Investigating Cross-Platform Robustness for Machine Learning Based IDSs on 802.11 Networks

Adetokunbo Makanju, A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
B3H 1W5
Canada
makanju@cs.dal.ca,zincir@cs.dal.ca

***Abstract*— Security and Intrusion detection in 802.11 networks is currently an active area of research where WiFi specific Data Link layer attacks are an area of focus. While these attacks are very simple in implementation, their effect on WiFi networks can be devastating. Recent research has focused on producing machine learning based IDSs for these attacks. Such IDSs have shown promise. Our work investigates the Cross-Platform robustness of such machine learning based solutions. By cross-platform robustness we mean the ability to train a solution on one network and run it seamlessly on another. We demonstrate that machine learning based IDSs could potentially suffer when employed across different platforms. In order to solve this, we propose a MAC address mapping technique which can achieve a Cross-Platform detection rate, for machine learning based IDSs, on average of 100% and a false positive rate on average of 0.1%.**

***Index Terms*— Intrusion Detection, Wireless Networks, Machine Learning, Robustness.**

I. Introduction

802.11 networks, Wireless Fidelity (WiFi) networks, wireless ethernet, depending on the literature all refer to same thing. They are all networks based on the IEEE 802.11 protocol. The security vulnerabilities of networks based on the IEEE 802.11 wireless network standard have been widely attested in literature [1] and with thier growing deployment in more locations, this ought to be of great concern to everyone. IEEE 802.11 is by far the most widely used wireless networking standard in the world today and its popularity increases by the day. The fact that 802.11 networks (and other wireless communication protocols) transmit information through open air waves is majorly responsible for their seeming openness to intrusions.

The Data Link layer is the second layer of the Open System Interconnect (OSI) protocol stack, it sets above the physical layer. Therefore as the name implies this layer is the target of Data Link layer attacks. Apart from being designed specifically for WiFi networks, most of the security features incorporated into the WiFi protocol such as data encryption and client authentication are not able to guard against these attacks. 802.11 Denial-of-Service (DoS) attacks are a subset of these attacks and they are the focus of work. While 802.11 data link layer DoS attacks are myriad, our work focuses specifically on the de-authentication attack. The de-authentication attack causes a Denial-of-Service (DoS) by injecting a subset of the IEEE 802.11 Management Frames i.e. the de-authentication frame into the network traffic.

Recent research has proposed machine learning based solutions for data link layer attacks, [2] Genetic Programming (GP) based solutions while [3] proposed a solution based on neural networks. In this paper, we investigate the Cross-Platform robustness of machine learning based IDSs for wireless data link layer attacks. Signature based IDSs e.g. Snort-Wireless and Kismet, that can be used for detecting data link layer attacks are Cross-Platform robust i.e. they can be used between networks with little or no change to their signatures. Machine learning based solutions will have to be Cross-Platform robust if they are to be seriously considered as an alternative to conventional signature based systems considering that recent work has showed that Genetic Programming (GP) based solutions can detect such attacks in situations where conventional systems cannot [4].

Our work demonstrates that machine learning based solutions are indeed susceptible to diminished performance when used across different platforms. To this end, we propose a possible solution to this problem. By focusing on the training feature set and the way this feature set is processed for presentation to the learning algorithms, we are able to discover that a significant degree of Cross-Platform robustness can be achieved by focusing on the Media Access Control (MAC) address mapping technique used. This should not be surprising as the 802.11 MAC is integral to of the 802.11 specification, indeed aside form the 802.11 MAC layer and the 802.11 physical layer, the 802.11 protocol does not differ much from other members of IEEE 802 protocol family.

The remainder of this paper is organised as follows. Section 2 discusses WiFi networks and data link layer attacks. Section 3 discusses the methods of detecting data link layer attacks investigated. Section 4 outlines the problem, proposed solution, experiments and explains our approach. Section 5 presents the results and conclusions are given in Section 6.

II. Data Link Layer Attacks and WiFi Networks

A. WiFi Networks

WiFi networks generally consist of one or more Access Points (APs) and a number of clients, which can be any device from laptop computers to wireless Personal Digital Assistants (PDAs), which communicate over a wireless medium using the IEEE 802.11 standard. Network technologies based on the IEEE 802.11 standard include 802.11b, 802.11g and others. These technologies differ from each other, amongst other things, by the frequency at which they operate and the bandwidth that they are able to deliver. In this paper, we deal specifically with 802.11b networks [5].

WiFi APs act as base stations or servers for wireless Local Area Networks (WLANs). Using Beacon Frames, they periodically broadcast their Service Set Identifier (SSID), a character string, which identifies the AP. This way, any authorised client machine that is within the range of the AP and that can pick up the SSID signal can choose to join the network of the AP.

WiFi networks have many advantages, one of which is their ease of deployment. This has made WiFi technology one of the fastest growing wireless technologies to reach its consumers[6].

However, security is of great concern in WiFi networks. WiFi networks are particularly susceptible to attacks, which their wired counterparts are not susceptible. In particular, transmitting data over open airwaves is responsible for this. Data transmitted in this fashion can easily be intercepted. Indeed, research suggests that security is the major inhibitor to the future growth of the wireless network market. Several protocols, which use authentication and cryptographic techniques like Wireless Encryption Protocol (WEP), WiFi Protected Access (WPA) and wireless Virtual Private Networks (VPN) have been proposed to ameliorate these vulnerabilities. These protocols, however, do not deal with attacks that target the physical and data link layers of the OSI protocol stack. Most of these attacks are DoS attacks, which usually exploit MAC frames, and their end effect results in the network being unusable or inaccessible to legitimate clients.

B. Data Link Layer Management Frames

The 802.11 standard defines various frame types that stations (clients and access points) use for communication, as well as for management and control of their connections [5]. This gives rise to three broad classes of frames i.e. management frames, control frames and data frames. Management frames are used by stations to establish and maintain connections. This makes them the target of most attacks, which aim to make a WiFi network unusable. Types of management frames include: Association, Disassociation, Authentication, De-authentication, Beacon and Probe frames. Full discussion on the uses of these frames is beyond the scope of this paper, we however briefly discuss the Association, Disassociation, Authentication and De-authentication frame subtypes below.

- **Authentication frame:** This frame is used by clients to enable an AP to identify them as legitimate stations on a WiFi network. The client sends an authentication request and the AP replies with an authentication response, which either accepts or rejects the identity of the client.
- **De-authentication frame:** A station sends a de-authentication frame to another station if it wishes to terminate secure communications. The station can either be the client or the AP.
- **Association request frame:** This frame is used by clients to associate with an AP. When a client is associated with an AP, the AP allocates resources for and synchronizes with the client. Association frames can either be requests (from the client to the AP) or responses (from the AP to client).
- **Disassociation frame:** This is sent when a station wishes to terminate an association between itself and another station. The station can either be the client or the access point.

C. De-authentication Attack

As mentioned earlier, this paper focuses on the De-authentication attack. This attack, like other MAC layer attacks is very easy to implement. An attacker simply eavesdrops on a network and gathers information about the stations on the network. The attacker then uses this information to spoof the MAC address of a station or AP on the network. If the attacker targets a specific client, it creates a de-authentication frame with the MAC address of the target as the destination and the MAC address of the AP as the source. This frame causes the client to loose its connection to the AP; this prevents the target from communicating any further as a legitimate client on the network. This scenario is outlined in Fig. 1.

Apart from the scenario outlined above the attacker can vary the scope of the attack i.e. focusing on the AP to take down the entire network, targeting a specific client or group of clients, as well.

D. Void11

Void11 is a free software implementation of some common 802.11b attacks [7]. The basic implementation works in a

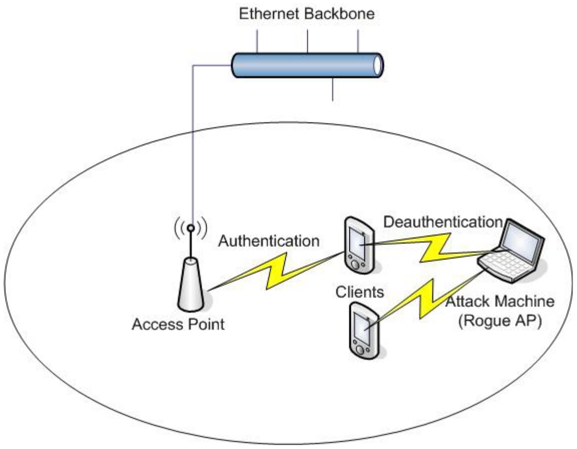


Fig. 1. De-authentication Attack

command line Linux/Unix environment (though it has a GUI implementation called gvoid11, too). For void11 to work on a computer, the computer must have a prism based wireless Network Interface Card (NIC) and must have hostap drivers installed. The hostap drivers allow the machine to act as a wireless AP [8].

Void11 implements three data link layer attacks, which use management frames. They are De-authenticate Flood (default mode), Authentication Flood and the Association Flood[1]. The basic goal of each of the attacks is to flood the network with management frames causing random clients to loose their connection with the AP or keep the AP busy dealing with client requests which slows down the network. The end result of each of the attack types differs based on the rate of injection of the frames and on the type of client involved.

All the de-authentication attacks which were launched to create the datasets used in our experiments used the default values of command line arguments of void11.

III. Detecting Data Link Layer Attacks

Intrusion Detection Systems (IDSs) are used to detect attacks against the integrity, confidentiality and availability of computer networks [2], [9]. They are analogous to burglar alarms, which monitor the premises to find evidence of break-ins. These operations aim to catch attacks and log information about the incidents such as source and nature of the attack. An IDS can be a combination of software and hardware, which collects and analyzes data collected from a network(s) or a host(s). IDSs are generally analyzed from two aspects:

- **Deployment:** Whether to monitor incoming traffic or host information.
- **Detection:** Whether to employ the signatures of known attacks or to employ the models of normal behavior.

The use of machine learning and artificial intelligence techniques in the building of IDSs is relatively new. Hitherto, building IDSs required a human expert to construct a set of rules, which when triggered, would indicate malicious activity. In this section, we briefly discuss intrusion detection systems compared in this work i.e. Snort-Wireless and Machine Learning based IDSs.

A. Snort-Wireless Based Data Link Layer Attack Detection

There are several open source and commercial IDSs available in the market today but Snort stands out as being one of the most popular. Developed in 1998 by Martin Roesch, Snort is an open source, real-time intrusion detection system [10]. Using signature and anomaly based metrics it detects and prevents attacks by utilizing a rule-driven language. It is the most widely deployed open source IDS in industry and research.

With the appropriate patches applied, Snort can be transformed into Snort-Wireless [11]. These patches enable Snort (Snort-Wireless, after patches are applied) to detect WiFi specific attacks. Signatures that detect the de-authentication attack (and other WiFi MAC Layer attacks) are among the patches included in Snort-Wireless.

Snort-Wireless is employed in this work for two reasons:

- To put our proposed IDS system into context with the existing technolgy.
- To form a baseline on our datasets for comparison to other systems.

To achieve this we simply replayed the datasets in their raw tcpdump format. These datasets which were later processed to produce the datasets in Table III.

B. Machine Learning Based Systems

As stated earlier the use of machine learning and computational intelligence techniques in the building of IDSs is relatively new and so is the research into its use in detecting MAC layer attacks in 802.11 networks.

A significant amount of recent research has been focused on the use of machine learning solutions in the detection of 802.11 MAC layer attacks. Specifically Genetic Programming (GP) and Artificial Neural Networks (ANNs) have been used. We briefly discuss these works below.

GP is an extension of the Genetic Algorithm (GA); which is an evolutionary computation (EC) method proposed by John H. Holland [12]. GP extends the GA to the domain of evolving complete computer programs [13]. Using the Darwinian concepts of natural selection and fitness proportional breeding, populations of programs are genetically bred to solve problems. These populations of programs can either be represented as tree like LISP structures or as binary strings, which represent integers. These integers are then mapped onto an instruction set and a set of source and destination registers. Each individual can thus be decoded into a program, which takes the form of assembly language type code for a register machine. This is known as the Linear Page Based GP (L-GP) [14].

In [2] L-GP alongside the Random Subset Selection - Dynamic Subset Selection (RSS-DSS) algorithm [15],was successfully used to detect the de-authentication attack. Building on previous work in using GP based IDSs [9], [16], the work focused on developing an appropriate fitness function and feature set for use in detecting the de-authentication attack. The results of that work showed promise as the L-GP based solution was able to achieve a 100% detection rate.

While it is difficult to give a definition to an ANN, we can safely say that an ANN is a non-linear statistical data modeling tool which consists of artificial neurons which are modeled on biological neurons. These individual neurons are connected to each other in a hierarchical manner to form a neural network.

In [3] a ANN, specifically a Dynamically Growing Neural Network (DGNN) was used in the training of anomaly based wireless IDS. The work utilized the Improved Winner Takes It All (IWTA) algorithm to successfully select a feature set and train an anomaly based detector for wireless attacks.

It is clear from the above that machine learning based solutions show promise in wireless IDSs. This in part forms the motivation for our work. While our work follows in

[1] The command syntax for using the void11 tool to launch an attack is: ***void11penetration -D -t[type of attack] -d[delay] -s[station MAC] -B[BSSID] [interface]***

similar lines of [4], we also compare our system against Snort-Wireless and an ANN based wireless IDS. In order to employ an ANN based IDS, we used the ANN implementation from WEKA[17]. WEKA is a suite of Machine Learning and Data Mining algorithm implementations, which is developed at the University of Waikato by Ian H. Witten and Eibe Frank. The datasets used in the GP and ANN experiments were processed using the same feature set and techniques.

IV. Experiments

Our experiments require that we have appropriate datasets. These datasets have to be in tcpdump format for replaying through Snort-Wireless. Moreover, the tcpdump files need to be labeled for the training and testing of the GP and ANN based IDS. In order to generate such datasets, we had to setup two separate networks. The first network (Network I) is outlined in Table I, while the second network is outlined in Table II. Both networks were setup in the same manner, see Figure 2, all the clients are connected to the APs via 802.11 connections on channel 6. Attacks were generated on both networks using void11. The data was collected on the monitoring machine using Kismet Wireless [18].

The only difference between the two networks is the APs. In Network I an Airport based AP is employed, whereas in Network II a Cisco based AP is employed. In doing so our aim is to simulate to seperate networks. An AP is central to any infrastructure based wireless network, creating two networks with different APs simulates different network environments.

The de-authentication attack implemented is directed at the AP. From the attack machine, using void11, a stream of de-authentication frames with the source set to the MAC address of the AP and the target to that of the broadcast address **(ff:ff:ff:ff:ff:ff)** are intermittently released into the network stream. Normal traffic is also generated using our web crawling implementation, which is developed using the Java 2 Platform, Micro Edition (J2ME). The web crawler ensures a continuous stream of web browsing requests from the clients as the network data is logged.

TABLE I
Network Components I

Type	Description
Clients	Palm Tungsten C (5x)
	HP IPAQ 4700 (3x)
	Dell Inspiron 9300 Laptop
	Macintosh Mini
AP	Airport Base Station Extreme
Monitoring Machine	Intel Based Desktop
Attack Machine	Tablet PC

TABLE II
Network Components II

Type	Description
Clients	Palm Tungsten C (5x)
	HP IPAQ 4700 (3x)
	Dell Inspiron 9300 Laptop
	Macintosh Mini
AP	Cisco Aironet Base Station
Monitoring Machine	Intel Based Desktop
Attack Machine	Tablet PC

A. Feature Selection

The tcpdump traffic files collected by Kismet wireless could be automatically replayed through Snort-Wireless but needed further processing before they could be used for training and testing on the GP based and ANN based IDS. To this end, an appropriate feature set had to be selected from the features within the frames. 802.11 frames consist of several features but not all of them are related to this attack. Based on the feature selection in previous work [2], the following subset of features were selected for this purpose:

1) **Frame Control** - Defines the protocol version, type/subtype of the frame and any flags
2) **Destination Address** - MAC address of the destination of the frame
3) **Source Address** - MAC address of the source of the frame
4) **Basic Service Set Identifier (BSSID)**- Ethernet Address of the Access Point
5) **Fragment Number** - Defines the fragment number in a particular sequence of the frames
6) **Sequence Number** - Defines the sequence number of the frame
7) **Channel** - The transmission channel used for communication

B. Data Set Generation

A total of 20 different datasets are generated and employed in the following experiments. Table III details these datasets.

Datasets A1 - A10 were collected on Network I while datasets C1 - C10 were collected on Network II.

C. GP Based IDS Training Parameters

The parameter settings for the GP in all cases are given in Table IV. In addition to the GP parameters, the fitness function utilised in this work is the switching fitness function [2]. The switching fitness function assigns credit to a member

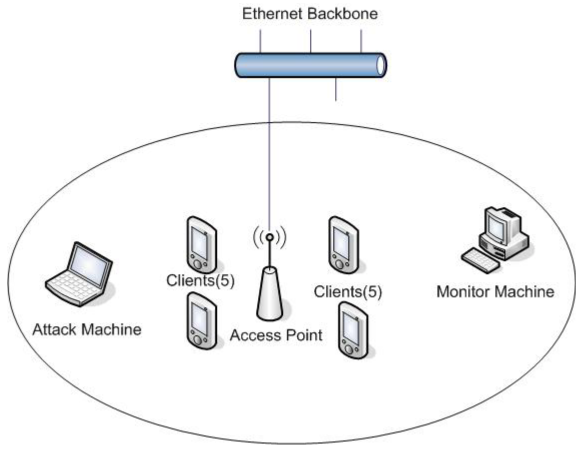


Fig. 2. Network Setup

TABLE III

DATASET CHARACTERISTICS

File	Size	Attack	% Attack	Network
A1	23960	3302	13.78	I
A2	20960	3440	16.41	I
A3	15880	3472	21.86	I
A4	15640	3258	20.83	I
A5	23160	3526	15.22	I
A6	17280	3505	20.28	I
A7	19120	3048	15.94	I
A8	18560	3835	20.66	I
A9	22080	3294	14.91	I
A10	21680	3253	15.00	I
C1	20160	3685	18.27	II
C2	24320	3669	15.08	II
C3	19040	3928	20.63	II
C4	27120	3890	14.34	II
C5	23880	3801	15.91	II
C6	24000	3466	14.44	II
C7	16040	4001	24.94	II
C8	23360	3297	14.11	II
C9	18760	3403	18.14	II
C10	17360	3086	17.77	II

TABLE IV

GP PARAMETERS

Parameter	Setting
Population Size	125
Maximum Number of Pages	32
Page Size	8 Instructions
Maximum Working Page Size	8 Instructions
Crossover Probability	0.9
Mutation Probability	0.5
Swap Probability	0.9
Tournament Size	4
Number of Registers	8
Function Set	(+,-,*,/)
Terminal Set	(0,...,255) ∪ ($r0,\ldots,r7$)
RSS Subset Size	5000
DSS Subset Size	50
RSS Iteration	1000
DSS Iteration	100

of the population depending on whether the execution of the individual on an exemplar produces a false positive (1) or a false negative (2). A higher credit value assignment at the end of the run indicates a poor performing individual.

$$Fitness \ += \frac{1}{TotalNumberofNormalConnections} \quad (1)$$

$$Fitness \ += \frac{1}{TotalNumberofAttackConnections} \quad (2)$$

D. ANN Based IDS Training Parameters

The parameter settings for the ANN in all cases are given in Table V. The neural networks used in the experiments used a very simple multilayer perceptron feed forward networks for building the classifiers. The multilayer perceptron had the input layer, one hidden layer and the output layer. The output layer had two outputs indicating whether the exemplar is an attack or otherwise. The network also uses a sigmoid function as its activation function and back propagation as its learning algorithm. These parameters are the default values for multilayer perceptrons in WEKA.

V. MAC ADDRESS MAPPING TECHNIQUES

This section explains the mapping schemes used for the three MAC identifier/address fields used in our feature sets. The MAC identifier fields are Destination Address, Source Address and BSSID.

The MAC address is a unique number address attached to most Network Interface Cards (NIC). The MAC address enables each station to have a unique name on a network. Though a machine on a network may identify itself differently depending on what Open System Interconnect (OSI) network layer the communication is taken place at, the MAC address acts as the name of a computer at layer 2. Indeed other names used at higher levels like IP addresses and hostnames all map back the MAC address. MAC addresses are usually shown with hexadecimal equivalent of each octet separated by a dash or colon. The following are valid MAC addresses : FF-FF-FF-FF-FF-FF , FF:FF:FF:FF:FF:FF. While the other fields of the frame used in our feature set have unique numbers which they can be mapped to in the processed datasets, this does not hold true for the MAC Address. This makes it imperative for us to come up with map address mapping schemes which can give unique numbers to the MAC addresses in the datasets while still providing meaningful patterns that can be deciphered by the learning algorithm in the training and testing procedure.

As stated earlier, since our aim is to explore impact of MAC address mapping schemes on the cross-platform robustness of machine learning based IDSs, we compare two different techniques for mapping MAC addresses. The results presented here show the results of our experiments using these two different techniques which are highlighted below.

*1) **Simple**:* This is the mapping scheme used in [4]. It maps the identifiers based on the ordinal position of the MAC addresses in a sorted list. The simple MAC mapping technique is detailed in Algorithm 1.

*2) **Role Based**:* This novel technique is devised by the authors to minimize the impact of the MAC address mapping on the cross-platform robustness issue. It maps MAC addresses based on the role which the machine of origin plays on the network in question or the role of the MAC address if it is special/reserved address. The recognized roles in our scheme are

- **Broadcast**
- **Access Point**
- **Station/Client**
- **Host**

TABLE V

NEURAL NETWORK PARAMETERS

Parameter	Setting
Momentum	0.2
Learning Rate	0.3
No. Epochs	500
Random Seed for Weights	0
No. of hidden nodes	30

- **Other**

The Role Based MAC mapping technique is detailed in Algorithm 2.

Algorithm 1 Simple Mapping

```
Input: Array $X[]$ containing all MAC addresses in dataset .
Output: Array $Y[]$ containing integer mappings of the MAC
    addresses in $X[]$. {Mapping of $X[i] = Y[i]$}
Sort(X)
$i = 1$
for every $macaddr\ in\ X$ do
    $Y[i] = i$
    $i++$
end for
Return(Y)
```

VI. Results

In intrusion detection, two metrics are typically used in order to quantify the performance of the IDS, Detection Rate (DR) and False Positive Rate (FP), equations (3) and (4) respectively. A high DR and low FP rate would be the desired outcomes. In the instance of an unbalanced data set (more of one type of exemplar then the other, in this case more normal then attack), an evolved solution can survive by simply learning to label all of the exemplars as the larger type in the data set. This survival technique will provide a high DR, but also a high FP rate, an undesirable result. Undesirable results of this kind are referred to as *outlier solutions*. The results presented do not include outlier solutions.

$$DR = 1 - \frac{\#FalseNegativeClassifications}{TotalNumberOfAttackConnections} \tag{3}$$

$$FP = \frac{\#FalsePositiveClassifications}{TotalNumberofNormalConnections} \tag{4}$$

A. Snort-Wireless Results

All the datasets listed in Table III are replayed through Snort-Wireless. Snort-Wireless is able to detect the attacks in the dataset without the need to change any of its detection metrics or configuration values. This an example of Snort-Wireless de-authentication alert:

```
[**] [211:1:1] (spp_deauthflood) Deauthflood
 detected! Addr src: 00:03:93:ec:64:55 ->
 Addr dst: ff:ff:ff:ff:ff:ff,
 Bssid: 00:03:93:ec:64:55. [**]
```

The presence of these alerts in the snort alert logs also serves to show that the attacks launched against the networks are effective.

Algorithm 2 Role Based Mapping

```
Input: Array $X[]$ containing all MAC addresses in dataset .
Output: Array $Y[]$ containing integer mappings of the MAC
    addresses in $X[]$. {Mapping of $X[i] = Y[i]$}
Sort(X)
$i = 1$
for every $macaddr\ in\ X$ do
    $Role$ = DetermineRole(macaddr) {Role can either
    be Broadcast, Access_Point, Host, Station or Other}
    if $Role = Broadcast$ then
        $Y[i] = 1$
        $i++$
    end if
    if $Role = Access\_Point$ then
        $Y[i] = 2$
        $i++$
    end if
    {Next IF is included only if data is been processed for
    a Host Based IDS}
    if $Role = Host$ then
        $Y[i] = 3$
        $i++$
    end if
    if $Role = Station$ then
        $Y[i] = 4$
        $i++$
    else
        $Y[i] = 5$ {Assumed that $Role = Other$}
        $i++$
    end if
end for
Return(Y)
```

B. Machine Learning Based Results

In order investigate the Cross-Platform robustness of our machine learning based solutions, we performed a 20-fold cross validation on the datasets. This implies that solutions are produced by training on each dataset and each solution tested on the each of the other datasets including itself. Moreover the ANN training on each dataset produced only one solution which was then tested on the other datasets. However, the GP training on each dataset produced 20 different solutions, which were trained using different seeding for the initial population. The final results were divided into four groups namely

1) Result of Testing On Network I datasets using solutions trained on Network I datasets
2) Result of Testing On Network II datasets using solutions trained on Network II datasets
3) Result of Testing On Network I datasets using solutions trained on Network II datasets
4) Result of Testing On Network II datasets using solutions trained on Network I datasets

Groups (1) and (2) are called results within-platform, whereas and (3) and (4) are cross-platform results. Figure 3 gives a breakdown of training times for all the runs of the GP and ANN. It shows the maximum, average and minimum training times using quartile charts. The results show that it takes relatively shorter periods of time to train GP based solutions when compared to ANN solutions. Our other results are presented in the following sections, which discusses results based on the MAC Address mapping techniques used.

C. Simple MAC Address Mapping

The algorithm for the Simple MAC Address Mapping technique is given in Algorithm 1. All the runs of the GP and ANN using this MAC mapping technique were able to produce best case solutions that achieved a 100% detection rate and a 0% FP rate, we however present the average case performance below.

TABLE VI
AVERAGE CASE PERFORMANCE WITHIN-PLATFORM: GP USING SIMPLE MAPPING

FP	DR	Time
0.015	0.99	42.285

TABLE VII
AVERAGE CASE PERFORMANCE ACROSS PLATFORM: GP USING SIMPLE MAPPING

FP	DR	Time
0.02	0.75	41.815

TABLE VIII
AVERAGE CASE PERFORMANCE WITHIN-PLATFORM: NEURAL NETWORK USING SIMPLE MAPPING

FP	DR	Time
0.02	1.0	83.055

TABLE IX
AVERAGE CASE PERFORMANCE ACROSS PLATFORM: NEURAL NETWORK USING SIMPLE MAPPING

FP	DR	Time
0.055	0.46	83.055

Our analysis we can see that FP rates remain pretty much constant for the GP and ANN for all solutions either within or across platforms, a difference can however be noticed when we evaluate the DR, Tables VI to IX. Within-Platform the average case DRs for both GP and ANN solutions are above 99% but this reduces to 75% and 46% respectively.

When we look at the average case DRs for both GP and ANN solutions across platforms we notice a significant drop in the performance. This fact is further corroborated by Figure 4 and Figure 5, where the degradation in performance can be clearly seen.

With these results we can state that cross-platform robustness is a problem for Machine Learning based IDSs. If Machine Learning based solutions are to be used in the real world, they would either perform below par or would have to be trained specifically for each network on which they run and re-trained every time there is a major change to the configuration of the network. Since this is not acceptable, in order to solve this problem the following is proposed. Based on our analysis of the situation, which was achieved by analysing the feature set and their representation in the datasets, we think that a major part of this Cross-Platform problem is due to MAC address representation. This lead to the design of the mapping technique which is based on network roles as a possible solution to the problem. The results of our experiments in this regard are presented in the following section.

D. Role Based MAC Address Mapping

The algorithm for the Role Based MAC Address Mapping technique is given in Algorithm 2. As stated above this mapping technique is proposed to solve the reduced detection capabilities of machine learning based IDSs when tested across platforms. Just like the Simple mapping technique, all the runs of the GP and ANN using this MAC mapping technique were able to produce best case solutions that achieved a 100%

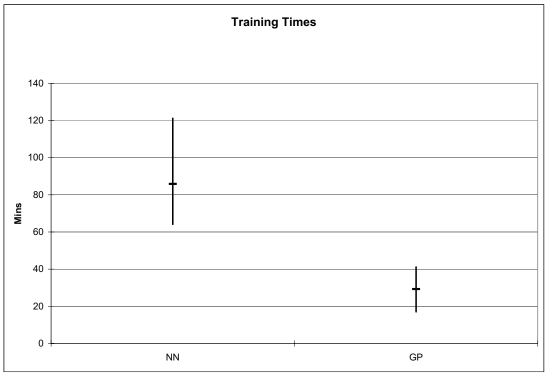


Fig. 3. Training Times

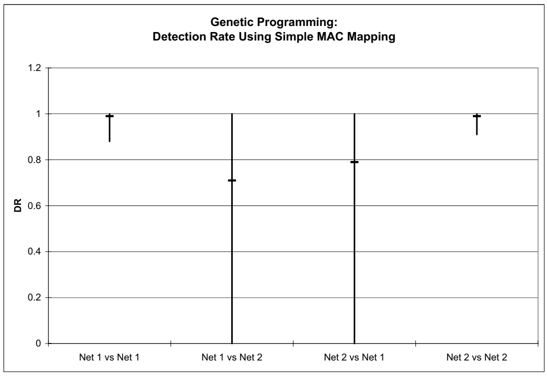


Fig. 4. GP: Detection Rate Using Simple MAC Mapping

detection rate and a 0% FP rate, we however again present the average case results below, Tables X to XIII

TABLE X
AVERAGE CASE PERFORMANCE WITHIN-PLATFORM: GP USING ROLE BASED MAPPING

FP	DR	Time
0.015	0.98	26.145

TABLE XI
AVERAGE CASE PERFORMANCE ACROSS PLATFORM: GP USING ROLE BASED MAPPING

FP	DR	Time
0.015	0.985	26.32

TABLE XII
AVERAGE CASE PERFORMANCE WITHIN-PLATFORM: NEURAL NETWORK USING ROLE BASED MAPPING

FP	DR	Time
0.0016	0.995	92.3975

Again we see that FP rates remain pretty much constant for the GP and ANN for all solutions either within or across platforms. We however do not notice the significant depreciation in performance when comparing DRs Within-Platform with those across platform as we noticed with the Simple mapping technique.

The bigger picture of the success of the role based mapping technique is further corroborated by quartile charts in Figure 6 and Figure 7, where sustained performance can clearly be seen. The average DRs remain relatively constant whether solutions are used Within-Platform or across platform.

VII. CONCLUSION AND FUTURE WORK

Using GP and ANN based solutions, we ascertained that issues exist with Cross-Platform robustness in machine learning based solutions for 802.11 Link Layer DoS Attacks. By focusing on the feature set and feature set presentation in previous work [2], [4], we narrowed a significant part of the problem to the representation of MAC addresses in the feature set in the training data. As we are dealing with Link Layer attacks our assertion has strong support, while the feature set in [2] might not be used in all instances, it is safe to say that whatever the feature set used, it would contain MAC addresses.

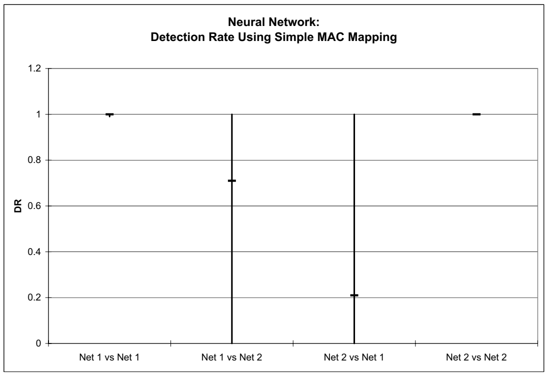


Fig. 5. ANN: Detection Rate Using Simple MAC Mapping

TABLE XIII
AVERAGE CASE PERFORMANCE ACROSS PLATFORMS: NEURAL NETWORK USING ROLE BASED MAPPING

FP	DR	Time
0.0025	0.999	92.3975

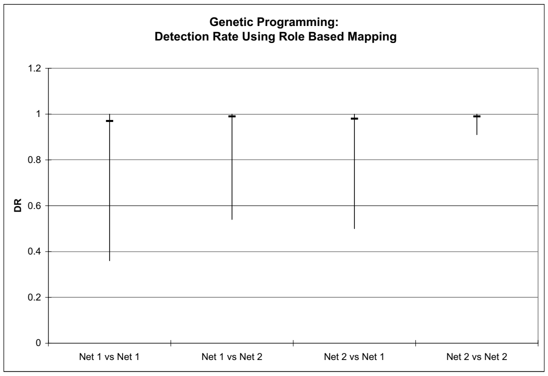


Fig. 6. GP: Detection Rate Using Role Based MAC Mapping

Our solution was to come up with a MAC mapping paradigm that represents the MAC in the training data based on role, to this end we proposed the Role Based Mapping technique as replacement for the Simple mapping technique used in [4]. Our results show that not only is the new technique able to maintain its DR capability Within-Platform from the Simple Mapping technique, it also vastly improves the DR in cross platform situations in both worst case and average case situations.

Our future work will explore the use of other MAC address mapping techniques that we develop and using data sets collected on larger networks with more than one AP. This will allow us to verify the effectiveness of our work over larger networks as well as a varied number and length of DoS attacks.

Furthermore, we plan on applying the same approach described here on other WiFi attacks, with the goal of developing an IDS that can be used to detect a variety of attacks. We also believe that the Role Based paradigm can be used to enhance cross platform robustness for network names at higher levels like IP addresses. We intend to achieve this by using our techniques on higher level attacks.

ACKNOWLEDGEMENTS

This research is supported by the NSERC Discovery and the CFI New Opportunities grants. The authors would also like to acknowledge the staff of Palomino System Innovations

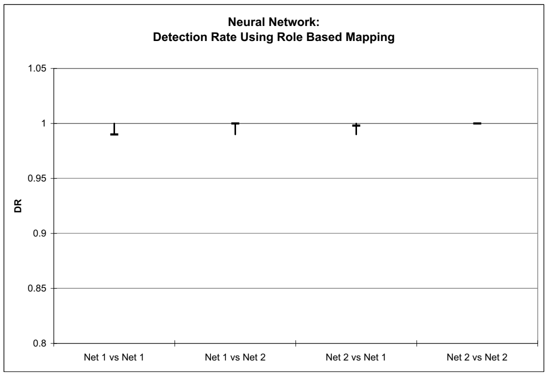


Fig. 7. ANN: Detection Rate Using Simple MAC Mapping

Inc., based in Toronto, Ontario and Telecoms Applications Reseeach Alliance (TARA), based in Halifax, Nova Scotia for their support in completing this work.

This work is conducted as part of the NIMS project at http://www.cs.dal.ca/projectx/.

References

[1] W. A. Arbaugh, N. Shankar, Y. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," *IEEE Wireless Communications*, vol. 9, pp. 44 – 51, December 2002.

[2] P. LaRoche and A. N. Zincir-Heywood, "Genetic programming based wifi data link layer attack detection," in *CNSR 2006*. Los Alamitos, CA 90720-1314: IEEE Computer Society, May 2006, pp. 285 – 292.

[3] B. L. Yanheng Liu, Daxim Tian, "A wireless intrusion detection method based on dynamic growing neural network," *1st International Multi-Symposium on Computer and Computational Sciences*, 2006.

[4] A. Makanju, P. Laroche, and N. Z. Heywood, "A comparison between signature and gp-based idss for link layer attacks on wifi networks," in *Proceedings of the 2007 IEEE Symposium Series on Computational Intelligence*, April 2007.

[5] IEEE-SA, *ANSI/IEEE Std. 802.11*, 1993rd ed., IEEE, New York, NY, USA, 2003.

[6] M. Maxim and D. Pollino, *Wireless Security*. McGraw Hill, 2002.

[7] R. Floeter, "Void11 main page, www.wirelessdefence.org/contents/void11mai Retrieved from the Web., August 2006.

[8] J. Malinen, "Host ap driver for intersil prism2/2.5/3, hostapd, and wpa supplicant, http://hostap.epitest.fi," Retrieved from the Web., 2006.

[9] M. Crosbie and E. Spafford, "Applying genetic programming to intrusion detection," in *AAAI Symposium on Genetic Programming*, J. K. E.V. Siegel, Ed., AAAI. Cambridge, MA, USA: MIT, 1995, pp. 1 – 8.

[10] Sourcefire-Inc, "Snort - the de facto standard for intrusion detection/prevention, http://www.snort.org," Retrieved from the Web., 2006.

[11] A. Lockhart, "Snort wireless, http://www.snort-wireless.org," Retrieved from the web., 2005.

[12] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbour, Michigan, USA: University of Michigan Press, 1975.

[13] J. Koza, "Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems," Computer Science Department , Stanford University, Tech. Rep., 1990.

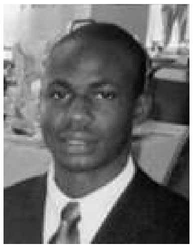
[14] M. Heywood and A. Zincir-Heywood, "Page-based linear genetic programming," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 3823 – 3828, 2000.

[15] C. Gathercole and P. Ross, "Dynamic training subset selection for supervised learning in genetic programming," *Parallel Problem Solving from Nature III*, vol. 866, pp. 312 – 321, 1994.

[16] D. Song, M. Heywood, and A. Zincir-Heywood, "Training genetic programming on half a million patterns: an example from anomaly detection," *IEEE Transactions on Evolutionary Computation*, pp. 225 – 239, 2005.

[17] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[18] M. Kershaw, "Kismet wireless, http://www.kismetwireless.net," Retrieved from the Web., 2006.



Adetokunbo Makanju obtained a BSc. in Computer Science from the University of Lagos, in Lagos Nigeria, in 1999 and is currently an MCS candidate at Dalhousie University, Halifax, NS, Canada. He also works part time as an Application Developer with Palomino Systems Innovations based in Toronto, ON, Canada. His research interests include but are not limited to the areas of Wireless Networks, Intrusion Detection, Genetic Programming and Cased Based Reasoning. In particular, his focus is on the applications of machine intelligence to real-world problems.

A. Nur Zincir-Heywood recieved the Ph.D degree in network information retrieval from the Department of Computer Engineering, Ege University, Izmir, Turkey, in 1998.

She is an Associate Professor with the Computer Science Department, Dalhousie University, Halifax, NS, Canada. From 1996 to 1997, she was a Visiting Researcher at the IIMS Research Center, School of Engineering, University of Sussex, Brighton, U.K. Previous to her current position, she was an Assistant Professor with the Department of Computer Engineering, Ege University (1998 - 2000). She has also been involved with Network Technology Workshops of Internet Society as an Instructor from 1997 to 2000. Her research interests include intrusion detection, network security, network management, and network information retrieval. She has published journal and conference papers in these areas, and has been involved in projects concerning network security and information systems.

Dr. Zincir-Heywood is a member of the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE).