.$nodeId.'_00)'; if($line =~ m|\$node_\($nodeId\)|i){#if the line contains the node in the blacklist $line =~ s\|^\|# \|; #comment that line out #$line =~ s|\$node_\($nodeId\)\|$badNodeReplacement|; #rename the culprit nodes to dummy nodes to mimic ip address takedowns } }$F_OUT->print($line); } $F_IN->close();	#!/usr/bin/Perl #Purpose:For editing the setdest file generated to eliminated attacking nodes for reconfiguration use warnings; use strict; use FileHandle; #GLOBAL VARIABLES my @blacklist = ('none',6.0); my $F_IN = FileHandle->new("MUTABLE_CODE/cbr_reduced_nodes_mutable"); my $F_OUT = FileHandle->new(">MUTABLE_CODE/finalCbr");#this first open cleans the file and writes afresh to it my $nodeId='first'; my $line=' '; while ($line = $F_IN->getline) { foreach $nodeId(@blacklist){ my $badNodeReplacement = '$node_('.$nodeId.'00)'; if($line =~ m|\$node_\($nodeId\)|i){#if the line contains the node in the blacklist $line =~ s\|^\|# \|; #comment that line out #$line =~ s|\$node_\($nodeId\)\|$badNodeReplacement|; #rename the culprit nodes to dummy nodes to mimic ip address takedowns } }$F_OUT->print($line); } $F_IN->close(); $F_OUT->close();

University of Ghana http://ugspace.ug.edu.gh

$F_OUT->close();	

Shell scripts (bash shell)

Table B-4 - Shell scripts for putting everything together

***"complete.sh"* file**	***"motherCleaner.sh"* file**
#!/bin/bash #Author: Kwadwo Boateng Ofori-Amanfo #Purpose: For bringing all the different parts of the final project together. a) the attack b) the detection and filter c) the reconfiguration #a) ns mitm_attack_reduced.tcl source motherCleaner.sh dataCleanerBetter.pl #running the ANN script and reconfiguring #b) cd ../MitmProtectorWithWeka/dist java -jar MitmProtectorWithWeka.jar #c) cd ../../mitmaodv (Perl setdestMutator.pl)&&(Perl cbrMutator.pl) cd MUTABLE_CODE ns mutableScriptTemplate.tcl	#!/bin/bash #Author: Kwadwo Boateng Ofori-Amanfo #Purpose: To clean the results and prepare them for weka #POSSIBLE REPLACEMENTS FOR LINE 12; THUS LINE 12 IS MOVE GENERIC WITH results.txt, sim1forNormalNetwork_reduced.tr, sim1forNormalNetwork.tr all as possible values #cat results.txt | grep -vP '^M' > resultsWithoutMovement.txt ##TEST #cat sim1forNormalNetwork_reduced.tr | grep -vP '^M' > resultsWithoutMovement.txt ##for filtering reduced node number simulation results #cat sim1forNormalNetwork.tr | grep -vP '^M' > resultsWithoutMovement.txt ##for filtering scaled up node number simulation results #GENERIC cat $1 | grep -vP '^M' > resultsWithoutMovement.txt #for filtering the trace file generated to remove movement/positioning data Perl dataCleanerBetter.pl Perl wekaPreprocessorEquivalent.pl #cat filteredResults.arff | head -$2 > reducedFilteredResults.arff #for picking the maximum number of lines to take out cat filteredResults1.arff | head -30001 > reducedFilteredResults.arff #for picking the 30001 as maximum number of lines to take out

Java codes for MitmProtectorWithWeka

Table B-5 - Implementation of feature selection and ANN using weka API

University of Ghana http://ugspace.ug.edu.gh

***"MitmProtectorWithWeka.java"* file**	***"PossibleFunctions.java"* file**
package mitmprotectorwithweka; import java.io.BufferedReader; import java.io.FileReader; import java.util.Random; import java.io.File; import java.io.IOException; import weka.classifiers.Evaluation; import weka.classifiers.functions.MultilayerPerceptron; import weka.core.Instances; /** * * @author daniel */ public class MitmProtectorWithWeka { public static void main(String[] args) { try { PossibleFunctions pF = new PossibleFunctions(); //String inputFile = new File("../mitmaodv/reducedFilteredResults.arff").getCanonicalPath(); String inputFile = new File("../mitmaodv/FINAL_DATA/POSSIBLE_FINAL_DATA/reducedFilteredResultsPreprocessedInWekaFromScaleDownAttackWithLatencyTrainingDataWithPacketMode.arff").getCanonicalPath(); //String inputFile = new File("../mitmaodv/FINAL_DATA/furtherReducedtrainDataForJava.arff").getCanonicalPath(); String outputFile = new File("../mitmaodv/FINAL_DATA/furtherReducedtestDataForJava.arff").getCanonicalPath(); System.out.println("Input: " + inputFile); System.out.println("Output: " + outputFile); BufferedReader trainer = new BufferedReader(new FileReader(inputFile)); BufferedReader tester = new BufferedReader(new	package mitmprotectorwithweka; import java.io.*; import java.util.*; import java.util.regex.*; import weka.classifiers.Evaluation; import weka.classifiers.functions.MultilayerPerceptron; import weka.core.Instances; import weka.core.Utils; import weka.filters.Filter; import weka.filters.unsupervised.attribute.Reorder; /** * * @author daniel */ public class PossibleFunctions { static String blacklistWithBrackets; //this function predicts the classification by looking at the probabilities assigned to each class and picking the maximum public void classOfInstanceByProbability(MultilayerPerceptron mP, Instances train, int instanceNo) throws Exception { String packet = ""; double[] prediction = mP.distributionForInstance(train.get(instanceNo)); int maxValueIndex = 0; for (int i = 0; i < prediction.length; i = i + 1) { //System.out.println("Probability of class " + train.classAttribute().value(i)+":"+Double.toString(prediction[i])); if ((prediction[i] > prediction[maxValueIndex])) { maxValueIndex = i; packet = train.classAttribute().value(i); } } System.out.print("\nclass " + train.classAttribute().value(maxValueIndex) + ":" + packet); } public void printSummary(MultilayerPerceptron base, Evaluation eval,

University of Ghana http://ugspace.ug.edu.gh

```
FileReader(outputFile));


Instances train = new Instances(trainer);//training data
//train = pF.removeColumnsData(train);//remove unwanted colums from training set and reorder remaining columns


Instances test = new Instances(tester);//testing data


train.setClassIndex(train.numAttributes() - 1);
test.setClassIndex(train.numAttributes() - 1);


trainer.close();
tester.close();


MultilayerPerceptron mP = new MultilayerPerceptron();
mP.buildClassifier(train);
Evaluation eval = new Evaluation(train);
eval.crossValidateModel(mP, train, 10, new Random(1));


pF.printSummary(mP, null, test);
//pF.classOfInstanceByProbability(mP, train,1);


} catch (Exception e) {
e.getMessage();
}
}


}
```

```
Instances data) throws Exception {
HashMap<String, Integer> counters = new HashMap<String, Integer>();


//output evaluation
System.out.println();
System.out.println("=== Setup ===");
System.out.println("Classifier: " + base.getClass().getName() + " " + Utils.joinOptions(base.getOptions()));
System.out.println("Dataset: " + data.relationName());


int counterInitializer = 1;
String blacklist = "{'none',"; //initialize this to none. It would be the only one in the array to be printed to the reconfiguration files if no other node is blacklisted
//output predictions
for (int i = 0; i < data.numInstances(); i++) {
//NB: OBSERVATIONS MADE FROM WEKA API
//System.out.print(" i0_4 : "+ data.instance(i).stringValue(data.attribute("t")));//did not work for numeric value
//System.out.print(" i0_2 : "+ data.instance(i).value(data.attribute("t")));//but this works for numeric value
//System.out.print(" i1 : "+ data.instance(i).classAttribute());//for the classification attributes
//System.out.print(" i2 : "+ data.instance(i).numClasses());//for the number of classes


String pred = String.valueOf(base.classifyInstance(data.instance(i)));
double[] dist = base.distributionForInstance(data.instance(i));
double classifyInstance = base.classifyInstance(data.instance(i));


System.out.print((i + 1));


System.out.print("\tactual: " + data.classAttribute().value((int) data.instance(i).classValue()));
System.out.println(" \tpredicted :" + data.classAttribute().value((int) classifyInstance));


//for storing up how many times a particular node is sending out aodv_mitm packets
if (!(data.instance(i).stringValue(data.attribute("It")).equals("?")) &&//first condition for removing unknown instances with unknown
```

University of Ghana http://ugspace.ug.edu.gh

```
classes

(data.instance(i).stringValue(data.attribute("PM")).equals("s"))

&& (data.classAttribute().value((int) classifyInstance).equalsIgnoreCase("aodv_mitm"))) {

if (counters.containsKey(Double.toString(data.instance(i).value(data.attribute("Hs"))))) {

counters.put(Double.toString(data.instance(i).value(data.attribute("Hs"))),

counters.get(Double.toString(data.instance(i).value(data.attribute("Hs")))) + 1);

} else {

counters.put(Double.toString(data.instance(i).value(data.attribute("Hs"))), counterInitializer);

}

}


/* System.out.print(" i4 : "+ data.instance(i).classValue());//for the value of the classes

System.out.print(" - ");

//System.out.print(Utils.arrayToString(dist));

System.out.print(" - ");

System.out.print(" i5 :"+Utils.maxIndex(dist));

System.out.print(" - ");

System.out.print(" i6 :"+classifyInstance);

System.out.println();*/

}


//for printing how many times a particular node is sent out packets

Iterator<String> keySelector = counters.keySet().iterator();

while (keySelector.hasNext()) {

String key = keySelector.next();

blacklist += key + ",";

//System.out.println("\n\nkey: "+key+" value: "+counters.get(key));

}

blacklist += "}";//classing the list


//formatting 'blacklist' by removing the last comma before the closing brace and then writing to the file

String stringContainingPattern = blacklist;

blacklist = replaceWithRegex(",}","}",stringContainingPattern);//the last comma would be removed
```

University of Ghana http://ugspace.ug.edu.gh

```
String blackListWithSpaceDelimiters = replaceWithRegex(",","
",blacklist);

System.out.println("blacklist:" + blackListWithSpaceDelimiters);

//printing the blacklist to it's possition in the file

String inputFile = new
File("../mitmaodv/MUTABLE_CODE/mutableScriptTemplate.tcl").
getCanonicalPath();

String outputFile = new
File("../mitmaodv/MUTABLE_CODE/mutableScriptTemplate_tem
p.tcl").getCanonicalPath();

String patternToReplaceInScript = "set.blacklist.\\{.*\\};";

String replacement1 = "set blacklist " +
blackListWithSpaceDelimiters + ";";

editFileWithRegex(patternToReplaceInScript,replacement1,inputFile,
outputFile);


//printing the blacklist to itsPerl script

inputFile = new
File("../mitmaodv/setdestMutator.pl").getCanonicalPath();//original
input; currently present - on disk file

outputFile = new
File("../mitmaodv/setdestMutator_temp.pl").getCanonicalPath();//fo
r creating a temporary output file that is finally rename to the original
input file to replace the original file



//changing the braces to brackets for Perl


patternToReplaceInScript = "my.@blacklist.=.\\(.*\\);";

blacklistWithBrackets = replaceWithRegex("\\{","(",blacklist);

blacklistWithBrackets =
replaceWithRegex("\\}",")",blacklistWithBrackets);

//patternToReplaceInScript = "blacklist";

System.out.println(blacklistWithBrackets);

replacement1 = "my @blacklist = " + blacklistWithBrackets + ";";

editFileWithRegex(patternToReplaceInScript,replacement1,inputFile,
outputFile);



//writing the blacklist to the cbrgen file also

patternToReplaceInScript = "my.@blacklist.=.\\(.*\\);";

inputFile = new
File("../mitmaodv/cbrMutator.pl").getCanonicalPath();//original
```

University of Ghana http://ugspace.ug.edu.gh

```
input; currently present - on disk file

outputFile = new File("../mitmaodv/cbrMutator_temp.pl").getCanonicalPath();//for creating a temporary output file that is finally rename to the original input file to replace the original file

editFileWithRegex(patternToReplaceInScript,replacement1,inputFile, outputFile);



}


public String replaceWithRegex(String patternToReplace,String replacement,String stringContainingPattern){

Pattern p = Pattern.compile(patternToReplace);

Matcher m = p.matcher(stringContainingPattern);

return m.replaceAll(replacement);//the last comma would be re



}


public void editFileWithRegex(String patternToReplaceInScript, String replacement1, String editableFile, String tmpFile) throws Exception {

String _newstring,_matchingString;

String _patternToReplaceInScript = patternToReplaceInScript;

String _replacement1 = replacement1;

String _editableFile = editableFile;

String _tmpFile = tmpFile;

PrintWriter writer = new PrintWriter(new BufferedWriter(new FileWriter(_tmpFile)));

BufferedReader trainer = new BufferedReader(new FileReader(_editableFile));


try {


Pattern p1 = Pattern.compile(_patternToReplaceInScript);

while ((_matchingString = trainer.readLine()) != null) {

Matcher m1 = p1.matcher(_matchingString);

_newstring = m1.replaceAll(_replacement1);

//System.out.println(_newstring);

writer.println(_newstring);

writer.flush();

}

} catch (IOException e) {
```

University of Ghana http://ugspace.ug.edu.gh

```
e.getMessage();
}
File realName = new File(_editableFile);
realName.delete();
new File(_tmpFile).renameTo(realName);


}



//for removing columns that are not wanted
public Instances removeColumnsData(Instances data){

//remove and reordering these columns
Instances newData = new Instances(data);
String[] reorderOptions = new String[2];
reorderOptions[0] = "-R";
reorderOptions[1] = "1,2,3,4,6,7,8,10,14,16,19,20,22,23,26,9";
Reorder reorderer = new Reorder();
try{
//reordering the remaining columns
reorderer.setOptions(reorderOptions);
reorderer.setInputFormat(data);
//Instances newData =  Filter.useFilter(data, remove);
newData = Filter.useFilter(data,reorderer);
//  for (int i = 0; i < newData.numInstances(); i++) {
//     System.out.println(" i2 : "+ newData.instance(i));//for the number of classes
//  }
}catch(Exception e){System.out.println(e.getMessage());}
return newData;
}
}
```

University of Ghana http://ugspace.ug.edu.gh

Results after simulation - unfiltered

University of Ghana http://ugspace.ug.edu.gh

s -t 7.668618722 -Hs 1 -Hd -2 -Ni 1 -Nx 189.48 -Ny 227.06 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id 2.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

r -t 7.668618722 -Hs 1 -Hd -2 -Ni 1 -Nx 189.48 -Ny 227.06 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id 2.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

s -t 7.668618722 -Hs 1 -Hd -2 -Ni 1 -Nx 189.48 -Ny 227.06 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

s -t 7.668733722 -Hs 1 -Hd -2 -Ni 1 -Nx 189.48 -Ny 227.06 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 106 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669581840 -Hs 0 -Hd -2 -Ni 0 -Nx 221.99 -Ny 212.84 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669581906 -Hs 18 -Hd -2 -Ni 18 -Nx 163.62 -Ny 178.21 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582022 -Hs 17 -Hd -2 -Ni 17 -Nx 237.11 -Ny 150.74 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582072 -Hs 4 -Hd -2 -Ni 4 -Nx 292.73 -Ny 208.32 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582206 -Hs 5 -Hd -2 -Ni 5 -Nx 83.89 -Ny 326.79 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582230 -Hs 11 -Hd -2 -Ni 11 -Nx 208.59 -Ny 378.33 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582320 -Hs 6 -Hd -2 -Ni 6 -Nx 98.88 -Ny 382.12 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669582324 -Hs 7 -Hd -2 -Ni 7 -Nx 297.15 -Ny 372.03 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669606840 -Hs 0 -Hd -2 -Ni 0 -Nx 221.99 -Ny 212.84 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669606906 -Hs 18 -Hd -2 -Ni 18 -Nx 163.62 -Ny 178.21 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669607022 -Hs 17 -Hd -2 -Ni 17 -Nx 237.11 -Ny 150.74 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669607072 -Hs 4 -Hd -2 -Ni 4 -Nx 292.73 -Ny 208.32 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

r -t 7.669607206 -Hs 5 -Hd -2 -Ni 5 -Nx 83.89 -Ny 326.79 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 800 -Is 1.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 2 -Pds 0 -Ps 1 -Pss 4 -Pc REQUEST

University of Ghana http://ugspace.ug.edu.gh

University of Ghana http://ugspace.ug.edu.gh

C. Appendix C - Results

sample unfiltered results after simulation

Table C-1- Sample of unfiltered results after simulation

Results after filtering with **"*dataCleanerBetter.pl*" - "*filteredResults.arff*"**
@relation attackProfile @attribute PM @attribute Hd @attribute Hs @attribute Id @attribute If @attribute Ii @attribute Il @attribute Is @attribute It @attribute Iv @attribute Ma @attribute Md @attribute Ms @attribute Mt @attribute Ne @attribute Ni @attribute Nl @attribute Nw @attribute Nx @attribute Ny @attribute Nz @attribute Pf @attribute Pi @attribute Pn @attribute Po @attribute t @data s,-2,18,19.0,0,0,512,18.0,cbr,32,0,0,0,0,-1.000000,18,AGT,---,585.67,415.30,0.00,0,0,cbr,1,69.230459700

University of Ghana http://ugspace.ug.edu.gh

r,-2,18,19.0,0,0,512,18.0,cbr,32,0,0,0,0,-1.000000,18,RTR,---,585.67,415.30,0.00,0,0,cbr,1,69.230459700

s,-2,18,-1.255,0,0,48,18.255,AODV,30,0,0,0,0,-1.000000,18,RTR,---,585.67,415.30,0.00,?,?,?,?,69.230459700

s,-2,18,-1.255,0,0,106,18.255,AODV,30,0,ffffffff,12,800,-1.000000,18,MAC,---
,585.67,415.30,0.00,?,?,?,?,69.230574700

r,-2,16,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,16,MAC,---
,567.07,435.45,0.00,?,?,?,?,69.231422791

r,-2,13,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,13,MAC,---
,592.41,371.46,0.00,?,?,?,?,69.231422848

r,-2,9,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,9,MAC,---
,662.79,387.89,0.00,?,?,?,?,69.231422973

r,-2,6,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,6,MAC,---
,604.00,300.23,0.00,?,?,?,?,69.231423088

r,-2,5,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,5,MAC,---
,468.26,487.11,0.00,?,?,?,?,69.231423159

r,-2,17,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,17,MAC,---
,557.65,550.23,0.00,?,?,?,?,69.231423159

r,-2,8,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,8,MAC,---
,679.99,567.68,0.00,?,?,?,?,69.231423297

r,-2,12,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,12,MAC,---
,418.03,345.28,0.00,?,?,?,?,69.231423306

r,-2,4,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,4,MAC,---
,653.02,601.36,0.00,?,?,?,?,69.231423360

r,-2,19,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,19,MAC,---
,448.72,259.72,0.00,?,?,?,?,69.231423391

r,-2,1,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,1,MAC,---
,389.08,344.58,0.00,?,?,?,?,69.231423396

r,-2,15,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,15,MAC,---
,401.86,283.24,0.00,?,?,?,?,69.231423454

r,-2,16,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,16,RTR,---
,567.07,435.45,0.00,?,?,?,?,69.231447791

r,-2,13,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,13,RTR,---
,592.41,371.46,0.00,?,?,?,?,69.231447848

r,-2,9,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,9,RTR,---
,662.79,387.89,0.00,?,?,?,?,69.231447973

r,-2,6,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,6,RTR,---
,604.00,300.23,0.00,?,?,?,?,69.231448088

Table C-2 - re-filtered results using Perl scripts after simulation

Results after re-filtering with **"wekaPreprocessorEquivalent*.pl*"** - ***"filteredResults.arff"***
@relation 'attackProfile-weka.filters.unsupervised.attribute.Remove-R11-13,15,17-18,21,24-25-weka.filters.unsupervised.attribute.Remove-R5-weka.filters.unsupervised.attribute.Reorder-R1,2,3,4,5,6,7,9,10,11,12,13,14,15,16,8'

University of Ghana http://ugspace.ug.edu.gh

@attribute PM {r,s,d,f}

@attribute Hd numeric

@attribute Hs numeric

@attribute Id numeric

@attribute If real

@attribute Ii numeric

@attribute Il numeric

@attribute Is numeric

@attribute It {cbr, AODV, AODV_MITM}

@attribute Iv numeric

@attribute Ma string

@attribute Md string

@attribute Ms string

@attribute Mt numeric

@attribute Ne real

@attribute Ni numeric

@attribute Nl string

@attribute Nw string

@attribute Nx numeric

@attribute Ny numeric

@attribute Nz real

@attribute Pf numeric

@attribute Pi numeric

@attribute Pn string

@attribute Po string

@attribute t numeric

@data

s,-2,18,19.0,0,0,512,18.0,cbr,32,0,0,0,0,-1.000000,18,AGT,---,585.67,415.30,0.00,0,0,cbr,1,69.230459700

r,-2,18,19.0,0,0,512,18.0,cbr,32,0,0,0,0,-1.000000,18,RTR,---,585.67,415.30,0.00,0,0,cbr,1,69.230459700

s,-2,18,-1.255,0,0,48,18.255,AODV,30,0,0,0,0,-1.000000,18,RTR,---,585.67,415.30,0.00,?,?,?,?,69.230459700

s,-2,18,-1.255,0,0,106,18.255,AODV,30,0,ffffffff,12,800,-1.000000,18,MAC,---
,585.67,415.30,0.00,?,?,?,?,69.230574700

r,-2,16,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,16,MAC,---
,567.07,435.45,0.00,?,?,?,?,69.231422791

r,-2,13,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,13,MAC,---
,592.41,371.46,0.00,?,?,?,?,69.231422848

r,-2,9,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,9,MAC,---
,662.79,387.89,0.00,?,?,?,?,69.231422973

University of Ghana http://ugspace.ug.edu.gh

r,-2,6,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,6,MAC,---
,604.00,300.23,0.00,?,?,?,?,69.231423088

r,-2,5,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,5,MAC,---
,468.26,487.11,0.00,?,?,?,?,69.231423159

r,-2,17,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,17,MAC,---
,557.65,550.23,0.00,?,?,?,?,69.231423159

r,-2,8,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,8,MAC,---
,679.99,567.68,0.00,?,?,?,?,69.231423297

r,-2,12,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,12,MAC,---
,418.03,345.28,0.00,?,?,?,?,69.231423306

r,-2,4,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,4,MAC,---
,653.02,601.36,0.00,?,?,?,?,69.231423360

r,-2,19,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,19,MAC,---
,448.72,259.72,0.00,?,?,?,?,69.231423391

r,-2,1,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,1,MAC,---
,389.08,344.58,0.00,?,?,?,?,69.231423396

r,-2,15,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,15,MAC,---
,401.86,283.24,0.00,?,?,?,?,69.231423454

r,-2,16,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,16,RTR,---
,567.07,435.45,0.00,?,?,?,?,69.231447791

r,-2,13,-1.255,0,0,48,18.255,AODV,30,0,ffffffff,12,800,-1.000000,13,RTR,---
,592.41,371.46,0.00,?,?,?,?,69.231447848

File worked on by ANN solution by Java

Table C-3- results after third level of filtering with "MitmProtectorWithWeka.jar"

Results after re-filtering with **"MitmProtectorWithWeka.jar"** - ***"filteredResults.arff"***
@relation 'attackProfile-weka.filters.unsupervised.attribute.Remove-R11-13,15,17-18,21,24-25-weka.filters.unsupervised.attribute.Remove-R5-weka.filters.unsupervised.attribute.Reorder-R1,2,3,4,5,6,7,9,10,11,12,13,14,15,16,8'
@attribute PM {r,s,d,f}
@attribute Hd numeric
@attribute Hs numeric
@attribute Id numeric
@attribute Ii numeric
@attribute Il numeric
@attribute Is numeric
@attribute Iv numeric
@attribute Mt numeric
@attribute Ni numeric

University of Ghana http://ugspace.ug.edu.gh

```
@attribute Nx numeric
@attribute Ny numeric
@attribute Pf numeric
@attribute Pi numeric
@attribute t numeric
@attribute It {cbr,AODV,AODV_MITM}

@data
s,-2,5,6.1,0,512,5.1,32,0,5,357.76,455.11,0,0,84.456631,cbr
r,-2,5,6.1,0,512,5.1,32,0,5,357.76,455.11,0,0,84.456631,cbr
s,-2,5,-1.255,0,48,5.255,30,0,5,357.76,455.11,?,?,84.456631,AODV
s,-2,5,-1.255,0,106,5.255,30,800,5,357.76,455.11,?,?,84.456746,AODV
r,-2,1,-1.255,0,48,5.255,30,800,1,245.92,328.36,?,?,84.457594,AODV
r,-2,4,-1.255,0,48,5.255,30,800,4,546.82,437.86,?,?,84.457594,AODV
r,-2,1,-1.255,0,48,5.255,30,800,1,245.92,328.36,?,?,84.457619,AODV
r,-2,4,-1.255,0,48,5.255,30,800,4,546.82,437.86,?,?,84.457619,AODV
s,-2,1,-1.255,0,48,1.255,29,800,1,245.91,328.37,?,?,84.4585,AODV
s,-2,1,-1.255,0,106,1.255,29,800,1,245.91,328.37,?,?,84.458575,AODV
r,-2,2,-1.255,0,48,1.255,29,800,2,103.17,345.7,?,?,84.459424,AODV
r,-2,5,-1.255,0,48,1.255,29,800,5,357.75,455.11,?,?,84.459424,AODV
r,-2,2,-1.255,0,48,1.255,29,800,2,103.17,345.7,?,?,84.459449,AODV
r,-2,5,-1.255,0,48,1.255,29,800,5,357.75,455.11,?,?,84.459449,AODV
s,-2,4,-1.255,0,48,4.255,29,800,4,546.81,437.82,?,?,84.462031,AODV
```

University of Ghana http://ugspace.ug.edu.gh

D. Appendix D- Other diagrams

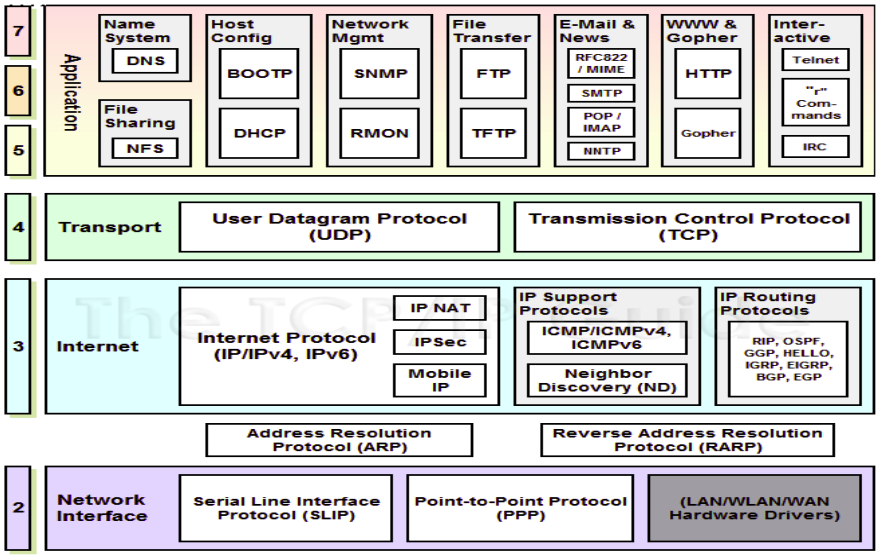


Figure D.1 - TCP-IP protocol suite

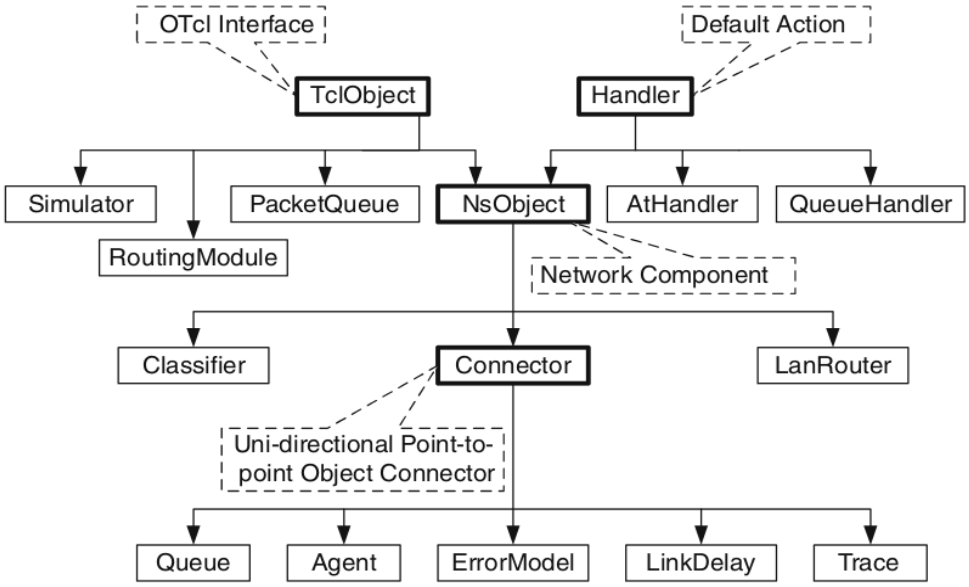


Figure D.2 - NS2 class structure