# Using Metrics From Multiple Layers to Detect Attacks in Wireless Networks

by

Francisco Javier Aparicio Navarro

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy

of

Loughborough University

28th February 2014

# Abstract

The IEEE 802.11 networks are vulnerable to numerous wireless-specific attacks. Attackers can implement MAC address spoofing techniques to launch these attacks, while masquerading themselves behind a false MAC address. The implementation of Intrusion Detection Systems has become fundamental in the development of security infrastructures for wireless networks. This thesis proposes the designing a novel security system that makes use of metrics from multiple layers of observation to produce a collective decision on whether an attack is taking place.

The Dempster-Shafer Theory of Evidence is the data fusion technique used to combine the evidences from the different layers. A novel, unsupervised and self-adaptive Basic Probability Assignment (BPA) approach able to automatically adapt its beliefs assignment to the current characteristics of the wireless network is proposed. This BPA approach is composed of three different and independent statistical techniques, which are capable to identify the presence of attacks in real time. Despite the lightweight processing requirements, the proposed security system produces outstanding detection results, generating high intrusion detection accuracy and very low number of false alarms. A thorough description of the generated results, for all the considered datasets is presented in this thesis. The effectiveness of the proposed system is evaluated using different types of injection attacks. Regarding one of these attacks, to the best of the author knowledge, the security system presented in this thesis is the first one able to efficiently identify the Airpwn attack.

**Keywords:**

IEEE 802.11, Wireless Security, Dempster-Shafer, Basic Probability Assignment, Data Fusion, Multi-layer Measurements, Intrusion Detection Systems

*A Noelia,*
*por querer compartir una vida juntos*

# Acknowledgments

First and foremost, I would like to thank my supervisor Professor David J. Parish for giving me guidance and constant support throughout my research. I express to him my deepest gratitude for giving me the opportunity to progress, not only in my career, but also at a personal level. I am also grateful to Dr. Konstantinos Kyriakopoulos for his collaboration during this research, his constant support and invaluable help. His friendship is one of the most valuable assets I will keep from all these years. I also owe special thanks to Dr. John Whitley and Dr. Raphael C.-W. Phan for their advices.

As well as this, I owe all my gratitude to Noelia, my wife. Noelia has been a tremendous source of encouragement and motivation during these years. She gave me strength when I needed it most. Also, I am deeply grateful to my parents and sister, for their support and endless love. I would have never been able to complete my research without them.

I would also like to thank all of my colleagues in High Speed Networks group, Dr. William Johnson, Abdulrazaq Almutairi, Rui Li, Segun Aina, Jozef Woods, Ginés Escudero and Ansar Jamil for their kindness and support. I would also like to express my thankfulness to my family and friends in Mazarrón, and my friends from all around Europe, for their constant interest in my research and their relentless moral support. Special thanks to Tim, Ambrogio, Moritz, and Chris.

Lastly, I would like to include an apology to my wife, my family and friends for having to put them up with long months of silence and brooding preoccupation during my research. All in all, I think all the effort has definitely paid off.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **ACK** | Acknowledgment |
| **ANN** | Artificial Neural Network |
| **AP** | Access Point |
| **Bel** | Belief Function |
| **BPA** | Basic Probability Assignment |
| **CRC** | Cyclic Redundancy Check |
| **CSMA/CA** | Carrier Sense Multiple Access with Collision Avoidance |
| **CTS** | Clear-to-Send |
| **D-S** | Dempster-Shafer Theory of Evidence |
| **DDoS** | Distributed Denial-of-Service |
| **DoS** | Denial-of-Service |
| **DR** | Detection Rate |
| **FN** | False Negative |
| **FP** | False Positive |
| **HTML** | Hypertext Markup Language |
| **HTTP** | Hypertext Transfer Protocol |
| **IDS** | Intrusion Detection System |
| $\mathbf{INJ_{Rate}}$ | Injection Rate |
| **IPS** | Intrusion Prevention System |
| **ISP** | Internet Service Provider |

| | |
|---|---|
| **ITS** | Intrusion Tolerant System |
| **k-NN** | k-Nearest Neighbour |
| **LTE** | Long Term Evolution |
| **MAC** | Medium Access Control |
| **MitM** | Man In The Middle |
| **NAV** | Network Allocation Vector |
| **NIC** | Network Interface Controller |
| **WEP** | Wired Equivalent Privacy |
| **WiFi** | Wireless Fidelity |
| **WiMAX** | Worldwide Interoperability of Microwave Access |
| **WLAN** | Wireless Local Area Network |
| **WPA** | WiFi Protected Access |
| **WSN** | Wireless Sensors Network |
| **PHY** | Physical layer |
| **Pl** | Plausibility Function |
| **RADIUS** | Remote Authentication Dial In User Service |
| **RSSI** | Received Signal Strength Indication |
| **RTS** | Request To Send |
| **RTT** | Round Trip Time |
| $\mathbf{SEQ_{Dif}}$ | Sequence Numbers difference |
| **SNMP** | Simple Network Management Protocol |
| **SNR** | Signal to Noise Ratio |
| **SSID** | Service Set Identity |
| **SSL** | Secure Sockets Layer |
| **SVM** | Support Vector Machine |
| **TN** | True Negative |
| **TP** | True Positive |
| **TTL** | Time To Live |
| **VPN** | Virtual Private Network |
| **ΔTime** | Frame Interarrival Time |

# Chapter 1

# Introduction

## 1.1 Background

Wireless Local Area Networks (WLANs) have experienced a tremendous growth in popularity over the last decade. WLANs, based on the IEEE 802.11 standard, have become very affordable and have been increasingly deployed in businesses of any kind, educational institutions, governmental buildings, public places, and private properties. They offer to users significant mobility advantage, usage flexibility, fast and cheap network deployment, and easier scalability than conventional wired networks. More recently, 4G technology has gained relevance in the mobile and broadband connectivity market. Two 4G wireless technologies, Long Term Evolution (LTE) and Worldwide Interoperability of Microwave Access (WiMAX), based on the IEEE 802.16 standard, are forecasted to dominate the broadband connectivity market.

The number of mobile broadband subscribers is expected to reach 9.3 billion by 2018 [1]. Despite the growth in popularity of the 4G technologies, IEEE 802.11 networks, known as WiFi, still attracts great interest among companies that provide Internet access, and final users. Infonetics explains that mobile operators want to see a closer integration of WiFi with mobile networks in coming years. The mobile operators perceive WiFi as a key solution for more intelligent mobile offload [2]. This market research firm forecasts that mobile WiFi access points will experience in 2016 a growth rate of the 86%, as mobile operators want a closer integration between WiFi and the mobile networks [3]. Private users of wireless-ready electronic devices still perceive WiFi as the main source of data connectivity for their smartphones and tablets. According to the report in [4], 58% of smartphone users and 93% of tablet owners use WiFi for Internet access, rather than using mobile broadband connectivity.

## 1.2 Insecurity of Wireless Networks

Unfortunately, the fast change in technology, the general trend towards mobility and Internet accessibility anytime-anywhere are playing a very important role in the insecurity of wireless devices. The vulnerabilities present in the wireless standard protocols have exposed wireless network users to an increasing number of sophisticated, easy to launch and untraceable attacks [51]. A discussion about the security of wireless network is presented in [53].

The IEEE 802.11 standard proposed different security protocols, establishing traffic encryption and integrity protection to the network infrastructure [5], as well as avoiding unauthorised access to the wireless networks. However, wireless networks cannot rely on these security protocols to protect the content of the communications. All these security protocols are vulnerable to decryption analysis processes [7] [8] [10]. In addition to be vulnerable to attacks and intrusion attempts that generally affect the devices connected to the wired networks such as viruses, malware, spam, web-based attacks, Denial of Service (DoS), Distributed DoS (DDoS), identity theft, etc., the wireless network users suffer also from new types of attacks that specifically affects the wireless networks. These attacks are known as Wireless-Specific attacks. The wireless-specific attacks target the wireless vulnerabilities in the two lower layers of the network protocol stack, the Physical (PHY) and the Medium Access Control (MAC) layers. In contrast to the cyber-attacks that come from the core of the network and reach the wireless devices through the Access Point (AP) for WiFi networks (eNodeBS for LTE and Base Station for WiMAX), wireless-specific attacks are those that are launched with a third party device and reach the wireless devices through the wireless link of the network.

The difficulty of launching cyber-attacks that specifically compromise the wireless communications is decreasing fast. The required knowledge, software and devices to launch these attacks are very easily available [9]. Both the software tools and the essential instructions can be easily found on the Internet. Similarly, the appropriate wireless Network Interface Controller (NIC) can be bought and the precise chipset driver can be also found on the Internet.

Bringing together the fact that the wireless networks are significantly more unsecure and easier to compromise by cyber-attacks than wired networks, the forecasted increase in the number of users that will make use of wireless networks to access to the Internet, and the fact that the number of cyber-crimes has experienced tremendous growth in recent years, creates the perfect scenario possible for hackers to carry out their malicious actions. The authors of [12] present a complete taxonomy of the most common threats that WLANs may encounter. Currently, the number of attacks that specifically target mobile devices remains small. However, information security experts have alerted that it is only a matter of time before these figures change [6]. It is expected that the next generation of wireless networks, particularly those serving a wide range of users (e.g. open/public wireless networks) will have to operate during prolonged periods of time under threat of attack or facing long periods of active attacks. Therefore, it is clear that special effort must be made to provide more reliable protection mechanisms to such networks and the devices that access to them.

### 1.2.1 Mechanisms to Enhance Wireless Networks Security

The design of secure and reliable wireless networks presents a major challenge to security system designers. Security mechanisms that have been commonly used to protect the networks, and especially wireless networks, have not been completely efficient. Whichever the implemented security mechanisms, it has been a matter of time before these mechanisms have been circumvented or overpassed by attackers. All the security mechanism should be designed with the following security objectives in mind. These security objectives have been nicely documented in [10] [12].

- Confidentiality

  The network security systems should ensure that only authorised users have access to transmitted information, and safeguard the privacy of the transmitted information from unauthorised users and attackers. Encryption, authentication and access control mechanisms are the most commonly utilised approaches to provide confidentiality to the wireless communications [13]. Authentication and access control mechanisms prevent unauthorised access to the wireless network resources. Encryption mechanisms prevent attacker or unauthorised users from reading transmitted information.

- Integrity

  The wireless devices within the WLANs should be aware of any intentional or unintentional modification in the transmitted information. The network security systems also should guarantee that unauthorised users and attackers do not modify the transmitted information, and indicate whether the information has been replayed. Cryptographic hashes are commonly utilised to provide integrity for wireless communications.

- Availability

  The wireless network should guarantee that legitimate devices could always be accessed and be able to use the network resources, upon demand. Network security systems should ensure that unintended users or attackers could not block the access to the wireless network resources. Because the availability of a network could be compromised by attacks of diverse nature, different security mechanisms might be required to safeguard the availability of the network.

Traditional network security mechanisms, such as cryptography protocols, firewalls or antivirus, are not efficient enough to protect the wireless network infrastructure. These static mechanisms are unable to dynamically adapt their detection capabilities to either the changing characteristics of the protected communication systems or the constantly changing complexity of the cyber-attacks. Although different amendments of the wireless technology standards have been released and modified to provide stronger cryptographic mechanisms and more robust security policies, there are still numerous attacks able to compromise the privacy of the wireless communications. Antivirus systems are intended to protect Personal Computers (PCs) from viruses, malware or Trojan horses, mainly at the application layer. These systems are effective identifying and removing from the systems infections that have already occurred, but inappropriate to protect wireless networks and wireless devices at the lower layers of the protocol stack. The firewalls, commonly allocated between the local network and the Internet Service Provider (ISP) backbone, cannot protect wireless networks from wireless-specific attacks because the lack of physical boundaries allows the attacker to directly interact with the wireless devices without passing through the firewall. Therefore, an improved or alternative solution for securing the wireless networks is required.

### 1.2.2 Intrusion Detection Systems

Performing an analysis of the vulnerabilities of the wireless networks it is easy to conclude that there exist three main characteristics that make the implementation of the wireless-specific attacks feasible. These are the ability to intercept and analyse the wireless communications content, the capability to inject malicious information into the wireless communication, and the capacity to impersonate the identity of legitimate wireless members of the network. An attacker can easily implement techniques of MAC address spoofing. These characteristics allow the attackers to implement an immense number of attacks that compromise the confidentiality, integrity and availability of the wireless communications and wireless networks infrastructure. The efforts to provide reliable security to the wireless networks should focus on discovering the real identity of the device that transmits the frames, even if the attacker masquerades itself behind a false identity.

The implementation of wireless network monitoring tools, such as Intrusion Detection Systems (IDSs), able to identify the presence of attackers intercepting and injecting information, as well as impersonating the identity of legitimate wireless devices, has become fundamental in the development of security infrastructures for wireless networks. IDSs are security systems designed to protect networks or computer systems by constantly monitoring and detecting malicious actions that compromise the resources of the monitored system [14]. In [86], the authors describe the main functions performed by IDSs as gathering activity information from the monitored system, analysing the gathered information and assessing the nature of this information, and raising an alarm if the outcome of the detection process indicates the presence of attack. These systems incorporate sophisticated information analysis techniques that allow processing diverse types of information.

IDSs outperform the protection capabilities of cryptography protocols, firewalls or antivirus. As will be explained in Chapter 4, certain types of IDSs dynamically update their detection capabilities, enhancing the network protection capabilities against attacks [15], in contrast to the traditional static security mechanisms; in particular, the anomaly-based IDSs. Different mechanisms could be utilised by the IDSs to analyse the datasets. Some mechanisms would produce more efficient results than others, depending on the different factors, such as the behaviour of the monitored system, the type of attacks, or the gathered metrics. An approach that has gained wide interest among the research community is the use of Data Mining techniques in tasks of intrusion detection [36].

The utilisation of data mining techniques has been proved to improve the detection capabilities of IDSs, making the datasets analysis process more efficient. Data mining techniques are able to identify previously unknown and useful information in datasets applying different mathematical and statistical analysis. However, almost every single data mining technique used in tasks of intrusion detection suffers from important drawbacks that need to be overcome. For instance, some of these techniques require preprocessing the analysed datasets to produce acceptable results, or require performing a thorough training process before carrying out the data analysis.

### 1.2.3 Multi-Layer Detection and Data Fusion

IDSs can make use of any observable and measurable metric of the monitored system to detect attacks. According to [16], common sources to extract these metrics are audit logs, network traffic, user commands or system calls. Focusing on wireless networks, the different features could be extracted from the frame headers, the frames payload or constructed traffic Netflow data. There is no restriction in the number of layers from which the metrics could be extracted. Either a single metric or multiple metrics could be extracted from one or multiple layers throughout the whole protocol stack. These gathered metrics compose the dataset analysed by an IDS to detect the presence of attacks or evidence of intrusion attempts.

The utilisation of an appropriate number of metrics is a very important step for IDSs towards an efficient intrusion detection process. Although there are cases in which IDSs that utilise information from a single metric might give precise detection results, the presence of attacks is rarely accurately detectable by examining a single metric from one layer of the protocol stack. As many researchers have previously demonstrated [37] [38] [39], the combined use of multiple metrics from the same or different protocol layers may result in higher detection accuracy rate with lower numbers of false alarms. Hence, utilising a multi-layer approach may help towards improving the process of detecting and mitigating wireless network attacks.

Using the information from multiple metrics from multiple layers could be managed in different ways. One of the mechanisms is the utilisation of data fusion methods to make a combined use of the information from the different metrics. Data fusion can be defined as the process of collecting information from multiple and heterogeneous sources, and combining them towards obtaining a more accurate final result [39]. Among different data fusion methods that could have been utilised, the Dempster-Shafer (D-S) Theory of Evidence has been chosen in this work. One of the reasons for this selection is the ability of managing uncertainty, which allows tackling a large range of problems.

The D-S theory has been previously used in the intrusion detection field to enhance the detection accuracy of IDSs [39] [40] [41]. Despite been proven as a powerful and efficient technique, a very important step to allow D-S theory to be used in practice remains to be investigated. This is to find an automatic and self-adaptive process of Basic Probability Assignment (BPA), based on the measured characteristics of the network. The major challenge for applying D-S theory in IDS is to automatically determine the beliefs assigned to each of the considered hypothesis, based on the information extracted from the network measurements [42]. None of the previous works that use D-S has found an efficient solution to this challenge.

Among all the works on IDS that investigate the use of the D-S theory, there exist multiple ways of assigning belief to each of the considered hypotheses. However, few of them could be used off-the-shelf without either prior thorough training or fine tuning period. Furthermore, most of the alternative techniques have to be trained with completely clean datasets. Expert opinion to manually assign the belief probabilities is often the utilised approach, which is inadequate for automatic and self-adaptive IDSs. The utilisation of fixed functions or linear functions to assign the belief is another approach often utilised, which cannot automatically adapt to changes. Other group of approaches to assign belief to each of the hypotheses is based on the uses of data mining techniques, which require the gathering of large amounts of data traffic and the completion of a training period before being able to perform BPA tasks. These systems may be unable to automatically adapt to changes in real time.

## 1.3 Motivation and Objectives

Despite the development of many mechanisms to provide security for the wireless networks, these networks remain insecure. Wireless communications still suffer from security vulnerabilities [10]. Designing a security system that could provide better levels of protection for the wireless networks is one of the principal objectives of this thesis. Any effort to provide an extra level of protection to a network has become an issue of critical importance. In an optimum situation, security system should be autonomous and unsupervised, able to work with the intervention from a system administrator. Designing a system with these characteristics has become one of the objectives of this research work.

Security systems that work in off-line or nearly real time are one step behind the attackers. In order to provide significant security protection against attacks, the proposed security mechanism should work in real time, analysing network frames as soon as they arrive to the protected system. The security system should be capable of detecting the presence of attacks without the need for thorough training process and the computational cost of the system should be lightweight to be applied in real time. Therefore, the proposed system should be computationally low cost. In addition, it is desirable for a security system to be scalable and applicable to other wireless technologies.

As explained previously, the utilisation of multi-layer approaches may help towards improving the process of detecting and mitigating wireless network attacks. Hence, in order to maximise the accuracy of the security system, this system should implement multi-layer and data fusion techniques to increase the intrusion detection capabilities. The D-S theory has been chosen in this work, for the ability of managing uncertainty that this technique provides. In order to utilise the D-S theory as part of an autonomous and unsupervised security system, able to perform the detection in real time, an automatic and self-adaptive BPA must be found. Currently, there exist multiple ways of assigning belief to each of the considered hypotheses. However, few of them could be used off-the-shelf without either prior thorough training or fine tuning period. Therefore, the implementation of novel BPA techniques able to automatically adapt their beliefs assignment to the current characteristics of the wireless network, without intervention from an IDS administrator became another of the most important objectives of this thesis.

## 1.4 Design Challenges

The experiments for the development of this thesis have been implemented using wireless network traffic datasets gathered in a live operational wireless network. WiFi has been the wireless network technology selected for the development of this work. This decision has been taken for different reasons. Firstly, the IEEE 802.11 standard still attract great interest among companies and final users. As explained previously, despite the growth in popularity of the 4G technologies, it is expected that wireless networks based on this standard will experience an 86% growth rate by 2016 [2]. Second, the wireless equipment based on the IEEE 802.11 standard was more easily accessible than WiMAX equipment. A limited number of WiMAX cards using open source drivers were available, making the implementation of wireless-specific attacks in a live operational WiMAX network unfeasible. Also, the compatibility of TShark [17] gathering network traffic is more advanced and better integrated with the IEEE 802.11 standard than the 4G technology standards.

Attempts were made to evaluate the proposed methodology on a WiMAX network. Although a base station emulator was used for this purpose, the capabilities of this emulator were highly restricted, and did not allow the implementation of the wireless-specific attacks that have been examined in this thesis. The level of restriction of this WiMAX network emulator made it difficult to acquire real WiMAX network traffic, and the experiments that have been implemented in this work were therefore very limited. However, the potential of the approach for use on a WiMAX network has been investigated to some degree.

It should be noted that the LTE and WiMAX network traffic datasets required could not have been created directly utilising the different network simulation software available. This option was discarded because of two main reasons. On the one hand, the network simulation software tools are unable to take into account all the parameters and environmental conditions that wireless communication may experience in a physical network testbed. The simulated network traffic datasets would lack relevant information to implement the experiments and to conduct the attack detections. On the other hand, the available network simulation software tools do not allow direct implementation of the wireless-specific attacks that are examined in this thesis. Therefore, the main experiments for the development of this thesis have been implemented in a physically deployed WiFi network testbed.

## 1.5 Thesis Contributions and Outcomes

- Novel BPA methodology

  The most important contribution of this thesis is a novel BPA methodology able to automatically adapt its probabilities assignment to the current characteristics of the wireless network, without intervention from an IDS administrator. The novel automatic, unsupervised and self-adaptive BPA methodology developed for this thesis is composed of three different and independent statistical techniques, able of generating predictions regarding the presence of attacks, based on historical parameters.

- Multi-layer security system

  Another major contribution of this thesis is the development of a novel automatic and self-adaptive security system, based on multi-layer data fusion architecture, which provides real time protection to the wireless networks against wireless-specific attacks. The security mechanism developed for this thesis makes use of metrics from multiple layers of observation to protect IEEE 802.11 networks from wireless-specific attacks. The developed anomaly-based IDS provides high level of protection to the wireless networks against this type of attacks. In addition, the system has been proved scalable and applicable to other wireless technologies.

- Real time protection

  The presented security system only requires a lightweight process for generating a baseline profile of normal utilisation. The lightweight processing requirements of the three proposed techniques allow implementing the BPA process for the detection in real time, concluding the real nature of each single analysed frame is just a few µseconds. This timeframe allows the presented system to perform any countermeasure action before a new incoming frame reaches the protected system.

- High intrusion detection accuracy

  Despite the lightweight processing requirements, the proposed security system produces outstanding detection results, generating high intrusion detection accuracy and very low number of false alarms. It also ratifies the improvement in the intrusion detection process provided by the combined use of information from multiple protocols stack layers. The complete description of the IDS framework and the detailed description of the three developed BPA methodologies are presented in this thesis.

- Detection of different injection attacks

  The effectiveness of the presented security system has been evaluated using different types of injection attacks. Regarding one of these attacks, to the best of the author knowledge, the security system presented in this thesis is the first one able to efficiently identify the Airpwn attack [18]. Still there has not been reported evidence that cyber-attackers are using the Airpwn attack actively. Nonetheless, in this thesis it is speculated that cyber-attackers might see in this tool an effective and easy to implement mechanism to gain substantial economic benefit.

- Publicly available datasets

  A series of network traffic datasets have been gathered from a live operational IEEE 802.11 network, physically deployed in the laboratory of the High Speed Network Group, at Loughborough University. The network traffic was gathered and stored in the form of pcap files [19], using TShark. These datasets have been made accessible and publicly available in [20]. Considering the lack of publicly available network traffic datasets with which different IDSs can be evaluated, the gathered datasets is another contribution of this thesis.

- Source code

  The presented security system has been written in the C language, during the completion of this thesis, which provides great flexibility to be easily adapted or integrated to other security implementations.

## 1.6 Thesis Outline

The outline of the thesis is organised as follows:

Chapter 2 provides an overview of the most important published works in the field of wireless security and intrusion detection. Existing detection methodologies to identify attacks in WLANs that make use of MAC spoofing are central to the content of this chapter. Different data mining techniques are described. Additionally, systems that use of multi-layer approach and data fusion methodologies are also discussed. This chapter also describes the different methods to determine the BPA values in D-S theory present in the IDS literature.

Chapter 3 presents a description of the security protocols recommended by the IEEE 802.11 standard, along with a brief description of the current security systems. This chapter also presents an extensive overview of the most common wireless-specific attacks. These are the attacks that more commonly compromise to wireless networks, in order to find a common implementation pattern that could help to identify a common detection or countermeasure mechanism against these attacks. The chapter finishes with a description of the particular wireless-specific attacks that have been practically evaluated in this thesis.

Chapter 4 introduces the concept of IDSs. This chapter presents a detailed taxonomy of the most relevant characteristics considered when an IDS is designed, as well as the pros and cons of each of the characteristics. Next, this chapter describes the characteristic included in the final architecture design of the IDS presented in this thesis, and discusses the principal reasons for selecting each of the characteristics that have been included in the final architecture of the system. The purpose of this chapter is to find the most convenient architecture for the presented detection system.

Chapter 5 introduces a thorough description of the detection techniques and internal architecture design of the IDS presented in this thesis. It describes the way the information if administrated within the IDS, and explains the sliding window technique developed to implement the IDS training process. It also introduces the concepts of multi-layer intrusion detection and data fusion techniques. The chapter concludes with the description of a novel BPA methodology able to automatically adapt its probabilities assignment to the current characteristics of the wireless network, without intervention from an IDS administrator.

Chapter 6 provides a thorough analysis of the wireless network traffic datasets analysed in this thesis. This chapter starts with a decision of whether utilise well-known network datasets, synthetically generated dataset, or network traffic datasets gathered from a live operational network. It also describes the testbed from which the different datasets have been gathered. Then, all the metrics used to carry out the intrusion detection are individually described. For each of the datasets, a thorough statistical analysis of each of the metrics is presented. The chapter concludes with a brief description of the concept of feature selection and curse of dimensionality.

Chapter 7 evaluates the effectiveness of the unsupervised anomaly based IDS framework presented in this thesis. This chapter presents a series of experiments and a thorough description of the generated results, for all the considered datasets and all the possible metric combinations. The experiments compare the detection results generated using the multi-layer approach (i.e. when all the considered metrics are used) against the same methodology utilising different sets of metrics. Also, the experiments evaluate the system configuration that best result generates, as well as the most appropriate sliding window length. The presented results empirically prove the efficiency of the proposed IDS and the different BPA techniques. In addition, the results empirically prove the intrusion detection can be implemented in real time.

Finally, Chapter 8 provides the conclusions of this thesis and discusses the possible future research work.

# Chapter 2

## Related Work

The research literature based on network security and IDSs is extensive. This chapter starts with an overview of the most important works published in the field of wireless security. It also provides an overview of the most important published works in the field of intrusion detection. Next, a brief description of some of the most widely used intrusion detection techniques is presented. Different techniques could be utilised by the IDSs to analyse network traffic and to identify attacks. Given that the efficiency of the IDSs depends on these techniques, it is important to consider the available intrusion detection techniques that could have been used in this thesis. For each technique, some advantages and disadvantages have been listed. This chapter pays special attention to the techniques mainly used by anomaly IDSs. Additionally, systems that use a multi-layer approach and data fusion methodologies are also discussed. Finally, the chapter concludes with the description of the different methods to determine the BPA values in D-S theory present in the IDS literature, and their inadequacy in being determined automatically and in real time, based on the measured characteristics of the analysed network traffic.

## 2.1 Wireless Security Background

The insecurity of IEEE 802.11 networks has been addressed in multiple research works. The authors of [12] present a complete taxonomy of the most common threats that WLANs may encounter. A number of wireless-specific attacks are described, along with a series of countermeasures. In [49], the authors address the insecurity of WiFi networks, describing some of the security protocols included in the IEEE 802.11 standard and some of the vulnerabilities. Similarly, the authors of [50] also describe some of the vulnerabilities of the IEEE 802.11 standard.

One of the most relevant works about the WiFi network vulnerabilities is presented in [51]. The authors describe a series IEEE 802.11 network vulnerabilities that could be exploited by an attacker. Then, the feasibility of implementing some wireless-specific attacks and the effect that these attacks produce on the wireless networks are studied with practical experiments. Finally, a number of countermeasures are also suggested. Other research work that describes an overview of the vulnerabilities in WiFi networks is presented in [52].

A discussion about the security of wireless network is presented in [53]. In this work, the authors discuss concepts about the security mechanisms of both, 3G and WiFi networks, and describe some of the standards vulnerabilities. Another research on the insecurity of wireless networks is presented in [5]. Similar to [53], this work discusses concepts about the security mechanisms of 3G and WiFi networks. A wide description of different security mechanisms described in the IEEE 802.11 standard is presented.

## 2.2 Research on Intrusion Detection Systems

IDSs have been a very active research topic for more than a decade. There are numerous publications that propose novel IDS approaches, for instance, to increase the detection efficiency of IDSs or to design more effective architecture of these systems. In [55], an extensive description of multiple concepts about IDSs is presented. The authors provide a complete guidance about the design, implementation and deployments of IDSs.

In [56], the authors describe multiple general concepts in the field of IDSs. This work describes some of the most common categories of IDS, provides some detection techniques, and presents the concept of data fusion in IDSs. Similarly, an extensive description of multiple concepts of IDSs is presented in [57] and explains some of the issues that remain open in the field of IDSs. Multiple concepts about IDSs are also presented in [58]. IDS categories, advantages and disadvantages of some of these categories and detection techniques are described in this work.

One of the most complete researches on IDSs is presented in [10]. This dissertation initially introduces the IEEE 802.11 standard, describes some of the vulnerabilities and feasible attacks, and gives a complete taxonomy of IDSs. Then, the author proposes a novel detection approach to detect intrusion in WLANs, using information from the PHY and MAC layers. In the PHY layer, the difference between consecutive Received Signal Strength Indication (RSSI) values is calculated and compared against a defined threshold. In the MAC layer, [10] measures the Round Trip Time (RTT) that takes to complete the Request-to-Send (RTS) - Clear-to-Send (CTS) handshake between the AP and the wireless client. This approach would be ineffective if the wireless devices do not implement the RTS-CTS process. Additionally, the detection threshold is calculated using averages of historical information. In contrast to the work presented in this thesis, both approaches operate independently. An alarm is raised after both approach consider there is an attack.

## 2.2.1 Detecting Attacks

A complete description of DoS attacks has been presented in [59]. The authors make a short description of IEEE 802.11 networks before describing a number of DoS attacks at the PHY and MAC layers. This work also presents a series of countermeasures that could be implemented against this type of attack. In [60], the authors present a brief description of different DoS that can be implemented in wireless networks. DoS attacks in WiFi networks, launched at the PHY and MAC layers, have been studied in many other research works. In [61] [62], the authors practically demonstrated the effect of the DoS attacks at both layers, stopping the legitimate wireless communication. Other researches have used simulation environments to study the DoS attacks in WiFi networks. For instance, [63] [64] [65] present works that use OPNET network modeller [66] to simulate DoS attacks. These works also propose countermeasure mechanisms against this type of attacks, and describe possible drawbacks for the proposed mechanisms.

In [67], another type of DoS attacks is presented. The authors have practically evaluated the effect of authentication request and association request flooding on the network performance. The solution against this attack proposed by the authors, MAC address filtering, may work against the legitimate network users.

One of the research works that employs the monotonic increment of the sequence number value to identify spoofing attacks is presented in [68]. Initially this work presents an extensive study about the change pattern of the sequence number in non-malicious situations. This is compared against the change pattern of the sequence number when spoofing attack takes place. Then, the authors propose and practically evaluate a methodology that uses sequence number value to identify attacks. Although the results show this is an effective approach, the authors explain that the proposed methodology is unable to detect spoofed frames if the attacker replicates the sequence number value of the last legitimate frame transmitted.

The methodology proposed in [72] also conducts the analysis of the sequence number to identify the presence of MAC spoofing attacks. Initially, the proposed detection system computes the theoretical maximum number of frames that can be transmitted per second. Through the use of the sequence number and the frame arrival time, the proposed methodology calculates whether the theoretical maximum number of frames is surpassed. In that case an alarm is reported. As highlighted by the authors, the effectiveness of this methodology relies on an arbitrary threshold defined by the system administrator, which is a major drawback. In addition, this methodology is unable to manage retransmitted frames, which, according to the authors, always produce false alarms. The authors of [72] also describe the utilisation of MacSpoof, a sequence number analysis tool. MacSpoof identifies gaps between the sequence numbers of two consecutive frames. If a certain number of gaps occur within a specified time period, an alarm is reported. This is another example of an inefficient detection mechanism that identifies the presence of attacks counting the number of times that the protected system has been attacked during a defined period of time.

In [69], the authors tackle the detection of spoofing attacks in a wireless ad hoc network. This work describes a method to detect MAC spoofing attacks, using the sequence numbers at the MAC layer and the interarrival time between frames. Described by the authors, in a situation in which the spoofing device transmitted before the legitimate wireless device, the detection system that uses the sequence numbers would consider that the legitimate device is implementing a spoofing attack. Additionally, the presented methodology makes uses of a sliding window scheme. For the implementation of the experiments, the authors explain that the length of the window ‘should be large enough’. However, the appropriate window length is never specified. In addition, the implementation of the severity region proposed in this work requires the administrator to specify the threshold levels.

Another approach that proposes using the sequence number to detect MAC spoofing attacks is presented in [70]. The authors present an adaptive threshold methodology to identify abrupt changes in the sequence number sequences. This approach is enhanced using timestamps of the monitored frames. Once more, the proposed system makes use of a threshold, which is empirically specified by the authors. Besides, the timestamp analysis is applied only on Beacon or Probe management frames.

The authors of [71] also present a methodology to detect MAC spoofing attacks using the sequence number and $RSSI$. This work requires generating a profile of normal behaviour in advance only with non-malicious information. The authors assume that the attacker is not present during the profiling period, and it is also assumed that the legitimate device is positioned in a fixed geographical location. Similar pieces of information are used in [72] to detect identity spoofing attacks.

In [73], the authors also tackle the detection of spoofing attacks in wireless networks using the single metric $RSSI$ to detect MAC spoofing attacks. Their approach requires the wireless devices to monitor the $RSSI$ value of communication in both directions of the communication between two legitimate devices in order to detect spoofing attacks. Also, the detection methodology requires the legitimate devices to keep and exchange records of the measured $RSSI$. Both parties compare their own records with the received information to detect the presence of identity spoofing attacks. This approach implementation exposes the exchanged information to be compromised by an attacker. Additionally, presented experiments involve ICMP protocol communication between the wireless devices. Whilst this type of information may be useful for the evaluation of the proposed methodology, this may not be representative of a real wireless network communication.

The authors of [74] also show that individual $RSSI$ fingerprints can be used to identify the presence of MAC spoofing attacks. The presence of separate $RSSI$ fingerprints generated by a single MAC address indicates the presence of MAC spoofing attack. The authors of [75] also make use of the $RSSI$, along with the geographical location of the wireless device, to detect the presence of identity spoofing attacks. After measuring the mean value of the $RSSI$ at multiple specific locations, the proposed methodology makes use of K-means clustering technique to identify the attack. The authors need to define the particular threshold value of the distance between centroids in the clusters to discriminate between non-malicious and spoofed information. The main drawback of this methodology is the assumption that the geographical location of the devices is accurately known. Additionally, the detection threshold needs to be empirically specified.

The authors of [76] make a critic against the mechanisms that use the sequence number and $RSSI$ to identify spoofing attacks. As an attempt to overcome the drawbacks of these methodologies, the authors propose an alternative methodology that associates the sequence number with some Quality-of-Service (QoS) parameters, with different wireless devices manufacturers, along with location parameters.

## 2.3 Detection Technique for Intrusion Detection Systems

The efficiency of the detection systems depends on the detection engine and the intrusion detection technique that it employs. Over the years, researchers and private companies have proposed and developed numerous techniques, which allow IDSs to analyse datasets and categorise the network traffic instances either as normal or malicious [77]. Some of the techniques would produce more efficient results than others, depending on different factors, such as the behaviour of the monitored system, the type of attacks, or the gathered metrics.

This section is not intended to describe in detail all the available detection techniques, but to give a brief insight into some of the most widely used intrusion detection techniques that could have been used in this thesis. For each of the techniques, the advantages that make them appropriate for intrusion detection, and the disadvantages that make these techniques inadequate for intrusion detection have been listed. The authors of [57] [58] indicate that the three most common techniques used by the anomaly-based IDSs are statistical methods, machine learning and data mining techniques. The most common techniques used by misuse-based IDSs are pattern recognition, implication rules and data mining techniques. Since the system proposed in this thesis is primarily an anomaly IDSs, this chapter pays special attention to the techniques used by this type of detection systems, and leaves aside the techniques mainly used by misuse IDSs.

### 2.3.1 Statistical-based Detection Techniques

Statistical-based IDSs are the most widely used systems [78]. Generally, these systems construct two models or profiles [58], along with an alarm threshold. The first of these profiles is the reference of normal network traffic behaviour, constructed during the training process or training phase. Statistical techniques are used to generate a stochastic model that represents the behaviour of previously gathered information [36]. The second profile is the statistical representation of the currently analysed information, during the detection process. The amount of data required to generate this profile should not be as predominant as for the profile of normal network traffic behaviour. In addition, the alarm threshold is a value that establishes the boundary between normal and malicious information [55].

Two of the main difficulties for statistical-based IDSs are the creation of accurate profiles and accurately comparing both profiles. Statistical-based IDSs compare both profiles and calculate the level of deviation. An intrusion is identified if the level of deviation between the two profiles exceeds the alarm threshold. The level of deviation is usually calculated based on a distance function [78] such as the Euclidean distance, Manhattan distance or Hamming distance. In frequently changing environments such as wireless networks, new statistical profiles for the current analysed information should be repeatedly generated. The reference of normality requires a good training process. Whilst it is not always necessary to implement the training highly frequently, the statistical profile of the currently analysed information requires to be constantly updated every time a new frame is gathered. Also, the procedure of defining the alarm threshold represents high difficulty [79]. The value of the alarm threshold has an important and direct impact on the performance of the detection system. If the alarm threshold was not large enough, the IDS would tend to generate False Positives (FPs). In contrast, if the alarm threshold was too large, the IDS would tend to produce a high number of False Negatives (FNs) [78] [80].

In [36], the authors highlight that statistical-based IDSs do not require prior knowledge about the normal behaviour of the protected system. Based on the assumption that the normal traffic is predominant in the network traffic, the model representing the behaviour of previously gathered information would represent, in turn, the behaviour of non-malicious network traffic. This property gives the statistical-based IDSs the ability to adaptively learn the normal behaviour of the protected network traffic [56].

However, this type of approach is not exempt of disadvantages. Statistical-based IDSs require gathering network traffic over a period of time [79]. If the time required for gathering the information is excessively large, detection could not be implemented in real time. Another disadvantage highlighted by the authors of [36] is that the reference of normality that these systems generate is susceptible to gradual modification over a period of time by the attacker. This modification causes an IDS to believe that the properties of the malicious information represent normality. In addition, [81] suggest that traditional statistical methods perform well when analysing homogeneous data measurements but their efficiency decreases when non-homogeneous data measurements are analysed.

#### 2.3.1.1 Counting Repeated Events

Not all statistical-based IDSs implement the intrusion detection similarly. A differential aspect of the statistical-based IDSs is the method employed to generate the reference of normality. In the intrusion detection literature diverse numbers of statistical methods have been proposed. A particular type of technique implemented by numerous IDSs is counting repeated events during specific periods of time.

For example, in [82], the authors present an intrusion detection framework based on statistical detection techniques to detect attacks at the MAC layer in WiFi networks and the presence of wireless devices not following the indication of the IEEE 802.11 standard. According to [55], this is a methodology able to detect, for instance, DoS attacks such as virtual jamming. The methodology presented in this work is based on detecting intrusions by counting events during periods of time. It establishes a set of alarm thresholds to some metrics. Then, the detection system calculates how often a particular event occurs during the defined period of time. The frequency value is compared to the alert threshold. If the frequency value exceeds the predefined threshold value, the attack has been identified and an alarm is flagged. The IDS needs to identify the evaluated event repeatedly before considering the presence of attack. Using a very similar methodology to the one implemented in [82], the authors of [37] present a framework that detects intrusions by counting events during a period of time and identify the attacks if the values of some predefined thresholds are exceeded. The authors of [67] also consider utilising a similar statistical detection technique to detect authentication and association frame flooding attacks. This work establishes an alarm threshold of 5 authentication or association frames per second. If the monitored AP receives more than 5 authentication or association frames with the same source MAC address within a second, the detection system considers the presence of a flooding attack. This particular work filters out any frame with source MAC address similar to the one detected as the source of the authentication and association flooding attack. Another example of an IDS that detect intrusions by counting events during periods of time is the one presented in [83]. This methodology sets a timer for each received deauthentication frame. When the detector has captured at least 2 consecutive deauthentication frames from a given MAC address within 6 seconds, it reports an alert. A different statistical detection technique also based on the same concept of counting events is mentioned in [36]. The IDS collects information during a given period of time, and counts the number of times some particular events have been repeated. If the number of occurrences of the events is too low, it is considered abnormal.

This intrusion detection methodology is very intuitive methodology and relatively easy to implement, and would not require a high computational cost for the detection engine. Although it is widely used, this method entitles fundamental drawbacks. After all, this detection methodology identifies attacks by counting the number of times that the protected system has been compromised during specific periods of time.

### 2.3.2 Data Mining

Traditionally, the security system administrators were in charge of manually generating the signatures utilised by the misuse IDSs to identify intrusions [15]. Similarly, security experts were in charge of manually the training datasets to train the supervised IDSs. Currently, the advances in data gathering technologies means that databases contain hundreds of thousands of data traffic records to be processed. The vast amount of data generated by current networks makes it impossible to perform manual analysis to detect network intrusions. A feasible solution to automate the data analysis process could be to employ data mining techniques.

Data mining is an approach that has gained wide interest among the research community in tasks of intrusion detection [36]. The concept of data mining describes a group of data analysis techniques used to automatically extract descriptive knowledge, identify previously unknown and useful information, predicting data relationships, and discovering behaviour patterns and trends from large amount of audited data [84]. These techniques can include different mathematical algorithms, statistical analysis and machine learning methods [85]. However, the use of data mining techniques mostly focuses on processing large amounts of audit data traffic rather than performing real time detection.

Many IDS administrators use data mining techniques to automatically generate intrusion detection models to be used by IDSs. This approach has become very useful to improve IDSs performance [86] because it can automate the analysis of large datasets and finding process of important information [58]. Many researchers have made use of data mining techniques to label training datasets and to train supervised IDSs [87]. In the last few years, the concept of data mining has been slightly diluted because, according to the authors of [36], now almost every processing mechanism able to analyse dataset is nowadays considered a data mining technique.

The advantage of applying data mining with IDSs lies in the fact that these techniques can automatically generate accurate intrusion detection models and signatures, which can be used to effectively distinguish between normal and abnormal network behaviours [88]. In the case of anomaly IDS, the data mining techniques can automatically create the reference of normal behaviour from the analysed information and implement the posterior intrusion detection. Similarly, data mining can be applied to automatically generate new signatures that will allow identify attacks or intrusion attempts, in the case of misuse IDS. However, almost every single data mining technique used on intrusion detection tasks requires, firstly, to have datasets for prior analysis, and, secondly, preprocessing of the dataset to produce acceptable results. The period of time required to gather the datasets is a drawback for real time systems. Also, most data mining techniques require training datasets [36], or require performing thorough training. Additionally, these techniques are generally computational intensive [58], and time consuming.

There exist many data mining techniques for intrusion detection. Based on the implemented task, these techniques can be divided into association rule analysis, sequence pattern analysis, classification analysis and cluster analysis [15] [88] [89].

### 2.3.2.1 Association Rule Techniques

The association data mining techniques were the earliest techniques used in task of intrusion detection [15]. This type of data mining techniques finds relations between data instances in the datasets [88] [90] and generates correlation logical rules in the form of *if* ~ *then* [89] from a training dataset by employing rule extraction algorithms. Each rule should be mutually exclusive to avoid classification conflicts. The outcome rules of the association analysis data mining techniques can be used as signatures by the misuse IDSs. One of the main advantages of this approach is its simplicity. Instead of manually generating the *if* ~ *then* rules, the association data mining techniques could automatically generate these rules.

One of the first research works that applied association data mining techniques in the task of intrusion detection was [90]. The detection methodology presented in this work requires labelled training datasets to generate the *if* ~ *then* rules. After a supervised training process, the authors generate association rules that allowed identifying intrusions. However, this type of technique is too simplistic to be able to detect the attacks currently implemented. One of the reasons is that, the sophistication of current attacks may cause that two different types of attacks manifest similarly to each other. Therefore, one association rule may incorrectly classify two different attacks into the same category.

### 2.3.2.2 Sequence Pattern Techniques

The different stages that wireless devices pass through during their normal protocol operations may be represented using state transitions. A feasible approach is using a Markov chain. The IEEE 802.11 standard defines these stages of normal operations. The Markov chain is a state transition approach that uses probabilities to model the chances of changing from one state to another. During a training process, this technique creates the different states and calculates the probabilities associated to each of the state transitions [36]. IDSs based on this technique, also known as sequence pattern IDSs, could use inconsistencies in the state transitions to recognise that the protected system is not behaving, as it should.

The sequence pattern techniques present a number of drawbacks when applied for intrusion detection in wireless networks. Initially, it may be understood that the IEEE 802.11 standard accurately defines the operations of the WiFi devices. However, different WiFi device vendors develop their own implementation of the IEEE 802.11 standard. Therefore, IDSs using sequence pattern techniques would require the exact configuration of each individual monitored device.

#### 2.3.2.3 Classification Techniques

The classification data mining techniques categorise instances of the dataset into different categories previously defined. These techniques produce a set of classification rules, generated after a supervised training process, that unmistakeably separate each category from the rest. Since the data mining techniques in this category are supervised, the training process requires the use of previously labelled datasets.

##### 2.3.2.3.1 Classification – Decision Tree

The decision tree is one of the most efficient classification techniques, according to [110]. The decision tree is a hierarchical structure used to classify data instances from a dataset with similar features into one of the defined categories [91]. Similar to other classification data mining techniques, the decision trees are constructed through a supervised training process, using previously labelled datasets [91]. Once the decision tree has been built, this structure can be used to classify new pieces of information into the predefined categories [92]. After applying the decision tree, each data instance is allocated into the particular category, which attributes are more similar to the attributes of the analysed data.

#### 2.3.2.3.2 Classification – Artificial Neural Networks

The Artificial Neural Network (ANN) is another classification data mining technique. The ANNs are computational models composed of a set of internal interconnected processing elements or nodes, also known as neurons [9] [93]. Each of these internal interconnections is assigned a particular weight value, which are dynamically adjusted according to a supervised training process [9]. Some ANNs may also modify their internal structure based on the supervised training process. The effectiveness of the ANNs depends on the quality of the supervised training process [40]. The different outcomes of the ANN define the different categories into which the analysed information is allocated. These outcomes are based on the characteristics of the nodes, as well as the weights associated with the internal interconnections [94]. Instead of allocating the data instances into one particular category, the neural networks provide the resulting weight value for each of the defined outcomes [93].

#### 2.3.2.3.3 Classification – k-Nearest Neighbour

Another classification data mining technique is the k-Nearest Neighbour (k-NN). This technique is used by [86] in tasks of intrusion detection. The classification process of k-NN is based on finding the closest match of an unclassified data instance to a group of k instances in a training dataset [95]. The instances in the group of k instances have similar label. Given a data instance, k-NN measures the difference between the features that describe this data instance and the features of the k different instances in the training dataset. The difference between features is usually measured using the Euclidean distance [85] [95]. The data instance is classified as the instances in the group that generates the closer match.

#### 2.3.2.3.4 Classification – Support Vector Machine

The Support Vector Machine (SVM) is an unsupervised classification data mining technique that generates a linear hyperplane over a dataset [13]. The SVM divides the instances of the evaluated dataset into two different categories, being the generated linear hyperplane the separation limit between both categories. During a training process, the SVM generates the best hyperplane that maximises the separation between the two classes in the training dataset.

## 2.3.2.3.5 Classification Techniques Drawbacks

The classification data mining techniques are more appropriate to be used by misuse IDSs, rather than anomaly IDSs [57], because of the functions that these techniques implement. In fact, the data mining techniques that belong to this classification category act mostly as network forensics analysis rather than intrusion detection in real time [96]. The classification analysis techniques present a number of drawbacks. First of all, these are techniques that require supervised training. Unless the system is provided with previously labelled datasets or datasets completely composed of normal traffic, it will not be able to create the appropriate classification rules to categorise the information. Unfortunately, in real wireless networks the data traffic is not labelled, and even in controlled environments it is highly complicate that all the data is of non-malicious nature [87]. This makes the use of these data mining techniques impractical. Also, if a new feature of the data instances needs to be analysed, a new training process needs to be executed to generate an updated set of classification rules, because it cannot be categorised as any of the previously known categories.

In the case of a decision tree, the main complexity resides in the training process and defining an appropriate stop point in the construction of the tree [93], to avoid overtraining. This means that an excessive number of classification rules have been generated. The major drawback of the ANN is the high computational cost [9] and time consumption associated with the training process. These systems require large amounts of data to be properly trained [98]. For instance, the authors of [99] indicate that the ANNs require collecting training data for several days, before training the system, and perform the detection. The authors of [97] also highlight that the ANNs require a large period of time to train, due to its complex internal structure. In the case of k-NN, whilst being effective in the tasks of intrusion detection, this technique has a high computational cost to conduct the classification [85]. Also, the classification performance of k-NN is very sensitive to the value of the parameter k [85], which needs to be predefined. In the case of SVM, the major drawback is the long training time required that limits its use in real time approaches. Also, because SVM is an unsupervised technique, the system administrator needs to define each of the classes after generating the hyperplane.

### 2.3.2.4 Clustering Techniques

Clustering has been commonly used as a data analysis mechanism, mostly focused on data visualisation, image analysis and pattern recognition. The clustering data mining techniques can also be used in tasks of intrusion detection to categorise network traffic instances as normal or malicious [88]. Numerous researches have used clustering in IDSs [24] [29] [91] [100].

Clustering provides an effective mechanism to automatically construct a profile of normal network behaviour that can be used to categorise the network traffic as normal or malicious [89]. Unlike other data mining techniques (e.g. Decision Tree), clustering is an unsupervised [101] technique that can be used to classify unlabelled instances, without previous training process or having previously labelled the instances of the datasets. This technique creates groups of data instances from a given dataset based on the similarity of their attributes. Instances within the same cluster have high similarity, whereas the attributes of the instances within a cluster have significant degree of dissimilarity with the instances in other clusters.

The main advantages of using clustering are its capability to identify previously unknown evidences of attacks, as well as variations of known attacks [15]. IDSs that make use of clustering are more efficient than other techniques when implementing intrusion detection in real time, and require minimal interaction with the IDS administrator [91] [100].

#### 2.3.2.4.1 K-means Clustering Drawbacks

There exist different algorithms of clustering techniques. One of the simplest and most commonly used clustering algorithms is the K-means [91]. This algorithm divides the data instances from a given dataset into K different clusters. The specific value of K has to be specified prior to the implementation of the division process.

Despite of its effectiveness, K-means presents also some limitations. One of these limitations is the significant influence that the selection of the initial cluster centroids has over the final results. The selection of different initial centroids may lead to different clustering results in similar datasets. Also, the specific number of clusters K into which K-means divides data instances has to be specified prior to the implementation of the division process. However, this number may not be initially available and needs to be empirically defined, which is a major drawback for this technique to work in real time. In addition, K-means requires the empirical definition of the parameter μ for correctly labelling the data instances as normal or malicious. This parameter is used to define the threshold between normal and malicious clusters.

## 2.4 Multi-layer Data Fusion

Previous approaches on cross-layer/multi-layer data fusion have used simplistic combination techniques, such as averaging or majority voting [102], which does not require training or complex calculation. Data fusion can be defined as the process of gathering information from multiple and heterogeneous sources about diverse events, activities or situations, and combining them towards obtaining a more accurate final result [14] [39]. In [103], a study of the benefits and limitations of the cross-layer designs in the field of intrusion detection is presented. This research compares two different IDS architectures. Another interesting study is presented in [38], which experimentally compares the detection performance of cross-layer IDSs with the performance of single layer IDSs. The authors of [104] propose a novel collaborative detection approach that combines the final outcome of multiple IDSs. Similar approach is used in [80], in which the authors propose the use of neural networks to combine the outcome of three different IDSs.

In the field of intrusion detection, the combined utilisation of multiple metrics, extracted from a diverse number of sources is a very important step towards an efficient intrusion detection process. Previous researches [105] [106] use data fusion techniques in order to enhance the performance of their systems. The information from different layers could be sent to a shared database, and an IDS could infer the presence of intrusions using this common set of information. This is the structure that a centralised IDS would implement. In another cross-layer architecture, IDSs can also implement the task of intrusion detection independently, utilising information from only one source of information. Then, the decision of each individual IDS could be sent to a data fusion system to merge the individual decisions and reach a combined final result. In [80], the authors present a method that combines the outcome of multiple IDSs using data fusion techniques. Each IDS conducts independent detection processes and the individual decisions are combined to produce a collective final decision. Similarly, in [106] the authors present an approach that combines the outcome of three heterogeneous IDSs. In this particular work, the three classification data mining techniques are decision tree, Naïve Bayes [13] and ANN. In fact, a double data fusion process is implemented. The three detection techniques are initially used to analyse three independent features. The fusion outcome for each individual of the piece of information is initially combined. Then the three outcomes are successively combined to reach a final detection result.

## 2.4.1 Researchers Using Dempster-Shafer Theory

The application of D-S theory for improving the performance of IDSs is a very active research topic. This data fusion method has been previously used in multiple publications. In [102], one of the most thorough descriptions of D-S is presented. The authors present a comparative study between D-S theory and Bayesian inference as data fusion algorithms. Another complete description of the D-S theory is presented in [75]. The mathematical foundation of D-S theory, along with a description of the advantages and disadvantages of this data fusion approach is presented in this work. In [39], the authors present a prototype for DDoS detection over wired links, based on D-S theory. The system, periodically, fuses the knowledge collected from different sensors within the network, in order to infer the current state of the monitored network. The authors in [41] present and evaluate an IDS for detecting DoS attacks in a wireless network. The authors use the D-S theory to fuse the information from distinct nodes running two different algorithms. In [39] [102], the authors present a comparative study of different data fusion methods and conclude that D-S theory is more promising than Bayesian inference.

## 2.4.2 Current Basic Probability Assignment Methodologies

The BPA process is crucial to the effectiveness of D-S theory [107]. The BPA value should be based on the measured characteristics of the monitored environment. With regards to the topic of this thesis, the major challenge for applying D-S theory on IDS is to automatically determine the BPA values should be based on the characteristics of the wireless network traffic measurements [42].

In the IDS literature there exist multiple ways of assigning probabilities to each of the hypotheses in D-S theory, ranging from data mining techniques to empirical approaches. For instance, [108] utilises expert opinion to manually assign the belief probabilities to each of the hypotheses. This BPA process is completely subjective and might not be adequate for automatic and self-adaptive IDSs. The authors in [41] present a methodology that seeks changes in the Signal-to-Noise Ratio (SNR). The value of this single metric is measured from distinct nodes running two different local algorithms, single threshold and cumulative sum. Based on the measured information, their system generates the BPAs through the use of a linear function. One of the drawbacks of this methodology is that both local algorithms require the utilisation of diverse tuning parameters. In [98], the authors also present an IDS that make use of the D-S theory. In order to assign the BPAs, this work defines a specific equation based on the utilisation of thresholds. Apart from using fixed functions, which is a limited mechanism to calculate the beliefs, the authors do not provide either an explicit definition of how the thresholds are calculated, or a clear explanation of how the BPAs are assigned. A similar approach is used in [39], which uses fixed functions to define the BPA value after a fine-tuning process. Another example, [75] proposes two different ways of assigning belief probabilities, for two different datasets. In the first case, their method calculates a threshold based on the length of the dataset, and then utilises that threshold and fixed functions to assign the belief probabilities. In the second case, a scaled approach with pre-defined beliefs is used. Since the beliefs are assigned using fixed functions and pre-defined thresholds, the mechanisms proposed in this work would be unable to automatically adjust to changes in the dataset profile without the intervention of the IDS administrator. In [40], the authors use multiple manually defined thresholds, empirically defined after analysing non-malicious data. The authors do not describe the way the thresholds are defined. One last example, the methodology employed by [42] uses data mining to proceed with the BPA tasks.

From the presented results, all of these methods are effective in increasing the DR and reducing the number of false alarms of the IDSs. However, none of the referred works investigate methods to find an automatic and self-adaptive process of BPA, and few of them could be used off-the-shelf without a previous training or fine tuning period. On the one hand, systems that make use of data mining techniques for BPA require the gathering of large amounts of data traffic, processing it and to complete a training period before being able to perform intrusion detection tasks. These systems are unable to automatically adapt to changes in the network traffic behaviour in real time. On the other hand, systems that have been empirically assigned fixed probability values by the IDS administrator, or systems that employ fixed functions to assign the belief probabilities are unable to automatically adjust to changes in the network traffic behaviour, without the intervention from the IDS administrator.

## 2.5 Wireless Network Monitoring

The task of wireless network traffic monitoring has been addressed in [8] [44]. The authors of these works present extended researches about different issues in deploying wireless network monitoring systems, and the process of implementing the actual monitoring of the IEEE 802.11 networks. Diverse concepts, such as the advantages of wireless monitoring, placement of the monitoring devices or merge of information from multiple monitoring devices, are all addressed in these papers. The authors of [83] describe a detection system in WiFi networks. The approach followed in this research is to monitor all possible transmission channels using numerous monitoring devices. Then the information is merged to proceed with the detection analysis.

In [44], the authors discuss the correct number of monitoring sensors. This is whether to deploy a large number of sensors or a reduced number of these devices, close to the different devices in the protected wireless network. This work argues that deploying a reduced number of sensors is harmful for the correct monitoring of the network traffic, because of the severe network traffic measurement loss that the wireless sensors might experience. The authors of [116] highlight that traffic loss is a critical occurrence that could yield to undetected attacks. According to the authors of [44], multiple sensors would reduce the amount of traffic loss. The authors of [83] aggress that multiple sensors should be deployed to reduce or avoid traffic loss.

## 2.6 Summary

As it has been presented in this chapter, there are multiple research works that address the insecurity of IEEE 802.11 networks. These works mainly focuses on describing vulnerabilities of the wireless communication protocols, proposing enhanced security protocols, and describing attacks. Similarly, there are several published research works in the field of intrusion detection that propose more effective IDS approaches, and novel detection techniques to increase the detection efficiency of IDSs. A brief description of some of the most widely used intrusion detection techniques has been presented in this chapter. Special attention has been given to approaches designed to identify spoofing attacks in WLANs, and to techniques mainly used by anomaly IDSs. These are statistical methods, machine learning and data mining techniques. A brief description of some of these intrusion detection techniques has been presented, along with a list of some advantages and disadvantages.

The classification data mining techniques are more appropriate to be used by misuse IDSs, rather than anomaly IDSs. The classification analysis techniques present a number of drawbacks. Most of the described data mining techniques are supervised, and require extensive training process, as well as labelled datasets, to be effective. Unless the system is provided with previously labelled datasets, it will not be able to create the appropriate classification rules to categorise the information. This makes the use of these data mining techniques impractical for IDSs designed to implement the intrusion detection autonomously and in real time. Clustering is an unsupervised technique that can be used to identify previously unknown attacks without previous training process. This makes clustering an adequate technique when implementing intrusion detection autonomously. However, this technique also presents some limitations, such as the significant influence that the selection of the initial cluster centroids has over the final results. Statistical-based detection techniques are also adequate when implementing intrusion detection autonomously. These techniques are unsupervised and do not require prior knowledge about protected system. Traditional statistical methods perform well when analysing homogeneous data. However, their efficiency decreases when non-homogeneous data is analysed.

The chapter concludes discussing systems that use of multi-layer approach and data fusion methodologies as part of the detection system. Among different data fusion methods, special attention has been given to the D-S theory. A description of the different methods to determine the BPA values present in the IDS literature has been presented. Few of these methods could be used autonomously, off-the-shelf without either prior thorough training or fine tuning period, and able to automatically adapt to changes in real time. Therefore, this chapter highlights the need for the implementation of a novel BPA methodology able to automatically adapt its probabilities assignment to the current characteristics of the wireless network, without intervention from an IDS administrator.

# Chapter 3

## Protecting Against Wireless-Specific Attacks

### 3.1 Introduction

There exist different types of wireless-specific attack that can compromise wireless networks. Despite the diversity of these attacks, their objective can be still the same for different attacks. Similarly, there exist several mechanisms to protect wireless networks against these attacks. Some of these mechanisms will be better than others in protecting certain components of the wireless network or protecting against certain types of attacks. In some cases, a particular security mechanism will be the only mechanism able to effectively protect the wireless networks against a certain type of attack, or a particular intrusion attempt.

If the attacks that the wireless network is being protected from were known beforehand, a security mechanism that provided better protection against this particular attack could be implemented. In that case, the wireless network would suffer the least damage or even no damage at all. Unfortunately, this is nearly impossible for legitimate users and the security mechanisms of the network. Only the attacker knows the particular attack that will be launched. The devices in the wireless network and the security mechanisms are unable to know which attack will be launched until this had already been occurred.

Occasionally, some of the countermeasure security mechanisms that have been proposed to protect the network against certain type of attacks could produce undesirable results or even further damage to the protected network. For instance, the use of access control lists can prevent unwanted wireless devices from joining the wireless network, but the access control lists can also prevent legitimate wireless devices from joining the network if their identity has been compromised. Hence, a security mechanism to provide consistent security to the wireless networks, regardless of the attack that these are being protected from should be implemented.

This chapter presents an overview of a number of wireless-specific attacks. These are the attacks that more commonly compromise wireless networks. The purpose of this chapter is to find, if possible, a common implementation pattern, which could help to identify a common detection or countermeasure mechanism against these attacks. Prior to describing the attacks, a brief and simple description of the security protocols recommended by the IEEE 802.11 standard is given. The purpose of this description is to make the reader aware of the efforts that the IEEE 802.11 standard has made to safeguard the confidentiality, integrity and availability of the wireless network. In addition, some of the traditional network security mechanisms are also described.

## 3.2 Traditional Wireless Network Security Mechanisms

### 3.2.1 The IEEE 802.11 Security Mechanisms

A complete description of these security protocols is out of the scope of this thesis. For a more detailed description of these protocols, refer to the IEEE 802.11 standard definition [111], or other works such as [5] [10] [112].

The IEEE 802.11 standard proposed different security protocols, establishing a traffic encryption framework at the MAC layer and integrity protection to the network infrastructure [5], as well as avoiding unauthorised access to the wireless networks. These are the Wired Equivalent Privacy (WEP), the WiFi Protected Access (WPA) and the WPA2 (WPA Version 2), also known as the IEEE 802.11i standard. These security protocols provide a framework for communication encryption, devices or user authentication, and communication content integrity protection.

WEP was the first of the three security protocols defined by the IEEE 802.11 standard. This protocol provides confidentiality protection using the stream cipher protocol Rivest Cipher 4 (RC4) to encrypt the frames. The devices at both ends of the wireless communication must share the same secret key to communicate. In order to stop undesirable devices from joining the wireless network, WEP implements two different authentication mechanisms, the Open System Authentication and the Shared Key Authentication. Additionally, WEP enhances the integrity protection of the wireless communications by including Cyclic Redundancy Check (CRC) checksum protection to the encrypted frames.

WPA provides confidentiality protection to the wireless communications using the Temporal Key Integrity Protocol (TKIP), a new encryption protocol based on the stream cipher protocol RC4 [10]. Similar to RC4, the TKIP protocol requires a shared secret key to protect the transmitted information [5]. WPA supports two different authentication mechanisms, the Shared Key Authentication and the IEEE 802.1x Server Based Authentication. Additionally, WPA also enhances the integrity protection to the wireless communications including Message Integrity Codes (MICs), replacing the previously used CRC checksum [5].

WPA2, also known as the IEEE 802.11i standard, is the security protocol most recently recommended by the IEEE 802.11 standard to provide authentication, confidentiality and integrity protection to the wireless communications. The IEEE 802.11i standard provides stronger cryptographic protection than WEP and WPA, using the Advanced Encryption Standard (AES) algorithm [113]. Additionally, the IEEE 802.11i protocol also specifies the use of TKIP as encryption protocol, similar to WPA. In order to stop undesirable users from joining the wireless network, WPA2 uses the IEEE 802.1x server based authentication mechanism [10].

#### 3.2.1.1 The IEEE 802.11 Authentication Mechanisms

Authentication can be considered as the first security barrier that the wireless devices have to surpass before joining the wireless network. Every wireless device willing to join a network uses the authentication process to identify itself to the network. The wireless network relies on this identification process to ensure that only the authorised devices can join the network. Authentication occurs every time a user tries to join a wireless network. Each wireless device can be authenticated with multiple APs at the same time [59]. There is not restriction on the number of APs that a single device can be authenticated with.

The IEEE 802.11 standard defines three different authentication mechanisms. These are the open system authentication, the shared key authentication and the IEEE 802.1x server based authentication [70]. The former is a two-step process that grants authentication to any wireless device that requests so. The shared key authentication is a four-step process that requires the use of common cryptographic material between the AP and the wireless device willing to join the network. If both parties in the communication share the same secret key, the wireless network grants the authentication. The AP uses the source MAC address in the request frames to determine the identity of the wireless device, in both open system authentication and shared key authentication. The downside of these mechanisms is that none of the two implement mutual authentication between both ends of the communication. The wireless devices are authenticated to the AP but these nodes have not the capability to authenticate the identity of the AP. In addition, both frames involved in authentication, authentication request and authentication response, are sent in the clear, without any encryption mechanism [114].

The IEEE 802.1x server based authentication mechanism is the third authentication mechanism recommended by the IEEE 802.11 standard. In contrast to the previous mechanisms, IEEE 802.1x provides mutual authentication to both parties of the wireless communication [10]. In this authentication mechanism, an authentication server (e.g. a RADIUS server [115]) is the responsible to manage the authentication process between the AP and the wireless users [70]. The authentication server decides whether to grant authentication or not, based on the credentials of the wireless users requesting authentication. The actual authentication method is implemented using the Extensible Authentication Protocol (EAP), which provides mutual authentication to both, the AP and the wireless users [70].

#### 3.2.1.2 The Insecurity of IEEE 802.11 Security Protocols

The described security protocols were supposed to make the WLANs more reliable and assure cryptographic protection to the transmitted information that traverses the wireless medium. The utilisation of these security protocols forces the attacker to crack the cryptographic material and decrypt the protected communication in order to access to the transmitted information. It is down to the efficiency of these security protocols and how well implemented these protocols are, to prevent an attacker from decrypting and accessing to the content of the frames.

However, wireless networks cannot rely on these security protocols to protect the content of the communications. All the security protocols previously presented are vulnerable to decryption analysis processes. If the attacker is able to obtain the cryptographic material used to encrypt the wireless traffic, the attacker is free to monitor the entire communication in the wireless network [116]. WEP is probably the most unsecure security protocol that could be used to protect the WLAN communications. WEP suffers from well-documented security vulnerabilities [49] [55]. All these security vulnerabilities have been identified and demonstrated in many research publications [8]. There are numerous software tools able to crack the cryptographic material used in WEP or WPA. An attacker using the right tools can easily obtain the shared secret key in a matter of minutes [7]. Additionally, the author of [10] indicates that frames can be modified despite including integrity protection such as CRC checksum.

The IEEE 802.11i standard was proposed to address the vulnerabilities experienced by WEP [112] and WPA, providing stronger cryptographic protection than its predecessors. Although WPA2 has not been cracked until now, researchers from Air Tight Networks found the so-called ‘Hole 196’ vulnerability in the security protocol WPA2 that allows malicious devices with knowledge of the Group Temporal Key (GTK), to transmit spoofed group addressed data frames to other wireless devices in a WPA2-protected WiFi network [118]. This vulnerability exposes the legitimate users of a WPA2-protected WiFi network to a diverse number of attacks, such Man-in-the-Middle (MitM) or DoS. Explained by the same researchers, this vulnerability is also presented in the WPA security protocol.

Moreover, the security protocols do not encrypt the whole communication of the network. All the management and control frames are transmitted unauthenticated and unencrypted [114], leaving IEEE 802.11 networks vulnerable to numerous type of attacks [119]. The IEEE 802.11w standard is the first and only amendment of the IEEE 802.11 standard to include protection to the management frames [67] [70]. Additionally, the header of the frames is never encrypted. Besides, regardless of which security protocol is used, an attacker can easily compromise the availability and integrity of the wireless networks. For instance, none of these protocols can protect the wireless communications certain type of attacks, such DoS [5] [113], which directly compromise the availability of the networks.

The utilisation of authentication mechanisms or encryption techniques certainly helps to secure the network, making the implementation of attacks or intrusion attempts more difficult to some extent. However, none of these security protocols have been able to effectively and completely protect the wireless networks. Despite the different amendments of the IEEE 802.11 standard that have been released to provide more secure and reliable security protocols, still there exist numerous security vulnerabilities in the protocols that makes the wireless communications insecure [119]. The level of security of the wireless communications can also be enhanced at higher layers using end-to-end encryption mechanisms, such as Internet Protocol Security (IPSec) or Virtual Private Network (VPN). However, according to [5], these mechanisms cannot protect the network from the wireless-specific attacks.

Due to the capabilities of the attacker to surpass the security protocols, the feasibility to crack and obtain the cryptographic material, and the numerous vulnerabilities in the security protocols, it is evident that cryptographic protection cannot stop the attacker from carrying out the attacks and intrusions in WLANs. Therefore, all the wireless-specific attacks that are going to be described in the following Section 3.3 are described assuming that the attacker has complete access to the transmitted information, and assuming that the attacker is able to modify the content of the network traffic without being noticed.

### 3.2.2 Network Firewalls

One other security mechanism that is commonly employed to provide protection to the networks is the firewall. A firewall is a mechanism that controls the traffic that is allowed to pass through it. Firewalls can be utilised as security systems to protect local networks from Internet-based attacks and control the access to the network [79]. These network security systems are commonly allocated between the local network and the ISP backbone, which allows the network nodes to communicate with other network nodes through the Internet.

Firewalls filter network traffic based on a set of rules and known access control policies. Information extracted from the header of the network frames is analysed by the firewalls. One or more pieces of information, e.g. source/destination address or source/destination port, are used to determine whether or not to stop the frames [120].

Firewalls provide a level of security to the local networks, but these systems are not effective security mechanisms for WLANs, and are unable to provide perfect protection to these type of networks against attacks [120]. Although many attacks from the Internet can be stopped by a properly configured firewall, not all the attacks from outside the network can be stopped by firewalls due to the difficulty in generating correct and precise filtering rules. Firewalls are unable to protect the network against attacks that have bypassed the system [79]. Another major drawback is the fact that firewalls cannot protect the network from attacks originated inside the network, because these attacks do not pass through the firewall. Additionally, the use of VPNs and wireless networks introduce new challenges to firewalls. The VPNs utilise encryption protection in both the header and the payload of the transmitted frames, making impossible the analysis of the header of the network frames. Besides, the lack of physical boundaries in the wireless networks communication allows the attacker to directly interact with the wireless network users without passing through the firewall.

### 3.2.3 Antivirus Software

According to [55], an antivirus is a software used on PCs to identify viruses, malware and spyware files, and to prevent the effect that these files may cause to the protected system. This security software uses signature databases to identify the malicious files. These are commonly designed to monitor critical components of the Operating Systems (OSs) and to disinfect the malicious files [55]. Antivirus alone is not enough to provide reliable protection against current Internet threats [11]. The main drawback of these systems is that an antivirus is unable to identify malicious files for which signature has not been defined. These systems rely on the frequent update of the signature databases to be efficient.

## 3.3 Wireless-Specific Attacks

Wireless networks are vulnerable to similar threats to the wired networks. Cyber-attacks and intrusion attempts that come from the ISP backbone core of the network, and reach the devices connected to the networks, affect similarly both type of networks. These attacks can be implemented remotely from any geographical location. Apart from being vulnerable to viruses, malware, spam, Internet-based attacks, DoS, DDoS, or identity theft attacks, the wireless network users suffer also from wireless-specific attacks. These are attacks that specifically affect the wireless networks.

In contrast to the cyber-attacks that come from the wired part of the network and reach the wireless devices through the AP for WiFi networks (eNodeBS for LTE and Base Station for WiMAX), the wireless-specific attacks are attacks that reach the wireless devices through the wireless part of the network; the wireless link. The attacker acts as a third party transmitter device to launch this type of attacks. The wireless-specific attacks require the attacker to be located in a position in which the victim will be within its transmission range. Similarly, the attacker should be located within the transmission range of the victim in most of the attacks. However, there are some attacks that can be implemented with the attacker being off the transmission range of the victim.

The following sections describe the most common type of wireless-specific attacks that the wireless communications and wireless networks infrastructure. After performing an analysis of these attacks, it is easy to conclude that there exist three main characteristics that make the implementation of the wireless-specific attacks feasible. First, the ability to intercept and analyse the wireless communications content. Second, the capability to inject malicious information into the wireless communication. Third, the capability to conceal the identity of the attackers to avoid being identified when launching attacks.

## 3.3.1 Eavesdropping

Eavesdropping is the process through which attackers and unauthorised users illegitimately intercept and analyse transmitted information within a wireless network. Any device within the transmission range of the wireless network, with the capability to monitor the wireless medium, could access the transmitted information. An attacker could intercept the transmitted frames, perform an analysis of the content of these frames, and extract information about different parameters of the wireless network.

In order to implement the eavesdropping attack, the attacker requires a receiver device able to monitor the traffic in the wireless network or wireless NIC configured in monitoring or promiscuous mode. The attacker does not actively interact with the wireless network traffic at all. The fact that eavesdropping does not actively interact with the wireless network traffic makes this attack impossible to detect [116]. Although the eavesdropping attack only compromises the confidentiality of the transmitted information, this attack is an initial step towards more dangerous attacks. Usually, the interception of information about the wireless network and the communication content is a prerequisite for the attackers to perform more severe active attacks over the wireless network.

The only countermeasure against eavesdropping is making use of reliable cryptographic mechanisms. However, as explained previously in Section 3.2, all the cryptographic mechanisms proposed by the IEEE 802.11 standard are vulnerable to decryption analysis processes. Eavesdropping is also a feasible attack because the wireless communications are not entirely protected by cryptographic mechanisms. The payload of the data frames is usually encrypted but the header of the data frames is never encrypted, for delivery reasons, exposing all the information to be analysed by an attacker. From the data frame header, an attacker can obtain, for instance, the MAC addresses, frames type and subtype, or Network Allocation Vector (NAV) value. On top of that, the management frames and the control frames are entirely unencrypted [59]. Therefore, the attacker would be able to intercept lots of information, even if the legitimate user made every possible effort to protect the wireless communication.

Eavesdropping does not only allow the attacker to obtain information directly from the content of the frames. The information can also be composed of different statistical parameters from the characteristics of the wireless network and the wireless communication, in the form of Netflow data. Much more difficult to protect than the content of the frames is to hide the information that an attacker could extract from the characteristics of the wireless network communication, such as the signal power, transmission timing patterns, number of wireless devices in the network, or periodicity of the transmission. Despite utilising cryptographic mechanisms to protect the wireless communication, the legitimate user cannot hide all these parameters from the attacker.

### 3.3.2 Active Attacks

The term active attack represents all the attacks that actively interact with wireless communications, illegitimately injecting traffic or electromagnetic signals into the wireless medium. The attacker needs also to be able to create fully formed IEEE 802.11 frames or modify previously captured frames and inject them into the wireless network. Moreover, the attackers should be able not to follow the indications of the IEEE 802.11 standard, and to inject whenever is needed.

#### 3.3.2.1 Denial-of-Service

Denial-of-Service (DoS) is one of the most common types of active attacks used against wireless networks. Also, it is the most harmful attack that the wireless networks may suffer [60]. This type of attack actively prevents legitimate devices in the network from successfully transmitting or receiving any frame. DoS attacks target the availability and the integrity of the legitimate communications in the wireless network, and can be launched at multiple network layers of the protocol stack. For this work, only the DoS attacks that belong to the category of wireless-specific attacks are considered. These are the DoS attacks that can be launched either at the PHY layer or the MAC layer of the protocol stack. None of the cryptographic mechanisms can protect the wireless networks from this type of attacks [5] [113].

##### 3.3.2.1.1 Physical Jamming

Physical jamming is a particular type of DoS attacks launched at the PHY layer that exploits the functionality of the physical carrier sensing functions of the IEEE 802.11 standard, in the legitimate users. By injecting a suitable level of radio frequency noise into the wireless medium, an attacker is able to make the wireless network devices conclude that the wireless medium is being used. The legitimate devices defer any transmission if the physical carrier sensing functions detect signal activities in the medium. Therefore, the wireless network devices affected by the physical jamming are prevented from transmitting, and the IEEE 802.11 standard does not provide any mechanism that could stop it.

In [61], the authors state that there is no conventional security mechanism able to entirely protect wireless networks from physical jamming. Every wireless device within the transmission range of the attacker is vulnerable to this attack. Moreover, physical jamming can be implemented even if the attacker is not a member of the wireless network [63]. On the other hand, this type of attack can be detected by examining the wireless medium with a signal analyser. The attacker only requires a transmitter device able to inject electromagnetic noise signals or a wireless NIC able to inject illegitimate traffic into the wireless medium. The wireless NIC does not follow the specifications of the IEEE 802.11 standard and the MAC protocol.

There exist four different types of physical jamming attacks. These are brute force, periodic, random, and precision jamming.

## 3.3.2.1.1.1 Brute Force Jamming

Brute force jamming is the most simple of the physical jamming modes. The attacker continuously injects radio frequency noise into the wireless communications and disrupts any communication. Legitimate users are prevented from transmitting, as well as receiving frames.

The effectiveness of brute force jamming is strongly correlated with diverse parameters, such as the used frequency channel and the transmission power level of the injected noise. Also, the attackers are able to directly target specific users using directional antennas. The authors in [37] have empirically demonstrated that the distance between the attacker and the legitimate wireless nodes is a parameter directly correlated with the effectiveness of any mode of physical jamming attack. The shorter the distance between the attacker and the legitimate users, the higher the effectiveness of the attack.

This attack can be easily detected using wireless medium monitoring tools, such as signal analysers. There have been proposed diverse methods as countermeasure actions against brute force jamming. For instance, increasing the signal strength of the legitimate wireless communications could reduce the effect of this type of jamming attack on the communication. The physical carrier sense mechanism would consider the transmitted signal of the attacker launching the brute force attack as background noise if the legitimate wireless device uses stronger signal strength than the attack signal. This would practically depend on the receiver sensibility. Another countermeasure method would be to raise the threshold that determines whether there exists any transmission in the wireless medium. Consequently, the legitimate users in the wireless network would be able to initiate a communication despite the presence of activity in the wireless medium. The authors of [62] demonstrated that the wireless transmission modulation plays an important role in the effectiveness of a jamming attack over the wireless network. Therefore, utilising the most appropriate modulation to each attack scenario could be a possible countermeasure technique. Dynamically negotiated radio frequency hop sequences is also proposed as countermeasure technique by [62]. Nevertheless, the authors of this work also explain that radio frequency hop sequences is not appropriate because informing all stations to change to a specific frequency during a DoS attack would be unfeasible. One last proposed countermeasure technique is the use of spread spectrum techniques [123]. All these countermeasure actions could reduce the effect of the brute force jamming attacks.

## 3.3.2.1.1.2 Periodic Jamming

Another modality of physical jamming is periodic jamming. Instead of continuously injecting radio frequency noise, the attacker intermittently injects high levels of interference into the wireless medium during short and periodic intervals. The attacker alternates between constant injection and silence periods.

This attack may prevent legitimate network users from transmitting and receiving frames. However, periodic jamming is less efficient than brute force jamming. Since the attacker alternates injection and silence periods, some wireless users would be able to continue the communication during the attack silence periods. The duration of both periods would affect the effectiveness of the periodic jamming. The authors of [122] have demonstrated that attackers using shorter silence periods cause greater impact over the communication than attackers using larger silence periods.

Detecting the presence of periodic jamming is more difficult than detecting brute force jamming. The periodic behaviour of this attack reduces the exposition time of the attacker to be detected by monitoring tools. Nonetheless, if it was detected, the monitoring tools could deduce the periodicity of the injection periods and circumvent the attack transmitting during the silence periods of the attacker.

The authors of [64] demonstrated that transmitting small frames are more probable to be successfully delivered than large frames if a periodic jamming attack is taking place. Then, one feasible countermeasure mechanism against this attack would be reducing the size of the transmitted frames. In addition, the countermeasure mechanisms proposed previously for brute force jamming are also applicable against other physical jamming attack modalities.

## 3.3.2.1.1.3 Random Jamming

Random jamming is another modality of physical jamming in which the attacker arbitrarily injects high levels of interference into the wireless medium, for short intervals. The duration of the injection and silence periods in random jamming is arbitrary. Similar to the periodic attack, the larger the injection period, the greater is the damage that the attack produces over the communication.

The random behaviour of this attack reduces the exposition time of the attacker to be detected by any monitoring tool. Detecting the presence of random jamming is more difficult than detecting brute force or periodic jamming. Even if a monitoring tool is able to detect the presence of the random jamming, the random pattern behaviour of this attack makes circumventing the effect of attack extremely difficult. The countermeasure mechanisms proposed for brute force jamming and periodic jamming could still be utilised.

## 3.3.2.1.1.4 Precision Jamming

In precision jamming, the attacker targets specific components of the legitimate wireless communication. These components could be either frames transmitted to/from specific users, certain types of transmitted frames, or specific interframe space times. The attacker remains silent while monitoring the wireless medium and injects as soon as it detects a legitimate transmission.

Overall, precision jamming could be the most effective mode of physical jamming, in terms of exposition time to be detected by any monitoring tool, and effectiveness compromising the availability and integrity of the wireless network. Again, this attack may prevent legitimate users from starting transmitting or prevent the destination users from receiving frames. Since the attacker injects radio frequency noise during short period of time at the precise moment, this attack is very difficult to detect. On the other hand, precision jamming requires a more complex functionality from the attacker, which requires knowing the MAC protocol and needs to predict the precise moment that it has to inject. The countermeasure mechanisms proposed for the previous modalities of physical jamming could be similarly utilised to reduce the effect of the precision jamming attacks.

#### 3.3.2.1.2 Virtual Jamming

Virtual jamming is another type of DoS attack. This attack, launched at the MAC layer, exploits the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol and the virtual carrier sensing function of the IEEE 802.11 standard, in the legitimate users. The virtual jamming attack is based on the fact that the wireless network users defer any transmission if the virtual carrier sensing function indicates that the medium is being occupied.

Assuming the capability of an attacker to create and inject fully formed IEEE 802.11 frames, an attacker can create frames with a very high NAV value. By injecting frames with NAV set to its maximum value (32767 µseconds [114]) an attacker is able to make all the wireless devices in the network postpone any transmission. An attacker can use all MAC control frames (e.g. RTS), MAC management frames (e.g. Probe Request) and data frames to launch this attack. An attacker that persists in injecting these frames with high NAV value can stop the communication of the entire wireless network. In [51], the authors estimate that injecting 30 crafted control frames a second with the maximum NAV value is enough to stop the whole wireless communication.

The effectiveness of this attack varies depending on the targeted objective. The authors in [61] have demonstrated that if the attacker targets the AP, the average transmission rate in the entire wireless network is deteriorated. However, all the wireless devices could continue communicating during the attack. On the other hand, if the attacker injects the control frame destined to one specific wireless device, all the components in the network are unable to communicate. Unfortunately, the authors of this work do not explain the reasons of these dissimilar results, when the targeted device changes. Additionally, the authors in [62] also explained that if the crafted control frames are destined to a non-existing device, all the components in the network are also unable to communicate. In that case, the authors concluded that the cause of this behaviour resides in the fact that the wireless devices do not properly implement the indications of the IEEE 802.11 standard. Moreover, the effectiveness of virtual jamming is also related to the type of control frame used. According to [61], RTS frames produce more effective attack results than CTS and ACK frames. By using RTS frames, the wireless users propagate the effects of the attack to further number of users, which are outside the transition range of the attacker. The same propagation effect cannot be achieved using either CTS or ACK frames.

One possible countermeasure mechanism against virtual jamming attack could be filtering out the crafted control frames. This could be implemented by using access control lists. The access control lists are based on the MAC address of the devices connected to the network [50] [53]. If the MAC address of a device is included in the access control list, all the traffic transmited by this device can be ignored. By filtering out the frames from the attacker and ignoring the NAV value in these frames, the legitimate components of the network would be able to continue communicating. This countermeasure mechanism would be effective if the attackers did not modify their MAC address. But, since the MAC addresses can be easily spoofed, the use of access control lists can affect to those legitimate devices whose MAC address had been spoofed. Another countermeasure mechanism proposed by the authors of [65] is to modify the IEEE 802.11 standard. After receiving a RTS frame and replaying with a CTS frame, the AP monitors the wireless medium. If the device sending the RTS does not start transmitting after a given period, the AP transmits another control frames with NAV set to 0. The authors of the same work also describe possible drawbacks for the countermeasure mechanism that they propose.

### 3.3.2.1.3 Frame Flooding

Attackers could implement another type of DoS attack by injecting large amount of frames in a relatively short period of time. This is known as the frame flooding attack. Frame flooding actively works against the availability and the integrity of the wireless network resources.

The wireless nodes that compose the wireless networks are generaly devices with limited operational resources. Frame flooding intends to exhaust the computational resources of these devices, preventing legitimate users in the wireless network from communicating. There exist diverse modalities of flooding attacks, based on the type of frame transmitted by the attacker. The most common modalities of this attack are probe request flooding, authentication request flooding, and association request flooding, according to [67].

### 3.3.2.1.3.1 Probe Request Flooding

Legitimate wireless devices utilise the probe request frames to discover the presence of wireless networks and to request specific information about the transition properties of the AP. According to the IEEE 802.11 standard, every AP that receives a probe request frame has to respond with a probe response frame. Given that the AP responds to the received probe request frames, the probe request flooding attack tries to exploit this active scanning mechanism.

In probe request flooding, the attacker injects multiple probe request frames in a very short period of time. An attacker flooding the AP with a large number of probe requests will exhaust the resources of the AP. As long as the attacker keeps sending probe request frames, the AP will be unable to process all requests from legitimate devices being served. These devices will be deprived of an efficient service, and the communication quality in the entire wireless network will be deteriorated.

One feasible solution to this attack is that the AP would not process repeated requests from the same MAC address. The authors of [67] explain the utilisation of MAC address filtering or access control filter as countermeasure mechanisms against different types of frame flooding attacks. Similar to the case of virtual jamming, this countermeasure mechanism would be effective if the attacker did not modify its MAC address. However, the use of access control lists can affect to those legitimate devices whose MAC address had been spoofed.

### 3.3.2.1.3.2 Authentication / Association Request Flooding

Authentication request and association request flooding follows similar implementation to probe request flooding. Wireless clients willing to join the wireless network use the authentication process to identify themselves to the network. In response, the AP transmits an authentication response frame to the devices, approving or disapproving the authentication for each received authentication request frame. Similar to the authentication request, the AP responds to each received association request frame.

Processing the authentication requests and association requests involves the utilisation of memory and computational resources of the AP. The AP uses a buffer to store information about the wireless users during the authentication process. When the buffer is full, the AP would not be able to accept any new authentication request [67]. An attacker injecting large amount of consecutive authentication/association request frames in a short period of time will exhaust the resources of the AP, and would not be able process any other incoming request. Additionally, the communication quality of the already associated users will be deteriorated, or even lost [113].

The authors in [67] have conducted practical experiments to demonstrate the effect of these attacks against a wireless network. From the results that this work presents, both attacks are able to drop the communication throughput in an entire wireless network down to zero.

Again, MAC address filtering could be a solution to this attack. Another countermeasure could be the utilisation of a counter or a timer, along with an established threshold. For each new received authentication/association request frame, the AP initiates the counter or a timer. If the number of incoming authentication /

association request frame surpasses the established threshold, the AP stops processing any other incoming request. However, this countermeasure mechanism can also affect legitimate wireless devices that request authentication/association after the established threshold has been reached.

### 3.3.2.2 Frame Replay

Frame replay is another attack that compromises the availability and the integrity of the legitimate communications in the wireless network. An attacker could retransmit intercepted frames; this is known as a frame replay attack. The attacker can retransmit any type of frame without a considerable level of difficulty. The device, to which the intercepted frame was destined to in first instance, either the AP or one of the wireless users, receives the replayed frame and will assume that this is a legitimate frame.

The real threat of this attack against the wireless network resides on the type of frames that the attacker replays. For instance, if the attacker replays a deauthentication frame or a disassociation frame, it could force legitimate devices to leave the wireless network. As long as the attacker keeps replaying these frames, the legitimate devices will be unable to reauthenticate with the AP. Mutual authentication between both parties of the wireless communication would make frame replay attacks more difficult to succeed [12]. However, this attack is feasible because both management and control frames are unauthenticated. Also, the correct utilisation of sequence numbers in the transmitted frames would also make frame replay attacks more difficult to succeed.

### 3.3.2.3 Frame Modification

The success of the frame replay attack relies on the information included in the intercepted frames. The attacker needs to replay particular type of frames, such as a deauthentication frame, in order to succeed in the attack attempt. In order to achieve the maximum potential of the frame replay attack, the attacker should be also able to modify the content of certain fields in the intercepted frames before being replayed.

Using frame modification, attackers compromise the integrity of the legitimate communications in the wireless network.

An attacker could modify beacon frames or probe responses from the AP, and advertise some wireless communication properties that are not suitable to accomplish an efficient communication. The attacker could make the nodes in the wireless network leave the network if the communication properties advertised by the AP are not the optimum. Similarly, the attacker could modify the information in the beacon frames or probe response frames, indicating that the AP only supports the weakest security protocol, e.g. WEP.

The frame modification attack can be implemented since the attackers can easily surpass the security protocols implemented by the IEEE 802.11 standard to provide confidentiality, integrity and availability to the wireless communication. These protocols cannot stop the attackers from modifying the content of the network frames. One possible countermeasure mechanism against the frame modification attack is to implement mutual authentication between the AP and the wireless users [12].

## 3.3.2.3.1 Identity Spoofing

Every Ethernet card and wireless NIC is assigned a unique MAC address, at the time of manufacture, to be unequivocally identified. However, this value can be arbitrary changed using an appropriate software tool. There are plenty of options available on the Internet that allow easily changing of the MAC addresses of the NICs. The capability of an attacker to modify the source MAC addresses in the injected frames, and masquerade itself behind a fake MAC address is known as Identity Spoofing or MAC address Spoofing. The attackers can spoof any MAC address. It could use not only the MAC address of legitimate user of the wireless network, but also the MAC address of non-existing devices within the wireless network.

MAC address spoofing is among the most serious threats that wireless networks may face [73]. An attacker that spoofs its MAC address is able to perpetrate a diverse number of attacks. For instance, an attacker may be able to surpass the access control mechanisms of the security infrastructure by spoofing the MAC address of legitimate users [124]. There exist numerous attacks, ranging from DoS to session hijacking that can be implemented because an attacker may masquerade itself as a legal user [69]. Utilising MAC spoofing is not completely necessary to implement most of the wireless-specific attacks described in this chapter, but the effectiveness of these attacks will be increased if the attacker utilised MAC spoofing. Many other wireless-specific attacks need to conceal the identity of the attacker to avoid being identified.

The different examples of spoofing attacks presented in this section can be launched because both management and control frames are unauthenticated. The different frames used for these attacks can be easily spoofed. If the IEEE 802.11 standard provided a robust and secure authentication mechanism for the management and control frames, MAC spoofing could not be used to launch any type of attack.

### 3.3.2.3.1.1 Deauthentication Request

Deauthentication request attack is one of the attacks that require identity spoofing to be successfully implemented. The IEEE 802.11 standard defines the deauthentication management frames for requesting the deauthentication of a specific wireless device from the network either, in case the authenticated device wanted to leave the network, or the AP wanted a specific device to leave the network. In deauthentication request attacks, an attacker forces legitimate devices to leave the wireless network. This attack can be implemented because the IEEE 802.11 standard does not provide any mechanism for validating the authenticity of the deauthentication frames [51].

An attacker can inject deauthentication frames, destined to the AP, using the source MAC address of a wireless device already authenticated or associated. The AP will process the spoofed deauthentication frames and accept the deauthentication request because the source MAC address corresponds to a legitimate device. After leaving the network, the victims might reinitiate the entire authentication and association process. As long as the attacker keeps sending spoofed deauthentication frames, the victims will be unable to reauthenticate with the AP. Similarly, the attacker can also spoof the source MAC address of the AP and inject this deauthentication frames, destined to a particular wireless device. Upon receiving the spoofed deauthentication frames, the targeted wireless devices will automatically leave the network. In one final attack scenario, the attacker can also inject deauthentication frames destined to the broadcast address [10]. As a consequence, all the authenticated or associated wireless devices have to leave the wireless network. However, the authors of [9] found that some wireless devices do not follow the indications of the IEEE 802.11 standard and could ignore this type of deauthentication frame.

### 3.3.2.3.1.2 Disassociation Request

The IEEE 802.11 standard states that each wireless user could be authenticated to different APs at the same time. However, these users are only allowed to be associated with a single AP at a time [112]. In order to stop the association with the AP, the IEEE 802.11 standard defines disassociation frames.

In disassociation request attacks, an attacker utilises spoofed disassociation frames to force legitimate devices to disassociate from the currently associated AP. Similar to the deauthentication request attack, this attack can be easily implemented because the IEEE 802.11 standard does not provide any mechanism for authenticating the authenticity of the disassociation frames [51]. An attacker can inject disassociation request frames using the source MAC address of an already associated wireless device, destined to the AP. The victim of this attack will be forced to disassociate from the associated AP. Similarly, the attacker can also inject disassociation request frames destined to a particular wireless device or destined to the broadcast address, using the spoof MAC address of the AP. As long as the attacker keeps sending spoofed disassociation request frames, the victims will be unable to reassociate with the AP.

Similar countermeasure mechanisms can be implemented to protect wireless networks against both attacks. MAC filtering out may eliminate the effect of these attacks [67], but this solution may also cause a DoS attack on the legitimate device whose MAC address has been spoofed. Another countermeasure mechanism proposed is to authenticate the management and control frames [51]. The authors of [59] [51] have also proposed to modify the IEEE 802.11 standard. For instance, the authors of

[51] propose queuing the deauthentication or disassociation requests for a short period of time (i.e. 5-10 seconds) before processing them. If a data frame arrives from the same source MAC address as the deauthentication/disassociation request, during the queuing time, the request frame is discarded. Although the results presented by [51] prove this countermeasure mechanism efficient, there exist some drawbacks associated to this mechanism. First, the queuing mechanism adds an undesirable delay to wireless devices that are roaming between APs. Also, this countermeasure would be ineffective if any data frame is received during the queuing time.

### 3.3.2.3.1.3 Rogue Access Point / Network Hijacking

Rogue AP, also known as network hijacking, is a type of attack through which the attacker is able to take over the communication between the legitimate AP and wireless devices. This attack can be implemented as an extension of the deauthentication and disassociation attacks. Initially, the attacker spoofs the identity of the legitimate AP and forces the victim to leave the wireless network. Then, the attacker makes the wireless device authenticate and associate with itself rather than the legitimate AP. In order to achieve this, the attacker start transmitting spoofed beacon frames, including the MAC address of the AP and the same Service Set Identity (SSID) value. Probe response frames containing the same information, the spoofed MAC address and SSID, can be used to make the wireless device associate with the attacker [125]. The attacker needs to inject the frames using stronger signal strength than the AP, since the wireless devices try to associate to the AP with the strongest signal strength [125]. Before completely accomplishing this attack, the attacker needs to make sure that the wireless device does not reassociate back with the legitimate AP.

Once the attacker has taken over the wireless communication, it has complete control of the communication [91]. The attacker would be able to freely access the information transmitted from or destined to the victim wireless client. It would be able to deliberately filter out and modify transmitted information, and it would be able to inject completely fake information to the wireless client.

The attackers can implement this attack since the management frames are unencrypted and unauthenticated, and by taking advantage of the identity spoofing capability [10]. One feasible countermeasure mechanism against the network hijacking attack is the implementation of mutual authentication between the legitimate AP and the wireless users [12]. Another countermeasure mechanism proposed by [114] is to increase the transmission signal strength of the AP to provide better wireless communication properties than the attacker, making the wireless devices authenticate with the AP again.

## 3.3.2.3.1.4 Energy Saving

The energy saving attack also requires identity spoofing to be efficiently implemented. The IEEE 802.11 standard provides a functionality to allow the wireless devices to save their generally limited energy resources. The wireless devices can enter an energy saving mode during which they are unable to transmit or receive any frame [51]. The energy saving mode leads to great power savings in wireless devices. The AP periodically transmits beacon frames containing the Traffic Indication Map (TIM) [59]. The TIM is used by the AP to indicate which wireless users in the energy saving mode have data buffered for them [112]. The wireless users must leave the energy saving mode to receive the beacon frames, and examine the TIM. If there is any buffered data at this time, the users send a PS-Poll control frame to the AP to retrieve the buffered data [112]. The AP sends the buffered data to the users that requested it and, subsequently, removes the contents of its buffer [51].

An attacker could spoofs the source MAC address of the PS-Poll control frames to launch the energy saving attack. An attacker can cause the AP to remove the buffered data destined to the wireless devices in the network. Taking advantage of the identity spoofing capability, the attacker is able to inject a PS-Poll control frame destined to the AP and to retrieve the buffered data of a legitimate user, while this user is sleeping [51]. After transmitting the buffered data the AP removes the contents of its buffer. Therefore, the legitimate user will never receive that information. Similarly, the attacker is able to inject beacon frames, spoofing the MAC address of the AP, including a modified TIM value. If the TIM value indicates that there is not buffered data to be retrieved, the user will return back to the sleeping mode [51]. Given that the legitimate user does not retrieve the buffered data, the AP may collapse the buffering resources of the AP. One final attack that could compromise the implementation of the energy saving mode aims to break the synchronisation between the AP and the wireless device. The wireless devices in energy saving mode relies on its synchronisation with the AP to wake up and been able to receive TIM values. Before entering into the energy saving mode, the wireless device synchronises with the AP when to wake up. Given that the synchronisation information is transmitted in the clear, using unauthenticated management frames, an attacker can inject spoofed management frames to make the wireless device to wake up at the inappropriate times, falling out of synchronisation with the AP [51].

### 3.3.2.4 Man-in-the-Middle

Man-in-the-Middle (MitM) is a type of attack that works against the integrity of communication. By implementing this attack, the attacker is able to access the communication between two of the nodes in the network, and is capable of modifying the content of the transmitted information.

MitM attack can be implemented using different methodologies. One of these methodologies is implementing this type of attack at the application layer. An attacker could initially install a computer virus, malware, or Trojan horse in the victim machine. Although an attacker could use these pieces of information to carry out many other malicious actions, these can also be used to have access to the system, and acquire, modify or delete information. This methodology of MitM attack is not exclusive of wireless nodes. It also affects nodes connected to an Ethernet network.

Another methodology of MitM attack is implemented very similarly to network hijacking attack. In fact, this methodology, which affects only to the nodes in a wireless network, could be interpreted as a double implementation of the network hijacking attack. As part of this methodology, the attacker has to spoof the identity of both end devices of the wireless communication. On the one hand, after making the wireless devices leave the network, the attacker spoofs the identity of the AP and starts transmitting spoofed beacon frames. These frames include the MAC address of the AP and similar SSID value to make the wireless device to associate with the attacker. On the other hand, the attacker spoofs the identity of the wireless device and tries to associate with the AP. At this moment, both legitimate devices believe that they are directly connected with each other. However, both are connected to the attacker. The attacker will receive all the frames that each end device transmits to the other wireless device. The attacker has now the freedom to let the information reach its destination, modify the content of the transmitted frames or discard any frame [126]. Finally, the attacker must assure that there is no direct communication between both legitimate wireless devices to succeed in the implementation of the MitM attack. In order to achieve that, the attacker must assure that each wireless device is off the transmission range of the other device. Alternatively, the attacker must assure that both wireless devices start using different frequency channels [52].

Another methodology of MitM attack can be implemented by intercepting the wireless communication and injecting crafted information, using the spoofed MAC address of a legitimate device. Similar to the previously explained MitM attack, this methodology affects only to the nodes in a wireless network. The attacker intentionally replaces the content of the website that the wireless clients receive. This methodology is one of the attacks that have been evaluated in this thesis. Further description of this implementation can be found in the following section 3.4.3.

## 3.4 Evaluated Wireless-Specific Attacks

As will be presented in forthcoming chapters, the effectiveness of the IDS proposed in this thesis will be evaluated using five different wireless network datasets, generated with two particular wireless-specific attacks, as well as a dataset with non-malicious information. These are deauthentication request and a particular implementation of MitM. The attacks have been practically implemented in a live operational IEEE 802.11 network, physically deployed in the laboratory of the High Speed Network Group, at Loughborough University.

### 3.4.1 Deauthentication Attack

One of the evaluated attacks is the deauthentication attack. This attack has been previously described in detail. The attacker launches a succession of deauthentication frames with the purpose of disrupting the connection between the AP and the wireless client, encrypted using WPA2. Although the wireless communication between the AP and the wireless client was WPA2 encrypted, this attack has been able to disrupt the communication. The attacker spoofed the source MAC address in the management frames used in this attack, utilising the MAC address of the AP. Aircrack [121] is the suite of penetration testing tools that has been used to implement this attack.

### 3.4.2 Man-in-the-Middle Attack

Another attack that has been implemented in this thesis is the MitM attack, launched at the PHY layer. In fact, two different versions of the same attack have been implemented using the software tool Airpwn [18], which can be found as part of the suite of penetration testing tools Aircrack. Both implementations require the attacker to be physically located between the AP and the wireless device, intercepting the wireless communication.

First of all, Airpwn intercepts the transmitted frames and looks for HTTP requests from the legitimate wireless nodes. As soon as the attacker eavesdrops a website request from a legitimate node in the wireless network, it injects its own crafted frame. For instance, the crafted frames may contain illegitimate HTML code onto the wireless channel, using the spoofed MAC address of the AP. Airpwn takes advantage of the RTT that a web server takes to respond to legal website requests to inject its own crafted HTML code. Since the attacker is physically located closer to the victim than the AP, it takes the attacker much less time to respond to the HTTP requests than the website server. When the victim receives the data, it will assume the original request was answered and process the fabricated HTML code from the attacker. In the first version of the attack, referred as $Attack01$, the attacker replaces the whole content of the authentic website to a custom one. In the second version of the MitM attack, referred as $Attack02$, Airpwn replaces only the images in the website. The attacker listens for images requests, as part of the HTTP requests, and injects its own fabricated images.

By replacing the content of the website that the wireless nodes receive, Airpwn could cause harm of varying severity. For instance, an attacker could cause less dangerous effects such as replacing the adverts of a specific website with different ones, with the economic benefit that this change could provide. An attacker could also include disturbing images that could produce psychological distress to the wireless network users. In the other extreme, an attacker could also cause more dangerous activity such as redirecting the wireless device to a phishing website. Figures 3.1 and 3.2 show the effect caused by Airpwn to the website that the wireless client receives. Figure 3.1 shows the legitimate content of the Chinese website 'Dict' (http://www.dict.cn), when Airpwn is not active. This is how the legitimate website looks like in normal conditions. Figure 3.2 shows the same website. This time, the Airpwn is active and the figure shows the effect that the $Attack02$ produces in the wireless client web browser. In this example, it is easy to identify the crafted injected content. The banner "*Hello Defcon! Your*" in red replaces the legitimate images in the original website. However, the crafted injected content could be more elaborated and the wireless client might not distinguish the difference between legitimate and malicious images. To make things clearer, Airpwn does not attack the website content at the web server. The attacker never targets the web servers. The attack occurs in the last link of the wireless network, before reaching the wireless client.

### 3.4.2.1 Malvertising Using Airpwn

There exist several Airpwn implementations that attackers can benefit from. Special attention has been given to this type of attack in this thesis because of its capability to modify the content of the websites presented to the wireless client. The capability of Airpwn to replace the adverts of a specific website with different ones could provide substantial economic benefit.

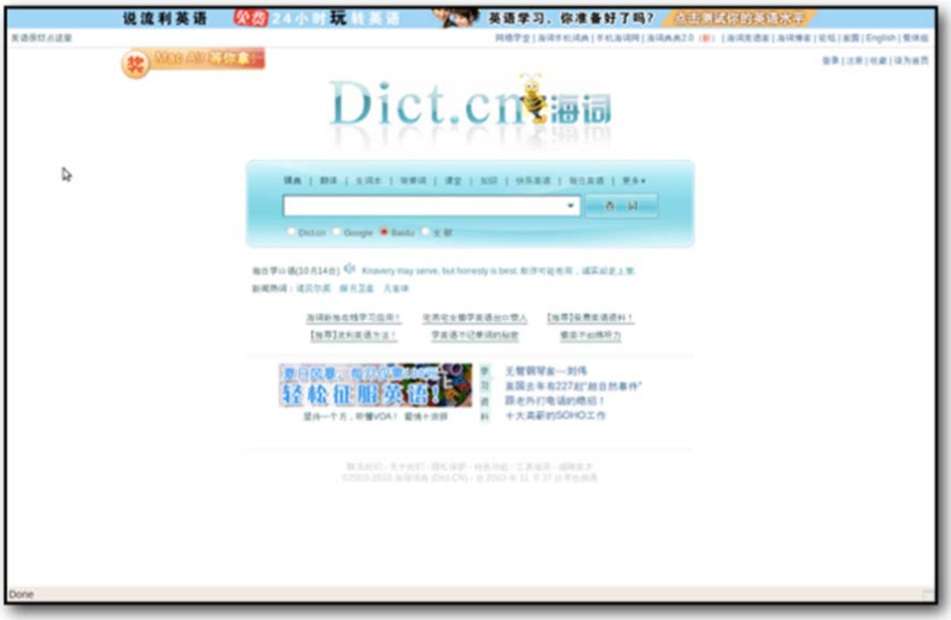
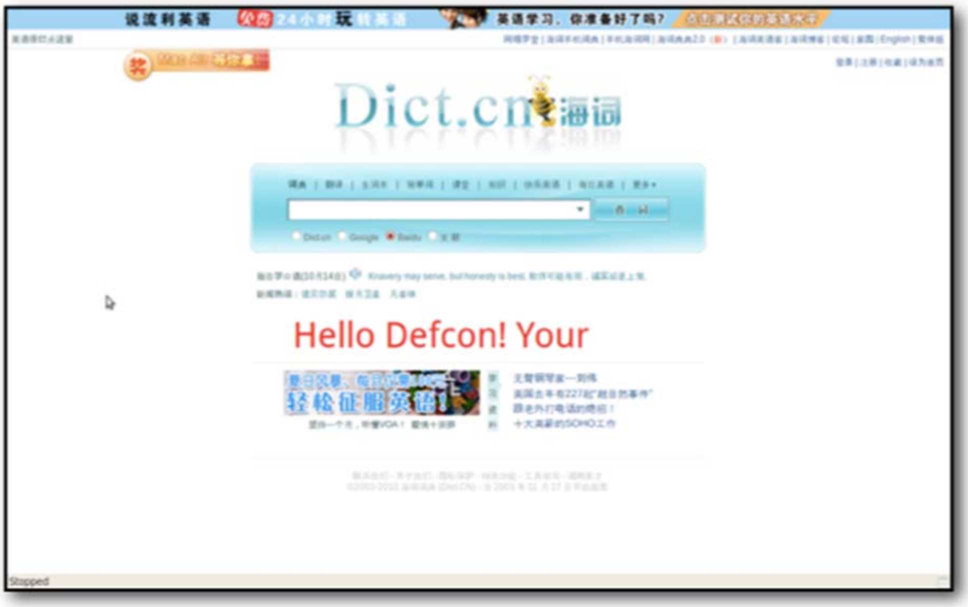


Figure 3.1 Website ‘Dict.cn’ Normal Content.



Hello Defcon! Your

Figure 3.2 Website ‘Dict.cn’ Injected Malicious $Attack02$ Content.

Currently, advertising in websites has become a high profitable business. In the last few years, cyber-attackers have gained presence in this billion-dollar environment trying to gain substantial economic benefit using illicit mechanisms. The authors of [25] explain that hackers can use the adverts embedded in websites to carry out malicious and fraudulent activities. This paper refers to this type of attack as malvertising. The authors highlight the fact that malvertising has become a very profitable illegitimate business, and they present an extensive study on the malvertising infrastructure. This work also shows that illicit adverts can be used by attackers to inject viruses, malware or Trojan horses into the devices accessing to the websites. Similarly, in 2013, a security research team exposed the existence of two software tools that injected unauthorised adverts in websites [26] [27] [28]. Some of these unauthorised adverts would lead users to malicious websites, or would even allow viruses and malware to gain access to the user devices [26]. In this particular case, the user is actively involved in the success of this attack. The user needs to install specific software prior the unauthorised adverts could be injected in the websites. Otherwise the malicious content could not be imbedded in the information displayed in the web browsers. Since the images used in these illegitimate adverts are, in some cases, the corporative brand logo of well-known multinational companies, users have no suspicion of this malvertising content appearing in their web browsers. In fact, this attack has passed undercover for a long time. It is of the interest of many parties to stop this type of actions, since it damages the reputation of both the websites in which the unauthorised adverts are injected and the brands which corporative image is used unauthorised, along with the damage caused to the final users.

Currently, there is no reported evidence that cyber-attackers are actively using the Airpwn attack in malvertising. However, this attack could achieve the same goal than the attack presented in the previous paragraph that injects unauthorised adverts in websites. In contrast to the previously presented attack, Arpwn does not require to install any specific software. Arpwn injects the crafted content into the wireless communication, spoofing the identity of a legitimate AP. This is just one of the multiple usages that could be given to this particular penetration testing tool. The lack evidences about Airpwn being actively used does not necessarily mean that cyber-attackers are not currently making use of it. Nonetheless, it has been practically demonstrated that Airpwn can be used in malvertising.

In this thesis it is speculated that cyber-attackers might see in this tool an effective and easy way to implement malvertising, as a mechanism to gain substantial economic benefit. Fortunately, this work presents a novel methodology that allow identify this attack, and allow the possible application of countermeasure techniques to jeopardise the success of Airpwn. To the best of the author knowledge, this is the first methodology able to efficiently identify this particular attack.

## 3.5 Discussion

In this chapter, several types of attacks have been described. The main purpose of this description has been to highlight the fact that, although many of these attacks can also affect wired devices connected to an Ethernet network, there exist a number of attacks specifically focused on wireless devices and WLANs. Traditional security systems that were initially designed to provide security to wired networks cannot provide a complete level of security to current wireless networks.

From the different wireless-specific attacks explained above, the most commonly proposed countermeasure mechanisms in this chapter are access control filters or MAC address filtering. It is easy to realise that the success of these mechanisms relies on the unequivocally identification of the attacker. The MAC address in the frames determines the identity of the wireless device. Unfortunately, this is not a reliable approach to assess the real identity of the wireless device. An attacker can easily implement techniques of MAC address spoofing, and masquerade itself behind a fake MAC address.

The efforts to provide reliable security against these attacks should focus on the identification of the device that has transmitted the frame. A possible solution might be using, not only the MAC address, but also multiple other parameters or metrics from the wireless device or the wireless communication to infer the real identity of these devices. The most effective approach would be to analyse features or metrics from every layer of the protocol stack, and try to unequivocally differentiate between legitimate wireless devices in the network and the attacker; starting from the PHY layer. This would still not be the definite solution. In order to succeed in its purpose, the attacker may try to imitate as many characteristics of a valid wireless client as possible throughout the entire protocol stack. If the attacker had the capability to replicate all the characteristics of a legitimate wireless client, the different security systems would not be able to provide protection, and the attacker would finally succeed. However, it is highly unlikely that the attacker could mimic every single feature of the legitimate devices. Therefore, the higher the number of parameters or metrics used to identify the real identity of the wireless devices, the higher the probability to identify any attempt of an attacker to masquerade itself behind the spoofed identity of a legitimate wireless device.

As will be explained in the following chapters, wireless network monitoring tools, such as IDSs, are the most appropriate network security mechanisms to identify the difference between a legitimate wireless device and an attacker impersonating the legitimate wireless device. If these network security mechanisms were able to unequivocally identify whether the frame has been transmitted by an attacker or by a legitimate node, then the most appropriate countermeasure mechanism could be applied. In order to prove that IDSs are appropriate mechanisms to protect networks against wireless-specific attacks, the deauthentication request and MitM attacks have been practically implemented in a live operational IEEE 802.11 network and evaluated. There might be a concern about whether the number of experiments is large enough to prove the efficiency of this security mechanism. Whilst evaluating all the existing wireless-specific attacks would be the most appropriate decision to assure that the IDSs can identify all these attacks, research wise, evaluating all these attacks would be impractical. The attacks implemented in this thesis are an adequate sample to showcase the efficiency of the IDSs.

Lastly, during the implementation of this thesis, a third type of attack was also implemented. This was the rogue AP in which the attacker tries to take over the wireless communication between the AP and wireless client. Similar to the previous two attacks, the attacker spoofed the source MAC address in the injected frames utilising the MAC address of the AP. The rogue AP attack was fully capable of getting sensitive information from the victim. The attack was implemented using the attacking tool HostAP [128]. However, the results of these experiments have not been included in this thesis for different reasons. The testbed in which the experiments were implemented was similar to the one in the deauthentication attacks. Therefore, these new results would not add to the idea that changes in the proximity of the attacker from the victim may have on the final results of the proposed detection system. Also, the experimental results using deauthentication and Airpwn attacks prove that the proposed methodology is able to efficiently detect different types of wireless-specific attack without making any adjustments to its implementation configuration, regardless of the implemented attack. Including the rogue AP experiment results would not add to prove the methodology efficiency. The experiment results for this type of attack, using the methodology proposed in this thesis have however been previously presented in [43]. The results published in this paper also prove the efficiency of the proposed detection system detecting this type of attack.

# Chapter 4

## Detection System Definition

### 4.1 Introduction

Due to the increasing number of attacks and intrusion attempts that target wireless networks, the implementation of wireless network monitoring tools, such as IDSs, has become fundamental in the development of security infrastructures for wireless networks. These systems outperform the protection capabilities of cryptography protocols, firewalls or antivirus software. IDSs incorporate sophisticated information analysis techniques that support the processing of any observable and measurable metric of the monitored system to detect attacks.

Different mechanisms could be utilised by IDSs to analyse the datasets. Similarly, multiple design configurations could be applied to the IDS architecture. Designing an IDS requires the consideration of multiple characteristics that define the final architecture of the system. The correct selection of these characteristics would establish the difference between an efficient and a poor detection process. This chapter introduces in detail the concept of IDSs, and provides a detailed taxonomy that describes the most relevant characteristics that need to be considered when an IDS is designed, as well as the pros and cons of each of the characteristics. Next, this chapter describes the characteristic included in the final architecture design of the IDS presented in this thesis, and discusses the principal reasons for selecting each of these characteristics. The purpose of this chapter is to find the most convenient architecture for the detection system presented in this thesis and to discuss the reasons for selecting each of the characteristics.

## 4.2 Intrusion Detection Systems

Intrusion Detection Systems (IDSs) are security systems that constantly monitor information from the protected environment, e.g. a computer or a network system, to identify evidence of attacks or intrusion attempts. The principal role of these systems is to detect malicious actions that compromise the confidentiality, integrity and availability of the resources of the protected systems [14]. In [86], the authors describe the main functions performed by IDSs as gathering activity information from the monitored system, analysing the gathered information and assessing the nature of this information, and raising an alarm if the outcome of the detection process indicates the presence of attack. The authors of [57] simplify the concept of IDS to just a classification problem in which a given piece of information is identified as normal or malicious. In my opinion, this last definition is excessively simplistic. The functions that IDSs implement are particularly complex to be defined as *just a classification problem*.

IDSs have become essential components of the security mechanisms. These systems should be an indispensable part of any security infrastructure, complementing and supplementing the weakness that the traditional security mechanisms may have. Although IDSs could operate independently, similar to other security systems, IDSs should not be considered an alternative for current security mechanisms that are currently widely deployed, such as firewalls or anti-virus. IDSs outperform the detection capabilities of traditional network security mechanisms. This is because IDSs incorporate more sophisticated information analysis techniques, able to analyse more diverse type of information. In addition, as is explained later in this chapter, certain types of IDSs are able to dynamically adapt their detection capabilities to the current characteristics of the resources being protected, in contrast to the traditional static security mechanisms.

It is important to understand that IDSs are not designed to prevent intrusions from occurring. Nonetheless, some IDSs allow the system administrator to specify a set of actions to be automatically implemented in case an attack is detected [55]. Based on the nature of the actions, IDSs that include response actions can be categorised as Intrusion Prevention Systems (IPSs) and Intrusion Tolerant Systems (ITSs). IPSs actively respond to the identified attacks by applying countermeasure techniques, attempting to stop them from persisting and preventing the attacks from succeeding [55]. However, actively stopping the attacks from persisting is not always possible for the detection systems. That is the main difference with ITSs. Instead of attempting to stop the attacks, the ITSs provide resilience against these attacks. ITSs adapt the operational capabilities of the protected system and allow the system to correctly continue working, despite the presence of the attacks [131].

### 4.2.1 Intrusion Detection Systems Classification

Designing an IDS requires considering multiple characteristics that define the final architecture of the system. The correct selection of these characteristics has direct effect on the final performance of the system, establishing the difference between an efficient and a poor detection process.

Different researchers in the field of intrusion detection have described numerous of these characteristics. The most widely utilised in the literature are the source where the information is gathered from, the methodology from which IDSs learn the difference between legitimate and malicious information, and the methodology utilised to conduct the intrusion detection. An extensive IDSs characteristics classification has been presented in [124]. This section provides a detailed taxonomy that, according to the author of this thesis, describes the most relevant characteristics for an efficient intrusion detection process. These characteristics are summarised in Table IV.I.

#### 4.2.1.1 Source of Information

Central to the intrusion detection process is the information analysed to infer the presence of attack. IDSs can make use of any measurable metric from a variety of sources. Based on the source of information, IDSs can be broadly divided into two main categories, network-based and host-based intrusion detection [56]. Router-based intrusion detection is another category described in [132], which is not generally used.

TABLE IV.I. CONSIDERED IDS CHARACTERISTICS.

| CHARACTERISTIC | DEFINITION | CATEGORY |
|---|---|---|
| **Source of Information** | **Defines from where the information used by IDSs is gathered.** | **Network-based** |
| | | **Host-based** |
| | | **Router-based** |
| **Learning Approach** | **Defines how IDSs learn the difference between normal and malicious information.** | **Supervised** |
| | | **Unsupervised** |
| **Detection Systems Cooperation** | **Defines the level of cooperation between different IDSs.** | **Autonomous** |
| | | **Cooperative** |
| **Cooperative Systems Deployment** | **Defines the way cooperative IDSs share the information.** | **Centralised** |
| | | **Hieratical** |
| | | **Distributed** |
| **Detection Timing** | **Defines how long takes to implement the intrusions detection.** | **Off-Line** |
| | | **On-Line** |
| **Detection Methodology** | **Defines the methodology utilised to implement the intrusions detection.** | **Misuse** |
| | | **Anomaly** |
| | | **Hybrid** |

The network-based IDSs gather information from the data traffic passing through the network. Both wired and wireless networks can be the source of this information.

This type of detection system can obtain information directly from the content of the frames (i.e. the frame header and the frame payload) or it can generate different statistical parameters from the characteristics of the network in form of traffic Netflow data (for instance, the duration of the connection). Multiple monitoring sensors could be deployed in one single IDS to provide multiple data streams for a multi-layer approach [133]. On the other hand, the host-based IDSs gather information exclusively from operations that occur in individual computers (e.g. PCs), such as operating system audit trails, system logs and calls, sequence of user commands, application logs, or resource usage [80] [86]. These systems allow the identification of unintended and misbehaving activity locally to the personal devices.

The host-based IDSs require be installed in the monitored PC. Current OSs have become highly complex [78]. Hence, for the proper operation of the host-based IDS, these systems may require great effort to be tailored to each respective system. In contrast, the network-based IDSs are able to protect the wireless devices in the network and the different components of the network from attacks, regardless of the specific OS installed in these systems. This type of system can operate without any knowledge at all or with little knowledge about the monitored network, such as the used network standard. In addition, the network-based IDSs do not require being installed in the network component devices. These systems could be installed in an independent monitoring device, not requiring direct access to any component of the monitored network. These facts make the network-based IDSs much easy systems to be developed.

The router-based IDSs collect information from within the network core of the ISP backbone, the link between routers. However, this category is not as commonly utilised by researchers as the other two categories. Whilst the resources required to develop and deploy host-based and network-based IDSs are easy available, the access to either the routers in the ISP backbone or the information passing though these devices is highly restricted. Therefore, the deployment or utilisation of this type of IDSs is a decision that only the ISPs can take. For instance, one of the research studies that have had access to the core of a major UK ISP network is [134].

#### 4.2.1.2 Learning Approach

Another characteristic that can be utilised to categorise IDSs is the approach utilised by these systems to learn the difference between legitimate and malicious information. IDSs can be categorised as supervised and unsupervised IDSs, based on the learning approach.

The supervised IDSs are provided with a set of learning samples or training datasets that helps the detection mechanisms in IDSs to determine whether the analysed information is malicious or normal. Predefined signatures of known attacks directly teach the detection systems what is normal information and what is not. Training datasets, commonly labelled, also allow the administrators of IDSs, not only to learn the difference between normal and malicious information, but also to define the specific types of attacks that IDSs have to learn. The labelled training datasets may contain instances of both, normal and malicious information. On the other hand, if the training dataset is not labelled, the supervised IDSs assume that the dataset only contains instances of normal information. Therefore, IDSs can compute the reference of normal behaviour and consider as malicious any analysed information that outlines the defined reference.

One complex task that supervised IDSs face is to assess whether the training datasets correctly represent the difference between normal and malicious information. In the case of utilising labelled datasets, the most important consideration is whether the instances in the datasets are correctly labelled. In addition, it is highly unlikely that the available labelled datasets or the attack signatures could cover all the existing attacks [135]. Therefore, this learning information needs to be frequently updated. In the case of utilising unlabelled dataset, the most important consideration is assuring that the dataset is completely clean, without any malicious instance. Again, obtaining completely clean training datasets can be extremely difficult or impossible to achieve. Even in a controlled testbed, assuring that the content of the training dataset is completely free of malicious instances is extremely hard [87]. If a supervised IDS was trained using an unlabelled dataset containing malicious instances, erroneously considering that the dataset is completely clean, the IDS would not be able to detect this particular attack [57].

The unsupervised IDSs learn the difference between legitimate and malicious information autonomously, solely based on the intelligence capabilities of the detection techniques analysing the attributes of the monitored information [100], without external support. The datasets used by the unsupervised IDSs to create the reference of normal behaviour are unlabelled, and may contain normal and malicious information instances. This saves the effort of assessing whether the instances in the datasets are correctly labelled, or whether the datasets contains any malicious instance.

Since these systems do not make use of learning samples or training datasets as do the supervised IDSs, the unsupervised IDSs rely entirely on the efficiency of their decision algorithms to determine the difference between legitimate and malicious information. Finding a decision technique that could precisely differentiate between both types of information is the biggest difficulty when designing unsupervised IDSs. In addition, the lack of learning information means that, if any attack was detected, the unsupervised IDSs would not be able to identify the specific types of attacks. On top of that, two conditions must be assured for the unsupervised IDSs to be efficient; conditions that IDSs cannot fully control. The first condition is that the number of legal instances in the dataset must be larger than the malicious instances. Second, the difference between normal and malicious instances in the dataset must be quantifiable. According to the authors in [87], these two conditions will make malicious instances in the dataset appear as rare outliners from the more predominant normal instances.

#### 4.2.1.3 System Architecture

The architecture of an IDS is composed of a diverse number of modules. The most common are a set of monitoring sensors, which perform information gathering functions, a database that stores the information gathered by the monitoring sensors, and a central engine that performs a set of operations to detect the presence of attacks [55] [136]. Many additional modules can be included to conduct efficient intrusion detection. For instance, it is also common for an IDS to include a preprocessing module [88], data cleansing module, an administration console to monitoring console to control the system [136], or data fusion module, among others.

##### 4.2.1.3.1 Detection Systems Cooperation

The architecture of IDSs provides numerous characteristics to categorise these systems. One of the characteristics is the level of cooperation between different IDSs. This means whether an IDS implements the detection individually or whether it cooperates with other detection systems. Based on the level of collaboration between diverse detection systems, these can be categorised as autonomous or cooperative intrusion detection [124].

One IDS can be designed with different levels of cooperation with other IDSs. In one end, there are IDSs primarily designed to conduct the intrusion detection using information shared by other detection systems. IDSs collaborate with other detection systems to conduct combined intrusion detection in a cooperative architecture. Sharing information among different IDSs can help to detect some ambiguous intrusions [137], reduce the number of false alarms by correlating different IDS outputs [104], and can help IDSs to anticipate possible attacks, previously detected by other IDSs. One of the best examples of this type of IDS architecture is a Wireless Sensors Network (WSN) looking for nodes whose internal behaviour have been compromised (i.e. Byzantine attack). The nodes in the WSNs share information about other nodes for the IDS to identify jeopardised devices. The level of cooperation among IDSs can be gradually reduced to the other extreme in which a single IDS would conduct the intrusion detection individually, without any information shared by any other IDS. In the other end, each IDS considered autonomous is responsible for performing the detection functions independently, and the effectiveness of these systems relies entirely on its own detection capabilities. There is no communication or information sharing with other IDSs.

Cooperative IDSs provide some advantages over the autonomous IDSs. For instance, these IDSs posses more information and, therefore, they have a clearer picture of the environment which they are monitoring, and the specific type of attacks that may be carried out. The authors of [104] propose a cooperative architecture in which multiple IDSs collaborate to make the intrusion detection more accurate and efficient. Despite the provided advantages, this type of system also has some drawbacks. First of all, the communication between IDSs may be vulnerable to attacks. An attacker could compromise the transmitted information, injecting faulty or false information, as well as colliding the communication between diverse IDSs. Also, a common communication language among all the detection systems should be defined for these systems to interpret the information received from other IDSs. This second requirement would not generate a great level of concern among a reduced number of cooperative IDSs. The administrator of these systems could agree on creating common communication language. However, this is not easily scalable to scenarios with high number of IDSs. In addition, the communication between cooperative IDSs increases the quantity of transmitted information, increasing the level of congestion in the communication bandwidth.

On the other hand, the efficiency of an autonomous system relies entirely on its detection capabilities. Given that this type of systems does not have access to extra information, the accuracy of the autonomous systems might be lower than the cooperative systems. Nonetheless, the autonomous IDSs eliminate the disadvantages that the cooperative IDSs present. The authors of [137] correctly argue that autonomous IDSs could only be installed in devices with enough resources. Hence, the computational requirement of the different detection engines used by IDSs should be as low as possible for the autonomous system to be efficient.

#### 4.2.1.3.2 Cooperative Systems Deployment

For IDSs using a cooperative architecture, the deployment of the systems may be also categorised as centralised, hieratical or distributed intrusion detection [58].

In a centralised system, individual monitoring sensors gather information locally and transmit this information to a centralised database. The sensors do not necessarily need to be in the same physical location. Then, the central detection engine of the IDS processes and analyses the information gathered by the different monitoring sensors, stored in the database [124]. The central IDS would identify attacks from the information collectively gathered. A centralised system can also be composed of different IDSs that carry out individual detection processes and send information about any detected attack intrusion to a central correlation system to corroborate the detection using information from other systems [58]. Common to all the cooperative IDSs, the information transmitted to the central database is vulnerable to a series of attacks. Also, the information gathered by different sensors may be duplicated in the database, increasing the computational cost of IDS dealing with duplicate instances. Additionally, the communication between the monitoring sensors and the central database may create a traffic bottleneck. On top of that, the central database becomes a unique failing point. The entire system would be unavailable if the central database is inaccessible as consequence of any failure. If the database fails, the entire IDS also fails.

In order to reduce the level of problem that a single central database or a single central detection engine may pose, a cooperative IDS can be deployed as a hieratical system. In this type of system deployment a diverse number of databases and detection engines are used. The different sensors or IDSs share information according to a hierarchic status. This approach can be envisioned as small cases of centralised systems, with similar vulnerabilities and drawbacks. However, the hieratical system splits the risk and reduces the probability of each database to fail, as well as reduces the probability of traffic bottleneck. In contrast to the centralised system, if one of the databases fails or is unavailable, the entire IDS do not completely fails. The communication between the different components is still vulnerable to attacks.

In a distributed system, each detection system shares information directly with the rest of the IDSs. Either the information gathered by the sensors or information about local detection process is shared in a completely distributed way, without the use of any central database [58]. Similar to the previous two approaches, the communication between the different components is vulnerable to attacks. Also, the process of dealing with duplicate instances and its computational cost becomes redundant. In a centralised system, the central database processes and deletes the duplicate instances. In a distributed system, each system deals with duplicate instances independently.

Therefore, all the distributed systems may need to process and delete similar pieces of information. In addition, the distributed system increases the quantity of transmitted information, increasing the level of congestion in the communication bandwidth.

#### 4.2.1.4 Detection Timing

IDSs can also be categorised according to the detection timing used by the system. This is whether a system works on-line or whether it works off-line [99] [124] [138]. The off-line IDSs work analysing network information in a non-real time manner. In contrast, the on-line IDSs work in real time, continuously analysing information as it is gathered from the monitored system, and detecting attack while they occur. The sooner an intrusion is detected, the less the amount of damage in infringed to the protected system.

The off-line IDSs could gather information from the protected system during hours, days, or even weeks, either periodically or continuously, before implementing the intrusion detection. Once the information has been gathered, this is processed and analysed by the detection system. However, off-line IDSs do not always need to wait until the complete dataset has been gathered. The off-line IDSs could also choose to use time intervals models to conduct the detection process, using a specific frequency.

The off-line approach is very convenient for researchers in the field of IDS to evaluate the effectiveness of the detection systems over the same dataset, or compare different detection approaches. However, the off-line IDSs fall into the category of forensic analysis. It becomes a matter of finding out whether the protected system has been attacked. By the time evidences of attack are found, these attacks may have already reached the protected system. The IDS would detect the attack after it has taken place. Even IDSs that work in nearly real time are one step behind the attackers.

Urgent attack detection is crucial to provide appropriate and prompt protection. The IDSs have to be able to detect the attacks as soon as they occur [116]. The detection timing for an IDS to be defined as real time, or on-line IDS, should not be defined using standard time units. In contrast, the detection timing should be defined by the time that an attacker needs to compromise the protected systems. The time that on-line IDSs need to detect intrusion attempts should be smaller than the time an attacker needs to compromise the protected system [139]. This would give the option to the protected system to implement any countermeasure action if required.

A reduced number of publications have presented an on-line IDS, according to [138]. This might be because the on-line IDSs generally posse high computational cost. The off-line tend to be less computational demanding than the on-line IDSs. However, the computational cost of the IDSs should be lightweight to be applied in real time [90]. But lightweight detection processes are less reliable than a thorough detection processes. Therefore, the requirement of urgent makes the on-line IDSs less reliable than the off-line IDSs [99].

#### 4.2.1.5 Detection Methodology

One last characteristic to categorise IDSs is the methodology utilised to implement the detection. Based on the detection methodology, IDSs can be categorised as Anomaly or Misuse intrusion detection [13] [57]. This is the most frequently used characteristic to categorise IDSs.

##### 4.2.1.5.1 Misuse Intrusion Detection

Similar to the antivirus, misuse IDSs make use of predefined signatures to identify known attacks. Misuse IDSs compare the analysed information against the signatures of known attacks, looking for any matching. In case any match is found, the systems will raise an alarm indicating the finding. A database is used to store the signatures. These signatures can be in the form of regular expressions or state transition models that characterise the implementation of particular malicious actions [10]. The attack signatures could be defined manually by the administrator of the detection system or it could be automatically defined as the outcome of an anomaly IDS. According to the authors in [100], the supervised IDSs are commonly used for misuse IDSs. The attack signatures define what normal information is and what malicious information is.

Traditionally, misuse IDS has been the preferred intrusion detection option for network security [57] [140]. Most of the commercial detection systems currently developed are based on misuse IDSs [78] [87] [106] [135] [141]. This is because the low percentage of FPs generated by this type of systems (commonly 0%), and the high degree of accuracy detecting known attacks. Also, the misuse IDSs are able to identify the specific type of attack that compromise the protected system. This fact is highly beneficial for the level of security provided to the protected system because, if the specific type of attack is known, the countermeasure mechanism that best protection provided against this attack can be implemented. In that case, the protected system would suffer low or even no damage at all. Misuse IDSs are also characterised by the short required time to detect the attacks. This is because it is assumed that the attack signatures are already stored in the signatures database before performing the intrusion detection analysis. The matching process carried out by these systems is done almost instantly. In addition, the implementation of misuse IDSs is conceptually simpler than anomaly IDSs. Evaluating the efficiency of these systems is also easier than assessing the efficiency of the anomaly IDS.

Unfortunately, these systems are victims of their own good performance. The high degree of accuracy detecting known attacks and the low percentage of FPs would bring the user of the IDS into a false sense of security against unknown attacks. The misuse IDSs are unable to detected previously unknown attacks and variations of known attacks [104]. These systems may suffer from high percentage of FN alarms. The occurrence of a FN means that an attack has passed unnoticed and reached protected system. Another disadvantage of these systems is that, for each new type of attack that is identified, a new signature describing the attack should be generated and included into the signatures database [57]. From the time an attack is created to the time the respective attack signature is created, the monitored system is vulnerable to this particular attack [87]. Furthermore, poorly defined signatures could make the IDS to identify normal behaviours as malicious. In addition, the authors of [58] emphasise that the larger number of signatures stored in the database, the higher the computational cost and the longer the time required for analysing the signatures. This is a very interesting remark, because clearly reflect the fact that for each new attack or modification of known attacks, a new signature ought to be generated. In an environment such as computer networks in which new attacks are constantly created, evaluating all the signatures in the database would be computational.

### 4.2.1.5.2 Anomaly Intrusion Detection

Researchers in the field of intrusion detection are mainly focused on anomaly IDS [36]. Anomaly IDSs create statistical references of normal behaviour of the protected systems using historical data [55]. Any deviation from the normal behaviour of the monitored systems is interpreted as an attack or intrusion attempt [40]. Anomaly IDSs are based on the assumption that the normal and malicious behaviour are differentiable from each other, and the difference must be quantifiable. If both behaviours were completely similar, any effort to identify the attacks would be impractical. In addition, anomaly IDSs require that the number of legal instances in the analysed dataset must be larger than the malicious instances. These two conditions are similar to the conditions required for the unsupervised IDSs.

The effectiveness of an anomaly IDS relies on the technique utilised to analyse the information and the accuracy of the technique used to perform the training process. The process of creating the statistical reference of normal behaviour is considered the training process or training phase [78]. Numerous techniques can be employed to generate this reference. Once the training process has finished, the anomaly IDSs calculate the level of deviation of the currently analysed information from the reference of normality. High level of deviation between both values is likely to correspond to evidence of intrusion [79]. The main difficulty is to define the boundary between normal and malicious information. According to [104], this is one of the most complicated, and the most crucial tasks for anomaly IDSs. As part of the training process, the anomaly detection systems also have to define an alarm threshold. This alarm threshold defines the boundary between normal and malicious actions. If the deviation overpasses the defined alarm threshold, the system concludes that an attack is taking place and raises an alarm.

In contrast to misuse IDSs, this type of IDS is able to identify previously unknown attacks and variations of known attacks. Unfortunately, anomaly IDSs are known for their poor detection efficiency [36]. The anomaly IDSs tend to produce high number of false alarms [78] [106] [142] [143]; mainly FPs. These alarms are caused because normal activity is misclassified as anomalous. This is because information that deviates from the reference of normality does not always correspond to the occurrence of attacks [104]. There would be sometimes in which legitimate behaviour might deviate from the reference of normality, causing the anomaly IDS recognise this deviation as evidence of attack [144]. Another drawback presented of the anomaly IDSs is the inability to identify the specific type of attack that compromises the protected system. Also, anomaly IDSs commonly require long training periods to efficiently detect malicious actions [55] [78]. In addition, the difficulty of training this type of systems increases in dynamic environment [143].

#### 4.2.1.5.3 Hybrid Intrusion Detection

The IDSs are commonly either misuse or anomaly detection systems only [15]. Nonetheless, a methodology that may enhance the detection capabilities of the IDSs is to combine the use of misuse or anomaly detection systems [15] [78] [124]. This is known as Hybrid intrusion detection [15].

The hybrid intrusion detection combines the advantages of both systems [124]. This type of systems seeks to improve the overall intrusion detection performance of the misuse IDS, while reducing the number of false alarms suffered by the anomaly IDSs [36]. The anomaly detection capabilities allow the hybrid system to detect previously unknown attacks, as well as to generate new signatures that can be used by the misuse system, whereas the misuse detection capabilities provide the celerity to detect the attacks and the high detection accuracy of known attacks. An example of hybrid detection system is presented in [145].

## 4.3 Discussing the Intrusion Detection System Designing

There is no one final IDS design that could be efficient in every possible scenario. The final architecture of the detection system is subjected to any initial need required from the monitored system. Each decision in the design involves modify the selected characteristics. As can be inferred from the previous IDS taxonomy, each of these characteristics offers a series of benefits over the rest, such as detecting certain attacks that the others may ignore or substantially increasing the detection capabilities of the system [55]. But the offered benefit is not always the same. This is tied to the configuration of the protected system, and the configuration varies according to the context in which the IDS has been defined. A clear example of this variation is the particular scenario in which the IDS is going to be deployed. An IDS designed to be deployed in a business office, most likely would have a different architecture than an IDS designed to be deployed in a battlefield. The targeted scenario adds some constrains to the final architecture of the IDS, defines some requirements that alter the benefit that each characteristic offers, and in turn, alters the final selection of these characteristics. Hence, the effect that these characteristics may have on the final performance of the detection system should be predicted when designing the IDSs.

The following section present a discussion about the specific characteristics that have been included in the IDS presented in this thesis, defines the architecture of the detection system, and offers the reasoning behind these selections. The chosen final architecture is the one that more efficiently adapts to the requirements defined for this work and best detection result produces, according to the author of this thesis.

### 4.3.1 Defining the Source of Information

The first decision about the design of the detection system presented in this thesis is to develop a network-based IDS. The main objective of this thesis is design a system to provide more reliable protection to wireless networks. Therefore, designing a network-based IDS has been a simple and straightforward decision. The system will monitor the traffic in the network to identify presence of attacks. This type of IDS would allow to be installed either in each of the wireless devices or in an independent third party monitor device. The source of information directly influences the type of information collected to identify the attacks. The proposed IDS has been designed to extract parameters only from different headers of the IEEE 802.11 frames. By doing so, the system is able to gather the required information, even if the traffic is encrypted, and does not commit any confidentiality infringement monitoring the payload content.

### 4.3.2 Defining the Detection Timing

Whereas this was not one of the initial requirements, the implementation of a detection system that could work in real time, or nearly real time, became one of the primary aims of the system proposed in this thesis. Instead of generating statistical parameters such as NetFlow data and SNMP statistics, or capturing network traffic datasets through sampling process, it was chosen for the detection system presented in this thesis to analyse the network traffic on a per-frame basis. This decision was taken because, in order to detect the wireless-specific attacks, the minimum unit of information that could compromise the protected system should be analysed.

There may be situations in which the computational resources of the wireless network could not deal with the cost of working on a per-frame basis in real time. Working on a per-frame basis also poses the problem whether the detection system is not able to manage each frame before the next frame reaches the system. Nonetheless, the computational cost depends, not on whether the detection system work on-line or off-line, but on the used detection technique. In general, the on-line IDSs are highly computational demanding and less reliable than the off-line IDSs [99]. If the implemented detection technique were able to infer the presence of attack using low computational cost, working on per-frames in real time could be a feasible approach.

### 4.3.3 Defining the Monitoring Sensors Deployment

One approach that could reduce the computational demand is to reduce the amount of analysed data traffic. The monitoring sensors should gather only network traffic local to the protected devices, and information that is transmitted in the radio frequency channel used by the protected device. It would not be necessary for an IDS to monitor network traffic in locations outside the transmission radio range of the protected nodes. IDSs should not consume computational resources protecting the wireless devices against attacks that cannot reach to the protected system. Similarly, it would not be necessary trying to provide protection against wireless-specific attacks launched in non-adjacent radio frequency channels that cannot interfere with the legitimate communication. The monitoring sensors should be located close or in the protected wireless device, monitoring the same radio channel.

### 4.3.4 Defining the Detection Systems Cooperation

For the presented detection system, it has been decided to design an autonomous IDS. The most important reason for designing an autonomous IDS is that information received from other IDSs may not be trustable. Similar to other wireless communication, an attacker may have compromised the communication. A similar situation would arise if different monitoring sensors distributed through the network had to transmit share information wirelessly. The communication between the different sensors could be compromised. In the IDS presented in this thesis, a single monitoring device collects all the network traffic. Therefore, it reduces the computational cost of merging and reorder information, cleaning duplicate frames or using any synchronisation method, as would occur if several sensors were used.

### 4.3.5 Defining the Learning Approach

One of the most important design requirements was that the presented IDS should be as independent from the system administrator as possible. Minimum or no human intervention should be required. Between the two possible options, using unsupervised IDSs is the only one that matches the requirements established for the design of the IDS presented in this thesis. This decision also benefits from the capability of the unsupervised IDSs to detect attacks using unlabelled datasets.

### 4.3.6 Defining the Detection Methodology

One of the last characteristics to consider is the methodology utilised to detect intrusions. The capability of being able to identify previously unknown attacks and variations of known attacks provides to the anomaly IDSs great advantage over the misuse IDSs to protect current network systems. Therefore, the implementation of an anomaly IDS has been chosen over a misuse IDS in this thesis.

## 4.4 Summary

This chapter has introduced in detail the concept of IDSs. The IDSs are security systems that constantly monitor information from the protected environment to identify evidence of attacks or intrusion attempts. These systems incorporate information analysis techniques able to analyse diverse pieces of information. In particular, anomaly IDSs are able to dynamically adapt their detection capabilities to the current characteristics of the resources being protected.

Designing an IDS requires the consideration of multiple characteristics that define the final architecture of the system. A number of characteristics that could be applied to the final architecture design of the IDS have been described. According to the opinion of the author of this thesis, these characteristics are the most relevant for an efficient intrusion detection process. The correct selection of these characteristics has direct effect on the final performance of the system, establishing the difference between an efficient and a poor detection process. Unfortunately, there is not a generic IDS architecture design that could be efficient in any situation, providing consistent security against any existing attack, or protecting any type of system. It has been obvious that when an IDS needs to be deployed, the final architecture of these systems must be designed very cautiously, always considering the system or device to which it is going to provide protection.

Although the particular attacks that IDSs protect from cannot be known beforehand, it is important to anticipate the attacks more likely compromise the confidentiality, integrity and availability of the resources of the protected systems. On the other hand, it has been clear that the IDSs must be tailored to the particular protected system in order to be efficient and accurate. With regards to the IDS presented in this thesis, the architecture design is an unsupervised network-based anomaly IDS that operates autonomously and on-line. The characteristic included in the final architecture design of the IDS presented in this thesis, and discusses the principal reasons for selecting each of these characteristics have been also presented in this chapter.

# Chapter 5

## Proposed Intrusion Detection Technique

### 5.1 Introduction

Any IDS requires some type of intelligence, in the form of detection technique to determine whether the analysed information is malicious or normal. The chosen technique should be able to construct an accurate reference of normal behaviour [89]. More important than constructing an accurate reference of normal behaviour is the requirement for these techniques to properly define the separation boundary between normal and malicious information. Chapter 2 provided a brief description of some of the most widely used intrusion detection techniques. The majority of these techniques work in a supervised manner. This fact makes them inefficient for a system that is intended to work in an unsupervised manner, without intervention from an IDS administrator.

This chapter describes the detection techniques and internal architectural design of the IDS presented in this thesis. This is a novel unsupervised detection system framework, able to automatically adapt its detection capabilities to the current characteristics of the wireless network, without intervention from an IDS administrator. This chapter also explains the sliding window technique developed to implement the IDS training process. Additionally, the concepts of multi-layer intrusion detection and data fusion techniques are also introduced. Among other data fusion techniques, special attention has been given to the Dempster-Shafer (D-S) theory of evidence. A thorough description of the D-S mathematical framework is presented, along with a series of practical examples. The chapter concludes with the most important contribution of this thesis. This is the description of a novel BPA methodology able to automatically adapt its probabilities assignment to the current characteristics of the wireless network.

## 5.2 Training Process

For an unsupervised anomaly IDS, the statistical definition of the reference of normality is created using historical information from the wireless network traffic. Implementing a robust training process is of crucial for the effectiveness of the anomaly IDSs [78]. This is carried out during the training process.

The training process starts with the monitoring sensors gathering information form the protected system. Then, the anomaly IDSs analyses this information to create a statistical reference of normality. Once the training process has finished and the reference of normal behaviour has been generated, the anomaly IDSs calculate the level of deviation of the currently analysed information from the reference. A high level of deviation between both values is likely to correspond to evidence of intrusion. If the training process is not implemented accurately, the anomaly IDS will not be able to define a proper reference of normality, and a large amount of false alarms will be generated.

The training process commonly requires collecting large amounts of historical data. How much historical data is actually required to conduct the training process is a feature that yet needs to be discussed. In [82], the authors indicate that the system has to gather ‘enough information’ to make the intrusion detection. But it is dependent on the detection system identifying what is enough. The idea that anomaly IDSs commonly require long training periods to efficiently detect malicious actions is widely assumed. However, using a system that would requires a long period of time to create the reference of normality would be a major drawback for an IDS that intends to provide intrusion detection in real time. In addition, the characteristics of normal behaviour in wireless network traffic, which is highly dynamic environment, frequently changes. If the system requires long training periods, the resulting profile of normal behaviour may be biased over time.

## 5.2.1 Sliding Window Scheme

The IDS in this thesis also builds the profile or reference of normality using historical data. The time and the amount of information required to effectively generate reference of normal wireless network traffic has been reduced as much as is possible. To manage the information and handle the non-stationary statistical distribution of the data, the proposed system operates on a sliding window scheme. Other researches, such as [38], also make use of a sliding window scheme. The content of the slots within the sliding window composes the historical data used by the IDS to conduct the training process. The length of the sliding window is represented by $n$. The system has one sliding window for each of the considered metrics and for each type of IEEE 802.11 frame. From each new incoming frame, the different metrics are extracted and stored in the last available slot, within the respective sliding window. Assuming the number of normal frames is larger than the malicious frames, the window would be mostly composed of metrics values from non-malicious frames. The content of the sliding window is then used to generate the reference of normality.

The sliding window scheme works as follows. The first time the IDS is run, the $n$ slots within the sliding window will be initially filled with frames metrics before being able to detecting intrusions. Once the $n$ slots within the sliding window have been filled, each of the $n$ frames metrics is analysed and the reference of normality is generated. After all the frames within the first sliding window have been analysed and the detection implemented, the system slides the window one single slot. The metric from the next incoming frame is stored in the slot that becomes empty after sliding the window. Then, a new reference of normality is calculated using the previous $(n-1)$ frames along with the last stored frame. After the new reference of normality has been calculated, only the last stored frame is analysed, since the previous $(n-1)$ frames have already been analysed. Next, the system slides the window one single slot again and a new frame is included. The described process is constantly repeated. This configuration allows detecting attacks as they occur. Figure 5.1 shows a diagram describing how the metrics are stored in the sliding window scheme. In the figure, the frame #20 is considered malicious; the frame is discarded and replaced by the new incoming frame #21.

#### 5.2.1.1 Benefits of the Sliding Window Scheme

The way this sliding window scheme is implemented is simple, robust and convenient to keep track of all the received frames. As an aside, it also perfectly integrates with the concept of array in the C programming language in which the author of this thesis has been developed the presented IDS.

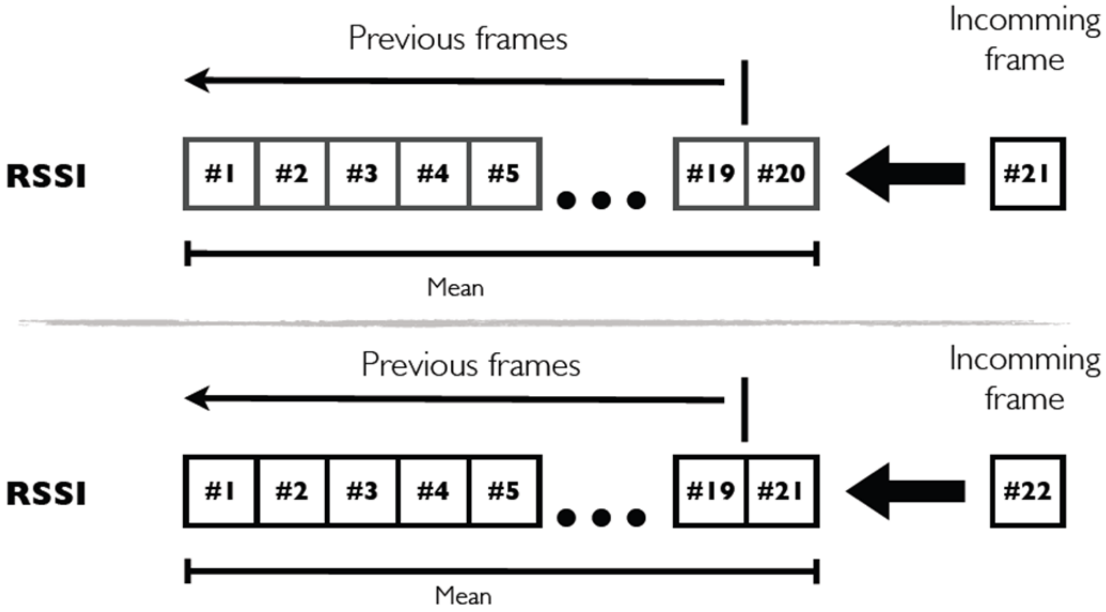


Figure 5.1 Sliding Window Scheme - 20 Slots Long.

From the operational point of view, there are diverse reasons for using this sliding window scheme. For instance, let suppose that an attacker launched a flooding attack in which numerous frames are injected in a very short period of time. If the metrics from all these frames were stored in the sliding window, the statistical reference of normality in the sliding window would be skewed, and the malicious frames would be misclassified as normal and the non-malicious frames would be erroneously classified as malicious. The use of the sliding window scheme avoids this from occurring by sliding only if the currently analysed frame has been classified as normal. Otherwise, if the currently analysed frame has been classified as malicious, the sliding window stays static, drops the frame classified as malicious and replaces the slot in the sliding window that has become unfilled with the next incoming frame.

There is also the possibility that an attacker could slowly corrupt the profile of normality. An attacker may try to mimic some of the communication features of normal wireless devices and gradually change the parameters, skewing the reference of normality. But it is difficult for an attacker to be able to correctly mimic all the measurable features throughout the protocol stack. Attacks could be undetected if an IDS only uses information from a single layer [150]. As explained in Section 5.3, multi-layer IDSs increases the chances for the detection system to identify the presence of intrusions. In turn, it would be highly difficult for an attacker to corrupt the reference of normality.

Despite the previous comments, still there exist one situation in which malicious frames could alter the reference of normality. This is the first time the IDS starts working, during the process of filling the initial sliding window. The system needs to fill the $n$ slots in the sliding window before being able to carry out the detection of new incoming frames. It was explained that the number of normal frames is larger than the malicious frames. However, if this condition is not met, the very first sliding window could contain more predominant number of malicious frames. In the unfortunate case that the number of malicious frames was larger than the number of non-malicious malicious frames and the detection reference of normality would be erroneously calculated. According to the authors in [55], including malicious information when the profile of normal behaviour is being created is one of the common problems of the anomaly IDS. Nonetheless, as will be explained in Chapter 7, because of the general structure of the proposed IDS, the proposed methodology produces good detection results, even if there exist high proportion of malicious frames within the first sliding window.

#### 5.2.1.2 Optimum Sliding Window Length

Finding the optimum sliding window length is very important for efficient and accurate intrusion detection results. The length $n$ of the sliding window will influence the overall detection performance of the system. A long window would include, on average, a larger proportion of normal frames and the statistics would average out to represent the normal profile. However, a larger window length may create inconsistency in the reference of normality. The physical properties of the transmitted signal in the wireless networks are unstable. In a network with moving wireless devices, any change in the geographical location or any obstacle between two nodes communicating may alter the metrics collected by the monitoring node, especially the metrics from the PHY layer. Therefore, building models of normal wireless network traffic over long-term periods may cause that the metrics of the currently monitored non-malicious frames could deviate from the reference of normality using frequently changing traffic information. This situation would make the IDS generate high numbers of FPs. Also, a small sliding window length will allow a prompt adaptability to legitimate changes in the network traffic characteristics. On the other hand, if the sliding window size is too small, it would possibly provide very little information for the training to generate profiles of normal behaviour that accurately represent the behaviour of the network, and in turn, the generated profile of normal network behaviour would probably be biased.

In addition, a large window length will also slow down the detection process. For each new incoming frame the reference of normality is calculated using all the frame metrics in the sliding window. Therefore, the processing time would increase along with the sliding window length. The larger the length of the sliding window, the larger the processing time. The IDS relies on the time required to reach a decision about the real nature of each single frame in order to be an on-line detection system. By using the sliding window scheme, if the detection process is fast enough, the frame analysed as malicious could be discarded from the sliding window before a new incoming frame arrives. As a consequence, numerous successive incoming frames would not saturate the detection system.

Through empirical experiments, it has been proved in Chapter 7 that the proposed IDS requires as little as 50 frames approximately to generate a reference of normality that produces highly accurate intrusion detection in real time. The reduced amount of information required to effectively generate reference of normal wireless network traffic also is another benefit that the sliding window scheme provides.

## 5.3 Multi-layer Intrusion Detection

In Chapter 6 it will be described that six different metrics are extracted from the gathered wireless network traffic datasets. All these metrics will be used by the detection engine to infer the presence of intrusions. Although each of these metrics are going to be initially treated and evaluated independently, the final decision of whether there exist evidences of attacks will be reached using the combination of all the metrics information. This method of using the combined information is known as a multi-layer or cross-layer approach.

IDSs that make use of a multi-layer approach have shown outperforming results, against single-layer detection systems, both in terms of DR and false alarms [37] [38] [39]. Although there are cases in which IDSs that utilise the information from a single metric might give good detection results, the presence of attacks is rarely accurately detectable by examining a single metric from one layer of the protocol stack. Cross-layer systems can combine information from two or more layers of the protocol stack, either adjacent or non-adjacent layers [103].

An attacker may try to mimic some of the communication features of normal wireless devices. It depends on the configuration of the attack, and how well this mimic attempt is implemented to determine if the real identity of the attacker would be unnoticed. As explained previously, it is difficult for an attacker to correctly mimic all the measurable features throughout the protocol stack. Therefore, the higher the number of monitored metrics, the greater the chances to identify inconsistencies in one of these features. Similarly, even if the attack is implemented at one layer in particular, this attack may have an effect on different layers of the protocol stack. Again, using information from the various sources that the multi-layer IDSs provide increases the chances for the detection system to identify the presence of attacks. Also, the greater the number of metrics used, the higher the chances for one of these metrics to show a difference between normal and malicious nature of the information.

In the process of combining information from multiple sources, different mechanisms could be used. The effectiveness of the combined approach changes from one mechanism to another, depending on the characteristics of information that is combined. Some mechanisms may produce more accurate results than others combining the same information. Therefore, selecting the most appropriate mechanism for each case is an essential step in order to obtain accurate detection results [151]. One of the mechanisms is the utilisation of data fusion methods to make a combined use of the information from the different metrics.

### 5.3.1 Data Fusion

Data fusion can be defined as the process of gathering information from multiple and heterogeneous sources about diverse events, activities or situations, and combining them towards obtaining a more accurate final result [14] [39]. In the data fusion framework presented in this thesis, the detection engine performs an independent detection process for each of the considered metrics. Then the independent decisions are sent to the data fusion system to merge all these decisions and reach a combined final conclusion.

### 5.3.2 Data Fusion Techniques

Two of the most commonly used data fusion techniques are Bayesian Theory and the Dempster-Shafer (D-S) theory of evidence. The former technique is based on probabilistic information whilst the second technique is based on evidential information. According to [40], Bayesian theory computes the occurrence probability of an event, assuming that the a priori probability of occurrence for this particular event is known. On the other hand, D-S theory mathematically represents evidence of occurrence of an event based on current observations, without additional a priori knowledge.

#### 5.3.2.1 Bayesian Theory

Bayesian inference is a mathematical discipline used to calculate the probability of occurrence of a certain event, based on the experience extracted from previous events. This theory uses evidences of information, from previous experience, to infer a combined probability of occurrence of a certain event. The operations of Bayesian theorem can be included into the category of conditional probability. The concept of conditional probability defines a probability of an event obtained using additional information, extracted from events that have previously occurred [22]. In tasks of intrusion detection, Bayesian would require the a priori probability that an attack is present in the analysed dataset.

##### 5.3.2.1.1 Bayesian Mathematic Framework

This mathematical discipline provides the probability of an event $A$ to be true, given that certain evidences $E$ are already known [23]. The required evidences to calculate the conditional probability are extracted from previous events, which previously occurred under similar experimental conditions to the event $A$. The events are mutually exclusive states of a system. This means that the system can be in only one of these states at a time [39]. The conditional probability provided by the Bayesian theory, also known as posterior probability, is written as in Equation 5.1:

$$P(A|E) = \frac{P(E|A)\,P(A)}{P(E)} = \frac{P(E|A)\,P(A)}{[P(A)\,P(E|A)\;+\;P(\bar{A})\,P(E|\bar{A})\,]} \tag{5.1}$$

According to this definition, Bayesian theory is unable to assign probability in the considered event in the absence of any other knowledge. Only after evidence $E$ is obtained, can the posterior probability be computed. From the Equation 5.1, three terms can be described. The term $P(A)$ reflects the probability that a particular event is true in the absence of evidence. This is generally known as the a priori probability. The value of $P(A)$ would be updated along with the posterior repetition of the consider event $A$.

#### 5.3.2.1.2 Bayesian Theory Implementation Issues

Although Bayesian theory can be very efficient in certain situations, there exist some challenges associated with its utilisation. The main challenge that most of the researchers highlight is the fact that the Bayesian requires complete knowledge of both the prior and conditional probabilities. These probabilities are very difficult or, in some cases, impossible to determine in practice [102]. Furthermore, there may be more than one way to formulate a particular problem using Bayesian, considering additional or different evidences about the same event. If different or additional evidences are taken into consideration, the posterior probability is likely to differ from one case to the other. Similarly, the posterior probability may change if the evidences are extracted from events, which occurred under different experimental conditions to the considered event. In addition, Bayesian theory does not allow the assigning of a particular probability to uncertainty. It requires that the probabilities be assigned to the occurrence of a state at a time [108].

### 5.3.2.2 Dempster-Shafer Theory of Evidence

The Dempster-Shafer (D-S) theory of evidence is a mathematical discipline that combines evidences of information from multiple and heterogeneous events in order to calculate the belief of occurrence of another event. According to [39] [40], the D-S theory can be considered an extension of Bayesian inference. The main purpose of the D-S theory is to infer the combined belief of occurrence of a certain event, just based on the information provided by the observed evidences. The evidences represent all the information available to infer the real state of the system.

Among different data fusion methods, the D-S theory has been chosen in this thesis, due to a number of benefits that this data function method provides over the rest of the methods. One of the benefits for this selection is the potential for managing uncertainty. In contrast to Bayesian, D-S theory allows assigning belief, not only to the principal system conclusions, but also belief to $Uncertainty$ [108]. Another benefit is that D-S theory does not need a priori knowledge or a priori probability distributions on the possible system states like Bayesian. This is very useful in many vague and unknown environment scenarios [40]. The more important outcome of this property is that, in tasks of intrusion detection, D-S is suitable for detecting previously unseen attacks because it does not require a priori knowledge. The decision of using the D-S theory in this thesis is also supported by the conclusions presented in [39] [102], in which the authors present a comparative study of different data fusion methods and conclude that D-S theory is more promising than Bayesian.

Nevertheless, there are three main drawbacks associated with D-S theory. Firstly this is the computation complexity, which increases exponentially with the number of possible event outcomes. If the system has $i$ possible outcomes, there will be up to $2^i - 1$ hypothesis to analyse. With regards to this work, the system has two possible outcomes; a frame could be normal or malicious. Therefore, the computational complexity of the algorithm would be significantly low. The second main drawback is the conflicting beliefs management, which is a widely known problematic situation in D-S theory. The conflicting belief phenomenon is nicely illustrated with an example from [75]. Given three events $\{A, B, C\}$ and two sensors. The first sensor might assign the following beliefs to the three events, $A = 0.9$, $B = 0.1$ and $C = 0$. Similarly, the second sensor might assign the following beliefs to the three events, $A = 0$, $B = 0.1$ and $C = 0.9$. Applying the D-S theory on these values, the rule of combination presented in the next section will result in the event B having the highest belief value, which is clearly wrong. This situation is solved assigning only non-zero values in one of the proposed BPA methodologies. D-S also requires the statistical independence of the different evidences [40]. D-S theory also requires that the evidences should be completely independent. Regarding the independence of the evidences, there has been extended discussion whether, in practice, the independence of the evidences really affects the performance of D-S theory, and whether this is really necessary. In [21], the author explains that the independence is not necessarily assured in many cases.

#### 5.3.2.2.1 Dempster-Shafer Mathematical Framework

The D-S theory of evidence was first formulated in [30]. D-S theory starts by defining a frame of discernment, or universe of discourse, $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$, which is the finite set of all possible mutually exclusive outcomes about some problem domain [42]. All the consider observers must have the same frame of discernment when using D-S [39]. The power set of the frame of discernment, $2^\Theta$, refers to every possible mutually exclusive subset composed of the elements of $\Theta$. Each subset of the power set is defined as a hypothesis. If $\Theta$ is composed of two elements, $\Theta = \{\theta_1, \theta_2\}$, the total number of hypotheses that it comprises is $2^\Theta = \{\theta_1, \theta_2, \{\theta_1|\theta_2\}, \emptyset\}$. Similarly, if $\Theta$ is composed of three elements, $\Theta = \{\theta_1, \theta_2, \theta_3\}$, the total number of hypotheses that it comprises would be $2^\Theta = \{\theta_1, \theta_2, \theta_3, \{\theta_1|\theta_2\}, \{\theta_1|\theta_3\}, \{\theta_2|\theta_3\}, \{\theta_1|\theta_2|\theta_3\}, \emptyset\}$.

With regards to this work, the frame of discernment is comprised of two outcomes, $A = Attack$ and $N = Normal$, $\Theta = \{A, N\}$. The presented work is able to classify gathered frames either as malicious or non-malicious. Therefore, there will be a set of four different hypotheses in this work, $2^\Theta = \{A, N, \{A|N\}, \emptyset\}$. The hypotheses are $Attack$ and $Normal$, respectively. The hypothesis $\emptyset$ refers to the empty set, and the hypothesis $\{A|N\}$ refers to $Uncertainty$. According to [108], $U = Uncertainty$ is caused by ambiguity in the evidences.

Each hypothesis from the power set $2^\Theta$ is assigned a belief value, or belief, within the range $[0, 1]$. The belief is assigned, based on the evidence of information. This probability assignment is known as the Basic Probability Assignement (BPA), represented as the mass probability function $m$. This is:

$$m: \; 2^\Theta \rightarrow [0, 1] \qquad if \begin{cases} m\,(\emptyset) = 0 \\ m\,(A) \geq 0, \forall A \subseteq \Theta \\ \sum_{A \subseteq \Theta} m\,(A) = 1 \end{cases} \tag{5.2}$$

The function $m(A)$ is defined as the basic probability number of the hypothesis $A$. This function represents the measure of total belief that is exactly assigned to the element $A$ [108]. If the basic probability number of the hypothesis $A$ is a non-zero value, $m\ (A) > 0$, the hypothesis is also known as a focal element. Equation 5.2 shows three conditions that need to be assured. First, the basic probability number of the empty set is 0: $m\ (\emptyset) = 0$. Second, the basic probability number of each hypothesis could be 0 or any other value, up to 1: $m\ (A) \geq 0$. Third, the summation of the basic probability number of all the hypothesis in the frame of discernment must add 1: $\sum_{A \subseteq \Theta} m\ (A) = 1$.

In order to define the total belief given to a certain hypothesis, two functions are defined by the D-S theory. These are the Belief function ($Bel$) and the Plausibility function ($Pl$). The former is the total belief committed to all the subsets of the consider hypothesis. The total belief assigned to the hypothesis $A$ equals the sum of the basic probability numbers for all sets $B$ that are contained in $A$ [108]. This is:

$$Bel(A) = \sum_{B \subseteq A} m\ (B) \qquad \forall A \subseteq \Theta \tag{5.3}$$

One particular feature that makes the D-S theory different from probability theory is the absence of the Additivity Rule. If $Bel(A) < 1$, the remaining evidence $1 - Bel(\bar{A})$ is not equal to $Bel(A)$, $Bel(A) \neq 1 - Bel(\bar{A})$ [149]. Therefore, the $Pl$ function is used to assign the total belief that does not refute the considered hypothesis. This is:

$$Pl(A) = 1 - Bel(\bar{A}) = \sum_{B \cap A \neq \emptyset} m\ (B) \qquad \forall A \subseteq \Theta \tag{5.4}$$

Thus, the belief committed to the hypothesis $A$ is within the interval composed by the Belief and the Plausibility functions, $[Bel(A), Pl(A)]$. This interval is also known as the Belief Range. The amount of belief assigned to $Uncertainty$ is represented by the difference $|Bel(A) - Pl(A)|$ [39]. Nonetheless, the authors of [47] state that the Belief and the Plausibility are similar in the case of pure probabilistic information. For this work, the amount of belief assigned to $Uncertainty$ represents the amount of belief that cannot be assigned either to the hypothesis $Normal$ or $Attack$.

The main capability of D-S is combining independent evidences of information, from different observers with the same frame of discernment, into other single evidence that expresses a common belief assigned to one specific hypothesis. Let $m_1(A)$ and $m_2(A)$ be the BPAs in the hypothesis $A$, from observer 1 and 2, respectively. The combination of evidences is implemented using Dempster's rule of combination. Dempster's rule of combination calculates the orthogonal summation of the BPAs values in one hypothesis from two different observers into a single belief, and is defined in Equation 5.5 as:

$$m(A) = m_1(A) \oplus m_2(A) = \frac{\sum_{X \cap Y = A} m_1(X) * m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) * m_2(Y)} \quad \forall\, A \neq \emptyset \qquad (5.5)$$

The denominator of Dempster's rule of combination is generally denoted as $1 - K$. If the denominator is equal to one, $K = 0$, the orthogonal summation does not exist, and the BPAs of both sensors are said to be completely contradictory.

Dempster's rule of combination allows the combining of evidences of information from only two different observers at a time. In order to combine evidences from more than two observers, Dempster's rule of combination can be used several times in consecutive iterations. The output results of the initial combination process are used as input evidences in the next iteration, along with the evidences of information from a third observer.

**Example**

To easily understand how to apply the D-S theory, an example is presented here.

Let us consider one system with three sensors, Sensor 1, Sensor 2 and Sensor 3. These sensors monitor and gather frames from a WLAN. Using the combined evidences of information provided by the three sensors, the system needs to classify the gathered frames either as malicious or non-malicious.

In such scenario, the frame of discernment is comprised of two possible outcomes, $A = Attack$ and $N = Normal.$ Hence, the total number of hypotheses considered for this example would be $2^{\Theta} = \{A, N, \{A|N\}, \emptyset\}$. Each sensor provides an independent belief in each possible hypothesis. The beliefs assigned by the three sensors are combined to calculate a final decision; i.e. whether the gathered frames are malicious or not. The basic probabilities for one of the frames are tabulated in Table V.I.

The horizontal axis of the Table V.I represents the beliefs of the Sensor 1, for each hypothesis. Similarly, the vertical axis represents the beliefs of the Sensor 2, for all the hypotheses. The cells in the Table V.I represent the multiplication of the beliefs of both sensors.

TABLE V.I. EVENT PROBABILITIES ASSIGNED BY $m_1$ (HORIZONTAL X) AND $m_2$ (VERTICAL Y).

| $m_2$ \ $m_1$ | $m_1(N) = 0.25$ | $m_1(A) = 0.32$ | $m_1(A\|N) = 0.43$ | $m_1(\emptyset) = 0$ |
|---|---|---|---|---|
| $m_2(N) = 0.1$ | 0.025 | 0.032 | 0.043 | 0 |
| $m_2(A) = 0.35$ | 0.0875 | 0.112 | 0.1505 | 0 |
| $m_2(A\|N) = 0.55$ | 0.1375 | 0.176 | 0.2365 | 0 |
| $m_2(\emptyset) = 0$ | 0 | 0 | 0 | 0 |

Dempster's rule of combination is used to combine the beliefs and generate a final decision. The results for the first iteration of this example are:

$$m_{12}(N) = \frac{(0.025 + 0.1375 + 0.043)}{1 - (0.0875 + 0.032)} = 0.233$$

$$m_{12}(A) = \frac{(0.112 + 0.1505 + 0.176)}{1 - (0.0875 + 0.032)} = 0.498$$

$$m_{12}(A|N) = \frac{(0.2365)}{1 - (0.0875 + 0.032)} = 0.269$$

Then, the output results of this initial combination process are used as input evidences in the next iteration, along with the evidences of information from the

Sensor 3. The horizontal axis of the Table V.II. represents the beliefs of the Sensor 3, for each hypothesis. Similarly, the vertical axis represents the combined beliefs of the Sensors 1 and 2, for all the hypotheses.

TABLE V.II. EVENT PROBABILITIES ASSIGNED BY $m_3$ (HORIZONTAL X) AND $m_{12}$ (VERTICAL Y).

| $m_3$ / $m_{12}$ | $m_3(N) = 0.27$ | $m_3(A) = 0.41$ | $m_3(A\|N) = 0.32$ | $m_3(\emptyset) = 0$ |
|---|---|---|---|---|
| $m_{12}(N) = 0.233$ | 0.063 | 0.0955 | 0.0745 | 0 |
| $m_{12}(A) = 0.498$ | 0.134 | 0.2045 | 0.1595 | 0 |
| $m_{12}(A\|N) = 0.269$ | 0.073 | 0.11 | 0.086 | 0 |
| $m_{12}(\emptyset) = 0$ | 0 | 0 | 0 | 0 |

The results for this iteration are:

$$m(N) = \frac{(0.063 + 0.073 + 0.0745)}{1 - (0.134 + 0.0955)} = 0.273$$

$$m(A) = \frac{(0.2045 + 0.11 + 0.1595)}{1 - (0.134 + 0.0955)} = 0.615$$

$$m(A|N) = \frac{(0.086)}{1 - (0.134 + 0.0955)} = 0.112$$

According to the combined results of the evidences of information from the three sensors, the belief in the hypothesis $A$ is higher than the other two hypotheses. Therefore, the hypothesis more likely to be true is $A$, with 61.5% of belief in $Attack$.

## 5.4 Basic Probability Assignment

D-S theory has been previously used in the intrusion detection field to enhance the detection accuracy. In [39] [40] [41], the authors have proven D-S as a powerful and efficient technique to be applied in IDSs. However, a very important step to be investigated remains open in D-S theory. This is to find an automatic and self-adaptive process of BPA. The BPA process is crucial to the effectiveness of D-S theory [107]. The BPA value should be based on the measured characteristics of the monitored environment. With regards to the topic of this thesis, the major challenge for applying D-S theory on IDS is to automatically determine the BPA values should be based on the characteristics of the wireless network traffic measurements [42].

In the IDS literature there exist multiple ways of assigning probabilities to each of the hypotheses in D-S theory, ranging from data mining techniques to empirical approaches. These have been previously described in Chapter 2. However, none of the referred works investigates methods to find an automatic and self-adaptive process of BPA, and few of them could be used off-the-shelf without a previous training or fine tuning period. This section addresses the need for an automatic BPA by proposing a novel methodology able to automatically provide accurate belief values and able to self-adapt its capabilities to the current characteristics of the wireless network, without intervention from an IDS administrator. The low computational complexity of the proposed methodology allows implementation in real time.

### 5.4.1 Automatic and Self-Adaptive BPA Methodology

Three different methodologies are proposed in this thesis in order to automatically assign the BPA values to each hypothesis of Θ, $2^{\Theta} = \{Attack, Normal, Uncertainty, \emptyset\}$. One method generates the belief in $Attack$, and a second method generates the belief in $Normal$. Both work concurrently and independently, in order to meet the requirement of the independency of the beliefs. Then, based on the belief in $Normal$ and $Attack$, a third method calculates a balanced belief in $Uncertainty$, which is not completely independent. The two proposed methodologies to generate the belief in $Normal$ and the belief in $Attack$ could be interchanged with each other, but must be independent. The decision of using one methodology for one hypothesis and the other methodology for the other hypothesis has been entirely empirical. The utilisation of a common methodology to generate the belief in different hypotheses would produce situations in which the BPA values are inversely correlated, and the independency of the beliefs would not be met.

Special attention was given to clustering techniques in Chapter 2. The reason for that is that two of the three methodologies to calculate the BPA value proposed in this thesis are, up to some extent, based on this unsupervised data mining technique. Clustering could be described in two ways. First, clustering uses the dispersion of the data to generate the different clusters. Data instances closer to the centroid of the cluster could be easily categorised into the same category as the centroid. Data instances more distant from the centroid could cause misclassification, and be categorised incorrectly. Therefore, it could be said that clustering is based on the dispersion of the data, and the distribution level of the data within the clusters. Second, it could be said that clustering is based on the distance of each individual data instance to the centroid of the clusters; being the Euclidean distance the most common method to calculate the distance between instances. One of the approaches presented and described in the upcoming sections takes in consideration the distribution of the whole dataset, whilst the other takes in consideration the distance of individual data instances of the dataset to a particular reference point.

A remark common to both methods, the possible belief in each hypothesis has been limited to a maximum of 0.5 (50%). This decision has been based on one of the conditions of D-S theory; all the hypotheses must add 1 (100%). Two completely independent methodologies assign beliefs in $Normal$ and $Attack$, respectively. The methodology to assign beliefs in $Uncertainty$ is based on the outcome of the two previous methods, as an adjustment parameter to satisfy the required conditions of D-S theory. Limiting the maximum possible belief in each hypothesis either 50% or 100% would not alter the final results because, for both values, the final beliefs needs to be adjusted to satisfy the conditions of D-S theory.

### 5.4.1.1 Method to Assign Belief in *Normal*

The methodology that has been proposed to assign beliefs in $Normal$ is based on the degree of dispersion of the values in the dataset. This methodology implements the Bloxplot [127] data representation approach, which has been adapted for the purpose of BPA. Bloxplot is an approach used to represent how the data in a dataset are distributed, and is able to show the presence of outliners within the analysed dataset. Analogous to clustering that was initially envisioned as a methodology for data visualisation and currently is also used in tasks of intrusion detection, Bloxplot has been used in this thesis not only to represent the degree of dispersion of the values in the dataset, but also as a mechanism to assign beliefs.

The Bloxplot starts by defining a certain number of parameters, based on the data values. These are the total number of instances in the dataset ($n$), the first quartile ($Q_1$) that defines the boundary for the lower 25% of the data, the second quartile, or median ($Me$), that defines the boundary for the 50% of the data, and the third quartile ($Q_3$) that defines the boundary for the lower 75% of the data. To calculate these three parameters, the $n$ instances in the dataset are sorted from the lowest to highest value. The $Me$ is the data instance that, after being sorted, divides the dataset in half, leaving the lowest 50% of the dataset at one side and the highest 50% at the other side. The $Q_1$ is the data instance that, after being sorted, leaves the lowest 25% of the dataset at one side and the highest 75% at the other side. The $Q_3$ is the data instance that, after being sorted, leaves the lowest 75% of the dataset at one side and the highest 25% at the other side. Also, the interquartile range ($IQR$), the difference between $Q_3$ and $Q_1$ represented in Equation (5.8), as well as the $Min$ and $Max$ values are calculated. These two last parameters are calculated using the following Equations 5.6 and 5.7, respectively.

By plotting all these parameters, Bloxplot provides a clean and robust method to represent how the values in the datasets are distributed, based entirely on the values of the data instances. New datasets with data similarly distributed would produce Bloxplot diagrams with similar shape. In contrast, new datasets with data distributed differently would produce Bloxplot diagrams with different shapes. The way the Bloxplot is constructed provides a very dynamic method to represent the distribution of the dataset. The proposed methodology provides a dynamic and self-adaptive methodology to assign the BPA values. In this case, this methodology has been defined to assign beliefs in the hypothesis $Normal$.

$$Min = Q_1 - 1.5 \times IQR \tag{5.6}$$

$$Max = Q_3 + 1.5 \times IQR \tag{5.7}$$

$$IQR = Q_3 - Q_1 \tag{5.8}$$

Each of the parameters used to represent the Bloxplot diagrams are used to define the boundaries of different classes. A particular BPA value is assigned to each of these classes. Figure 5.2 illustrates the different classes and the belief value associated to each of them. If the value of the currently analysed frame metric coincides with $Me$, the belief is 50%. If the value is falls between the $Q_1$ and $Me$, or $Q_3$ and $Me$, the belief in $Normal$ is 40%. Values between $Min$ and $Q_1$, or $Q_3$ and $Max$ will acquire belief of 30%. The rest of the values will acquire belief of 15% in $Normal$. All these belief values have been empirically calculated. Considering the $Me$ the reference point in the dataset, the closer the data instances are to the $Me$, the higher the belief in $Normal$.

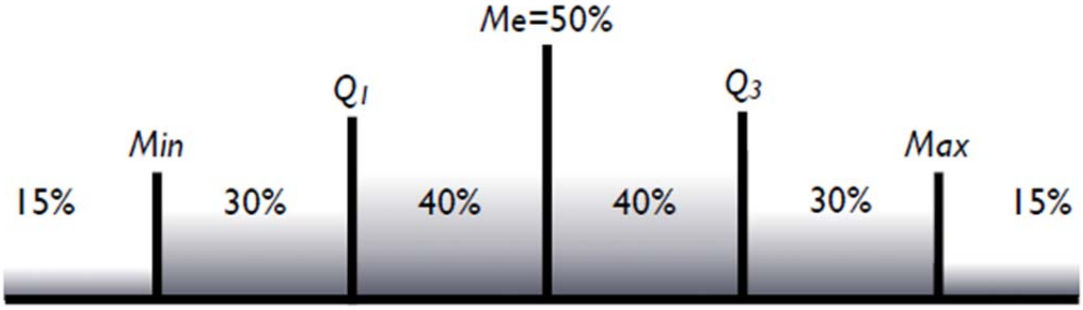


Figure 5.2 BPA Scale For Belief in $Normal$.

Any time a new metrics value is included into the sliding window, all the described parameters are recalculated, the boundaries of the classes are redefined and the belief is assigned. The metrics of each new incoming frame are allocated within one of these classes. Depending on the class that the currently analysed frame is allocated to, the system assigns the belief in $Normal$. Although the boundaries of the classes are fixed, their value changes every time a new frame is included in the sliding window, and the parameters are, again, calculated. This methodology always assigns a non-zero BPA value to the instances. Therefore, it solves the issue of the conflicting belief phenomenon.

**Example**

To easily understand how the proposed methodology to assign beliefs in $Normal$ works, an example is presented here. Let us consider a sliding window with length $n = 17$. The gathered metric values are as follows, with the 11th value, -26, being the only malicious value in the sliding window.

*-36, -36, -34, -34, -36, -36, -36, -34, -36, -36, **-26**, -34, -36, -36, -34, -36, -36*

After sorting all the values from lowest to highest:

*-36, -36, -36, -36, **-36**, -36, -36, -36, **-36**, -36, -36, -34, **-34**, -34, -34, -34, -26*

where $Q_1 = -36$, $Me = -36$, and $Q_3 = -34$. Using these values, $IQR = -34 - (-36) = 2$, $Min = -36 - 1.5 \times 2 = -39$, and $Max = -34 + 1.5 \times 2 = -31$. Hence, the boundaries of the classes are:

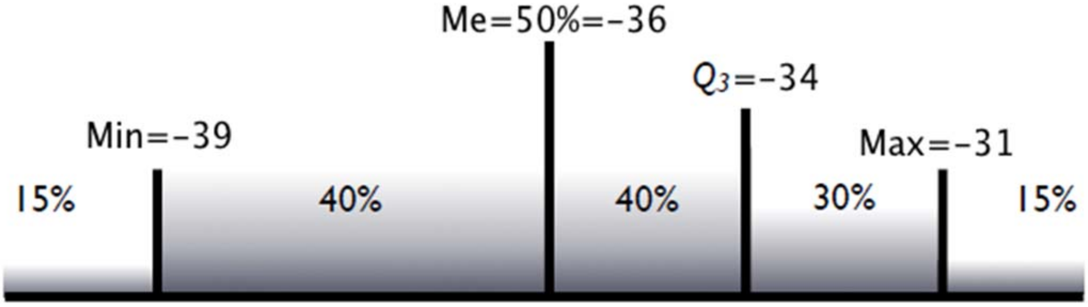


Figure 5.3 Example - BPA Scale For Belief in $Normal$.

The instances with value -36 are assigned 50% belief in $Normal$ and the instances with value -34 are assigned 40% belief in $Normal$, whereas the malicious instance with value -26 are assigned 15% belief in $Normal$. These results are in line with the desired BPA results to identify the malicious instance.

#### 5.4.1.2 Method to Assign Belief in *Attack*

The methodology that has been proposed to assign beliefs in $Attack$ is based on the distance from the currently analysed instance to a point of reference. It is necessary to start by defining a certain number of parameters. Again, $n$ represents the total number of instances in the dataset. It is required to identify a point of reference, as well as the data instance with the highest value ($Hi$) and the instance with the lowest value ($Lo$). After sorting the $n$ instances in the dataset, it is straightforward to select $Hi$ and $Lo$. To select the point of reference, both parameters, the Mean ($M$) and the Mode ($Mo$), have been proposed and evaluated. As will be presented in Chapter 7, selecting the $M$ and $Mo$ will produce slightly different detection results. The parameter $M$ is calculated using the Equation 5.9.

$$M = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{5.9}$$

where $n$ is the total number of instances.

Once the point of reference is selected, the Euclidean distances from the point of reference to the lowest value ($Lo$) and the highest value ($Hi$) are calculated. The value with the largest Euclidean distance ($D_{max}$) from the point of reference represents the maximum possible belief in the hypothesis $Attack$. Next, the Euclidean distance from the point of reference to the currently analysed data instance ($D$) is also calculated. Finally, the belief in $Attack$ is assigned using a simple linear function, making use of the different parameters calculated.

Similar to the previous methodology, any time a new frame is included into the sliding window, all the described parameters are recalculated and the new belief is assigned. Figure 5.4 illustrates the definition of the distance values.

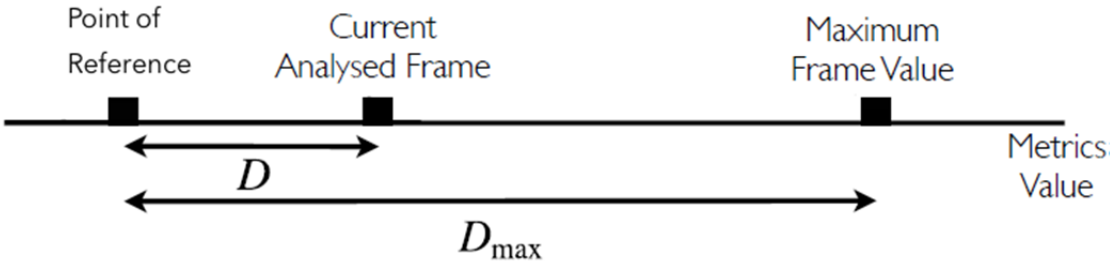


Figure 5.4 BPA Scale For Belief in $Attack$ Based on Distance.

**Example**

Another example is presented to easily understand how the proposed methodology to assign beliefs in $Attack$ works. The considered sliding window with length $n = 17$ is similar to the previous example. Again, the 11th value, -26, is the only malicious value in the sliding window. In this example, the $M$ defines the point of reference.

*-36, -36, -34, -34, -36, -36, -36, -34, -36, -36, **-26**, -34, -36, -36, -34, -36, -36*

After sorting all the values from lowest to highest:

***-36**, -36, -36, -36, -36, -36, -36, -36, -36, -36, -36, -34, -34, -34, -34, -34, **-26***

where $Lo = -36$ and $Hi = -26$. Using Equation 5.9, $M = \frac{\sum_{i=1}^{n} a_i}{n} = -34.823$. The distance from the point of reference to $Lo$ is 1.177, and the distance to $Hi$ is 8.823. Therefore, for this example, the $D_{max} = 8.823$ and represents the maximum possible belief in the hypothesis $Attack$, 50%. The distance to the instances with value -36 is $D_{-36} = 1.177$. For these values, the belief in the hypothesis $Attack$ is 6.67%. The distance to the instances with value -34 is $D_{-34} = 0.823$. For these values, the belief in the hypothesis $Attack$ is 4.664%. Finally, the distance to the instance with value -26 is $D_{-26} = 8.823$. For this value, the belief in the hypothesis $Attack$ assigned to these instances is 50%. Once more, these results are in line with the desired BPA results to identify the malicious instance.

## 5.4.1.2.1 Modified Method to Assign Belief in *Attack*

During the implementation of the experiments presented in this thesis, it appears that, in some particular situations, the proposed IDS did not always generate the best achievable intrusion detection results. Especially when the sliding window length was too small. The more significant reason is that the belief in $Attack$ is, some times, low. Some of the malicious frames were misclassified as non-malicious.

During the experiments implemented to improve the intrusion detection results, it was decided to include an additional parameter in the method to assign the belief in the hypothesis $Attack.$ This was the frequency $(F)$ of the data instances within the sliding window, which is the value with the highest frequency value. Using the previous example, the frequency value is $F = 11$, since -36 is repeated 11 times within the sliding window. However, including this additional parameter requires a method to correlate both parameters, the distance and the frequency, and produce the belief in $Attack.$

One feasible approach to correlate the frequency value and the distance is to use the angle generated by both parameters. Again, the instance with the lowest value $(Lo)$ and the data instance with the highest value $(Hi)$ are identified after sorting the values from lowest to highest. Similarly, selecting $M$ and $Mo$ will produce slightly different detection results, as will be presented in Chapter 7. The parameter $D_{max}$ and the parameter $D$ are calculated as previously explained. Now, as an addition to the previous methodology, the angle that $F$ and $D_{max}$ form is calculated. This angle, referred as the angle alpha (α), represents the maximum possible belief in the hypothesis $Attack.$ Next, the angle that $F$ and the distance to the currently analysed data instance $(D)$ form is also calculated. This angle is referred as the angle beta (β). Finally, the belief in $Attack$ is assigned using a simple linear function between both angles. The angles α and β are calculated using Equation 5.10 and 5.11, respectively. The minimum belief in $Attack$, 0%, is defined by the angle 0°.

$$\alpha = \cos^{-1}\left(\frac{F}{\left(D_{max}{}^{2} + F^{2}\right)^{\frac{1}{2}}}\right) \tag{5.10}$$

$$\beta = \cos^{-1}\left(\frac{F}{\left(D^{2} + F^{2}\right)^{\frac{1}{2}}}\right) \tag{5.11}$$

Figure 5.5 illustrates the definition of the different parameters and angles. Once more, any time a new frame is included into the sliding window, all the described parameters are recalculated and the belief is assigned.

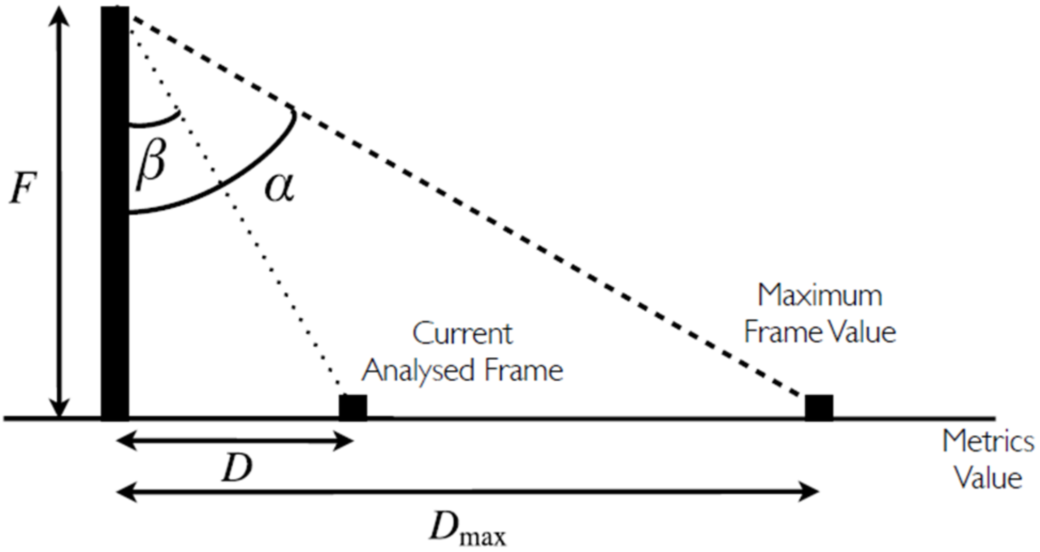


Figure 5.5 BPA Scale For Belief in $Attack$ Based on Angle.

**Example**

Similar to the previous methodologies, an example is presented to easily understand how the proposed methodology to assign beliefs in $Attack$ works. The considered sliding window with length $n = 17$ is similar to the previous examples. Again, the 11th value, -26, is the only malicious value in the sliding window. In this example, $M$ defines the point of reference.

*-36, -36, -34, -34, -36, -36, -36, -34, -36, -36,* ***-26****, -34, -36, -36, -34, -36, -36*

After sorting all the values from lowest to highest:

*__-36__, -36, -36, -36, -36, -36, -36, -36, -36, -36, -36, -34, -34, -34, -34, -34, __-26__*

where $Lo = -36$, $Hi = -26$, $M = -34.823$ and $F = 11$. The distance from the point of reference to $Lo$ is 1.177, and the distance to $Hi$ is 8.823. Therefore, for this example, $D_{max} = 8.823$. To represent the maximum possible belief in the hypothesis $Attack$, 50%, the angle $\alpha = \cos^{-1}\left(\frac{11}{(8.823^2+11^2)^{\frac{1}{2}}}\right) = 38°43'$. The distance to the instances with value -36 is $D_{-36} = 1.177$. For these values, the angle β is $\beta_{-36} = 6°6'$, and the belief in the hypothesis $Attack$ assigned to these instances is 7.884%. The distance to the instances with value -34 is $D_{-34} = 0.823$. For these values, the angle β is $\beta_{-34} = 4°16'$, and the belief in the hypothesis $Attack$ assigned to these instances is 5.523%. Finally, the distance to the instance with value -26 is $D_{-26} = 8.823$. For this value, the belief in the hypothesis $Attack$ assigned to these instances is 50%.

A comparison of the detection results generated by both methodologies, using the same dataset, is presented in Chapter 7. An extended discussion is presented in that chapter about which of the methodologies generate the best results.

#### 5.4.1.3 Method to Assign Belief in *Uncertainty*

One last methodology to assign beliefs in $Uncertainty$ needs to be proposed. The methodology that has been proposed to assign beliefs in $Uncertainty$ is based on the outcome of the two previous methodologies. The $Uncertainty$ has been considered as an adjustment parameter to satisfy the required conditions of D-S theory.

The outcome of the two previous methods could provide four different and mutually exclusive situations:

- Low belief in $Attack$ and high belief in $Normal$.
- High belief in $Attack$ and low belief in $Normal$.
- High belief in $Attack$ and high belief in $Normal$.
- Low belief in $Attack$ and low belief in $Normal$.

For the first and second cases, both methods have reached consistent conclusions. Hence, it is expected that the belief in $Uncertainty$ should be low. In contrast, in the third and fourth cases, both methods have reached contradictory conclusions. Therefore, the belief in $Uncertainty$ is expected to be high in both cases. The proposed methodology normalises the belief in $Attack$ and belief in $Normal$ to assign the belief in $Uncertainty$.

The methodology to assign the belief in $Uncertainty$ normalises the smaller of the other two beliefs ($Belief_{Min}$) to the largest ($Belief_{Max}$). In line with the previous two methodologies, the maximum BPA value has been limited to 50%. The belief in $Uncertainty$ is calculated using Equation 5.12.

$$Belief_{Unc.} = \frac{0.5 * Belief_{Min}}{Belief_{Max}} \tag{5.12}$$

**Example**

To easily understand how the proposed methodology to assign beliefs in $Uncertainty$ works, the results from the previous two examples are used. Table V.III shows the tabulated BPA results for the instances with value -36, -34 and -26.

For the instances with value -36, the belief in $Uncertainty$ is 7.884%. For the instances with value -34, the belief in $Uncertainty$ is 6.903%. Finally, the belief in $Uncertainty$ is 15% for the instances with value -26. In the three cases, the calculated belief in $Uncertainty$ is low.

TABLE V.III. EXAMPLE - INITIAL BPA VALUES.

| Metric Value | -36 | -34 | -26 |
|---|---|---|---|
| Belief in $Normal$ | 50 % | 40 % | 15 % |
| Belief in $Attack$ | 7.884 % | 5.523 % | 50 % |
| Belief in $Uncertainty$ | 7.884 % | 6.903 % | 15 % |

This example represents a situation in which both methodologies reach consistent conclusions. The belief in $Uncertainty$ is relatively low. Now consider another showcasing example in which that the belief in $Normal$ and $Attack$ are 40% and 49.7%, respectively. For these values, the belief in $Uncertainty$ is 40.2%. As expected, the belief in $Uncertainty$ is high.

### 5.4.1.4 Adjustment Value

One of the conditions of D-S theory indicates that the summation of the three BPA values, for each of the hypothesis must add to 100%. This is: $\sum_{A \subseteq \Theta} m\ (A) = 1$. None of the cases in the previous examples satisfy this condition. Therefore, an adjustment value ψ needs to be calculated. The value ψ will be subtracted to each of the three BPA values, and is calculated as follows:

$$\psi = \frac{X - 1}{3} \tag{5.18}$$

where $X$ is the summation of the BPA values. Continuing with the previous examples, for the instances with value -36, $\psi_{-36} = (0.65768 - 1)/3 = -\ 0.1141$. Therefore, the beliefs in $Normal$, $Attack$ and $Uncertainty$ are readjusted to 61.41%, 19.295% and 19.295%, respectively. For the instances with value -34, $\psi_{-34} = (0.52426 - 1)/3 = -\ 0.1586$ . Therefore, the beliefs in $Normal$ , $Attack$ and $Uncertainty$ are readjusted to 55.86%, 21.38% and 22.76%, respectively. Finally, for the instances with value -26, $\psi_{-26} = (0.8 - 1)/3 = -\ 0.067$. Therefore, the beliefs in $Normal$, $Attack$ and $Uncertainty$ are readjusted to 21.65%, 56.7% and 21.65%, respectively.

To summarise, the BPA results for the instances with value -36, -34 and -26 have been tabulated Table V.IV. According to these results, the proposed methodology considers the instances with value -36 and -34 as non-malicious because the belief in $Normal$ is the highest of the three hypotheses, which are, in fact, of non-malicious nature. On the other hand, the instance with value -26 is considered to be malicious because the belief in $Attack$ is the highest of the three hypotheses.

TABLE V.IV. EXAMPLE - ADJUSTED BPA VALUES.

| Metric Value | -36 | -34 | -26 |
|---|---|---|---|
| Belief in $Normal$ | 61.41 % | 55.86 % | 21.65 % |
| Belief in $Attack$ | 19.295 % | 21.38 % | 56.7 % |
| Belief in $Uncertainty$ | 19.295 % | 22.76 % | 21.65 % |

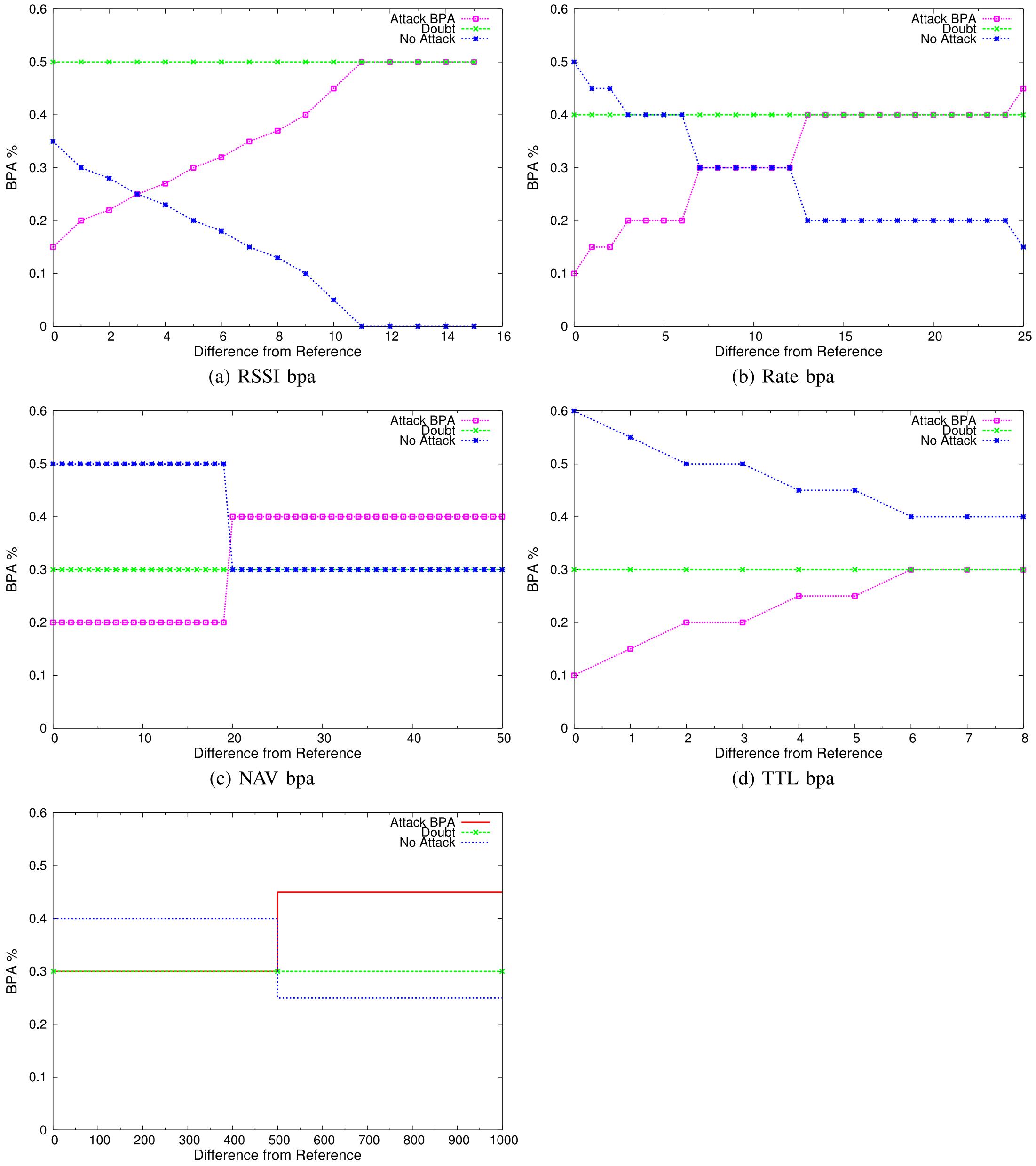
### 5.4.2 Manual BPA Methodology

Before designing the automatic and self-adaptive BPA methodologies presented in the previous section, multiple experiments were implemented using fixed functions. The graphical representation of these functions is presented in Figure 5.6. These functions were experimentally designed.

These fixed functions allowed verifying the correct implementation of D-S theory, and proving that the combined used of multiple metrics outperforms the intrusion detection results of single layer approaches. The different metrics will be described in Chapter 6. The first set of experiments and results generated using the manual BPA methodology was presented in [33]. A comparison study between the manual and automatic BPA methodologies, detecting wireless-specific attacks was presented in [32]. Although the manual BPA methodology proved effective, this approach presents the same drawbacks to the systems that use fixed functions, described in Chapter 2. This methodology is unable to automatically adjust to changes in the network traffic behaviour, without the intervention of the IDS administrator.

## 5.5 System Framework

The IDS proposed in this work is composed of a number of interlinked modules or components as shown in Figure 5.7.



(a) RSSI bpa

(b) Rate bpa

(c) NAV bpa

(d) TTL bpa

(e) Seq bpa

Figure 5.6 Manual BPA Assignment Functions.

The first of these components is the ‘Frames Collector’. This represents the monitoring module in which TShark [17] performs its functions, gathering the traffic within the wireless network. The gathered information is stored in the form of pcap files. The pcap files hold most information related to packets that pass through a network. They are the default format used when monitoring a computer network [43]. Although the IDS is able to implement the intrusion detection process on-line, in real time, storing the data in pcap files allows further analysis of the gathered wireless network traffic to be performed. It also allows the detection performance of the IDS to be subsequently evaluated.

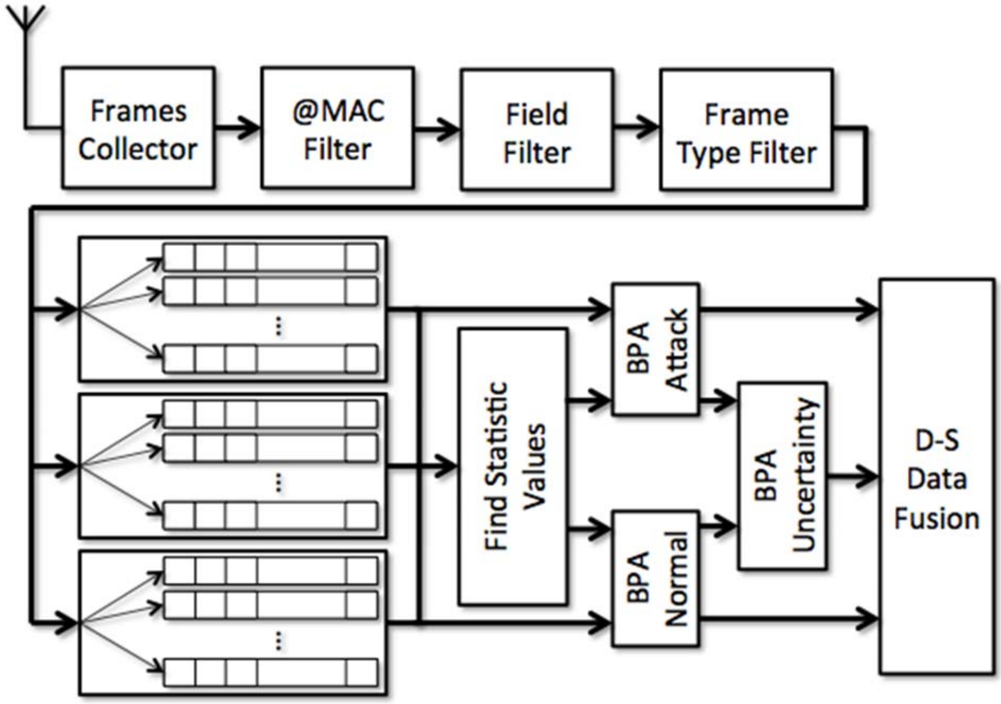


Figure 5.7 Schematic Representation of The Proposed IDS.

Once the raw traffic information has been gathered, the pcap files are sent to the ‘MAC Address Filter’. This module filters out the information that does not meet the filtering criteria. Only frames from the AP, and destination MAC address, broadcast or the wireless client, are kept. The following module is the ‘Field Filter’. For the remaining frames in the dataset, the six metrics described in Chapter 6, along with the type and subtype values are extracted from each of the frames. Each metric value is isolated and stored in a readable format for the detection system. All these filtering processes have been implemented using the filtering capabilities of TShark:

*MAC Addresses Filter:*

```
tshark -l -T fields -R "(wlan.da==2x:xx:xx:xx:xx:x6 || wlan.da==ff:ff:ff:ff:ff:ff)
&& (wlan.sa==0x:xx:xx:xx:xx:x3)" -r /path/…
```

*Field Filter:*

```
tshark -l -T fields -e wlan.fc.type -e wlan.fc.subtype -e radiotap.dbm_antsignal -e
radiotap.datarate -e ip.ttl -e wlan.duration -e wlan.seq -e frame.time_delta" - r
/path/…
```

The metric values are divided into three separate streams of information, based on the type of frames; IEEE 802.11 Data frames, Management frames and Control Frames. The module ‘Frame Type Filter’ is in charge of implementing this division. Similar to the work presented in [38], a sliding window scheme has been employed in this work. Figure 5.7 shows three sets of sliding windows, one for each type of WiFi frame. Each of these sets contains as many sliding windows as the metrics used. The module ‘Find Statistic Values’ is in charge of calculating the required statistical parameters. The same statistical parameters are calculated for each sliding window, each time a new frame is included. The statistical parameters are sent to the modules ‘BPA Normal’ and ‘BPA Attack’, along with the actual content of the sliding windows. Intuitively, the ‘BPA Normal’ is in charge of generating the belief in the hypothesis that the currently analysed frame is non-malicious, whereas the ‘BPA Attack’ is in charge of generating the belief in the hypothesis that the currently analysed frame is, in fact, malicious. Using the outcome of both modules, the module ‘BPA Uncertainty’ generates the belief in $Uncertainty$. Finally, the outcome of the three modules, ‘BPA Normal’, ‘BPA Attack’, and ‘BPA Uncertainty’ are fused in the module ‘D-S Data Fusion’, to generate final set of belief results. For each analysed frame, the outcome of the system is always a 3-fold result, Belief in $Normal$, Belief in $Attack$ and Belief in $Uncertainty$. The hypothesis with the highest beliefs of the three is considered as the most likely to be the correct hypothesis.

The only module that has not been specifically developed for this thesis is the filtering module, which makes use of functions provided by TShark. The modules listed in the previous paragraph have been explicitly developed for this thesis. The flowchart of the detection methodology is presented in Figure 5.8.

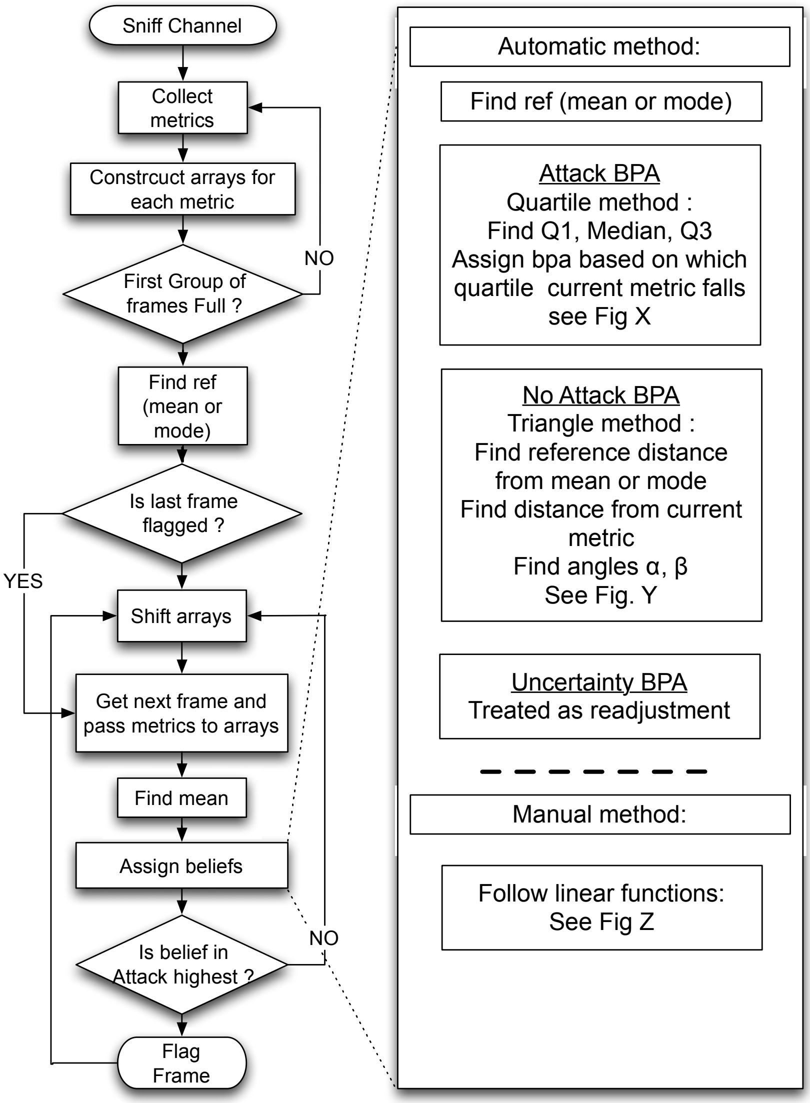


Figure 5.8 Flowchart of The Proposed BPA Methodology.

## 5.6 Summary

This chapter has described the detection techniques used to determine whether the analysed information is malicious or normal, and internal architecture design of the presented IDS. In this thesis, a sliding window scheme is used to manage the information, implement the training process and construct an accurate statistical reference of normal behaviour. The system has one sliding window for each of the considered metrics and for each type of IEEE 802.11 frame. This is a simple, robust and convenient technique to keep track of all the received frames. One example of the benefits provided by this scheme is that the use of the sliding window scheme avoids an attacker from altering the statistical reference of normality when launching a flooding attack.

This chapter has also explained the concepts of multi-layer intrusion detection and data fusion techniques have also been introduced. IDSs that make use of a multi-layer approach have shown outperforming results, against single-layer detection systems. The D-S theory of evidence is the data fusion technique chosen as the most appropriate for the presented detection system because this technique provides a series of benefits over other data fusion techniques. A thorough description of the D-S theoretical and mathematical framework is presented, along with a series of practical examples describing how this technique works. D-S theory has been previously used in the intrusion detection field to enhance the IDSs accuracy. However, an automatic, unsupervised and self-adaptive process of BPA, based on the measured characteristics of the monitored environment, able to operate in real time has not been proposed yet. Such BPA approach has been presented in this chapter.

Three different techniques have been proposed. Only two of these techniques work independently in order to meet the requirement of the independency of the beliefs. The third technique is based on the outcome beliefs of the other two. The decision of using one methodology that lacks of statistical independence was deliberate, supported by other researchers opinion (e.g. [21]), which explains that the independence is not necessary in many cases. Despite being rather simplistic, and dissimilar to well-known information management techniques, the combined utilisation of the three techniques applied to the different wireless network metrics produce a highly accurate detection system. In addition, the low computational complexity of the proposed methodology allows implementation in real time. The results presented in Chapter 7 will prove the efficiency and accuracy of the presented IDS.

# Chapter 6

## Wireless Network Traffic Datasets

### 6.1 Introduction

In this chapter, the analysed real IEEE 802.11 network traffic datasets, the live operational testbed and the data gathering process are described. Firstly, the chapter addresses the decision of whether utilise well-known and publicly available network datasets, synthetically generated dataset using network simulation software, or utilise network traffic datasets gathered from a live operational and physically deployed network. The experiments implemented in this thesis have been tested utilising datasets from a live operational IEEE 802.11 network, physically deployed in the laboratory of the High Speed Network Group, at Loughborough University. The description of the pros and cons of this decision have been presented.

The wireless network traffic is composed of numerous measurable metrics, which are in last instance, the information that would allow IDSs to identify the presence of attack. This chapter continues presenting an extensive description of the different metrics that have been considered for this thesis. This chapter also explains with a practical example the drawback that the D-S theory produces when too many metrics are fused. This is, the larger the number of metrics to be fused, the lower the influence of the Belief in $Uncertainty$ in the final results. Additionally, the statistical description of these metrics is presented, along with a discussion of whether these metrics are appropriate for the detection of wireless-specific attacks. Lastly, the chapter concludes with a brief description of the concept of feature selection and the general need for this approach, as well as the description of the concept of curse of dimensionality.

## 6.2 Using Real Network Traffic Datasets

Using network traffic datasets from a live operational network to evaluate the proposed IDS provides a series of advantages to the system. A physically deployed testbed provides more realistic parameters than a simulated scenario or synthetically generated dataset. However, generating network traffic datasets from live operational networks also has some disadvantages. This environment might not properly scale to a real-life network containing more nodes and more external factors, out of the control of the network administrator. In a real-life environment, it is hard to guarantee that the data does not contain traces of attacks that the administrator does not know about [141]. Even in a controlled network, these datasets are subject to the risk of containing external attacks [36], especially in wireless environments.

Also, the datasets generated in physically deployed testbeds generally contain large amounts of redundant and irrelevant information. Commonly, the datasets would require the application of preprocessing mechanisms to clean the dataset and to make it suitable for the analysis system. Additionally, it is important to know the real nature of all the instances in the datasets for training and evaluation purposes. Real network traffic datasets lack of labelling, which is a major drawback for the supervised detection systems. The real nature of all the instances in the analysed dataset must be known in order to evaluate the efficiency of IDSs correct detection performances. It is impossible to provide evaluation measurements without correctly labelled datasets. Collecting labelled datasets from live operational networks is highly complicated [146], and in many cases impossible. In normal conditions, real network traffic is not labelled. Currently, the task of labelling the instances in the datasets is primarily carried out using previous off-line forensic analysis.

### 6.2.1 Lack of Public Available Network Traffic Datasets

Despite the disadvantages, there are researchers that still prefer to generate their own datasets, for two particular reasons. On the one hand, the benefits explained about the use of real network traffic datasets. On the other hand, the lack of publicly available network traffic datasets. Among the research community there exists a general concern about the lack of publicly available network traffic datasets [57], to evaluate IDSs. Working with a common evaluation dataset would allow comparisons of the efficiency of different IDSs. The lack of an appropriate publicly available dataset impairs the evaluation of the efficiency of the different IDSs [147].

A common approach among researchers in the field of IDS has been to generate datasets in a controlled and deterministic environment to analyse, test, and evaluate IDSs [147]. In [57], the authors explain that there have been some efforts for providing a framework for the researchers to generate datasets in a way that these could be replicable by other researchers.

Although researchers are generating their own network traffic datasets, these are reluctant to share such information and to make the datasets publicly available. One of the main problems of releasing real network traffic datasets relies on privacy concerns [148]. According to the authors of [147], the main reason for this reluctance is privacy concerns related to sharing too much private information, such as passwords contained in the payload of the IP packets.

The option that has been most commonly used over the years to evaluate IDS is the DARPA 1999 dataset [45]. Nowadays, many researchers, e.g. [105], keep evaluating their systems in an off-line environment by using the DARPA 1999 dataset. This publicly available dataset is a common benchmark for IDSs evaluation, which provides a framework in which the number and type of attacks are accurately known, fully and correctly labelled. This dataset contains complete network traffic packets, stored in tcpdump format. One portion of the dataset was real network traffic, whereas the rest of the dataset was simulated. The fact that the background traffic of this dataset was simulated attracts most of the criticism that this dataset receives. During the five weeks of the collection process, four main particular types of attack were included. Additional descriptions about these attacks can be found in [148].

This dataset has been useful for evaluating IDSs because it provides a framework in which the number and type of attacks is accurately known. However, this approach remains an in vitro process, and does not consider the profile of real traffic of a wireless network. Numerous researchers [46] [148] highlight that the main disadvantage of the DARPA 1999 dataset is the uncertain accuracy of the simulated background traffic inserted into this dataset. For instance, the authors of [148] have found quantitative differences between the simulated background traffic of this dataset and real traffic. The authors of [46] also highlight that this dataset is obsolete for evaluating current IDS, since it was made publicly available almost fifteen years ago. In fact, the DARPA 1999 dataset is not representative of current wireless networks. For all these drawbacks, the option of using the DARPA 1999 dataset has been discarded in this thesis.

## 6.3 Wireless Network Testbed

The experiments implemented in this thesis have been tested in a live operational IEEE 802.11 network, physically deployed in the laboratory of the High Speed Network Group, at Loughborough University. This is a physically deployed testbed, yet also a controlled networked environment that allowed the generation of network traffic datasets composed of real IEEE 802.11 information. None of the datasets or any portion of the information compressed in them has been artificially simulated.

The principal components of this network are:

- An Access Point (AP);
- A monitoring node utilising the TShark software for collecting frames, and being responsible for performing the proposed intrusion detection;
- An attacker implementing a series of wireless-specific attacks;
- A client associated with the AP, accessing various websites hosted on the Internet across different geographical locations.

A schematic representation of the wireless network testbed and a picture of the actual testbed are presented in Figures 6.1 and 6.2, respectively.

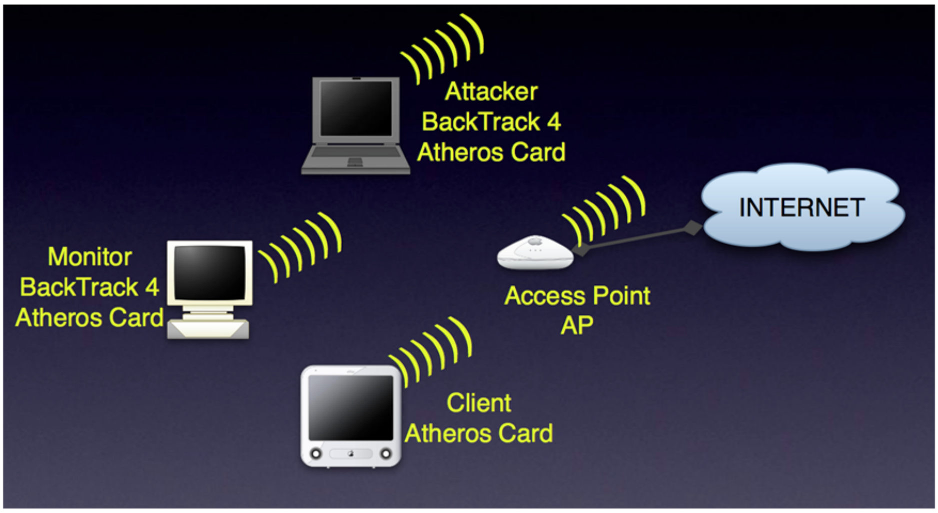


Figure 6.1 Schematic representation of the wireless network testbed.

Figure 6.2 Deployed real IEEE 802.11 network testbed.

Although the topology of this testbed has been modified for the implementation of some particular experiments, the main components of the testbed are those listed. The used AP is a Linksys WRT54GL AP, transmitting through channel 6 at a variable transmission rate, and using the IEEE 802.11b standard. The machine used by the attacker is a desktop PC running the BackTrack Linux OS. In this machine, a PCMCIA wireless card was installed, using the Atheros AR5213A chipset. Similar to the attacker, the monitoring device also used a desktop PC using a PCMCIA wireless card with Atheros chipset. The OS was Ubuntu 12.04 Linux. Both the traffic monitoring software and the detection system presented and implemented for this thesis were installed in the monitoring device. The machine used by the wireless client acting as the victim of the wireless-specific attacks a MacBook Air running the Snow Leopard OS X, and uses the built-in wireless card.

## 6.4 Wireless Network Traffic Gathering

The gathered frames were stored in the form of pcap files [19], using TShark. This type of file is the default format used when monitoring a computer network [43]. When monitoring the wireless network, a large amount of unnecessary information is also gathered. The room in which the testbed network has been deployed was surrounded by different adjacent research laboratories, using their own wireless networks. Some of them were even using the same wireless transmission channel as the one used in the deployed testbed. Hence, once the pcap files have been gathered, it is highly probable that these files would contain traces of communications from wireless networks that are not being protected. All the gathered datasets are accessible and publicly available in the personal website of Dr. Kyriakopoulos [20], from Loughborough University.

### 6.4.1 Filtering Wireless Network Traffic

The external information from adjacent networks may be characterised by different parameters or configuration from the protected network. Keeping these measurement values in the analysed information may distort the detection results. In order to provide protection to wireless devices, including information only transmitted from/to the protected device would be necessary. It would be necessary to discard all this external information that is not of the interest of the proposed detection system. Hence, a data cleansing process has been carried out before performing the intrusion detection. As part of this data cleansing process, the detection process is carried out only on the connection between the AP and the wireless client. The network traffic is filtered out using two criteria based on the source and destination MAC addresses of the network traffic. First, the source MAC address should be the MAC address of the AP. Second, the destination MAC address should be either the broadcast MAC address (ff:ff:ff:ff:ff:ff) or the MAC address of the wireless client. This is implemented using the filtering capabilities of TShark previously explained in the Section 5.5.

The use of the MAC address filter removes the information from the adjacent wireless networks, and helps to reduce the total amount of information that the detection system needs to process and analyse. In turn, the computational cost is also reduced. On the other hand, the MAC address filter provides a simplistic first mechanism of protection. The attacker could implement the attacks using its own MAC address or the MAC address of a non-existing device. Although this is not entirely realistic because the attacker would normally take the precaution of hiding its identity to avoid being identified, it is a feasible situation. In a real scenario, nothing stops an attacker implementing the attacks using a different MAC address from the MAC address of the AP. Therefore, filtering out traffic external to the protected connection would eliminate malicious frames injected not using MAC address spoofing capabilities.

In theory, the filter guarantees that the detection system analyses only information from the connection link between the protected devices. But the attacker is still capable to hide its identity by spoofing the MAC address of any of the legitimate devices, as explained in Chapter 3. In each of the different experiments implemented throughout this thesis, it is assumed that the attacker spoofs the MAC address of the AP. In these cases, the MAC address filter would be ineffective in filtering out the traffic injected by the attacker. Therefore, apart from the legitimate information from the wireless networks, the generated pcap files would include any malicious frames injected by the attacker.

## 6.5 Wireless Network Metrics

The audited information also plays an important role for achieving efficient intrusion detection results [79]. Apart from the detection technique, IDSs rely on the quality of the information datasets to produce accurate detection results. Central to the implementation of the proposed IDS is the analysis of different metrics from the IEEE 802.11 frame header. The monitoring node identifies a particular set of metrics from each frame, and sends the value of these metrics to the detection engine. One of the most important aspects about the set of metrics is that they should provide accurately evidence for the detection system to identify the difference between normal and malicious frames. The detection system would only have a limited set of metrics available to assess the real nature of the analysed frames.

### 6.5.1 Selection of Metrics

The presented IDS developed in this thesis obtains the information from a particular number of metrics extracted from the IEEE 802.11 frames. The selection of the metrics has been conducted experimentally, after the evaluation of all the available set of metrics. The relevance of the different metrics and selection of the most appropriate set of metrics can be evaluated empirically [94]. For instance, in the feature ranking techniques used in [91], the system administrator selects a specific set of metrics based on empirical approaches.

Among all the available metrics, six have been experimentally selected as the most appropriate for detecting the attacks. The six selected metrics are the $RSSI$ and the Frames Interarrival Time ($\Delta Time$) value at the PHY layer; the Injection Rate ($INJ_{Rate}$), the Network Allocation Vector ($NAV$) value and the Sequence Numbers difference ($SEQ_{Dif}$) at the MAC layer; and the Time To Live ($TTL$) value at the Network layer. These six metrics have been used in all the evaluated experiments. All these metrics are thoroughly described below in following sections. $TTL$ is the only metric that has not been used in the experiments with deauthentication attack, due to the fact that management frames lack this metric.

All these metrics show distinctive values between the wireless communication of a legitimate device and the wireless communication of an attacker spoofing the identity of the legitimate wireless device. This difference is based on two factors. The first factor is the technical implementation of the attack. Different attacking tools are implemented differently. Some attacking tool would be more efficient than others spoofing the identity of the legitimate devices. The capability to completely imitate the communication characteristics of a legitimate wireless relies on how well implemented the tool is. The second factor is the level of difficulty for an attacker to replicate the values of the metrics. The circumstance that five out of the six of the chosen metrics are collected at the two lower layers of the protocol stack makes these metrics more difficult to replicate than other available metrics.

Another aspect to be considered when choosing the metrics is the privacy of the transmitted information. Firstly, gathering metrics from the higher layers of the protocol stack and analysing the payload of the transmitted frames would arise privacy concerns over the transmitted information of the user. In addition, the wireless communication between the legitimate client and the AP may be encrypted. If an IDS had to analyse encrypted information, this system would need to know the cryptographic material to decrypt the content of the frames. On top of that, the IDS might not always be installed in the protected devices. It might be installed in a third party device, or it might work as a cooperative IDS and the information shall be shared with numerous devices. Hence, the most optimum metrics would be from the two lower layers of the protocol stack, which are never encrypted.

Using the same set of metrics repeatedly could be perceived as a risky implementation, in which the detection capability of the system is constrained by these metrics. However, as was empirically proven in [31], this set of metrics efficiently adapts the detection capability of the system to different attacks. There may be situations in which, one particular metric of the six would provide stronger evidences of intrusion than the rest of metrics for one particular type of attack. Although the evidences that the rest of metrics provide could be very weak, the strong evidence of intrusion of this single metric would dominate the detection decision of the IDS. Whereas, there may be situations in which this particular metric would be ineffective detecting another type of attack, because the evidences of intrusion that this metric provides are very weak. In such situation, another metric or metrics should provide strong evidence of intrusion to achieve efficient detection. Metrics that provide poor evidence of attack are compensated with the rest of metrics that provide strong evidences. By using the combination of all these six metrics, the best results overall in detecting malicious injected frames are produced.

#### 6.5.1.1 Too Many Metrics for Dempster-Shafer

During the development of this work, another factor has been identified that makes the reduction in the number of metrics necessary. It is related with the data fusion technique, D-S theory of evidence that has been introduced in Chapter 5. The problem with D-S is that the larger the number of metrics to be fused, the lower the influence of the Belief in $Uncertainty$ in the final results. One of the benefits provided by this fusion approach managing is not obtainable after successive fusions. Therefore, using the smallest possible set of metrics would take full advantage of the D-S theory, and its capability to manage $Uncertainty$.

In order to better understand the effect of the number of metrics on the data fusion technique clearer, we consider the following example using real belief measurements extracted from one of the experiments in this thesis. These belief values are calculated using the different techniques presented in Chapter 5. Six independent sensors monitor a particular parameter, and conclude the belief values in the three hypotheses (i.e. $Normal$, $Attack$ and $Uncertainty$) presented in Table VI.I. The belief values for the three considered hypotheses are sorted by the particular sensor that provides these beliefs.

The belief values of each sensor, for each hypothesis are sequentially fused to reach a final conclusion. Only the beliefs from two sensors can be fused at a time. The resulting beliefs of the current fusion are then fused with the beliefs of the next sensor. This process is repeated until the beliefs of the last sensor are fused. Using Dempster's rule of combination, the different beliefs have been fused. The results after the fusion process are presented in Table VI.II. As can be seen in the highlighted bottom row of the table, the values of the belief in *Uncertainty* follows a decreasing trend, as more fusion processes are implemented. This phenomenon works against one of the characteristics that make D-S theory attractive in the field of unsupervised IDS. This is the capability of managing *Uncertainty*, which is an advantage over other fusion approaches. Nevertheless, due to this phenomenon, due to the reduced value of the belief in *Uncertainty*, the D-S fusion process becomes almost similar to a process of Bayesian fusion.

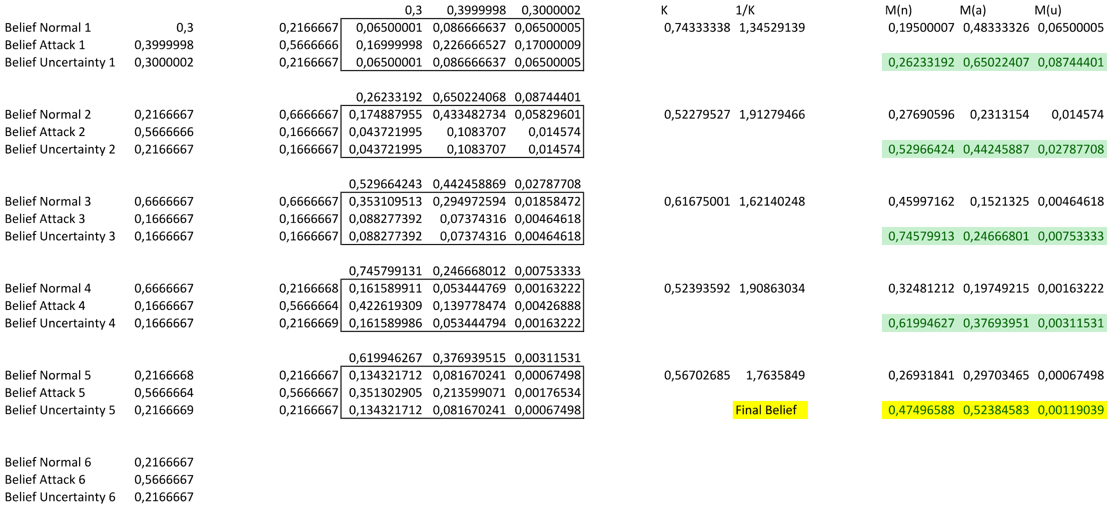
TABLE VI.I. BELIEF VALUES - EXAMPLE.

| | | Sensor | | | | | |
|---|---|---|---|---|---|---|---|
| | | *#1* | *#2* | *#3* | *#4* | *#5* | *#6* |
| **Hypothesis** | *Normal* | 0.3 | 0.217 | 0.667 | 0.667 | 0.217 | 0.217 |
| | *Attack* | 0.4 | 0.567 | 0.167 | 0.167 | 0.567 | 0.567 |
| | *Uncertainty* | 0.3 | 0.216 | 0.166 | 0.166 | 0.216 | 0.216 |

TABLE VI.II. RESULTS OF SUCCESSIVE BELIEF VALUES FUSION - EXAMPLE.

| | | Iteration | | | | |
|---|---|---|---|---|---|---|
| | | *#1 - #2* | *R - #3* | *R - #4* | *R - #4* | *Final Results* |
| **Hypothesis** | *Normal* | 0.262 | 0.857 | 0.187 | 0.746 | 0.475 |
| | *Attack* | 0.65 | 0.107 | 0.751 | 0.247 | 0.524 |
| | *Uncertainty* | 0.088 | 0.036 | 0.062 | 0.007 | 0.001 |

The presented phenomenon is another practical reason for selecting the smallest possible set of metrics that could allow the IDS detect the evaluated attacks, without compromising the efficiency of the system. The whole fusion process is presented in more detail in Figure 6.3. The column in the left hand side shows the beliefs that the 6 different sensors assign to each of the considered hypotheses. Although these are independent, some of the sensors assign similar belief values to the hypotheses. The second column shows the succession of beliefs combination, tabulated in tables. The first table shows the belief multiplication of the sensor 1 and sensor 2. The second table shows the belief multiplication for the outcome of the previous combination and the beliefs of the sensor 3, and so on. The third column shows the calculation of the denominator of the Dempster's rule of combination, for each of the combinations. Finally, the column in the right hand side shows the outcome belief values after each of the combinations.



| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0,3 | 0,3999998 | 0,3000002 | K | 1/K | M(n) | M(a) | M(u) |
| Belief Normal 1 | 0,3 | 0,2166667 | 0,06500001 | 0,086666637 | 0,06500005 | 0,74333338 | 1,34529139 | 0,19500007 | 0,48333326 | 0,06500005 |
| Belief Attack 1 | 0,3999998 | 0,5666666 | 0,16999998 | 0,226666527 | 0,17000009 | | | | | |
| Belief Uncertainty 1 | 0,3000002 | 0,2166667 | 0,06500001 | 0,086666637 | 0,06500005 | | | 0,26233192 | 0,65022407 | 0,08744401 |
| | | | 0,26233192 | 0,650224068 | 0,08744401 | | | | | |
| Belief Normal 2 | 0,2166667 | 0,6666667 | 0,174887955 | 0,433482734 | 0,05829601 | 0,52279527 | 1,91279466 | 0,27690596 | 0,2313154 | 0,014574 |
| Belief Attack 2 | 0,5666666 | 0,1666667 | 0,043721995 | 0,1083707 | 0,014574 | | | | | |
| Belief Uncertainty 2 | 0,2166667 | 0,1666667 | 0,043721995 | 0,1083707 | 0,014574 | | | 0,52966424 | 0,44245887 | 0,02787708 |
| | | | 0,529664243 | 0,442458869 | 0,02787708 | | | | | |
| Belief Normal 3 | 0,6666667 | 0,6666667 | 0,353109513 | 0,294972594 | 0,01858472 | 0,61675001 | 1,62140248 | 0,45997162 | 0,1521325 | 0,00464618 |
| Belief Attack 3 | 0,1666667 | 0,1666667 | 0,088277392 | 0,07374316 | 0,00464618 | | | | | |
| Belief Uncertainty 3 | 0,1666667 | 0,1666667 | 0,088277392 | 0,07374316 | 0,00464618 | | | 0,74579913 | 0,24666801 | 0,00753333 |
| | | | 0,745799131 | 0,246668012 | 0,00753333 | | | | | |
| Belief Normal 4 | 0,6666667 | 0,2166668 | 0,161589911 | 0,053444769 | 0,00163222 | 0,52393592 | 1,90863034 | 0,32481212 | 0,19749215 | 0,00163222 |
| Belief Attack 4 | 0,1666667 | 0,5666664 | 0,422619309 | 0,139778474 | 0,00426888 | | | | | |
| Belief Uncertainty 4 | 0,1666667 | 0,2166669 | 0,161589986 | 0,053444794 | 0,00163222 | | | 0,61994627 | 0,37693951 | 0,00311531 |
| | | | 0,619946267 | 0,376939515 | 0,00311531 | | | | | |
| Belief Normal 5 | 0,2166668 | 0,2166667 | 0,134321712 | 0,081670241 | 0,00067498 | 0,56702685 | 1,7635849 | 0,26931841 | 0,29703465 | 0,00067498 |
| Belief Attack 5 | 0,5666664 | 0,5666667 | 0,351302905 | 0,213599071 | 0,00176534 | | | | | |
| Belief Uncertainty 5 | 0,2166669 | 0,2166667 | 0,134321712 | 0,081670241 | 0,00067498 | | Final Belief | 0,47496588 | 0,52384583 | 0,00119039 |
| Belief Normal 6 | 0,2166667 | | | | | | | | | |
| Belief Attack 6 | 0,5666667 | | | | | | | | | |
| Belief Uncertainty 6 | 0,2166667 | | | | | | | | | |

Figure 6.3 Complete Results of Successive Belief Values Fusion - Example.

### 6.5.1.2 Received Signal Strength Indication – RSSI

The *RSSI* value is a value associated with the transmission and reception of the communication that represents the power level received by the wireless NIC, in dBm. A wireless NIC configured in monitoring mode can measure the *RSSI* value of the received information. The *RSSI* depends on many factors such as distance between source and destination, physical obstacles, WLAN equipment, used frequency channel and an environmental coefficient [32].

Despite being a volatile value, the $RSSI$ for most of the frames transmitted by a particular wireless device follows a fairly tight and predictable Normal distribution [72]. The measurement of consecutive $RSSI$ values from the same transmitter can generate a $RSSI$ fingerprint, which could be used to unequivocally identify individual transmitter devices [76]. Two different wireless devices positioned at the same geographical location, transmitting with similar signal strength will generate different distributions of $RSSI$ at the reception device [72]. The radio chipset and firmware of the wireless NIC characterises individual wireless devices with a unique $RSSI$ fingerprint [10]. Many circumstances must converge for an attacker to be able to replicate the $RSSI$ fingerprint of a particular legitimate wireless device, and most of them are out of the control of the attacker. Hence, it will be almost impossible for an attacker to imitate the $RSSI$ fingerprint of a specific wireless device [72].

Detection systems that employ the $RSSI$ to identify the presence of MAC spoofing attacks are very sensitive to mobile devices [76]. The effectiveness of the IDS that make use of the $RSSI$ fingerprint to detect MAC spoofing attacks may be reduced if the protected device is a moving wireless device. In that case, this type of methodology could generate a high number of FPs.

### 6.5.1.3 Injection Rate – $\mathbf{INJ}_{RATE}$

The $INJ_{Rate}$ is the speed at which the information is transmitted by the wireless devices. Several $INJ_{Rate}$ have been standardised in the IEEE 802.11. Some are mandatorily supported whilst others are optional [112]. The $INJ_{Rate}$ at which each frame is transmitted depends on many factors such as the distance between the source device and destination device, physical obstacles in the propagation path of the radio signal, the environmental conditions and the particular selected modulation scheme utilised by the source of the wireless communication [32]. The $INJ_{Rate}$ directly affects other parameters of the communication, such as throughput or the probability of packet loss.

Multipath fading is a propagation phenomenon in which the transmitted signal traverses multiple different paths from the transmitter to the receiver. The effect is that the receiver signal is distorted by reflexion of the same transmitted signal. In order to deal with this phenomenon, many vendors configure their NICs to reduce the $INJ_{Rate}$ [112]. Using low $INJ_{Rate}$ provides higher throughput in the wireless communications. The lower the used $INJ_{Rate}$, the higher the probability for a transmitted signal to be properly received by the receiver. The approach followed by legitimate NIC vendors is also used by most of the attacking tools, which tend to inject forged frames at low $INJ_{Rate}$ in order to be more efficient [33].

Transmitting at low $INJ_{Rate}$ is not exclusive to malicious devices. A legitimate device would commonly transmit management frames using low $INJ_{Rate}$. However, the $INJ_{Rate}$ value can also be used to reveal the presence of an attacker implementing a MAC spoofing attack.

#### 6.5.1.4 Frame Interarrival Time – $\Delta\mathrm{Time}$

The $\Delta Time$ is defined as the time lapse between the two consecutive received frames. The $\Delta Time$ has been previously used in tasks such as Internet traffic classification [34] or congestion bottleneck in wired networks [35]. Additionally, $\Delta Time$ has been also used to identify the presence of MAC spoofing attacks [69].

The description presented in [69] explains the statistical distribution difference that would produce a non-malicious wireless device transmitting at a fixed interval and an attacker implementing MAC spoofing attacks. The statistical distribution of the time lapse between two consecutive frames, when only the legitimate device is transmitting, would be different from the distribution generated when both devices are transmitting. Using similar approach, the $\Delta Time$ could also be used by the proposed IDS framework to identify the presence of MAC spoofing attacks.

#### 6.5.1.5 Network Allocation Vector – NAV

The $NAV$ value is a 16 bits-long field transmitted in the MAC header of the frames that indicates the amount of time a wireless node reserves the wireless medium to complete a communication. The wireless nodes aiming to transmit should compete with other nodes for the control of the transmission medium. The $NAV$ value is specified in µseconds, with a maximum value of 32767 µseconds. A non-zero $NAV$ value informs to all the nodes in the wireless network to defer to complete a communication. The nodes are allowed to transmit only if the $NAV$ value reaches 0.

The definition of this value is primarily based on the length of the transmitted frames and the $INJ_{Rate}$ [32]. Different hardware vendors and different software drivers define the $NAV$ value differently. The different between $NAV$ values could help towards detecting attackers spoofing the identity of legitimate wireless devices. On top of that, the $NAV$ value is central to the implementation of the virtual jamming attack explained in Chapter 3. An attacker intending to occupy the wireless transmission medium and make all the wireless devices in the network postpone any transmission would use $NAV$ values higher than for normal frames. Some publications in the field of IDS have analysed this metric in order to identify misbehaving wireless nodes. In [82], the authors compare the $NAV$ value against the actual duration of the current transmission to identify inconsistencies in both values. Therefore, finding inconsistency in the $NAV$ value in the analysed frames may evidence the presence of attacks.

#### 6.5.1.6 Sequence Number Difference – SEQ$_{Dif}$

The Sequence Number at the MAC layer is a value included in all the management and data frames, which acts as a counter for each frame that is transmitted by a wireless device. Every single wireless device keeps its own sequence number stream. The sequence number value monotonically increments, from 0 to 4095, every time a non-fragmented data or management frame is transmitted [72]. The sequence number remains constant in all the retransmitted frames [70].

The analysis of the sequence number stream is a common method to identity MAC spoofing attacks. Abrupt changes in the sequence number may indicate the presence of a MAC spoofing attack [10] [69] [71]. If an attacker implements a MAC spoofing attack, the monotonic incrementing series of sequence numbers will produce two different streams of sequence numbers [69] [72]. Two different streams, along with MAC address of the transmitting node can be used to detect this attack.

Traditional detection systems that use the sequence number to identify MAC spoofing attacks are based on the analysis of the difference between the sequence numbers of two consecutive frames, the $SEQ_{Dif}$ [72]. In theory, the value of $SEQ_{Dif}$ should be constantly 1. One simple methodology is keeping tack of the monotonic increment of the sequence number value, and reporting an alarm if the $SEQ_{Dif}$ differs by more than 1. However, this methodology may generate excessive number of false alarms [68]. The occurrence of frame loss, duplicated frames or frames retransmitted out of order makes the $SEQ_{Dif}$ different from 1. Most of the $SEQ_{Dif}$ values are from 0 to 2 [68]. The difference between the sequence numbers of two consecutive frames can also be a negative value. The authors of [68] also report that, if an AP gives priority to beacon and probe response frames before the data frames, the $SEQ_{Dif}$ could be -1 or -2. Therefore, alternative methodologies must be implemented if $SEQ_{Dif}$ is used for intrusion detection. The work presented in [68] proposes a MAC spoofing detection system based on the analysis of the sequence numbers. This system takes into account the fact that the $SEQ_{Dif}$ can be different from 1. It defines bounds in the $SEQ_{Dif}$ and considers that a sequence number is abnormal if the $SEQ_{Dif}$ is any value between 3 and 4092. The authors of this work explain that the proposed system is unable to detect each spoofed frame if the attacker replicates the sequence number of the previous legitimate transmitted frame.

However, some NICs allow changing the sequence number in each frame [71]. Attackers could try to replicate the sequence number of the previous legitimate transmitted frame, in order to avoid being detected. Also, the sequence number cannot be used to identify the attacks that utilise crafted control frames, since these frames do not include sequence number [10]. Additionally, the analysis of the sequence number streams is ineffective in case the legitimate wireless device is not transmitting [71]. In addition, IEEE 802.11e-enabled devices would generate high number of false alarms because this standard allows multiple sequence number streams from the same wireless device [76]. This fact could lead this methodology to generate high number of FPs. Hence, the implementation of detection systems that use the only sequence numbers are insufficiently robust [119].

#### 6.5.1.7 Time To Live – TTL

Finally, the $TTL$ is another metric that should be analysed because it provides evidence of attacks exploiting HTTP sessions. The metric $TTL$ is an 8-bit long field in the IP packets that determines the maximum numbers of hops a packet can make, between the source and destination, before being dropped from the network. Every time an IP packet is forwarded by a router or switch in the network, the $TTL$ value is decremented by 1. Decrementing the $TTL$ value requires modification of the header of the IP packet. The IP packet whose $TTL$ value reaches zero is dropped from the network. The $TTL$ is a mechanism to avoid an IP packet from entering in a loop, hopping indefinitely between routers. The sender of the IP packet sets the $TTL$ value. Commonly, the initial $TTL$ value is set to 32, 64, 128 or 255, depending on the OS of the sender device [109]. Different OSs define different $TTL$ values. Outdates OSs such as Windows 95 use 32 as the $TTL$ value. Linux OS or MacOS use the $TTL$ value 64, whereas recent versions of Windows OS use $TTL$ value 128, as default [134].

An Internet user accessing different websites generates multiple HTTP request packets. Although different websites may be hosted on different web servers across different geographical locations, the content of some websites are hosted in the same web server. In that case, the content of the website will follow an almost similar path through the Internet, before being received by the user. Therefore, all the packets will be assigned similar initial $TTL$ value, and will be forwarded by the almost similar number of routers through the Internet. For instance, in a simple conducted experiment, it was evaluated that the number of hops required to access to Google (http://www.google.co.uk) and the BBC website (http://www.bbc.co.uk), was in both cases 14 hops. The initial $TTL$ value was 64, and the $TTL$ value of the IP packets when arriving to the laptop was 50.

An attacker implementing attacks exploiting HTTP sessions may implement the functionality of assigning the most common initial $TTL$ value to the crafted packets, to try to replicate the behaviour of legitimate web servers. The attacker could be also configured to dynamically adjust the $TTL$ value. These mechanisms would make the receiver believe that the IP packets are actually received from the legitimate web server. Identifying gaps in the $TTL$ value when requesting the content of a website may evidence the attacks exploiting HTTP sessions. The use of the $TTL$ value in task of intrusion detection has been also implemented by [117] [134].

### 6.5.2 Analysis of the Network Traffic Datasets

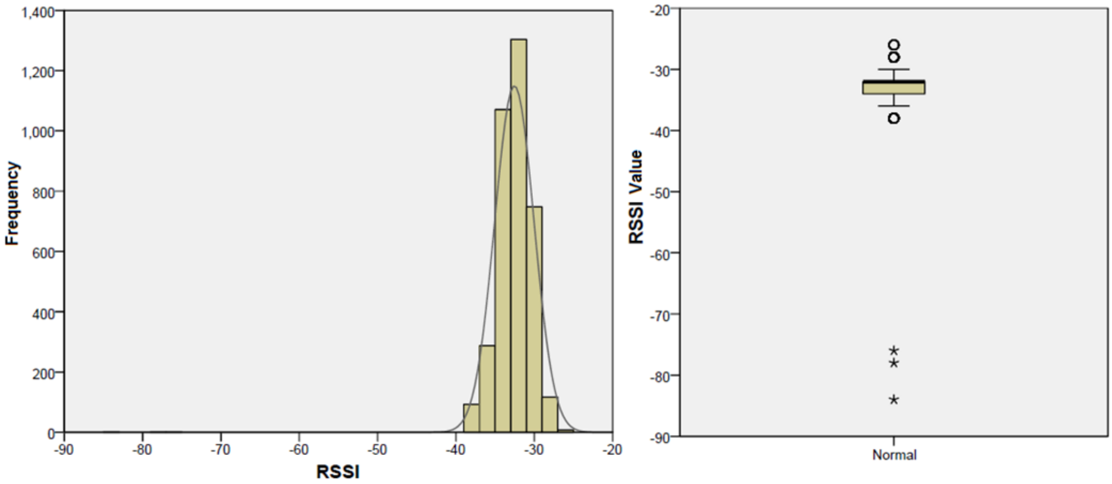
The purpose of this section is to individually analyse each of the described metrics, for each of the datasets and for each type of attack. The analysis will provide a clear understanding of the distribution shape of these metrics, and will indicate if the normal and malicious traffic are statistically differentiable from each other. One dataset has been gathered when no attackers were present in the wireless network. Three separate datasets have been gathered when the Airpwn attack experiments were implemented. Another two datasets have been gathered when the deauthentication attack experiments were implemented.

#### 6.5.2.1 Normal Dataset

The first analysed dataset contains only non-malicious communication traffic between the AP and the wireless client. This experiment is used to evaluate the performance of the proposed detection system, in situations of normal operations. It is important that the proposed detection system does not produce FP alarms when no attacks exist. In total, 3631 network frames compose this dataset of non-malicious wireless network traffic information. 70.8% of this dataset, 2572 instances, is composed of data frames whilst 29.2% of this dataset, 1059 instances, is composed of management frames.

- $RSSI$

The average value for the $RSSI$ in this normal dataset is -32.54, and the standard deviation value is 2.52. Figure 6.4.a shows a histogram representing the frequency of the metric $RSSI$ in this dataset free of malicious instances. The skewness and the kurtosis are -5.454 and 100.084, respectively. Figure 6.4.b also shows the distribution of the $RSSI$ values.

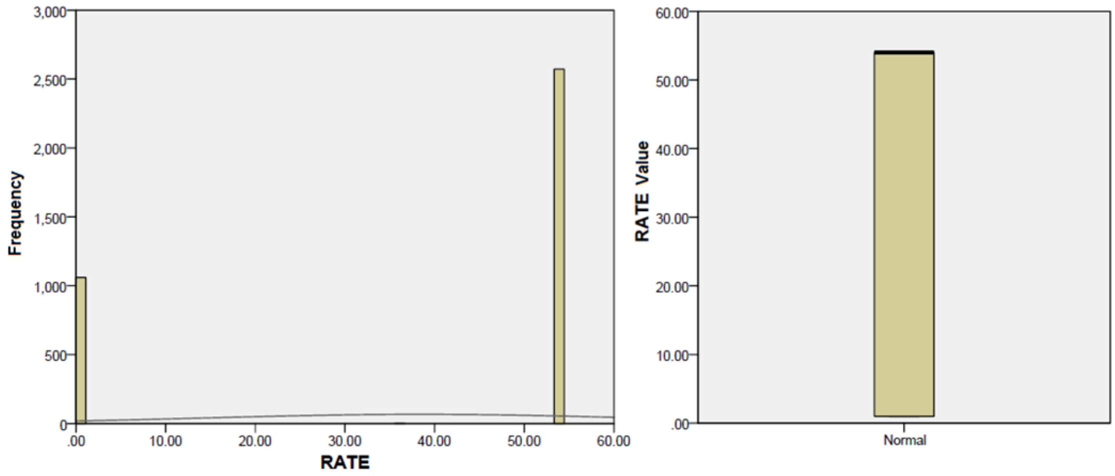


(a) (b)

Figure 6.4 Normal Dataset - $RSSI$: (a) Histogram, (b) Boxplot.

- $INJ_{Rate}$

For the $INJ_{Rate}$, the average value is 38.5373, and the standard deviation value is 24.0918. Figures 6.5.a and 6.5.b show a histogram representing the frequency of the $INJ_{Rate}$ and the boxplot representing the distribution of the $INJ_{Rate}$ in this dataset. The skewness is -0.917, and the kurtosis is -1.16. As can be clearly appreciated, the distributions of both metrics are very dissimilar. In contrast to the values of the $RSSI$ that are mostly concentrated on one particular value, the values of $INJ_{Rate}$ concentrate on two distinct values, distant from each other.

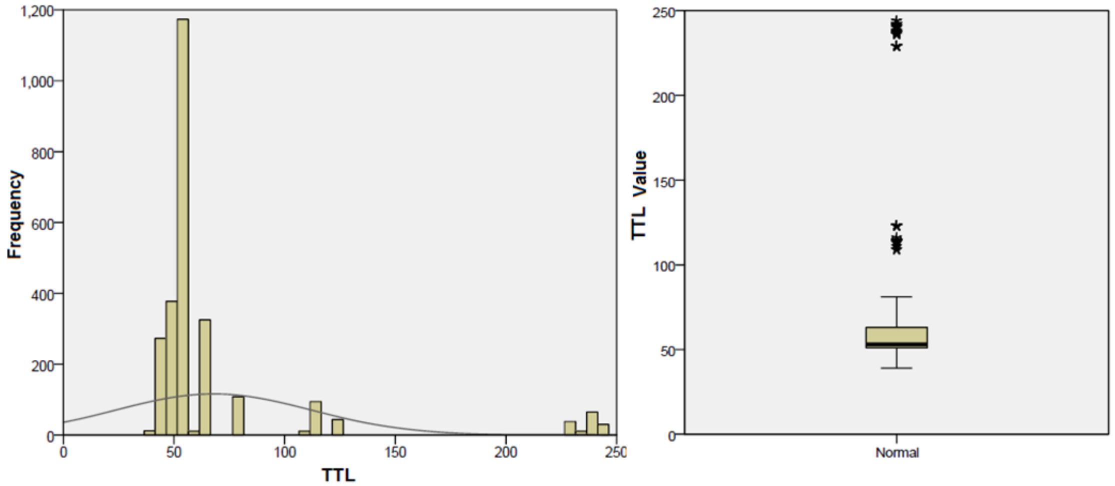


(a) (b)

Figure 6.5 Normal Dataset - $INJ_{Rate}$: (a) Histogram, (b) Boxplot.

- $TTL$

For the $TTL$, the average value is 48.07, and the standard deviation value is 48.356. Again, Figures 6.6.a and 6.6.b show a histogram representing the frequency of the $TTL$ in this dataset, and the boxplot representing the distribution of these values. The skewness is 2.196, and the kurtosis is 6.66. The distribution of this metric has a shape dissimilar to the two previous metrics. Although most of the measurements are also concentrated on one particular value, the values of the $TTL$ are more distributed than the values of the $RSSI$.

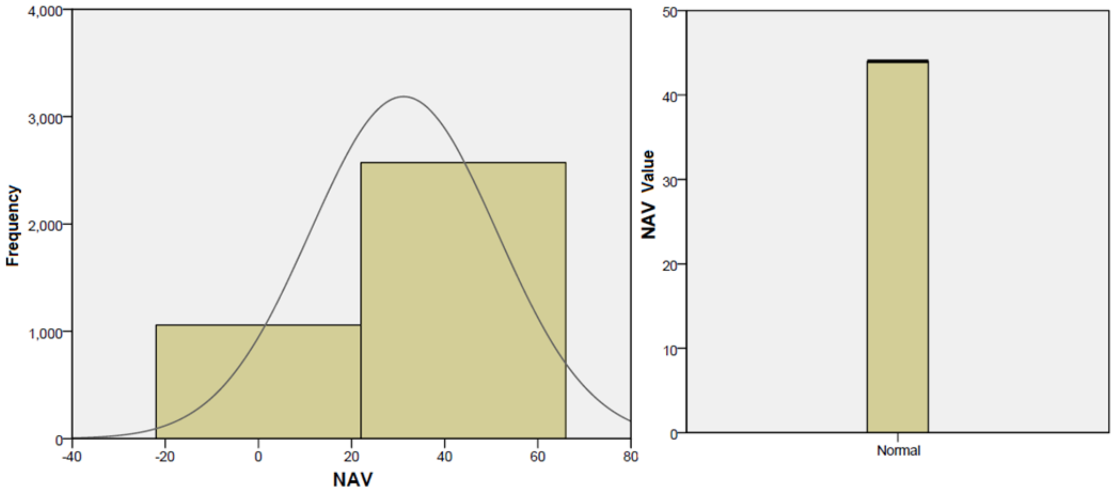


(a) (b)

Figure 6.6 Normal Dataset - $TTL$: (a) Histogram, (b) Boxplot.

- $NAV$

The average value for the $NAV$ is 31.17, and the standard deviation value is 20.002. The frequency of the $NAV$ in this dataset is shown in Figure 6.7.a. The skewness is -0.917, and the kurtosis is -1.159. Figure 6.7.b also shows the distribution of the $NAV$ values. Similar to the $INJ_{Rate}$ , the values of $NAV$ concentrate on two distinct values. However, in this case, the values are not distant from each other.



(a) (b)

Figure 6.7 Normal Dataset - $NAV$: (a) Histogram, (b) Boxplot.

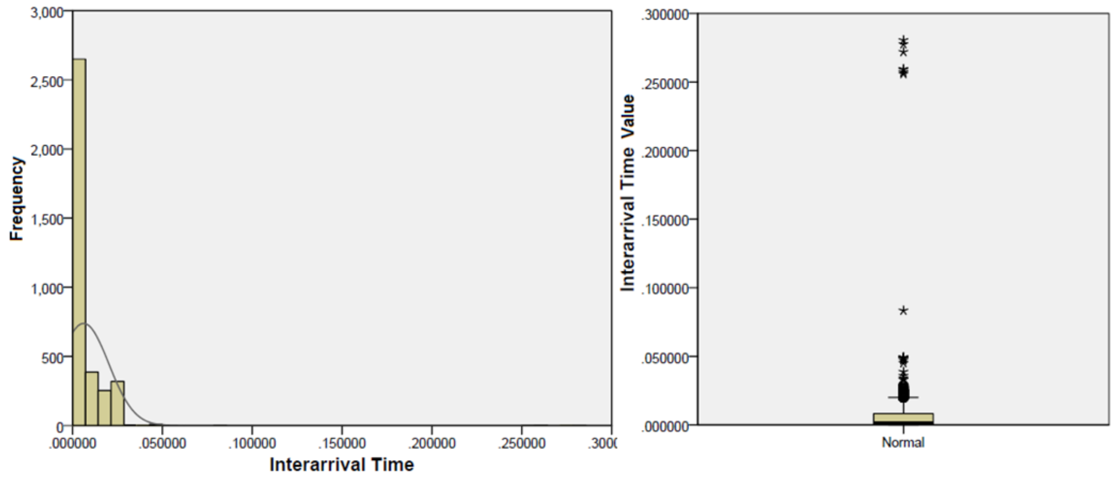
- $\Delta Time$

The average value for $\Delta Time$ is 0.0061, and the standard deviation value is 0.014. The frequency and the distribution of $\Delta Time$ in this dataset are shown in the Figures 6.8.a and 6.8.b. The skewness is 12.713, and the kurtosis is 229.215. Most of the $\Delta Time$ measurements are also concentrated in one particular value, similar to $RSSI$ and $TTL$.

- $SEQ_{Dif}$

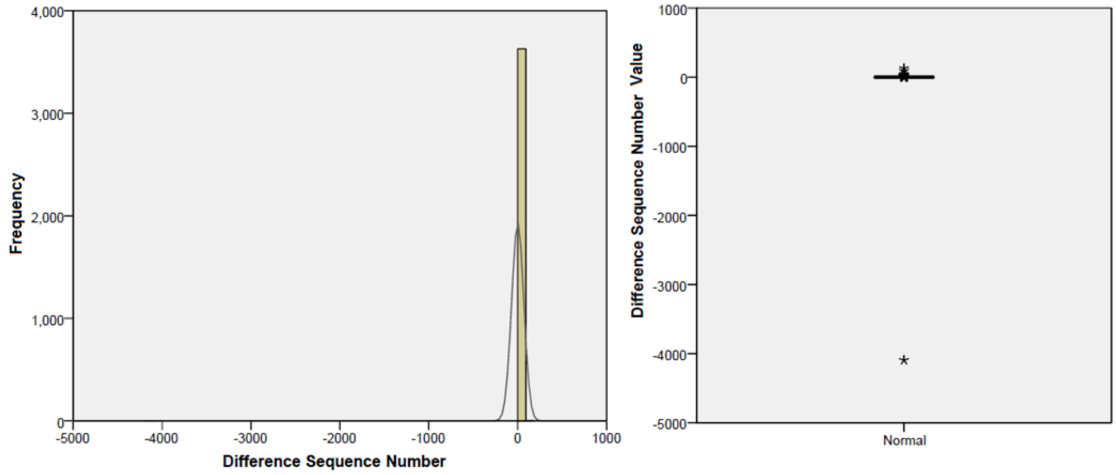
Finally, for $SEQ_{Dif}$, the average value is 0.38, and the standard deviation value is 68.058. The frequency distribution of the $SEQ_{Dif}$ in this dataset is shown in Figures 6.9.a and 6.9.b. The skewness is -60.055, and the kurtosis is 3614.931.

Once more, most of the $SEQ_{Dif}$ measurements are concentrated in one particular value, with outliners produced by normal communication operations.



(a) (b)

Figure 6.8 Normal Dataset - $\Delta Time$: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.9 Normal Dataset - $SEQ_{Dif}$: (a) Histogram, (b) Boxplot.

It is clear that the distributions of the different histograms are statistically dissimilar from each other. Generating a common statistical profile of normal traffic that could adapt to the distribution of all the considered metrics would not be feasible. Numerous FP alarms may be generated. Therefore, an individual statistical profile should be generated for each of the metrics, independently.

### 6.5.2.2 Airpwn Attack Experiments Datasets

#### 6.5.2.2.1 Airpwn Attack 01 Dataset

The testbed from which the second dataset was gathered includes the attacker. This dataset comprises network traffic information from the attacker, and from the communication between the AP and the wireless client. For the following three datasets, the main target is to evaluate the performance of the proposed detection system detecting different types of attacks.

In this particular experiment, the attacker has implemented the first type of Airpwn attack, $Attack01$. This dataset is composed of 1361 network frames in total, 20.1% data frames and 79.9% management frames. 99.2% of this dataset, 1350 instances, is of non-malicious nature. The wireless client sent these frames. The remaining 0.8% of this dataset, 11 instances, is malicious information.

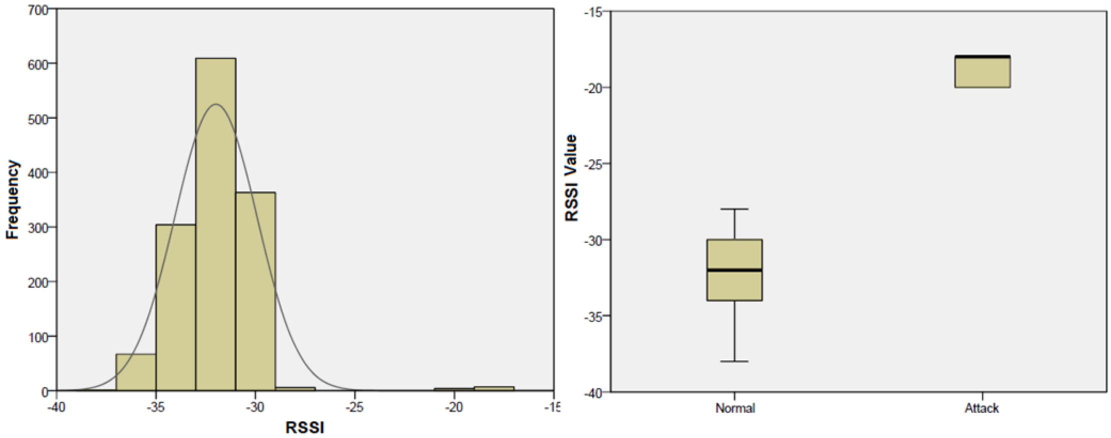
- $RSSI$

  The average value for the $RSSI$ in this dataset containing both malicious and non-malicious instances is -31.99, and the standard deviation value is 2.069. Figure 6.10.a shows a histogram representing the frequency of the metric $RSSI$ in this dataset. The skewness is 1.854, and the kurtosis is 12.383. Figure 6.10.b also shows the distribution of the $RSSI$ values, based on the real nature of the frames. Considering only the value for the $RSSI$ of the non-malicious frames in this dataset, the average value is -32.1, and the standard deviation value is 1.691. Considering only the value for the $RSSI$ of the malicious frames, the average value and the standard deviation value are -18.73 and 1.009, respectively.

- $INJ_{Rate}$

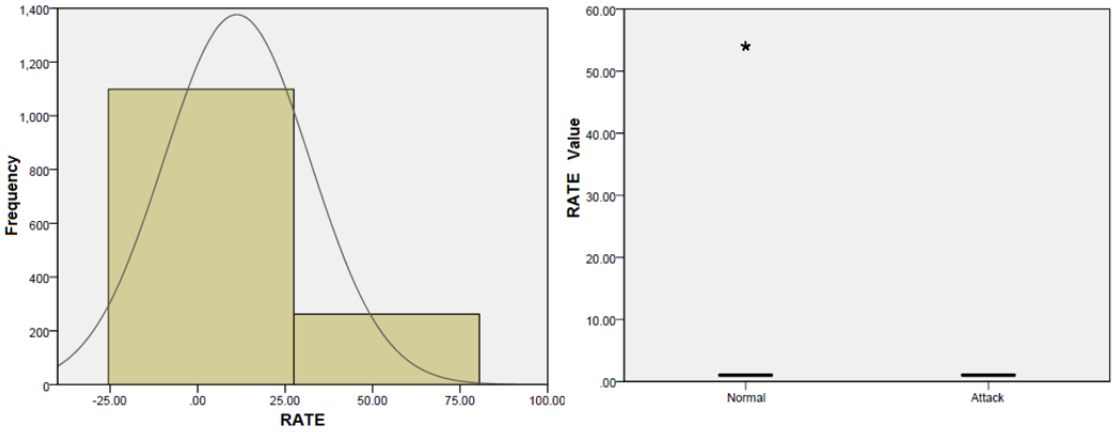
  For the $INJ_{Rate}$, the average value is 11.2028, and the standard deviation value is 20.9038. Figure 6.11.a shows a histogram representing the frequency distribution of $INJ_{Rate}$ in this dataset. The skewness is 1.562, and the kurtosis is 0.439. Figure

6.11.b shows the distribution of the $INJ_{Rate}$ values, based on the real nature of the frames. Considering only the value for the $INJ_{Rate}$ of the non-malicious frames in this dataset, the average value is 11.2859, and the standard deviation value is 20.9685. Whilst, considering only the value for the $INJ_{Rate}$ of the malicious frames, the average value and the Standard Deviation value are 1 and 0, respectively.



(a) (b)

Figure 6.10 Airpwn Dataset - $RSSI\ Attack01$ Traffic: (a) Histogram, (b) Boxplot.

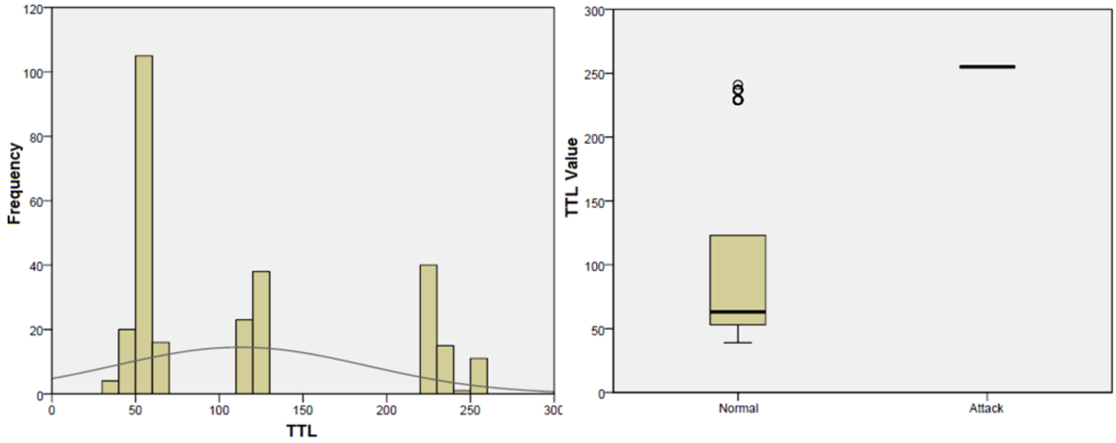


(a) (b)

Figure 6.11 Airpwn Dataset - $INJ_{Rate}\ Attack01$ Traffic: (a) Histogram, (b) Boxplot.

- $TTL$

For the $TTL$, the average value is 22.57, and the standard deviation value is 56.249. Again, Figures 6.12.a and 6.12.b show a histogram representing the frequency of the $TTL$ in this dataset, and the distribution of the $TTL$ values, respectively. The skewness is 2.881, and the kurtosis is 7.663. Considering only the value for the $TTL$ of the non-malicious frames in this dataset, the average value is 20.68, and the standard deviation value is 52.399. Considering only the value for the $TTL$ of the malicious frames, the average value and the standard deviation value are 255 and 0, respectively.



(a) (b)

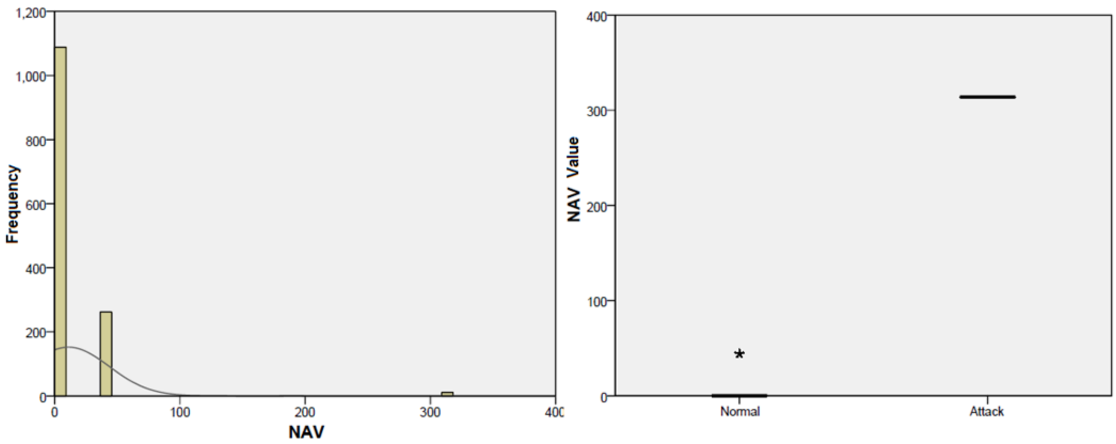
Figure 6.12 Airpwn Dataset - $TTL$ $Attack01$ Traffic: (a) Histogram, (b) Boxplot.

- $NAV$

The average value for the $NAV$ is 11.01, and the standard deviation value is 32.391. The frequency distribution of the $NAV$ in this dataset is shown in Figure 6.13.a. The skewness is 6.803, and the kurtosis is 59.415. The distribution of the $NAV$ values, based on the real nature of the frames, is shown in Figure 6.13.b. Considering only the value for the $NAV$ of the non-malicious frames in this dataset, the average value is 8.54, and the standard deviation value is 17.408. Considering only the value for the $NAV$ of the malicious frames, the average value and the standard deviation value are 314 and 0, respectively.

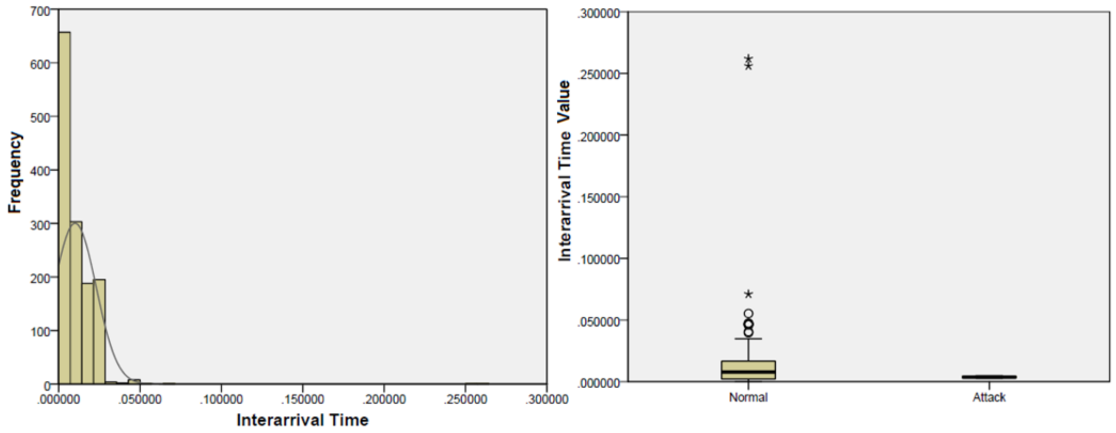
- $\Delta Time$

The average value for $\Delta Time$ is 0.101, and the standard deviation value is 0.0129. The frequency of $\Delta Time$ in this dataset is shown in the Figure 6.14.a. The skewness is 10.886, and the kurtosis is 202.704. The boxplot representing the distribution of the $\Delta Time$ values is shown in the figure 6.14.b. Considering only the value for the $\Delta Time$ of the non-malicious frames in this dataset, the average value is 0.0102, and the standard deviation value is 0.0129. Considering only the value for the $\Delta Time$ of the malicious frames, the average value is 0.0036 and the standard deviation value is 0.0009.



(a) (b)

Figure 6.13 Airpwn Dataset - $NAV\ Attack01$ Traffic: (a) Histogram, (b) Boxplot.

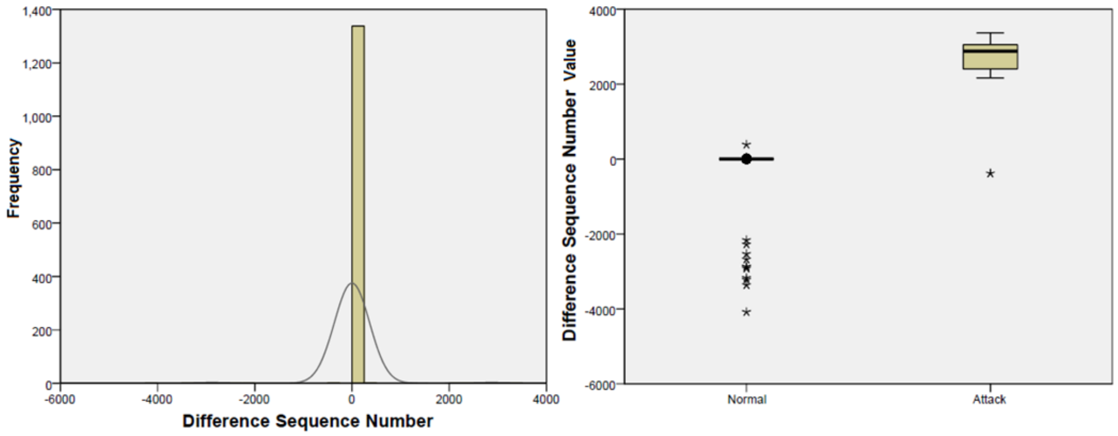


(a) (b)

Figure 6.14 Airpwn Dataset - $\Delta Time\ Attack01$ Traffic: (a) Histogram, (b) Boxplot.

- $SEQ_{Dif}$

Finally, for $SEQ_{Dif}$, the average value is -1.07, and the standard deviation value is 361.446. Figure 6.15.a shows a histogram representing the frequency of the metric $SEQ_{Dif}$ in this dataset, whilst Figure 6.15.b shows the distribution of the $SEQ_{Dif}$ values, based on the real nature of the frames. The skewness is -1.046, and the kurtosis is 68.64. Considering only the value for $SEQ_{Dif}$ of the non-malicious frames in this dataset, the average value is -21.61, and the standard deviation value is 267.468. Considering only the value for $SEQ_{Dif}$ of the malicious frames, the average value and the standard deviation value are 2519.82 and 1034.159, respectively.



(a) (b)

Figure 6.15 Airpwn Dataset - $SEQ_{Dif}$ $Attack01$ Traffic: (a) Histogram, (b) Boxplot.

Similar to the datasets that only contains non-malicious information, the distributions of the different histograms are statistically dissimilar from each other. Comparing the distributions of the histograms of the dataset $Attack01$ against the distributions of the histograms of the Normal traffic dataset, by metric have a very similar distribution. Focusing on the difference between the normal and malicious instances, both types of information are statistically differentiable. The difference is more noticeable in some of the metrics such as $NAV$ than others. This statistical difference is visually represented by the boxplots.

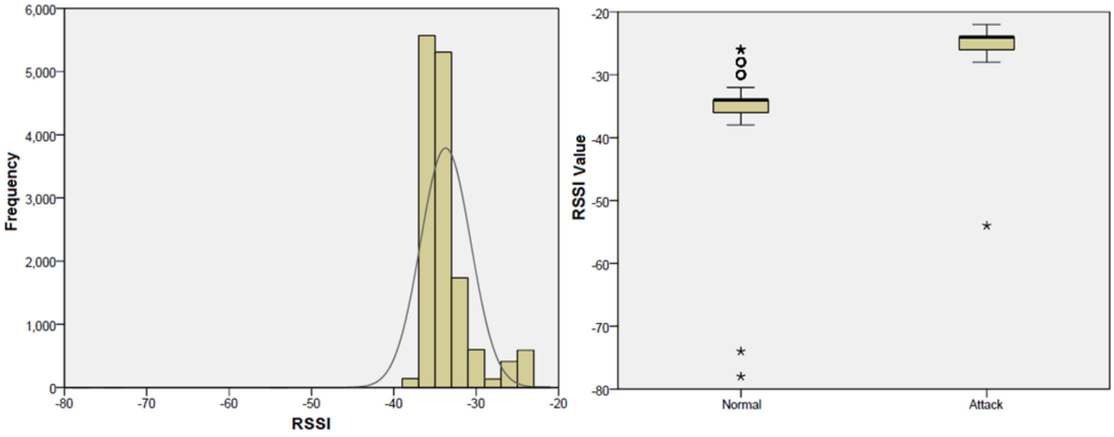
#### 6.5.2.2.2 Airpwn Attack 02 Dataset

For the third dataset, the attacker implemented the second type of Airpwn attack, $Attack02$. This dataset, which is substantially larger that the two previous datasets, also comprises network traffic information from the attacker, and from the communication between the AP and the wireless client. This dataset is composed of 14493 network frames in total, 90.6% data frames and 9.4% management frames. 93.1% of this dataset, 13498 instances, is of non-malicious nature, whilst the remaining 6.9% of this dataset, 995 instances, is malicious information. The analysis of different metrics in this dataset is tabulated in Table VI.III. Histograms representing the frequency different metrics in this dataset, as well as bloxplot diagrams representing the distribution of the metrics are shown in Figures 6.16.a – 6.21.b.

TABLE VI.III. AIRPWN $Attack02$ DATASET ANALYSIS RESULTS

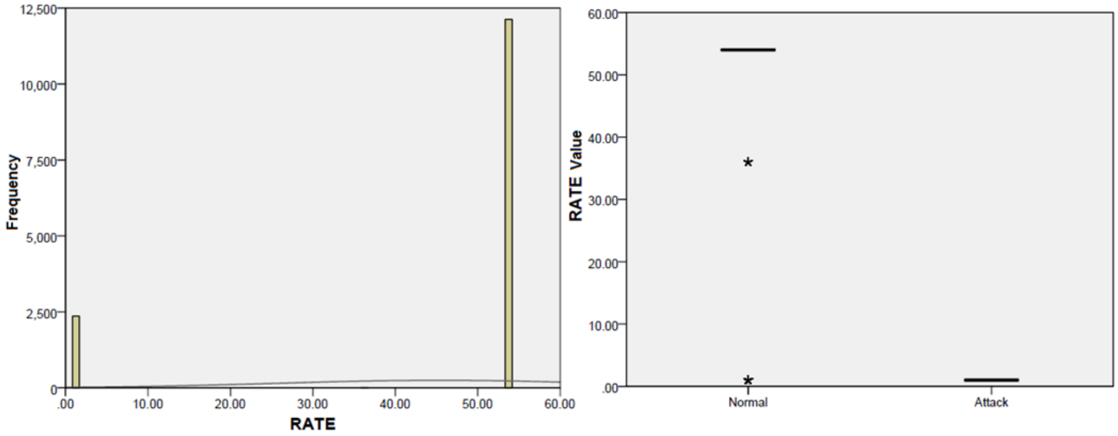
| *Dataset Content* | *Average* | *Standard Deviation* | *Skewness* | *Kurtosis* |
|---|---|---|---|---|
| $RSSI$ | | | | |
| *Normal - Malicious* | -33.72 | 3.049 | 1.378 | 8.135 |
| *Normal* | -34.37 | 1.894 | | |
| *Malicious* | -24.82 | 1.374 | | |
| $INJ_{Rate}$ | | | | |
| *Normal - Malicious* | 45.3609 | 19.566 | -1.825 | 1.334 |
| *Normal* | 48.6309 | 15.9778 | | |
| *Malicious* | 1 | 0 | | |
| $TTL$ | | | | |
| *Normal - Malicious* | 73.25 | 67.461 | 1.895 | 2.379 |
| *Normal* | 59.85 | 47.663 | | |
| *Malicious* | 255 | 0 | | |

| | | | | |
|---|---|---|---|---|
| $NAV$ | | | | |
| *Normal - Malicious* | 58.4 | 70.57 | 3.02 | 8.866 |
| *Normal* | 39.55 | 13.262 | | |
| *Malicious* | 314 | 0 | | |
| $\Delta Time$ | | | | |
| *Normal - Malicious* | 0.0023 | 0.0094 | 29.526 | 1285.574 |
| *Normal* | 0.0019 | 0.0079 | | |
| *Malicious* | 0.0077 | 0.0204 | | |
| $SEQ_{Dif}$ | | | | |
| *Normal - Malicious* | -0.05 | 410.425 | -0.397 | 43.102 |
| *Normal* | 9.05 | 302.615 | | |
| *Malicious* | -123.49 | 1093.643 | | |
| *Dataset Content* | *Average* | *St. Deviation* | *Skewness* | *Kurtosis* |



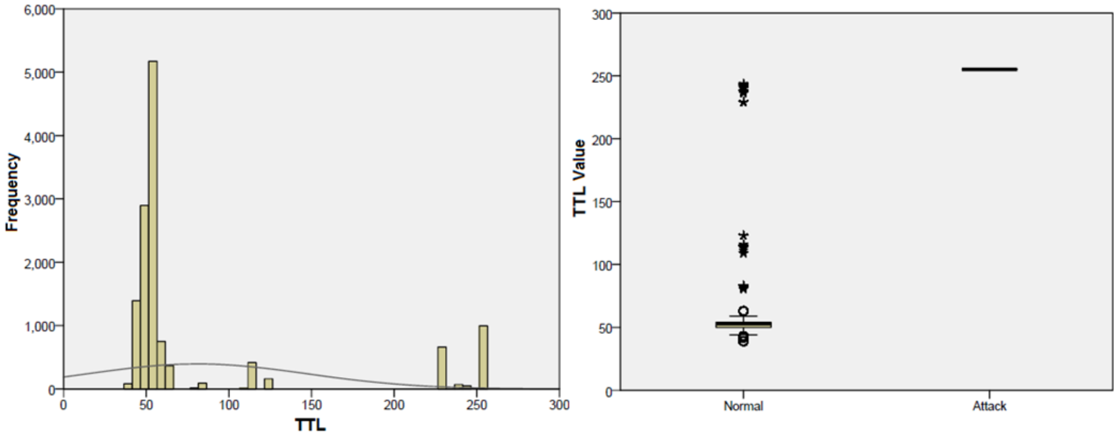
(a) (b)

Figure 6.16 Airpwn Dataset - $RSSI\ Attack02$ Traffic: (a) Histogram, (b) Boxplot.



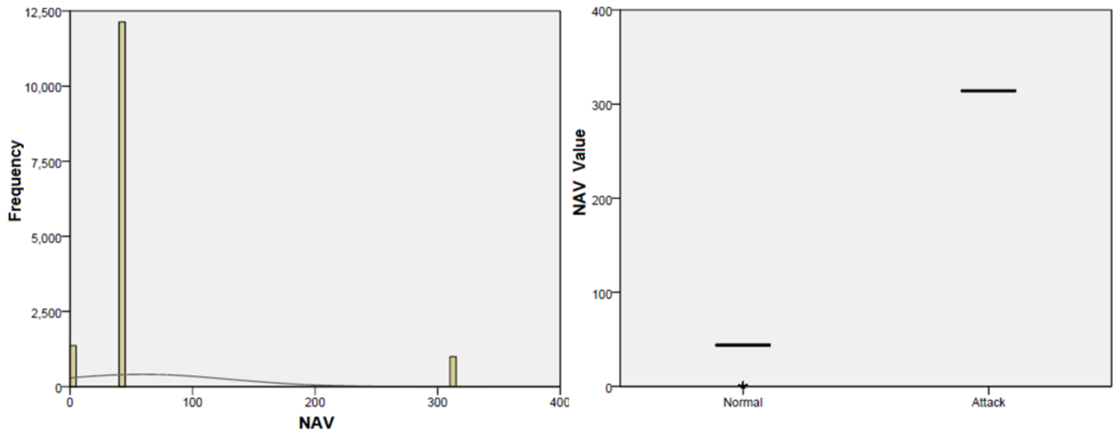
(a) (b)

Figure 6.17 Airpwn Dataset - $INJ_{Rate}$ $Attack02$ Traffic: (a) Histogram, (b) Boxplot.



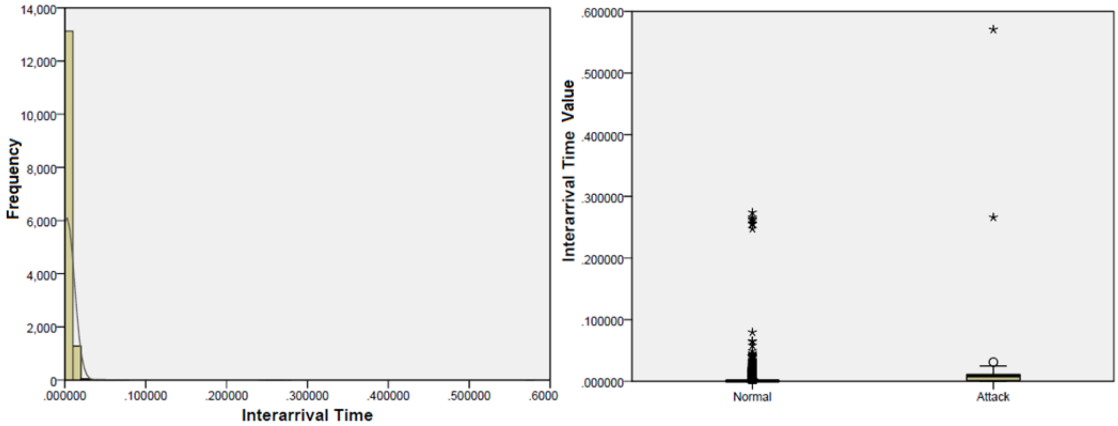
(a) (b)

Figure 6.18 Airpwn Dataset - $TTL$ $Attack02$ Traffic: (a) Histogram, (b) Boxplot.



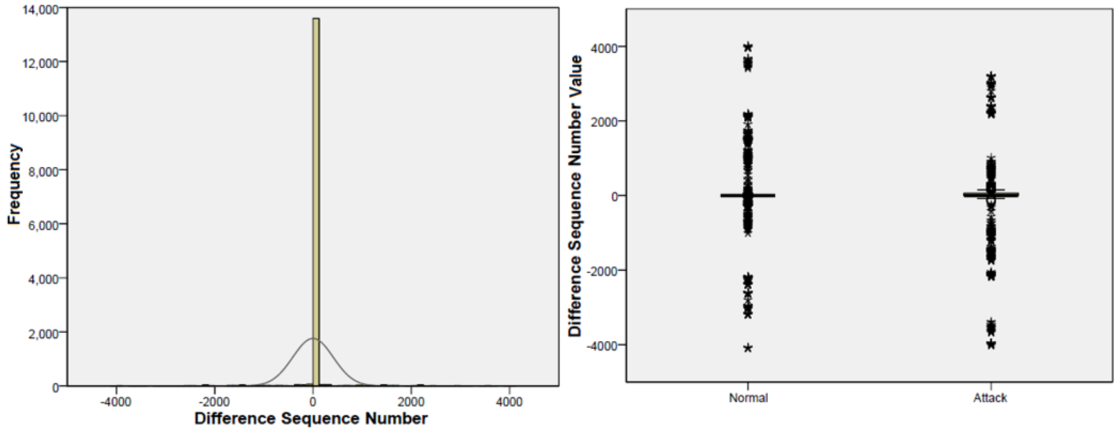
(a) (b)

Figure 6.19 Airpwn Dataset - $NAV$ $Attack02$ Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.20 Airpwn Dataset - $\Delta Time$ $Attack02$ Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.21 Airpwn Dataset - $SEQ_{Dif}$ $Attack02$ Traffic: (a) Histogram, (b) Boxplot.

The distributions of the different metrics in the dataset $Attack02$ have apparent similarities to the previous two datasets. The difference in the amount of information contained in this dataset does not significantly change the statistical characteristics of the metrics distributions. Focusing on the difference between the normal and malicious instances, both types of information are also statistically differentiable, similar to the dataset $Attack01$. This statistical difference is visually represented by the different boxplots.

#### 6.5.2.2.3 Mixed Airpwn Attack Dataset

One last dataset contains both network traffic information from the attacker and from the wireless communication between the legitimate devices. In contrast to the previous datasets, this experiment alternates periods of only normal wireless traffic with periods in which the attacker implements attacks. An initial period of only normal wireless traffic was followed by a defined period of time in which the attacker implemented the $Attack01$. Then, another period of normal traffic was followed by another period of malicious activity. This last time, the attacker implemented the $Attack02$.

The main purpose was to evaluate the performance of the proposed detection system in situations in which different attacks were present. The adaptability of the proposed detection system to different attacks is verified through this experiment. This dataset is composed of 12130 network frames in total, 88.1% data frames and 11.9% management frames. 99.1% of this dataset, 12016 instances, is of non-malicious nature. The remaining 0.9%, 114 instances, is malicious information. Focusing on the particular type of attack, 6.14% of the malicious information in this dataset, 7 instances, corresponds to the $Attack01$, whereas 93.86% of this dataset, 107 instances, corresponds to the $Attack02$. Figure 6.22 provides a graphical representation of the proportion of the instances present in the dataset.

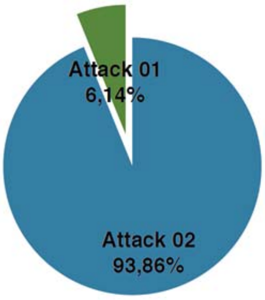


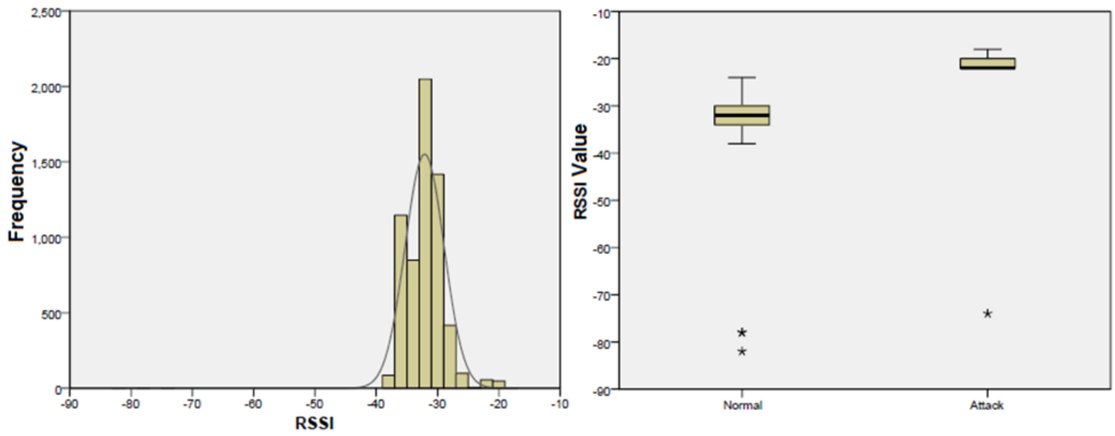
Figure 6.22 Mixed Airpwn Attack Dataset Information Proportion.

The analysis of different metrics in this dataset, $Mixed\ Attack$, is tabulated in Table VI.IV. Histograms representing the frequency different metrics in this dataset, as well as bloxplot diagrams representing the distribution of the metrics are shown in Figures 6.23.a – 6.28.b.

TABLE VI.IV. AIRPWN $Mixed\ Attack$ DATASET ANALYSIS RESULTS

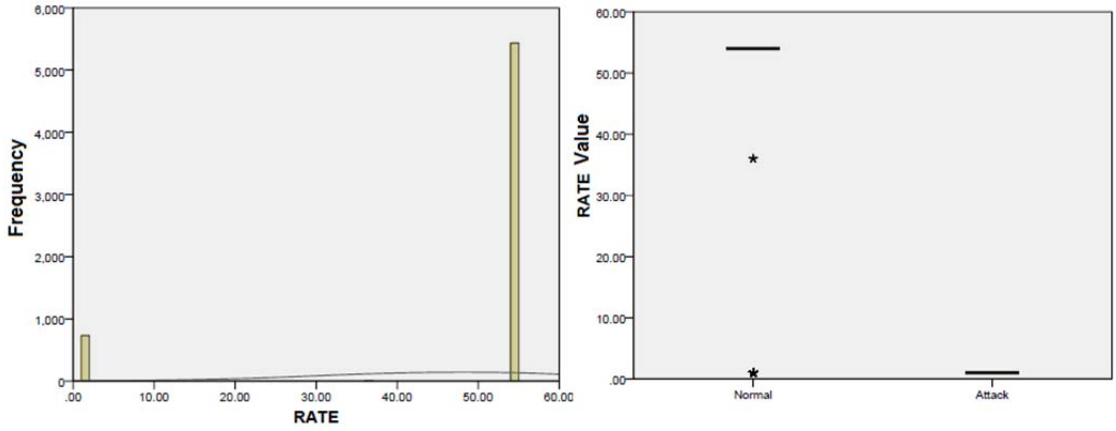
| *Dataset Content* | *Average* | *Standard Deviation* | *Skewness* | *Kurtosis* |
|---|---|---|---|---|
| $RSSI$ | | | | |
| *Normal - Malicious* | -32.11 | 2.606 | -1.176 | 34.203 |
| *Normal* | -32.21 | 2.354 | | |
| *Malicious* | -21.53 | 5.065 | | |
| $INJ_{Rate}$ | | | | |
| *Normal - Malicious* | 47.1866 | 17.7299 | -2.22 | 2.933 |
| *Normal* | 47.6248 | 17.2308 | | |
| *Malicious* | 1 | 0 | | |
| $TTL$ | | | | |
| *Normal - Malicious* | 62.76 | 54.651 | 2.268 | 4.89 |
| *Normal* | 60.93 | 51.584 | | |
| *Malicious* | 255 | 0 | | |
| $NAV$ | | | | |
| *Normal - Malicious* | 41.3 | 30.137 | 6.659 | 60.459 |
| *Normal* | 38.72 | 14.304 | | |
| *Malicious* | 314 | 0 | | |
| $\Delta Time$ | | | | |
| *Normal - Malicious* | 0.0022 | 0.0072 | 24.358 | 858.046 |
| *Normal* | 0.0021 | 0.0072 | | |
| *Malicious* | 0.0066 | 0.0049 | | |
| $SEQ_{Dif}$ | | | | |

| $SEQ_{Dif}$ | | | | |
|---|---|---|---|---|
| *Normal - Malicious* | -0.11 | 159.712 | -6.792 | 227.98 |
| *Normal* | 5.31 | 127.928 | | |
| *Malicious* | -571.83 | 815.421 | | |
| *Dataset Content* | *Average* | *St. Deviation* | *Skewness* | *Kurtosis* |



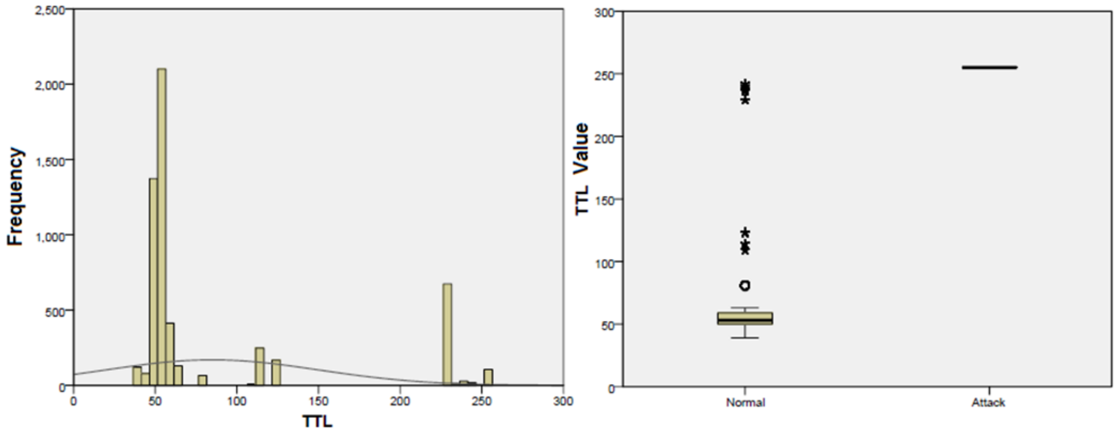
(a) (b)

Figure 6.23 Airpwn Dataset - $RSSI\ Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.



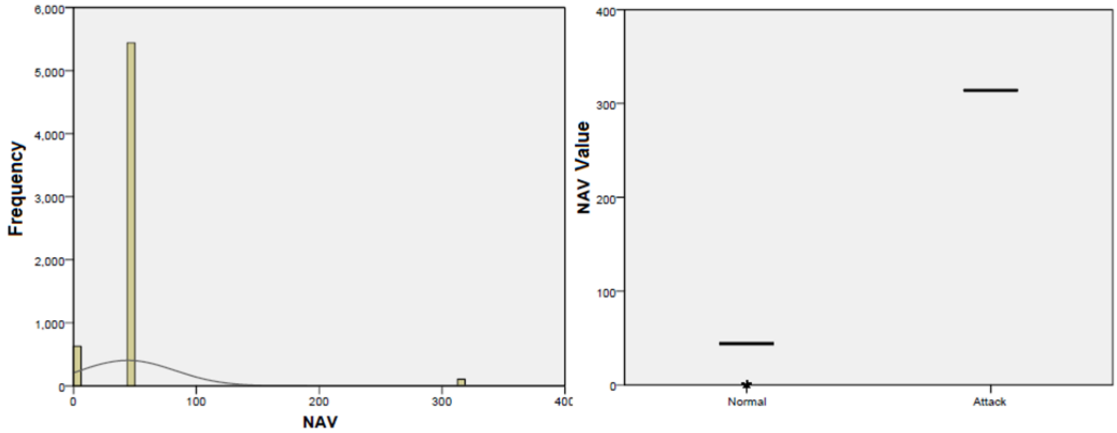
(a) (b)

Figure 6.24 Airpwn Dataset - $INJ_{Rate}\ Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.



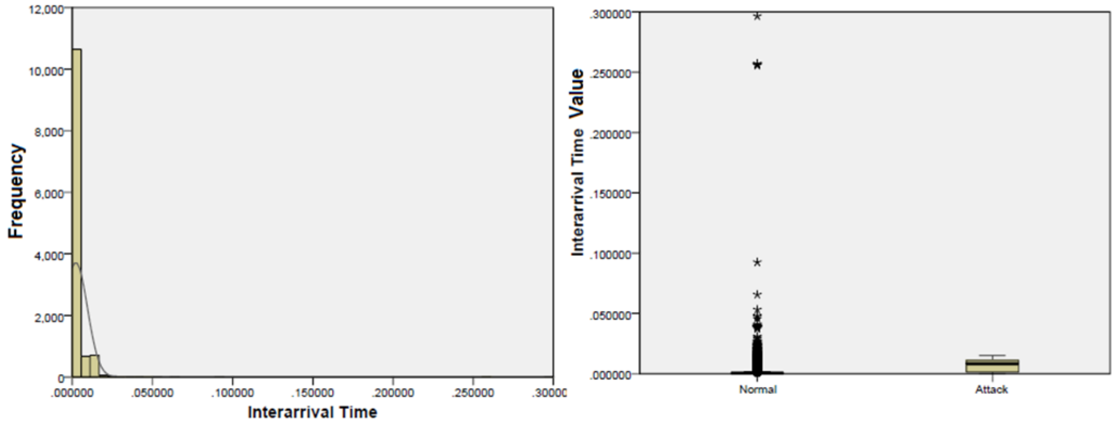
(a) (b)

Figure 6.25 Airpwn Dataset - $TTL\ Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.



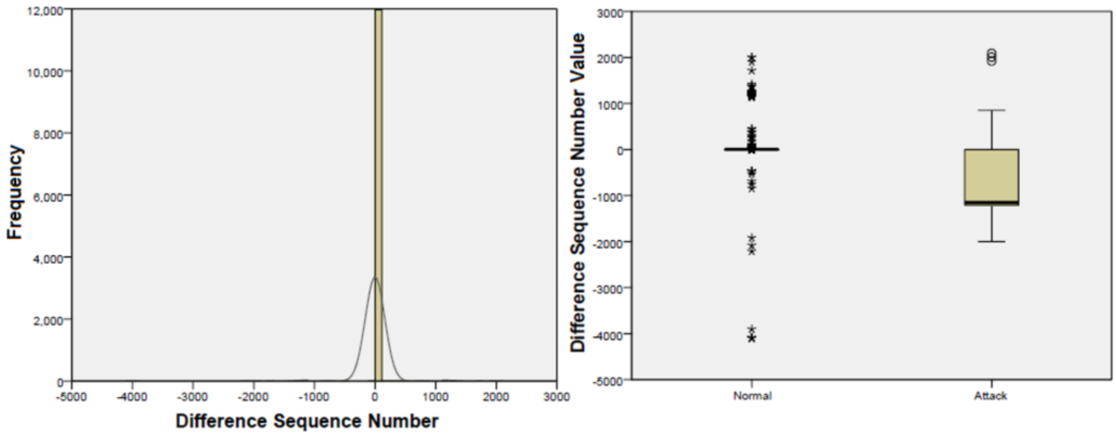
(a) (b)

Figure 6.26 Airpwn Dataset - $NAV\ Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.27 Airpwn Dataset - $\Delta Time\ Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.28 Airpwn Dataset - $SEQ_{Dif}$ $Mixed\ Attack$ Traffic: (a) Histogram, (b) Boxplot.

### 6.5.2.3 Deauthentication Attack Experiments Datasets

#### 6.5.2.3.1 Short Distance Datasets

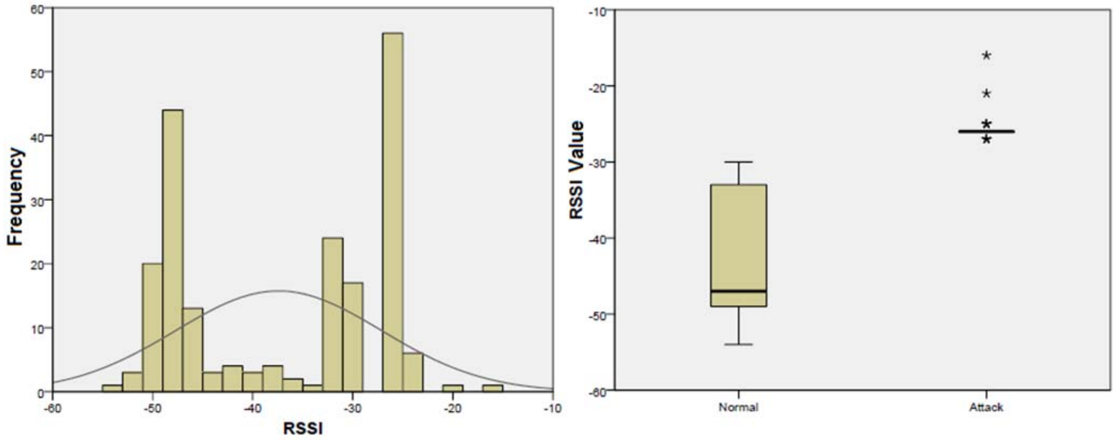
Another attack implemented in this thesis was the deauthentication attack. For this type of attack, two separate datasets were gathered. In the first dataset, the attacker was placed close to the AP, around 1.5 metres away. This dataset is composed of 203 network frames in total. 139 instances, 68.5% of the dataset, are normal frames from the communication between the AP and the wireless client. The remaining 31.5% of the dataset, 64 deauthentication frames, is malicious. The analysis of different metrics in this dataset is tabulated in Table VI.V.

TABLE VI.V. DEAUTHENTICATION SHORT DISTANCE DATASET ANALYSIS RESULTS

| *Dataset Content* | *Average* | *Standard Deviation* | *Skewness* | *Kurtosis* |
|---|---|---|---|---|
| *RSSI* | | | | |
| *Normal - Malicious* | -37.41 | 10.295 | -0.075 | -1.69 |
| *Normal* | -42.78 | 7.877 | | |
| *Malicious* | -25.75 | 1.447 | | |

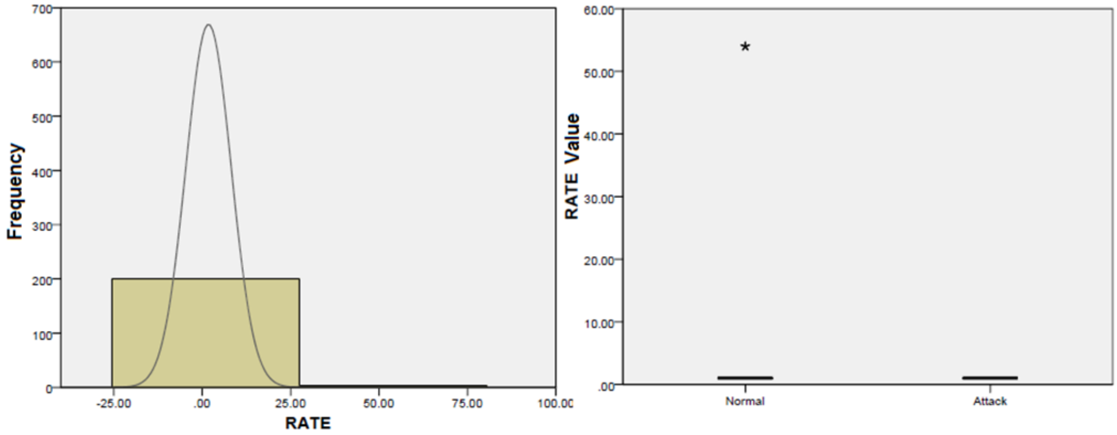
| | | | | |
|---|---|---|---|---|
| $INJ_{Rate}$ | | | | |
| *Normal - Malicious* | 1.7833 | 6.411 | 8.102 | 64.284 |
| *Normal* | 2.1439 | 7.7296 | | |
| *Malicious* | 1 | 0 | | |
| $NAV$ | | | | |
| *Normal - Malicious* | 129.03 | 154.301 | 0.37 | -1.878 |
| *Normal* | 43.87 | 108.066 | | |
| *Malicious* | 314 | 0 | | |
| $\Delta Time$ | | | | |
| *Normal - Malicious* | 0.0499 | 0.0487 | 0.105 | -1.976 |
| *Normal* | 0.0711 | 0.0446 | | |
| *Malicious* | 0.0038 | 0.0095 | | |
| $SEQ_{Dif}$ | | | | |
| *Normal - Malicious* | 0.86 | 23.246 | -1.025 | 20.039 |
| *Normal* | -4.12 | 20.905 | | |
| *Malicious* | 11.67 | 24.531 | | |
| *Dataset Content* | *Average* | *St. Deviation* | *Skewness* | *Kurtosis* |

Histograms representing the frequency different metrics in this dataset, as well as bloxplot diagrams representing the distribution of the metrics are shown in Figures 6.29.a – 6.33.b.



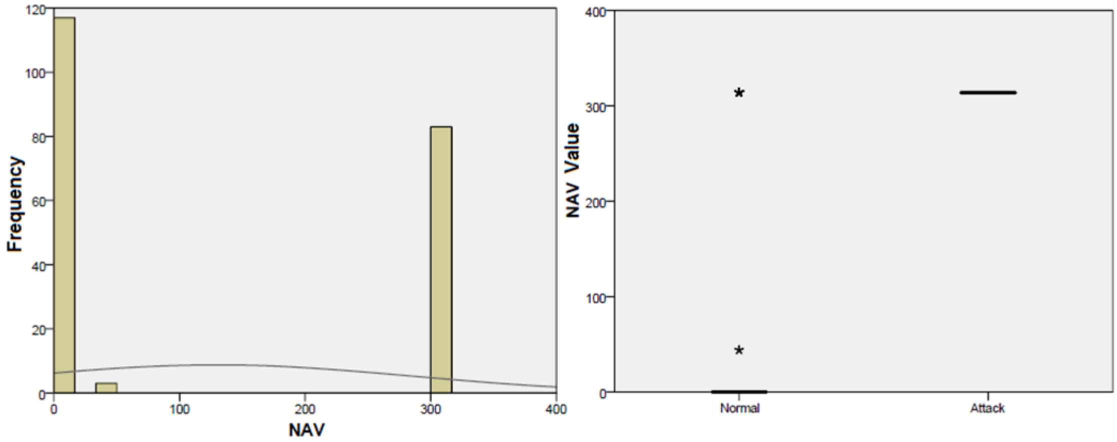
(a) (b)

Figure 6.29 Deauthentication Attack - $RSSI$ Short Distance Traffic: (a) Histogram, (b) Boxplot.



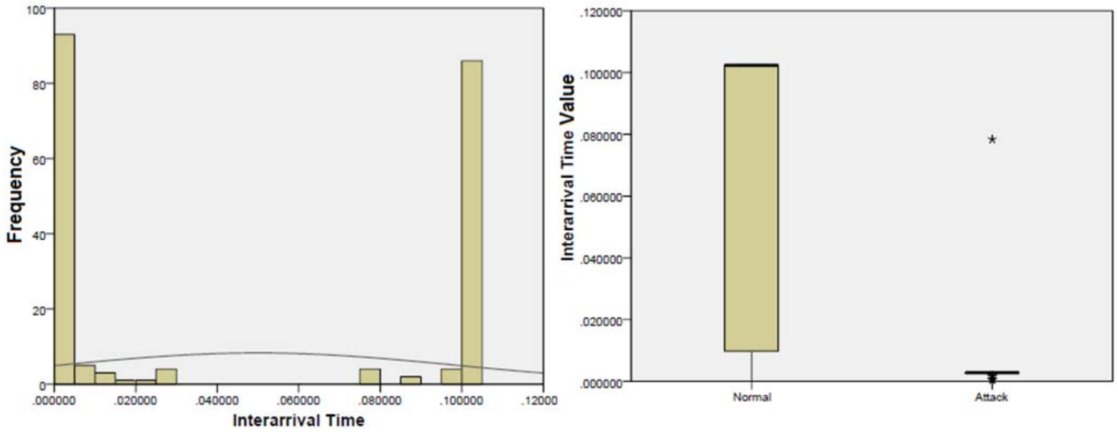
(a) (b)

Figure 6.30 Deauthentication Attack - $INJ_{Rate}$ Short Distance Traffic: (a) Histogram, (b) Boxplot.



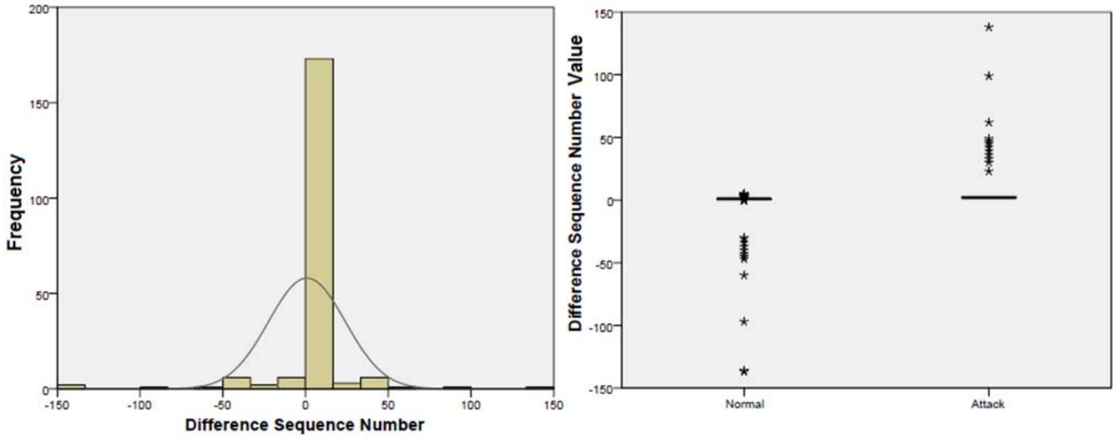
(a) (b)

Figure 6.31 Deauthentication Attack - $NAV$ Short Distance Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.32 Deauthentication Attack - $\Delta Time$ Short Distance Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.33 Deauthentication Attack - $SEQ_{Dif}$ Short Distance Traffic: (a) Histogram, (b) Boxplot.

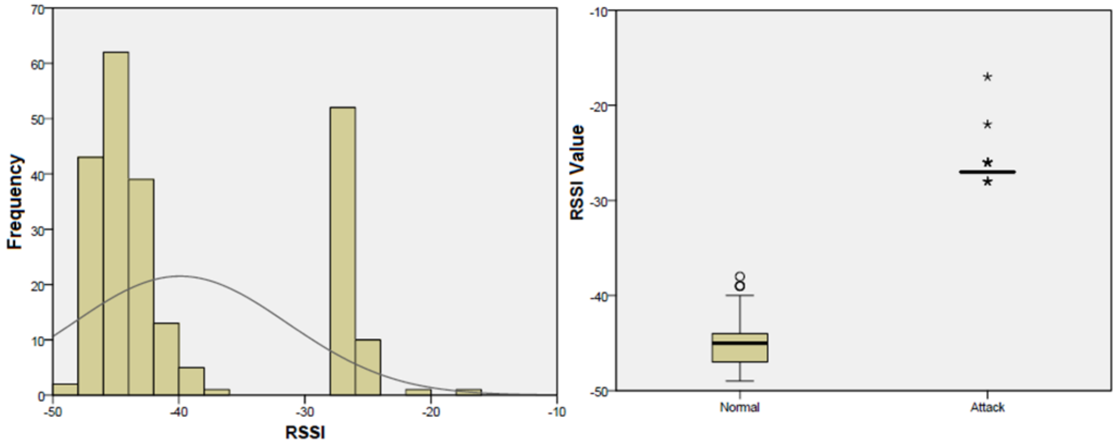
### 6.5.2.3.2 Long Distance Datasets

In the second experiment, a similar deauthentication attack was implemented. However, the testbed from which the second dataset was gathered included a modification in the distances between the attacker and the AP. The attacker was placed farther away from the AP, around 10 metres away. This modification of the position of the attacker will alter the metric values of the network traffic datasets. Therefore the main objective for this change is to evaluate how changes in the testbed topology might affect the performance of the proposed detection system. This dataset is composed of 229 network frames in total. 72.1% of this dataset, 165 instances, is of non-malicious nature. The remaining 27.9%, 64 instances, is malicious information. The analysis of different metrics in this dataset is tabulated in Table VI.VI. Histograms representing the frequency different metrics in this dataset, as well as bloxplot diagrams representing the distribution of the metrics are shown in Figures 6.34.a – 6.33.b.

TABLE VI.VI. DEAUTHENTICATION LONG DISTANCE DATASET ANALYSIS RESULTS

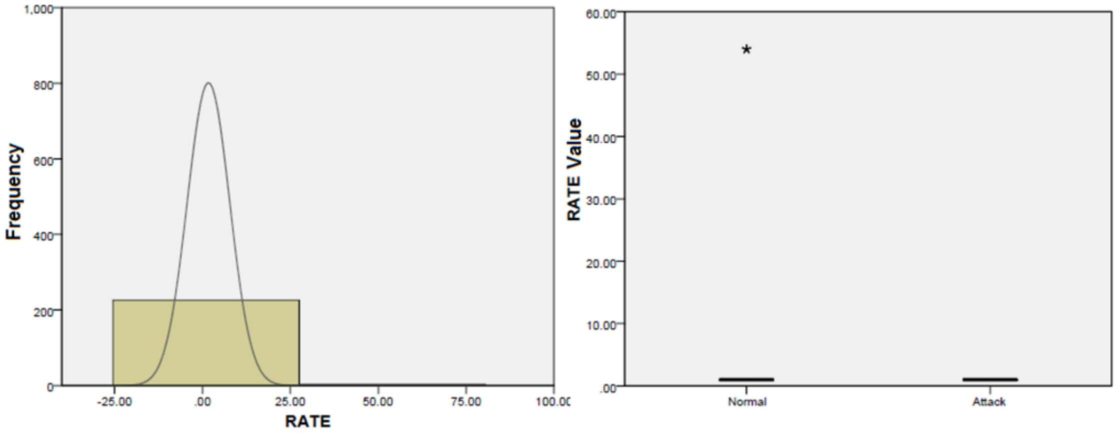
| *Dataset Content* | *Average* | *Standard Deviation* | *Skewness* | *Kurtosis* |
|---|---|---|---|---|
| | | $RSSI$ | | |
| *Normal - Malicious* | -39.91 | 8.495 | 0.879 | -0.976 |
| *Normal* | -45.04 | 1.072 | | |
| *Malicious* | -26.67 | 1.448 | | |
| | | $INJ_{Rate}$ | | |
| *Normal - Malicious* | 1.6943 | 6.0395 | 8.621 | 72.956 |
| *Normal* | 1.9636 | 7.1028 | | |
| *Malicious* | 1 | 0 | | |
| | | $NAV$ | | |
| *Normal - Malicious* | 111.64 | 150.115 | 0.613 | -1.635 |
| *Normal* | 33.15 | 95.656 | | |
| *Malicious* | 314 | 0 | | |
| | | $\Delta Time$ | | |
| *Normal - Malicious* | 0.0558 | 0.0484 | -0.137 | -1.947 |
| *Normal* | 0.0761 | 0.0417 | | |
| *Malicious* | 0.0035 | 0.0096 | | |

| | | | | |
|---|---|---|---|---|
| $SEQ_{Dif}$ | | | | |
| *Normal - Malicious* | 0.93 | 461.067 | -0.035 | 7.625 |
| *Normal* | -97.99 | 372.782 | | |
| *Malicious* | 225.97 | 562.039 | | |
| *Dataset Content* | *Average* | *St. Deviation* | *Skewness* | *Kurtosis* |



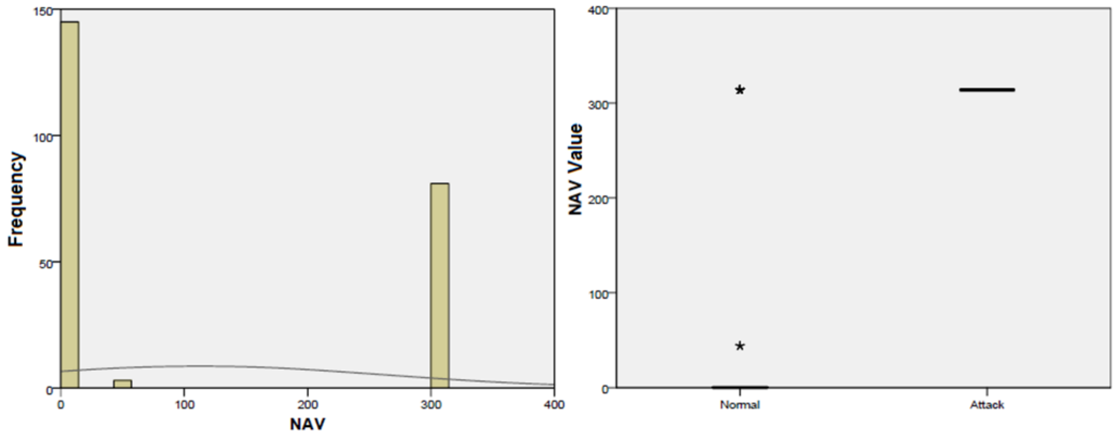
(a) (b)

Figure 6.34 Deauthentication Attack - *RSSI* Long Distance Traffic: (a) Histogram, (b) Boxplot.



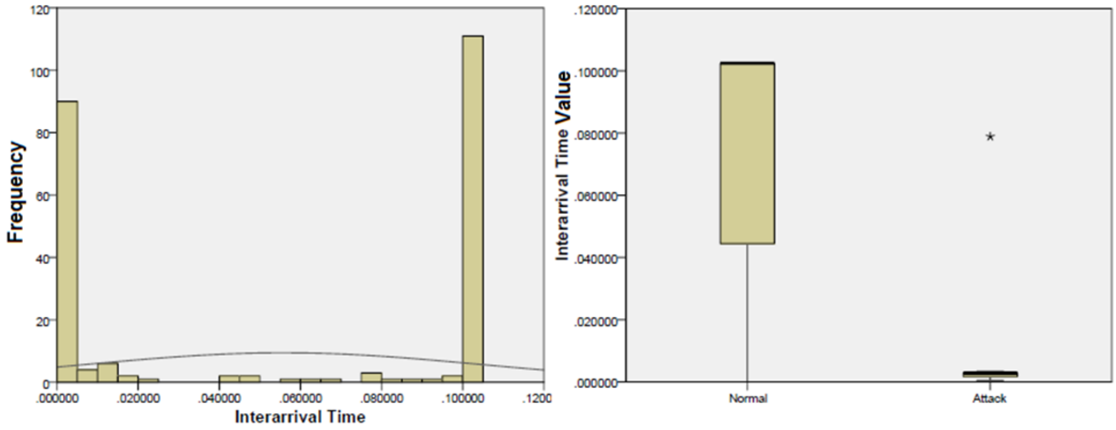
(a) (b)

Figure 6.35 Deauthentication Attack - $INJ_{Rate}$ Long Distance Traffic: (a) Histogram, (b) Boxplot.



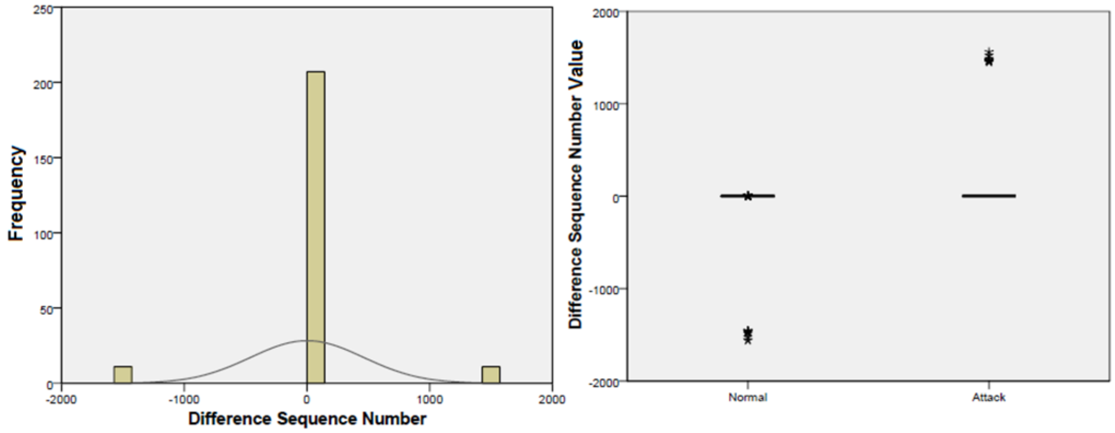
(a) (b)

Figure 6.36 Deauthentication Attack - $NAV$ Long Distance Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.37 Deauthentication Attack - $\Delta Time$ Long Distance Traffic: (a) Histogram, (b) Boxplot.



(a) (b)

Figure 6.38 Deauthentication Attack - $SEQ_{Dif}$ Long Distance Traffic: (a) Histogram, (b) Boxplot.

Similar to the evaluated datasets when the Airpwn attacks were launched, the distributions of the different histograms generated when the Deauthentication attack was implemented are statistically dissimilar from each other. Again, trying to generate a common statistical profile of normal traffic that could adapt to the distribution of all the considered metrics may cause numerous FP alarms.

With the statistical results presented through this section, it has been evidenced that all of the metric measurements follow non-homogeneous distributions. The characteristic measures for each of the six used metrics are statistically dissimilar from each other. In addition, there are no statistical distributions that the different metrics values could adapt to. Therefore, trying to generate a common statistical profile of normal traffic that could adapt to the distribution of all the considered metrics may cause numerous FP alarms. Hence, for all the evaluated datasets, all the metrics should be analysed and treated independently.

In order to simplify the analysis of the parameters that describe the different datasets, a summary is presented in Table VI.VII. Figure 6.39 represents the number of frames within each dataset. Figure 6.40 represents the proportion of the frames within each dataset.

TABLE VI.VII. DATASETS CHARACTERISTICS.

| **Attack** | **Type of Attack** | **Number of Frames** | | **Proportion** | |
|---|---|---|---|---|---|
| | | **Normal** | **Malicious** | **Normal** | **Malicious** |
| *Normal* | n/a | 3631 | n/a | 100% | n/a |
| *Airpwn* | Attack01 | 1350 | 11 | 99.2% | 0.8% |
| | Attack02 | 13498 | 995 | 93.1% | 6.9% |
| | Mixture | 12016 | 114 | 99.1% | 0.9% |
| *Deauthentication* | Long Distance | 165 | 64 | 72.1% | 27.9% |
| | Short Distance | 139 | 64 | 68.5% | 31.5% |

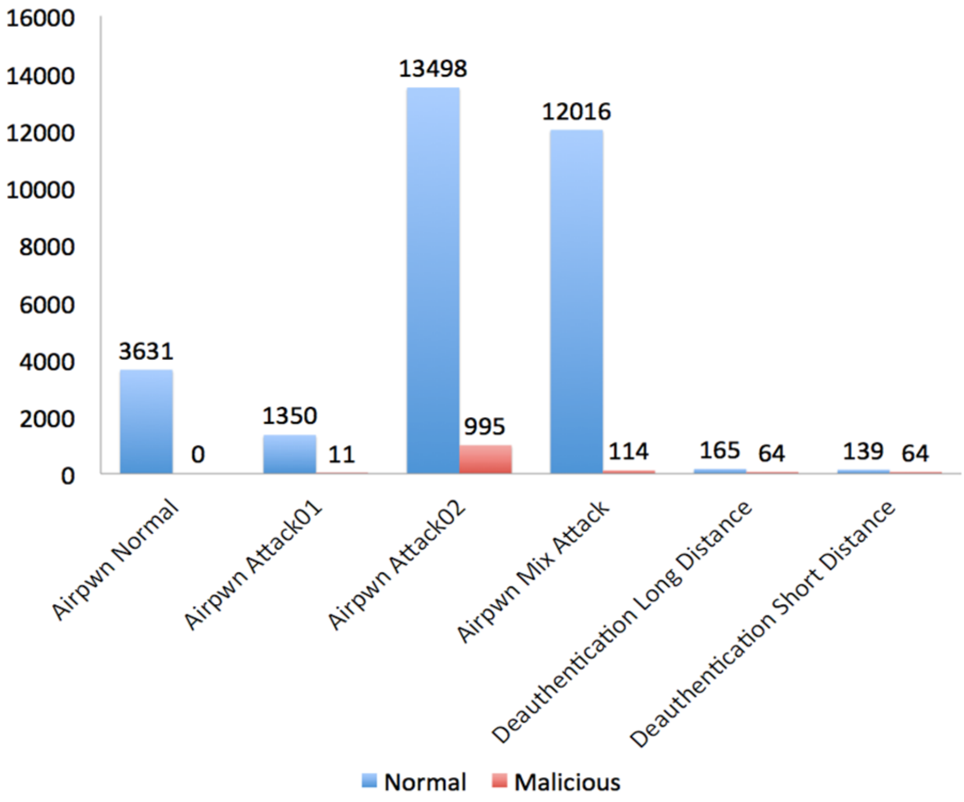


Figure 6.39 Number of Frames In Each Dataset - Bar Chart.

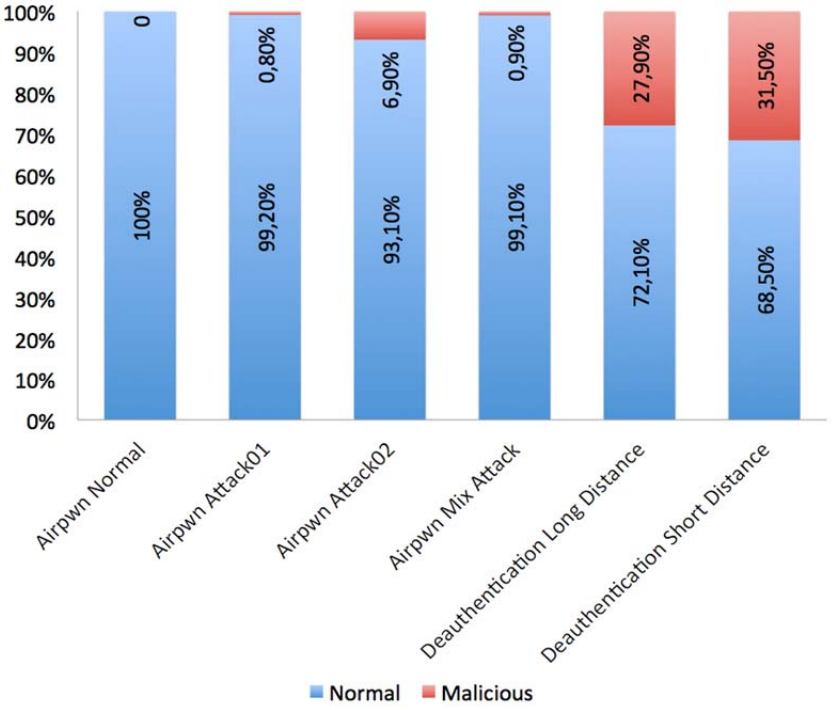


Figure 6.40 Proportion of Frames In Each Dataset - Bar Chart.

### 6.5.3 Metric Conditions

Previously, it was explained that anomaly IDSs require that the number of normal frames must be larger than the malicious frames, and the difference between metrics of the normal and malicious frames in the assessed datasets must be statistically differentiable and quantifiable.

Regarding the first condition, the authors of [80] indicate that the number of normal data instances is more predominant than malicious data instances in real network data traffic. The analysis of the network traffic datasets in the previous section has proven that the generated datasets are, in fact, composed of more normal data instances than malicious. Therefore, the first of the condition does not entitle a problem for the proposed detection system. Regarding the second condition, in the next section, it has been also presented the results of the unpaired $t$-test analysis conducted to prove that both type of metrics are statistically differentiable. This statistical analysis has been implemented with the statistical analysis software 'IBM SPSS Statistics' [152].

#### 6.5.3.1 Unpaired $t$-Test for Metric Analysis

The $t$-test is a parametric test used to determine whether the means of two uncorrelated data samples differ from each other [153]. Since the instances evaluated in this method are independent, the particular type of test is the unpaired $t$-test.

Two hypotheses have been defined as part of the Hypothesis Testing Framework. The Null hypothesis $H_0$ assumes that the means of the normal and anomalous instances of the metrics to be analysed are not statistically differentiable. On the other hand, the Alternative hypothesis $H_A$ assumes that the means of both instances are statistically differentiable between each other. The significance level considered in the following analyses is 5%, which is the default value for the $t$-test in IBM SPSS Statistics. The results of the unpaired $t$-test for each metric and each dataset have been tabulated in the following tables.

The first dataset to be analysed is the Airpwn $Attack01$ dataset. The results of this analysis are presented in the Table VI.VIII. For most of the metrics ($RSSI$, $TTL$, $NAV$ and $SEQ_{Dif}$) there exist evidence of a difference between the instances of malicious and non-malicious nature. The significance level ($p$ = 0) is lower than 5%. Hence, these four metrics can provide evidence of distinction between both types of information, for this dataset. However, the significance level for the metrics $INJ_{Rate}$ ($p$ = 0.104) and $\Delta Time$ ($p$ = 0.095) are both higher than 5%. Hence, these two metrics cannot provide evidences of distinction between both types of information, for this dataset.

TABLE VI.VIII. AIRPWN DATASET - $Attack01$ - $t$-TEST ANALYSIS

| | $t$-test for Equality of Means | | | |
|---|---|---|---|---|
| Metrics | Sig. (2-tailed) | Mean Difference | 95% Confidence Inter. of the Diff. | |
| | | | Lower | Upper |
| $RSSI$ | 0.000 | −13.371 | −14.373 | −12.368 |
| $INJ_{Rate}$ | 0.104 | 10.286 | −2.121 | 22.692 |
| $TTL$ | 0.000 | −234.323 | −265.327 | −203.319 |
| $NAV$ | 0.000 | −305.461 | −315.761 | −295.161 |
| $SEQ_{Dif}$ | 0.000 | −2541.429 | −2708.227 | −2374.63 |
| $\Delta Time$ | 0.095 | 0.0065 | −0.0011 | 0.0141 |

The next dataset to be analysed is the Airpwn $Attack02$ dataset. The results of this analysis are presented in the Table VI.IX. For this dataset there exists evidence of a statistical difference between the instances of malicious and non-malicious nature, for all the metrics. The significance level ($p$ = 0) is lower than 5%. Hence, the six metrics can provide evidences of distinction between both types of information, for this dataset.

TABLE VI.IX. AIRPWN DATASET - $Attack02$ - $t$-TEST ANALYSIS

| | $t$-test for Equality of Means | | | |
|---|---|---|---|---|
| Metrics | Sig. (2-tailed) | Mean Difference | 95% Confidence Inter. of the Diff. | |
| | | | Lower | Upper |
| $RSSI$ | 0.000 | $-9.547$ | $-9.667$ | $-9.427$ |
| $INJ_{Rate}$ | 0.000 | 47.63 | 46.638 | 48.623 |
| $TTL$ | 0.000 | $-195.149$ | $-198.111$ | $-192.187$ |
| $NAV$ | 0.000 | $-274.446$ | $-275.27$ | $-273.622$ |
| $SEQ_{Dif}$ | 0.000 | 132.534 | 106.194 | 158.874 |
| $\Delta Time$ | 0.000 | $-0.0057$ | $-0.0063$ | $-0.0051$ |

The results of the unpaired $t$-test analysis for the Airpwn $Mixed\ Attack$ dataset are presented in Table VI.X. Again, there exists evidence of a statistical difference between the instances of malicious and non-malicious nature, for all the metrics. The significance level (which is $p = 0$) is lower than 5%. Hence, the six metrics can provide evidences of distinction between both types of information, for this dataset.

TABLE VI.X. AIRPWN DATASET - $Mixed\ Attack$ - $t$-TEST ANALYSIS

| | $t$-test for Equality of Means | | | |
|---|---|---|---|---|
| Metrics | Sig. (2-tailed) | Mean Difference | 95% Confidence Inter. of the Diff. | |
| | | | Lower | Upper |
| $RSSI$ | 0.000 | $-10.68$ | $-11.122$ | $-10.239$ |
| $INJ_{Rate}$ | 0.000 | 46.624 | 43.461 | 49.788 |
| $TTL$ | 0.000 | $-194.069$ | $-203.539$ | $-184.598$ |
| $NAV$ | 0.000 | $-275.284$ | $-277.91$ | $-272.658$ |
| $SEQ_{Dif}$ | 0.000 | 577.143 | 549.531 | 604.755 |
| $\Delta Time$ | 0.000 | $-0.0045$ | $-0.0058$ | $-0.0032$ |

The results of the unpaired $t$-test analysis for the Deauthentication Short Distance dataset are presented in Table VI.XI. For most of the metrics ($RSSI$, $NAV$, $SEQ_{Dif}$ and $\Delta Time$) there exists evidence of a difference between the instances of malicious and non-malicious nature. The significance level ($p$ = 0) is lower than 5%. Hence, these four metrics can provide evidences of distinction between both types of information, for this dataset. However, the significance level for the metric $INJ_{Rate}$ ($p$ = 0.238) is higher than 5%. Hence, this metric cannot provide evidences of distinction between both types of information, for this dataset.

TABLE VI.XI. DEAUTHENTICATION DATASET - SHORT DISTANCE - $t$-TEST ANALYSIS

| | $t$-test for Equality of Means | | | |
|---|---|---|---|---|
| Metrics | Sig. (2-tailed) | Mean Difference | 95% Confidence Inter. of the Diff. | |
| | | | Lower | Upper |
| $RSSI$ | 0.000 | −17.034 | −18.993 | −15.075 |
| $INJ_{Rate}$ | 0.238 | 1.144 | −0.764 | 3.052 |
| $NAV$ | 0.000 | −270.129 | −296.801 | −243.458 |
| $SEQ_{Dif}$ | 0.000 | −15.794 | −22.379 | −9.21 |
| $\Delta Time$ | 0.000 | 0.0674 | 0.0562 | 0.0785 |

Finally, the results of the unpaired $t$-test analysis for the Deauthentication Long Distance dataset are presented in Table VI.XII. Again, for the four metrics $RSSI$, $NAV$, $SEQ_{Dif}$ and $\Delta Time$ there exists evidence of a difference between the instances of malicious and non-malicious nature. The significance level ($p$ = 0) is lower than 5%. Hence, these four metrics can provide evidences of distinction between both types of information, for this dataset. However, the significance level for the metric $INJ_{Rate}$ ($p$ = 0.28) is higher than 5%. Hence, this metric cannot provide evidences of distinction between both types of information, for this dataset.

TABLE VI.XII. DEAUTHENTICATION DATASET - LONG DISTANCE - $t$-TEST ANALYSIS

| | $t$-test for Equality of Means | | | |
|---|---|---|---|---|
| Metrics | Sig. (2-tailed) | Mean Difference | 95% Confidence Inter. of the Diff. | |
| | | | Lower | Upper |
| $RSSI$ | 0.000 | −18.371 | −18.946 | −17.795 |
| $INJ_{Rate}$ | 0.28 | 0.964 | −0.788 | 2.715 |
| $NAV$ | 0.000 | −280.848 | −304.441 | −257.256 |
| $SEQ_{Dif}$ | 0.000 | −353.963 | −479.801 | −228.124 |
| $\Delta Time$ | 0.000 | 0.0725 | 0.0621 | 0.0829 |

All the results presented in this section assure the two conditions described as necessary for the proposed methodology to be effective.

### 6.5.4 Feature Selection

Feature Selection is a widely used concept among the intrusion detection, machine learning and data mining systems, which refers to a group of techniques able to reduce the number of metrics in a given dataset to the minimum and optimise the selection process of the most relevant set of metric [92]. Optimising the selection of network traffic metrics has also a significant impact on the performance and speed of the IDSs [91] [142]. Analysing datasets containing irrelevant and redundant information may slow the intrusion detection process down [130]. In [142], the authors practically showcases that the utilisation of feature selection techniques contributes to improve overall accuracy of their system, reducing the number of false alarms and improving the DR. Ideally, all IDSs should implement feature selection as part of their framework to improve the attack detection accuracy.

Irrelevant and redundant metrics introduce inappropriate information in the training or analysis dataset, degrading the accuracy of the intrusion detector [9]. These metrics will cause what is known as 'curse of dimensionality' [48]. When implementing feature selection techniques, IDSs should have the precautions not to unintentionally delete more metrics than should be deleted. In that case, relevant metrics could be removed and the IDS may end up misclassifying attacks. IDSs need to verify that results with similar accuracy are produced using both the complete and the preprocessed dataset.

The implementation of automatic feature selection techniques for unsupervised IDSs is still a great challenge for researchers in intrusion detection [54], especially if the IDSs perform the detection in real time. It requires a period of time to be executed. The larger the number of metrics in the dataset, the longer the required time. An IDS that preforms attack detection in real time may not be able to afford too much time on this preprocessing procedure.

#### 6.5.4.1 Curse of Dimensionality

The term curse of dimensionality refers to the degrading effect that the utilisation of an excessive number of metrics causes to the results of data analysis, intrusion detection, machine learning and data mining systems. Curse of dimensionality is caused by the utilisation of datasets containing irrelevant and redundant information.

One of the main characteristics for an IDS to produce accurate detection results is to have enough network traffic metrics to analyse [48]. IDSs rely on the relevance of the evidences provided by these metrics to produce accurate intrusion detection results. But having an excessive number of metrics could be more prejudicial for the results than having too few metrics to analyse. For instance, in [9], the author presents the results of implementing K-means clustering on a given dataset. The results that this work presents show that selecting 8 specific metrics out of all the available metrics produce the best intrusion detection results. The use of this set of 8 metrics maximises the accuracy of the detection systems. The author of this work assessed the results that would be obtained if more metrics were utilised. The results show that increasing the number of metrics up to 19 metrics produces a drop of 17% in the detection accuracy. This deterioration of the results is the direct consequence of the curse of dimensionality.

## 6.6 Summary

This chapter has described several aspects regarding the IEEE 802.11 network traffic datasets used in this thesis. Firstly, this chapter addresses the decision of whether utilise publicly available network datasets, dataset generated using network simulation software, or utilise network traffic datasets gathered from a live operational network. The experiments in this thesis have been implemented with network traffic datasets gathered from a live operational IEEE 802.11 network. Using network traffic datasets from this network to evaluate the proposed IDS provides a series of advantages. The dataset would contain more realistic parameters than a synthetically generated dataset. This datasets would commonly require being processed to clean it and make it suitable for the detection process. As part of the preprocessing process, this chapter has explained the filtering procedure used to remove the unnecessary information gathered adjacent wireless networks, and helps to reduce the total amount of information that the detection system needs to analyse. The network traffic is filtered out using MAC address of the AP and the wireless client.

This chapter has also presented an extensive description of the different metrics that have been considered for this thesis. Six metrics have been experimentally selected as the most appropriate for detecting the attacks, after the evaluation of all the available metrics. The selection of the metrics plays an important role in the generation of accurate intrusion detection results. As it has been explained in this chapter, the number of the metrics is also an important to be considered. An example has been practically described that shows the problem with D-S using a large number of metrics. The influence of the belief in *Uncertainty* follows a decreasing trend, as more metrics are fused.

The chapter has also presented the statistical description of all the used metrics in all the datasets, along with a discussion of whether these metrics are appropriate for the detection of wireless-specific attacks. This analysis proves that the number of normal data instances is more predominant than malicious data instances in the analysed network traffic datasets. On the other hand, this analysis also statistically proves that the metrics can provide distinction between both normal and malicious frames. These are the two conditions that need to be met for the anomaly IDSs to be accurate. In addition, the statistical distribution of each metric is dissimilar from each other. Generating a common statistical profile of normal traffic that could adapt to the distribution of all the considered metrics would not be feasible.

Lastly, the chapter concludes with a brief description of the concept of feature selection and the general need for this approach, as well as the description of the concept of curse of dimensionality.

# Chapter 7

## Results Evaluation

### 7.1 Introduction

This chapter evaluates the effectiveness of the unsupervised anomaly based IDS framework presented in this thesis. The approach followed in this chapter to demonstrate the effectiveness of the detection system is to compare the detection results generated using the multi-layer approach (i.e. when all the considered metrics are used) against the same methodology utilising different sets of metrics. The purpose is to verify whether the combined use of all the selected metrics outperforms the same methodology, utilising other combinations with fewer number of metrics. For the *Normal* and Airpwn datasets (i.e. *Attack01*, *Attack02* and *Mixed Attack*), there exist $2^6 - 1$ different metrics combinations, whereas for the deauthentication datasets (i.e. *Long Distance* and *Short Distance*), there exist $2^5 - 1$ different metrics combinations.

The results are used to prove different aspects of the IDS proposed in this thesis. It is important that the detection system should be able to provide very high or even perfect detection capabilities. Hence, one of the aims of this results evaluation is to prove that the proposed IDS is accurate providing extra level of protection to wireless networks. Also, it is important to prove the effectiveness of the proposed system against different type of attacks, as well as to find the methodology that best results generates (i.e. mean or mode, and distance or angle). The application of one mechanism or the other may directly affect the final results of the proposed detection system. Therefore, a fundamental part of the evaluation experiments is to assess the performance of the system using each of the configurations. Another aspect to be evaluated is the sliding window length that generates the best detection results. In addition, it is important to verify whether or not the proposed methodology is capable of being implemented in real time, and to estimate the maximum number of malicious frames that could be included within the initial sliding window before the accuracy of the detection results deteriorate.

This chapter will present a thorough description of the generated results, for all the considered datasets and all the possible metric combinations. Using these results, this chapter will address all the points previously described. Nonetheless, before presenting the experiments results, this chapter introduces the evaluation functions used to examine these results.

## 7.2 Statistical Evaluation Parameters

The efficiency of IDSs could be evaluated using multiple parameters, such as the amount of resources (CPU, Memory, etc.) the system consumes, or the required time to conduct the detection. Nonetheless, the most important aspect to evaluate the effectiveness of the proposed intrusion detection methodology is its ability to make correct predictions [58]. This is achieved using a series of evaluation functions over the generated system outcome. These evaluation functions have been widely used among the researching community in the field of IDSs. Before listing the functions, it is also necessary to define four evaluation parameters utilised in the functions. These parameters provide quantifiable evidence of how effective are the IDSs at making correct detections. These are:

- True Positive ($TP$) refers to one attack frame that has been correctly classified as malicious.
- True Negative ($TN$) refers to one non-attack frame that has been correctly classified as legal frame.

- False Positive ($FP$) refers to one non-attack frame that has been misclassified as malicious.
- False Negative ($FN$) refers to one attack frame that has been misclassified as legal frame.

These parameters provide quantifiable evidence of how effective are the IDSs at making correct detections. All these parameters are utilised to calculate the following evaluation functions; Equations 7.1 – 7.6:

- Detection Rate ($DR$) is the proportion of attack frames correctly classified as malicious, among all the attack frames.

$$DR\ (\%) = \frac{TP}{FN + TP} \tag{7.1}$$

- False Positive Rate ($FP_{Rate}$) is the proportion of non-attack frames misclassified as malicious, among all the evaluated frames.

$$FP_{Rate}\ (\%) = \frac{FP}{TP + FP + TN + FN} \tag{7.2}$$

- False Negative Rate ($FN_{Rate}$) is the proportion of attack frames misclassified as legal, among all the attack frames.

$$FN_{Rate}\ (\%) = \frac{FN}{TP + FN} \tag{7.3}$$

- Overall Success Rate ($OSR$) or $Accuracy$ is the proportion of the total number of frames correctly classified, among all the evaluated frames.

$$OSR\ (\%) = \frac{TN + TP}{TP + FP + TN + FN} \tag{7.4}$$

- $Precision$ or $Recall$ is the proportion of attack frames correctly classified as malicious, among all the alarms generated.

$$Precision\ (\%) = \frac{TP}{TP + FP} \tag{7.5}$$

- $F\text{–}Score$ or $F\text{–}Measure$ is a tradeoff between Precision and DR. The $F\text{–}Score$ produces a high result when $Precision$ and DR are both balanced [14] [97]. The higher the $F\text{–}Score$, the better the $Precision$ and the DR.

$$F\text{–}Score\ (\%) = \frac{2 * Precision * DR}{Precision + DR} \tag{7.6}$$

The most important action for the IDSs is to generate the maximum number of $DR$ and not generating any false alarm. Generating either $FN$ or $FP$ is not desirable. Both situations show decrease of the effectiveness of an IDS detecting intrusions. However, cost of generating $FNs$ is often higher than the cost of $FPs$ [80]. An IDS that generates too many $FPs$ works against legitimate communications. However, the protected system is not actually compromised by any threat. The administrator of the IDS might ignore the raised attack alerts [78]. On the other hand, every single $FN$ is an attack that has gone undetected and reached the protected system.

## 7.3 Results Evaluation

This section describes the experiment results generated by the unsupervised anomaly based IDS framework presented in this thesis. The principal objective of the described experiments is to evaluate the effectiveness of the proposed mechanisms. There are a series of additional points that have been also evaluated and verified through the implementation of these experiments.

One of these points is to compare the detection results generated using the multi-layer approach (i.e. when all the considered metrics are used) against the same methodology utilising different sets of metrics. The purpose is to verify whether the combined use of all the selected metrics outperforms the same methodology, utilising other combinations with fewer number of metrics.

It is necessarily to evaluate the results that the proposed methodology produces, using all the possible combinations of metrics. For the *Normal* and Airpwn datasets (i.e. *Attack01*, *Attack02* and *Mixed Attack*) there exist $2^6 - 1$ possible metric combinations, since six different metrics have been selected. Therefore, to evaluate the results of the proposed methodology, the same wireless network dataset is evaluated 63 times. For the deauthentication datasets (i.e. *Long Distance* and *Short Distance*) there exist $2^5 - 1$ possible metric combinations, since five different metrics have been selected. Therefore, the same wireless network dataset is evaluated 31 times.

Another point that needs to be addressed is to find the sliding window length that generates the best detection results. The same dataset has been evaluated multiple times varying the length of the sliding window. In the case of Airpwn attack, the length value has been gradually increased from one slot to a length of 200 slots, $[1 \leq n \leq 200]$. In the case of the deauthentication attack, the length value increased only up to 135 slots, $[1 \leq n \leq 135]$. This is because the number of instances in both datasets, *Deauthentication Long Distance* and *Deauthentication Short Distance*, is too small. Using $n = 200$ would not allow the sliding window to slide. For each of the different values of the sliding window length, the same evaluation parameters are calculated evaluating the same dataset.

Another objective of the experiments presented in this section is to verify whether or not the proposed methodology is capable of being implemented in real time. A challenging requirement of the intrusion detection mechanism proposed in this thesis is to operate in a per-frame basis. As already mentioned in Chapter 5, the length value of the sliding window has a direct impact on the required time to implement the detection. Therefore, in order to prove that the system can operate in real time, the time required to assess each captured frame has been calculated.

In addition, the results of the experiments that use the automatic BPAs methodologies have been used also to evaluate the maximum number of malicious frames that could be included within the initial sliding window before the accuracy of the detection results were affected. An attacker could alter the first statistical reference of normality if an attack were launched before the initial sliding window is completed with captured frames. In that case, the characteristics of the malicious frames would dominate the sliding window and the detection system would classify normal traffic as malicious. This vulnerability makes it necessarily to assess the percentage of malicious frames that could be included within the initial sliding window, in correlation with the sliding window length, before the accuracy of the detection results were affected.

Also, it is important to find the methodology that generates the best detection results. As have been explained in Chapter 5, there are four possible system configurations that could be implemented to assign belief in $Attack$. One of these configurations uses the Euclidean distance of the current frame from the current reference of normality. The modified configuration uses the angle generated by the Euclidean distance and frequency of the data. The application of one mechanism or the other may directly affect the final results of the proposed detection system. Therefore, a fundamental part of the evaluation experiments presented is to assess the performance of the system using each of the configurations. Another parameter that needs to be evaluated through the implementation of the following experiments is the reference of normality. It is necessarily to assess whether the mechanism to determine the reference of normality is adequate or not to provide efficient results. Two simple statistical parameters (i.e. mean or mode) have been used to define this reference.

### 7.3.1 Deauthentication Attack Experiment Results

One of the implemented attacks is the deauthentication attack. The detection was possible only using management and control frames, and data frames information from the two lower layers of the protocol stack. Firstly, because the network was encrypted with WPA2, and with the assumption that the monitor node does not have the key, it was not possible or necessary to retrieve information above the MAC layer.

Two sets of experiments, using the same attack have been implemented. In a first experiment (*Long Distance*), the attacker is placed 10 metres away from the victim. In the second experiment (*Short Distance*) the attacker is placed 1.5 metres away from the victim. The experiment results presented in this section assess the effect of changes in the proximity of the attacker from the victim on the final results. The testbed for the detection of the deauthentication attack is the generic testbed described in Chapter 6. All the devices are located in a stationary geographical location. Since five metrics have been selected, the same wireless network dataset has been evaluated 31 times.

#### 7.3.1.1 Deauthentication Attack Results – Long Distance

The multi-layer results for the deauthentication *Long Distance* dataset, using the five considered metrics for the four assessed system configurations, are presented in Figure 7.1. The figure represents the $FP_{Rate}$ results, modifying the length value of the sliding window. In the figure, the Y-axis of the graph represents the percentage of $FP_{Rate}$. The X-axis of the graph represents the length value of the sliding window.

All the configurations provide completely perfect detection for any sliding window length larger than 3 slots. 100% of $DR$ is constantly achieved for $3 \leq n$. In terms of $FP_{Rate}$, the detection system using the mean as reference of normality and Euclidean distance, distance-mean, always generates results lower than 2% for any sliding window length larger than 3 slots, $11 \leq n$. After the sliding window reaches 31 slots long, $31 \leq n$, the $FP_{Rate}$ is constantly reduced to 0.87%, and for $79 \leq n$ the $FP_{Rate}$ is constantly reduced to 0.43%. Using the configuration angle-mean, the detection $FP_{Rate}$ results are slightly higher than the previous case. Only after the sliding window length reaches 81 slots long, $81 \leq n$, the $FP_{Rate}$ is constantly reduced to 0.43%. The $FP_{Rate}$ results for the distance-mode and angle-mode outperform the results of both, distance-mean and angle-mean for any length value of the sliding window. The percentage of $FP_{Rate}$ does not reach 1.75% for any $n$ value. Also, the $FP_{Rate}$ is constantly reduced to 0.43% in both configurations for $79 \leq n$.

As a method to compare the four different assessed configurations, Table VII.I shows the results for each of the four configurations when the length value of the sliding window is $n = 29$. The results show that the distance-mode configuration produces the best results overall for this dataset. Nonetheless, the difference with the other three methodology configuration variations is very small, almost unnoticeable. The value of the sliding window length that best results produces is $68 \leq n$. However, selecting a sliding window length $n = 29$ can be considered a good option, because the results 100% $DR$ and 0.87% $FP_{Rate}$ can be considered an acceptable result.

TABLE VII.I. DEAUTHENTICATION LONG DISTANCE - 5 METRICS RESULTS - 29 SLOTS

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time$ (μ$sec$) |
|---|---|---|---|---|---|---|---|
| Distance // Mean | 100 | 0.87 | 0 | 0.99 | 0.969 | 0.98 | 25 |
| Distance // Mode | 100 | 0.87 | 0 | 0.99 | 0.969 | 0.98 | 20 |
| Angle//Mean | 100 | 1.74 | 0 | 0.98 | 0.941 | 0.97 | 17 |
| Angle//Mode | 100 | 1.3 | 0 | 0.98 | 0.955 | 0.98 | 27 |

Figure 7.2 shows the average processing time required to provide a final decision for each frame. The four assessed configurations require almost the same processing time. For $n = 29$, the average processing time is between 17μsec and 27μsec, being 20μsec for the distance-mode configuration. For $n = 68$, the average processing time would increase to 76μsec. This average processing time represents the time from the moment a frame is captured to the moment a final decision is reached. Since the average interarrival time between two consecutive frames is 55msec, the intrusion detection can be implemented in real time.

Figures 7.3 – 7.6 represent the $FP_{Rate}$ results of any possible combination of four metrics when the system uses the methodology configurations distance-mean, distance-mode, angle-mean and angle-mode, respectively. The only set of 4 metrics that outperforms the combination of all the considered metrics is $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$ (No $\Delta Time$). The $FP_{Rate}$ is slightly lower than for the combination of 5 metrics. The detection results in terms of $DR$ are not shown in this graph but these are always 100% for any case when $12 \leq n$.

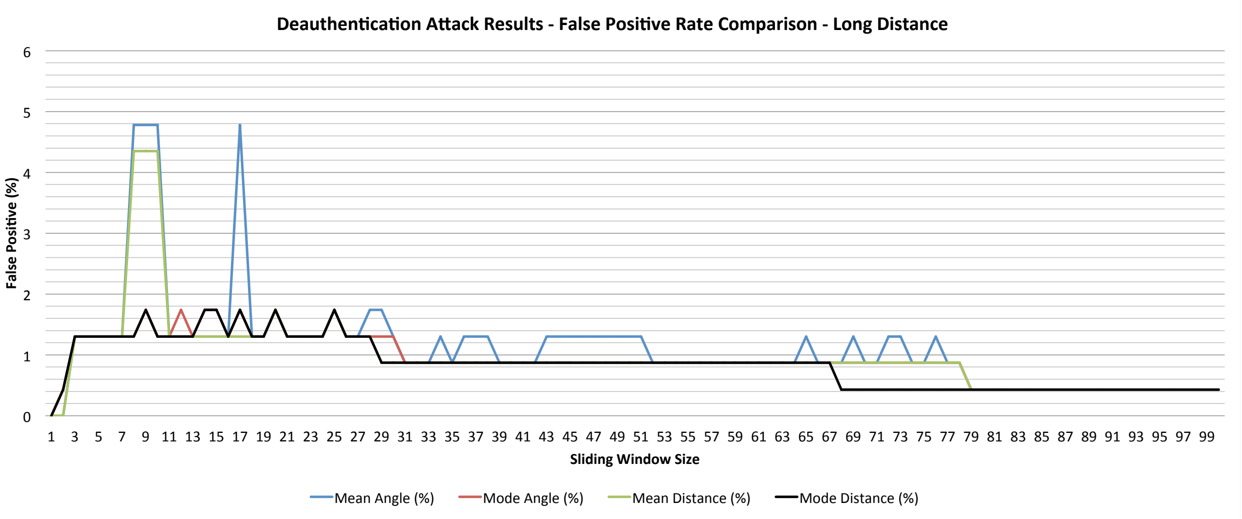


Figure 7.1 Deauthentication Long Distance - $FP_{Rate}$ Results Comparison.

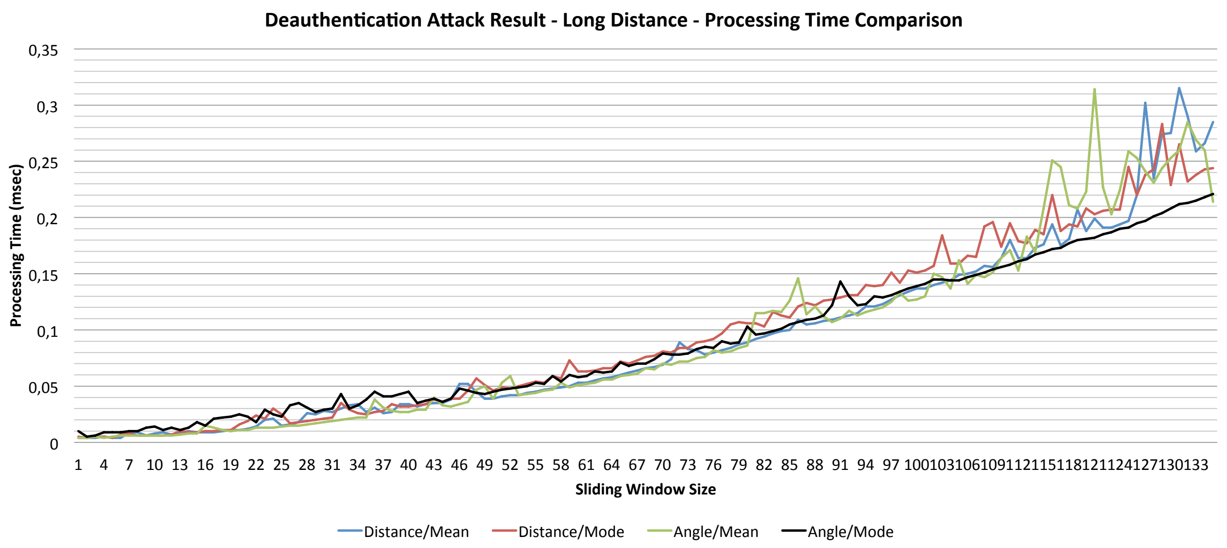


Figure 7.2 Deauthentication Long Distance - Per Frame Detection Analysis Processing Time.

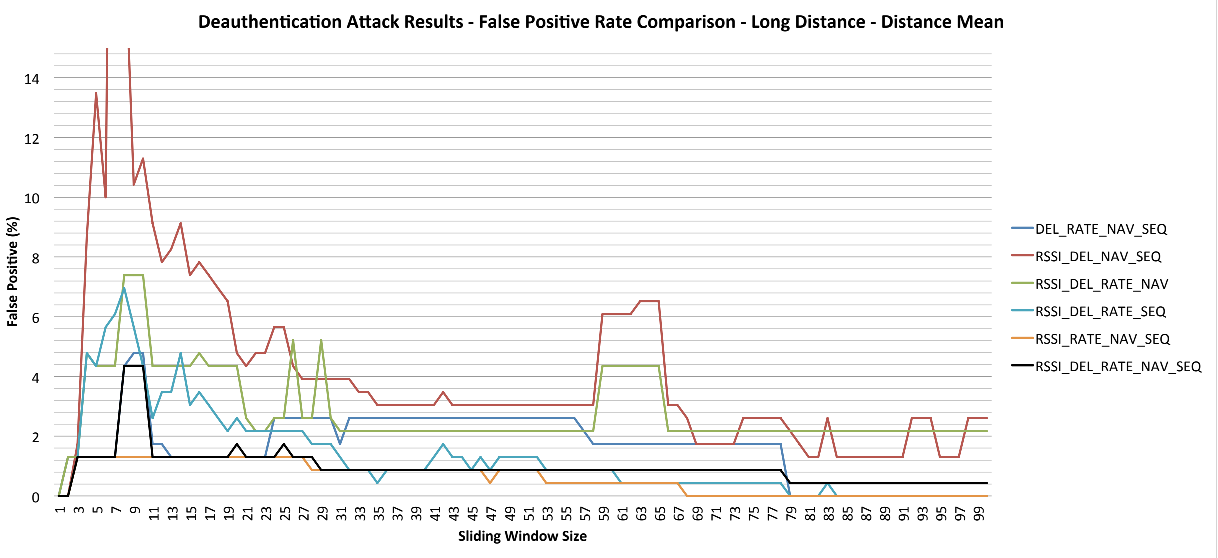


Figure 7.3 Deauthentication Long Distance - $FP_{Rate}$ 4 Metrics - Distance & Mean.

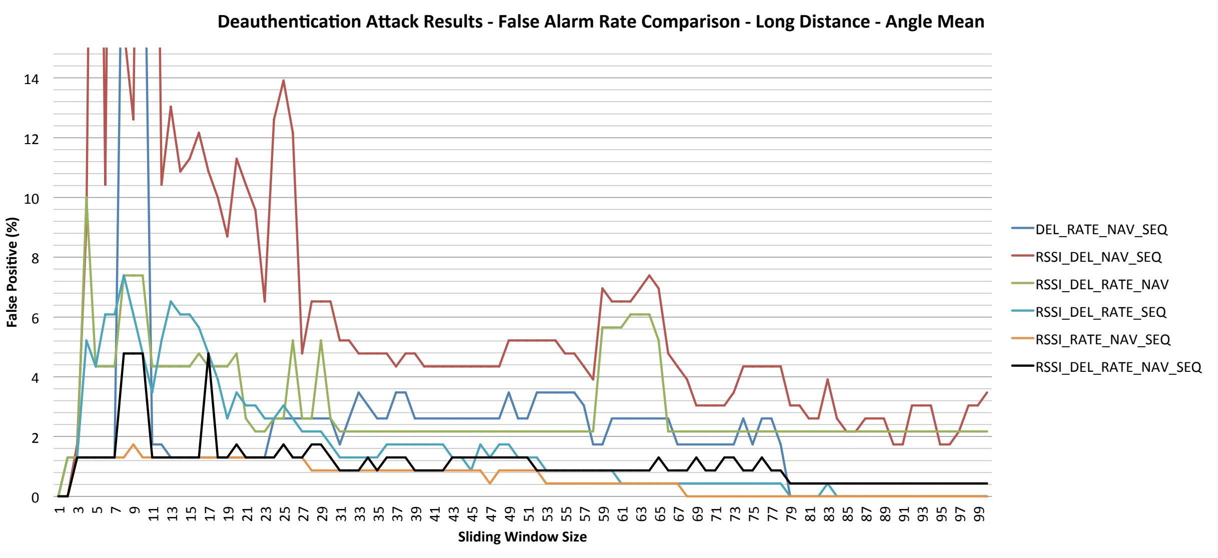


Figure 7.4 Deauthentication Long Distance - $FP_{Rate}$ 4 Metrics - Angle & Mean.

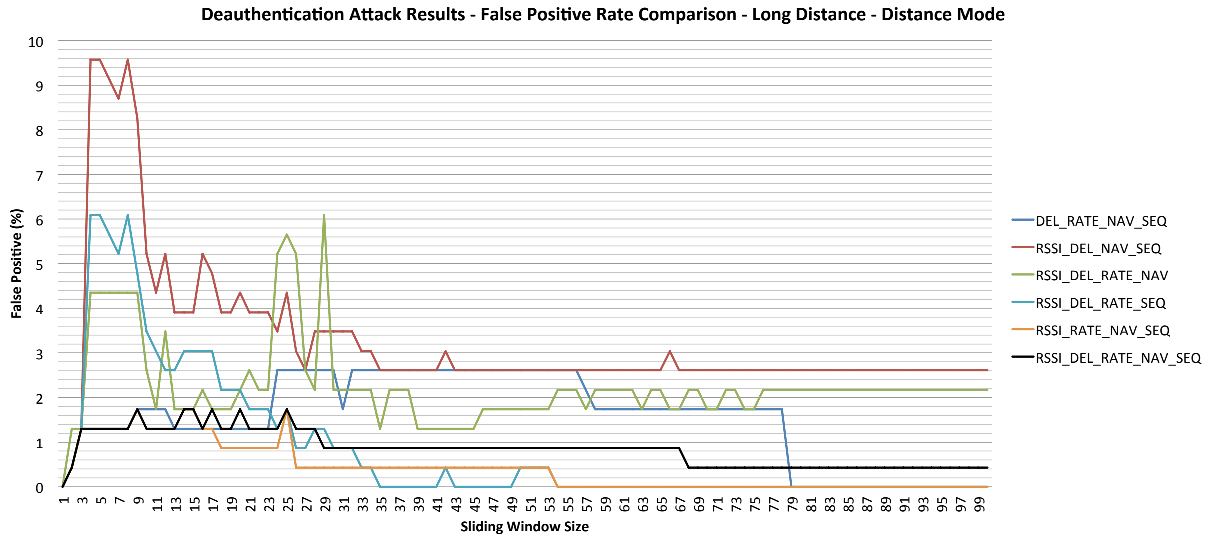


Figure 7.5 Deauthentication Long Distance - $FP_{Rate}$ 4 Metrics - Distance & Mode.

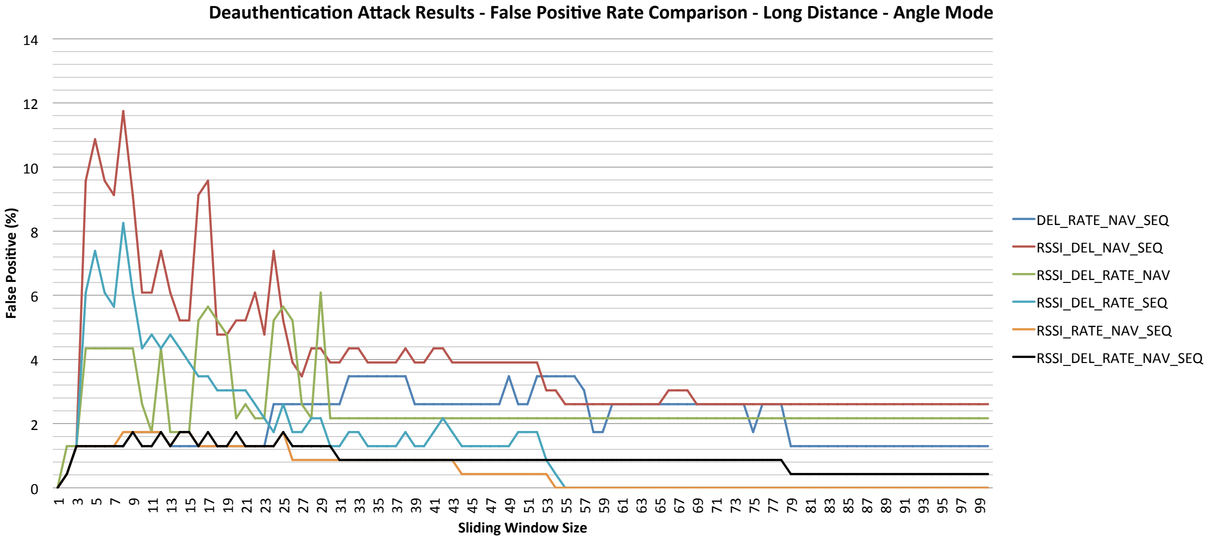


Figure 7.6 Deauthentication Long Distance - $FP_{Rate}$ 4 Metrics - Angle & Mode.

Figure 7.7 represents a close comparison of the $FP_{Rate}$ results for the $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$ combination. The $DR$ results are not represented because they are similar to the case of the 5 metrics. For $3 \leq n$, the system produces 100% $DR$. The detection system does not generate $FP_{Rate}$ results higher than 1.75% for any $n$ value. This shows an increase in the effectiveness of the 5 metrics results. The configuration distance-mode produces again the best result overall. Specifically, the system generates 0.43% $FP_{Rate}$when the sliding window is $[26 \leq n \leq 53]$, and for $54 \leq n$, the $FP_{Rate}$ result is 0%. For this particular sliding window length, the average processing time to produce perfect detection is 71µsec as shown in Figure 7.8.

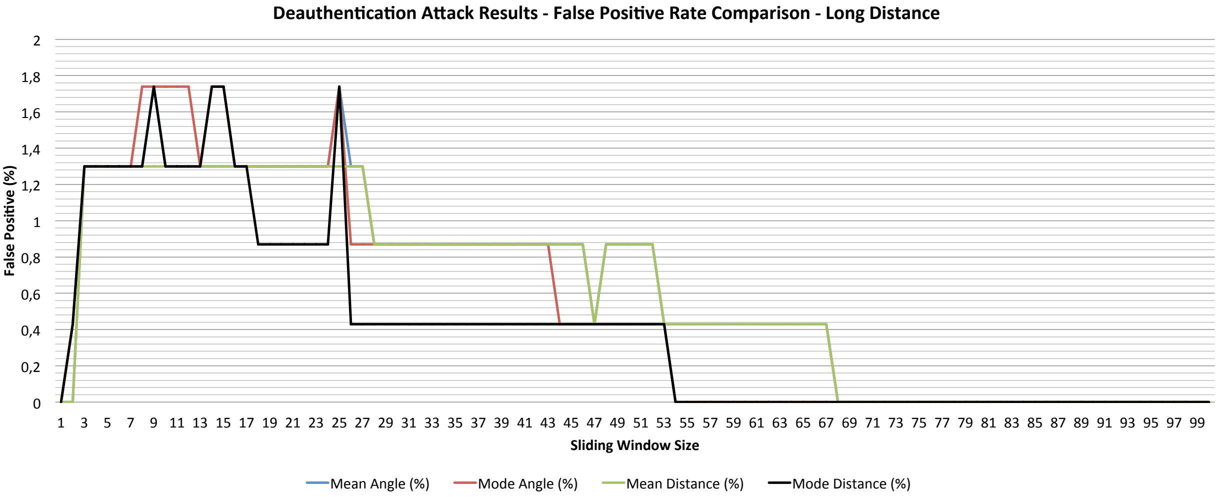


Figure 7.7 Deauthentication Long Distance - $FP_{Rate}$ Results Comparison - 4 Metrics - $RSSI\ INJ_{Rate}\ NAV\ SEQ_{Dif}$.

Table VII.II shows a comparison of the evaluation results for the metric combination $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$. The top four rows of the table show the results when $n = 29$. This is the length of the sliding window for which the system, using all the considered metrics, produced the best results. The bottom four rows of the table show the results when $n = 54$, which produces the best results for the metrics combination $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$. The presented results prove that not including $\Delta Time$ improves the final intrusion detection results. However, the length of the sliding window needs to be almost doubled to be able to produce these results, which also increases the probability to include malicious frames in the initial sliding window.

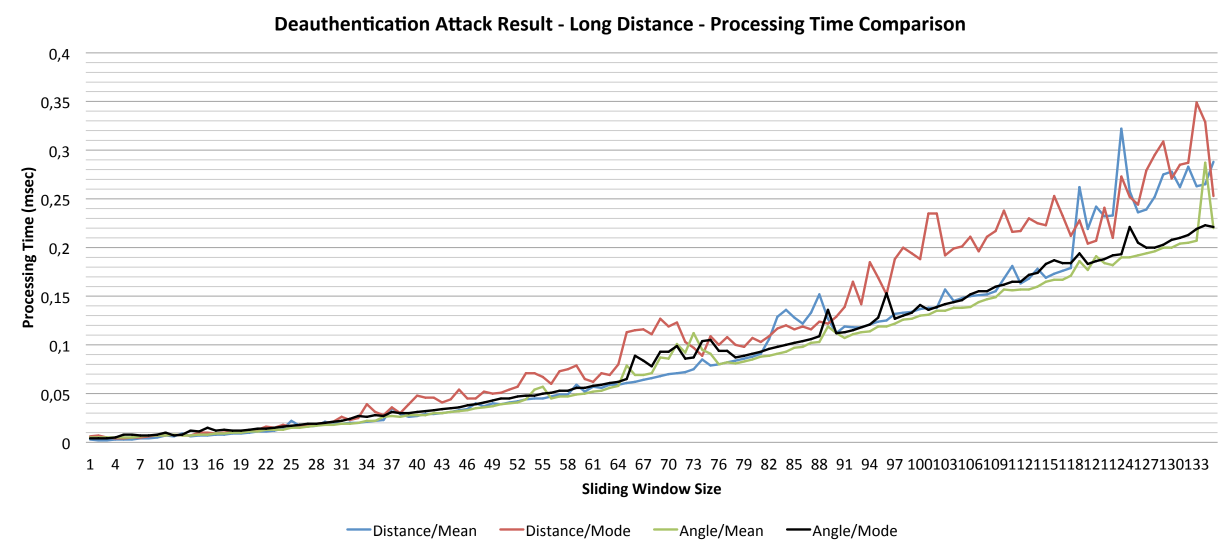


Figure 7.8 Deauthentication Long Distance - $RSSI\ INJ_{Rate}\ NAV\ SEQ_{Dif}$ - 4 Metrics - Per Frame Detection Analysis Processing Time.

TABLE VII.II. DEAUTHENTICATION LONG DISTANCE RESULTS - 4 METRICS $RSSI\ INJ_{Rate}\ NAV\ SEQ_{Dif}$ - 29 // 54 SLOTS.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time$ (μsec |
|---|---|---|---|---|---|---|---|
| Distance // Mean | 100 | 0.87 | 0 | 0.99 | 0.969 | 0.98 | 21 |
| Distance // Mode | 100 | 0.43 | 0 | 0.99 | 0.984 | 0.99 | 20 |
| Angle // Mean | 100 | 0.87 | 0 | 0.99 | 0.969 | 0.98 | 18 |
| Angle // Mode | 100 | 0.87 | 0 | 0.99 | 0.969 | 0.98 | 20 |
| Distance // Mean | 100 | 0.43 | 0 | 0.99 | 0.984 | 0.99 | 45 |
| Distance // Mode | 100 | 0 | 0 | 1 | 1 | 1 | 71 |
| Angle // Mean | 100 | 0.43 | 0 | 0.99 | 0.984 | 0.99 | 54 |
| Angle // Mode | 100 | 0 | 0 | 1 | 1 | 1 | 48 |

Figures 7.9 and 7.10 show the $DR$ and $FP_{Rate}$ results for the configuration distance-mean using all the possible combination of metrics, when the sliding window length is $n = 29$ and $n = 54$, respectively. In order to record all the possible combination of metrics for the experiments with the deauthentication attack, the different metric combinations have been categorised in the indexes represented in Table VII.III. The index D1 referrers to the set that combines all the considered metrics, and the index D31 referrers to a single metric set. Therefore, the best results are to be expected from the test index D1. The Y-axis of the graphs represents the percentage of $DR$ and $FP_{Rate}$. The X-axis of the graphs represents the indexes shown in the table, which correspond to the different metric combinations.

TABLE VII.III. INDEXES OF THE USED METRICS IN DEAUTHENTICATION ATTACK.

| Index-Metrics | Index-Metrics | Index-Metrics | Index-Metrics |
|---|---|---|---|
| D1 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | D9 - $RSSI$ - $\Delta Time$ - $SEQ_{Dif}$ | D17 - $RSSI$ - $\Delta Time$ | D25 - $INJ_{Rate}$ - $SEQ_{Dif}$ |
| D2 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | D10 - $RSSI$ - $INJ_{Rate}$ - $NAV$ | D18 - $RSSI$ - $INJ_{Rate}$ | D26 - $NAV$ - $SEQ_{Dif}$ |
| D3 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | D11 - $RSSI$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | D19 - $RSSI$ - $NAV$ | D27 - $RSSI$ |
| D4 - $RSSI$ - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | D12 - $RSSI$ - $NAV$ - $SEQ_{Dif}$ | D20 - $RSSI$ - $SEQ_{Dif}$ | D28 - $\Delta Time$ |
| D5 - $RSSI$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | D13 - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | D21 - $\Delta Time$ - $INJ_{Rate}$ | D29 - $INJ_{Rate}$ |
| D6 - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | D14 - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | D22 - $\Delta Time$ - $NAV$ | D30 - $NAV$ |
| D7 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ | D15 - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | D23 - $\Delta Time$ - $SEQ_{Dif}$ | D31 - $SEQ_{Dif}$ |
| D8 - $RSSI$ - $\Delta Time$ - $NAV$ | D16 - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | D24 - $INJ_{Rate}$ - $NAV$ | |

In Figures 7.9 and 7.10, using the distance-mean configuration there are a number of combinations that generate extremely poor $DR$ results, reaching $DR$ 0% in some cases. These metric combinations are D7, D11, D14, D16, D18, D21, D24, D25, and D29. One metric that is present in all the mentioned metric combinations is the $INJ_{Rate}$. This is the effect of the curse of dimensionality, because the frames are constantly transmitted at a fixed $INJ_{Rate}$ of 1Mbps by the client and the attacker. This fact makes the $INJ_{Rate}$ an irrelevant metric for the intrusion detection process.

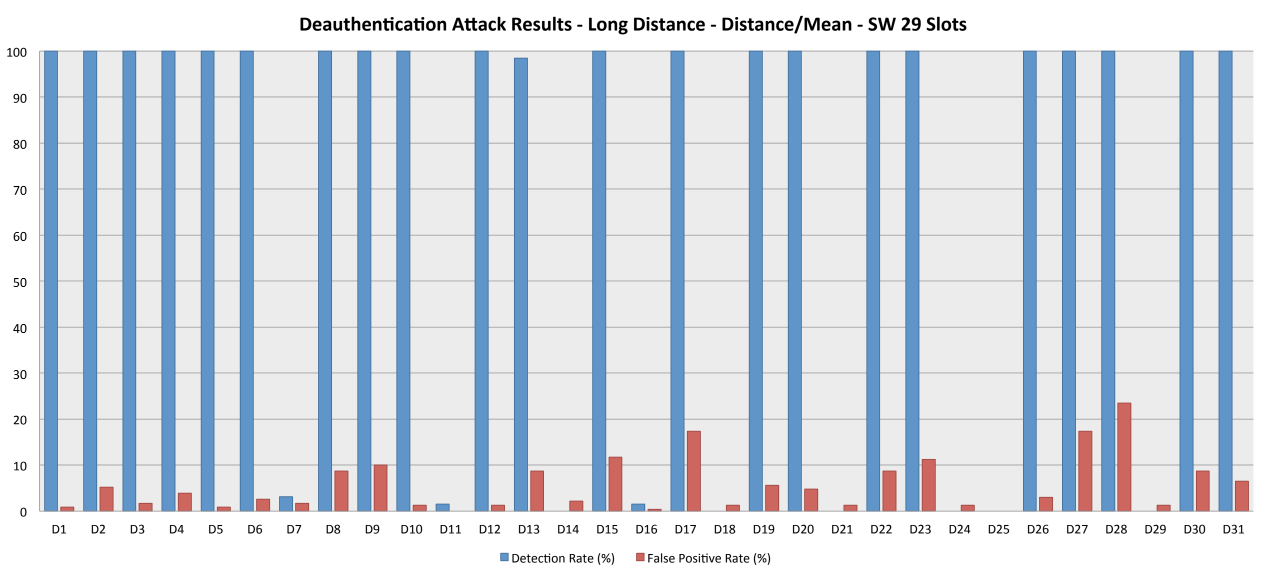


Figure 7.9 Deauthentication Long Distance - 29 Slots - Distance & Mean.

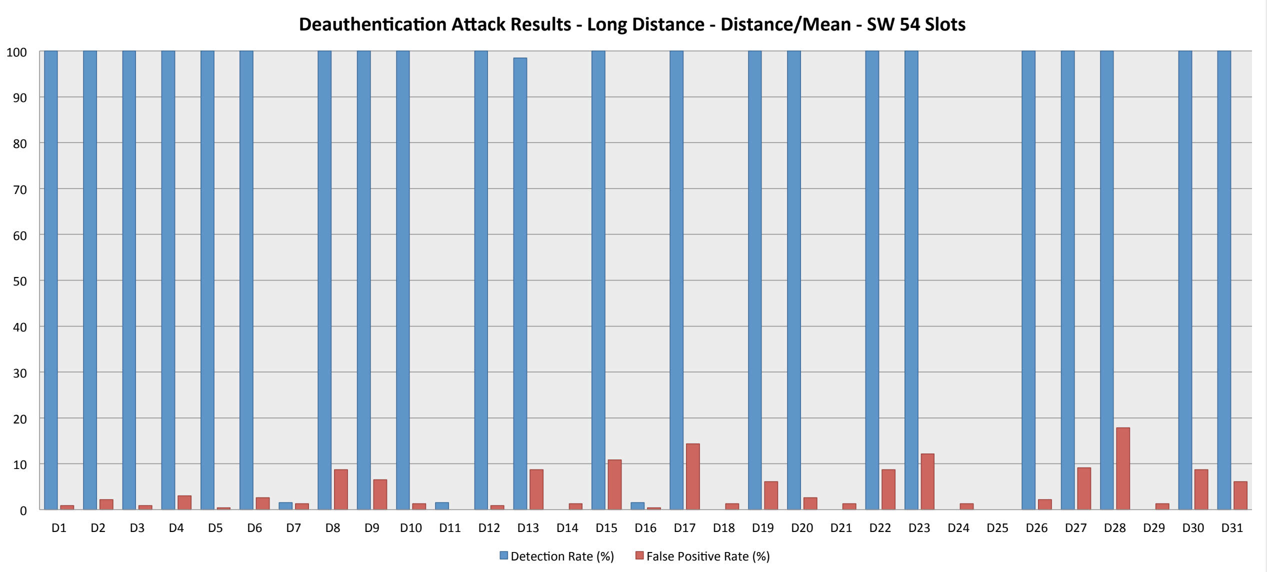


Figure 7.10 Deauthentication Long Distance - 54 Slots - Distance & Mean.

Comparing both figures helps to understand the effect that increasing the length of the sliding window produces on the detection results. In terms of $DR,$ the results for $n = 29$ and $n = 54$ remain are similar. In terms of $FP_{Rate}$, there are 13 metric combinations that produce $FP_{Rate}$ results higher than 5%, in the case of $n = 29$. Six out of these 13 metric combinations (D9, D15, D17, D23, D27, and D28) produce $FP_{Rate}$ results higher than 10%. In the case of $n = 54$, there are 12 metric combinations that produce $FP_{Rate}$ results higher than 5% and only 4 of these 12 metric combinations (D15, D17, D23, and D28) produce $FP_{Rate}$ results higher than 10%. In general, 15 metric combinations reduce the number of $FP$ alarms, and only 2 cases are increased. These results show that increasing the length value of the sliding window reduces the number of $FP$ alarms.

The $DR$ and $FP_{Rate}$ results for the configuration angle-mean, using $n = 29$ and $n = 54$, are presented in Figures 7.11 and 7.12. In the case of $n = 29$, the utilisation of angle-mean makes D7 and D14 generate 100% $DR$, as a difference to the distance-mean configuration that the $DR$ results for the two metric combinations were 3.12% and 0%, respectively. Similarly, increasing the length value of the sliding window reduces the number of $FP$ alarms. In the case of $n = 29$, there are 14 metric combinations that produce $FP_{Rate}$ results higher than 5%, and 5 out of these 14 metric combinations produce $FP_{Rate}$ results higher than 10%. These are D9, D15, D17, D27, and D28. In the case of $n = 54$, there are 13 metric combinations that produce $FP_{Rate}$ results higher than 5%, and only 5 out of these 13 metrics combinations produce $FP_{Rate}$ results higher than 10%. These are D9, D15, D17, D23, and D28. Although the difference is not very large, 16 metric combinations reduce the number of $FP$ alarms, and only 3 cases are increased. Again, increasing $n$ reduces the number of $FP$ alarms.

Comparing both configurations, the $FP_{Rate}$ results for both sliding window length are substantially increased using the distance-mean configuration. Using angle-mean, three particular metric combinations clearly exceed a 35% $FP_{Rate}$, when $n = 29$. These are (D17) $RSSI - \Delta Time$, (D27) $RSSI$, and (D28) $\Delta Time$. When $n = 54$, there are also three particular metric combinations that produce a drastic increase of $FP$ alarms. However, in this case, the metric combinations are (D17) $RSSI - \Delta Time$, (D23) $\Delta Time - SEQ_{Dif}$, and (D28) $\Delta Time$. Eventually, the $FP_{Rate}$ result of D27 is reduced to 9.13%. Hence, the presented $FP_{Rate}$ results indicate that angle, instead of distance, increases the number of $FP$ alarms for some combination of metrics with low number of metrics.

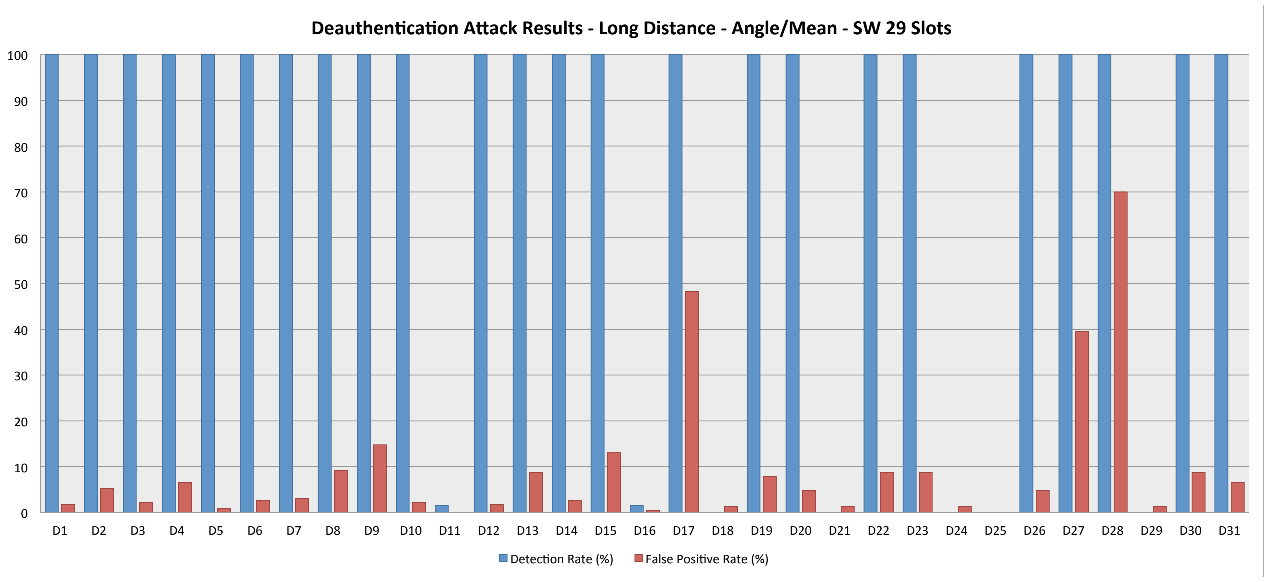


Figure 7.11 Deauthentication Long Distance - 29 Slots - Angle & Mean.

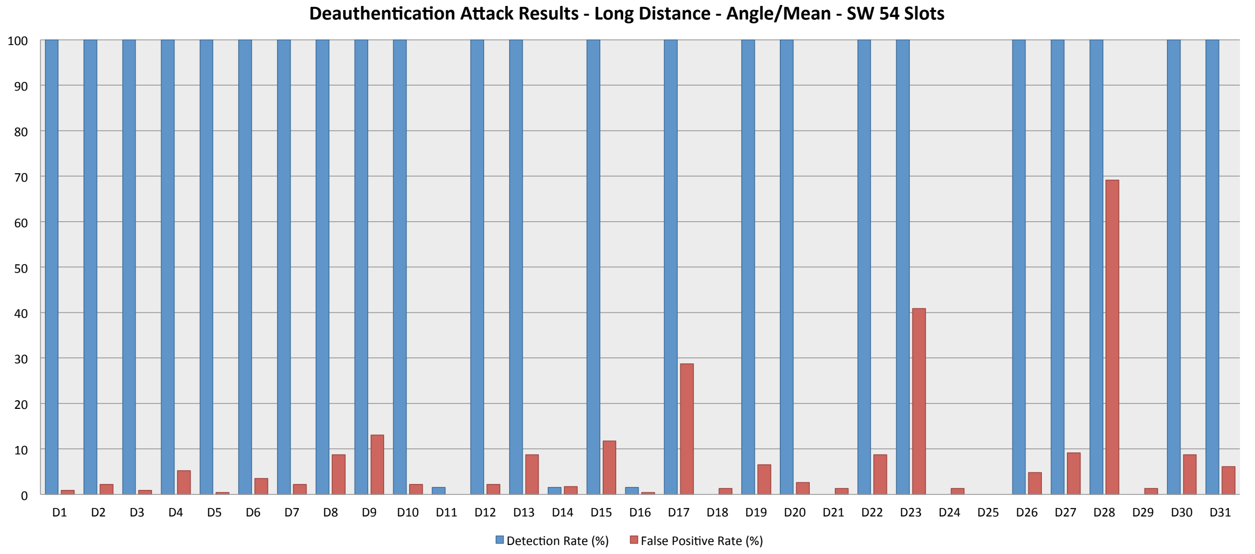


Figure 7.12 Deauthentication Long Distance - 54 Slots - Angle & Mean.

Figures 7.13 and Figure 7.14 show the $DR$ and $FP_{Rate}$ results for the distance-mode configuration, when the sliding window length is $n = 29$ and $n = 54$, respectively. Similar to what occurs with the distance-mean configuration, the metric combinations D7, D11, D14, D16, D18, D21, D24, D25, and D29 generate extremely poor $DR$ results, generating 0% of $DR$ in some cases; direct consequence of curse of dimensionality when using the $INJ_{Rate}$. In terms of $FP_{Rate}$ results, there are 12 metric combinations that produce results higher than 5%, in the case of $n = 29$. Four out of these 12 metric combinations (D15, D23, D27, and D28) produce $FP_{Rate}$ results higher than 10%. In the case of $n = 54$, there are 11 metric combinations which produce $FP_{Rate}$ results higher than 5%, and only 5 out of these 11 metric combinations (D9, D15, D17, D23, and D28) produce $FP_{Rate}$ results higher than 10%. Similar to the previous results, increasing the length value of the sliding window reduces the number of $FP$ alarms. In total, 11 metric combinations reduce the number of $FP$ alarms and 6 cases are increased, after increasing the sliding window length.

The utilisation of the distance-mode reduces the number of $FP$ alarms compared to the distance-mean, for the two assessed sliding window lengths. In the case of $n = 29$, 9 of the metric combinations produce fewer numbers of $FP$ alarms, and 14 of the metric combinations produce similar number of $FP$ alarms to the same metric combination for the case of distance-mean. On the other hand, 8 cases generate higher number of $FP$ alarms. In the case of $n = 54$, 5 of the metric combinations produce fewer numbers of $FP$ alarms, and 5 of the metrics combinations produce higher number of $FP$ alarms than the same metrics combination when using distance-mean.

Finally, Figure 7.15 shows the $DR$ and $FP_{Rate}$ results for the configuration angle-mode, when the sliding window length is $n = 29$, and Figure 7.16 shows the same results for the configuration angle-mode, when the sliding window length is $n = 54$. Again, 5 particular metric combinations generate 0% $DR$ (D18, D21, D24, D25, and D29), using both of the assessed sliding window lengths. These are the same metrics combinations that generate 0% of $DR$ for the configuration angle-mean. In terms of $FP_{Rate}$ results, there are 13 metric combinations that produce $FP_{Rate}$ results higher than 5%, in the case of $n = 29$. Six out of these 13 metrics combinations (D9, D15, D17, D23, D27, and D28) produce $FP_{Rate}$ results higher than 10%. In the case of $n = 54$, there are 12 metric combinations that produce $FP_{Rate}$ results higher than 5%, and only 5 out of these 12 metric combinations produce $FP_{Rate}$ results higher than 10%. These are D9, D15, D17, D23, and D28. Once more, increasing $n$ reduces the number of $FPs$. In total, 11 metrics combinations reduce the $FPs$ and only 4 metric combinations increased the $FPs$ after increasing the sliding window length.

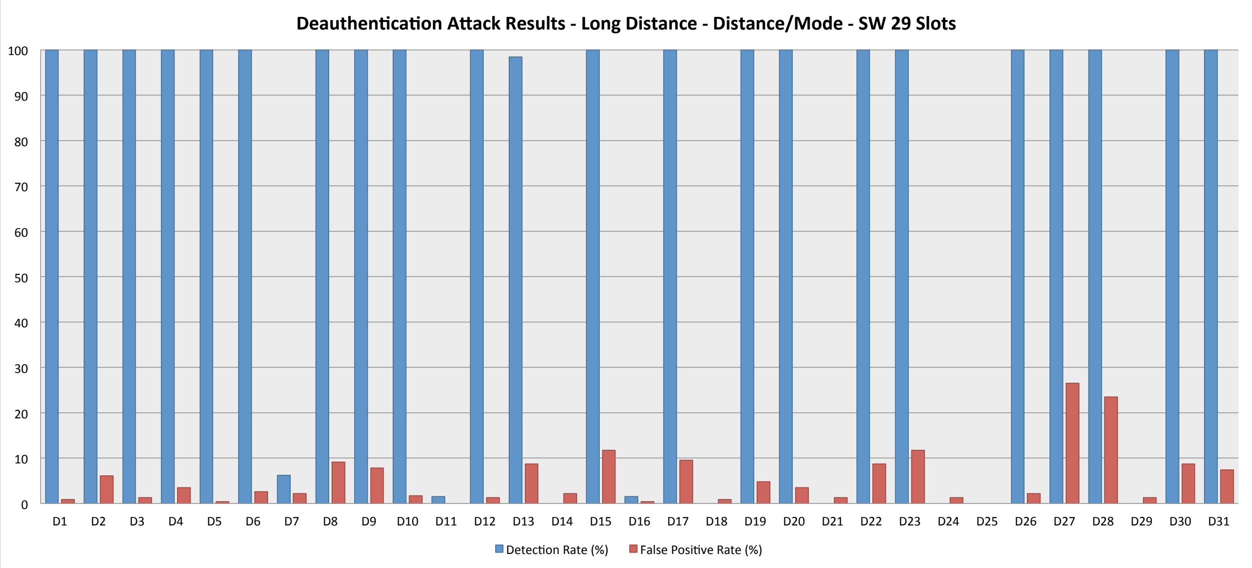


Figure 7.13 Deauthentication Long Distance - 29 Slots - Distance & Mode.

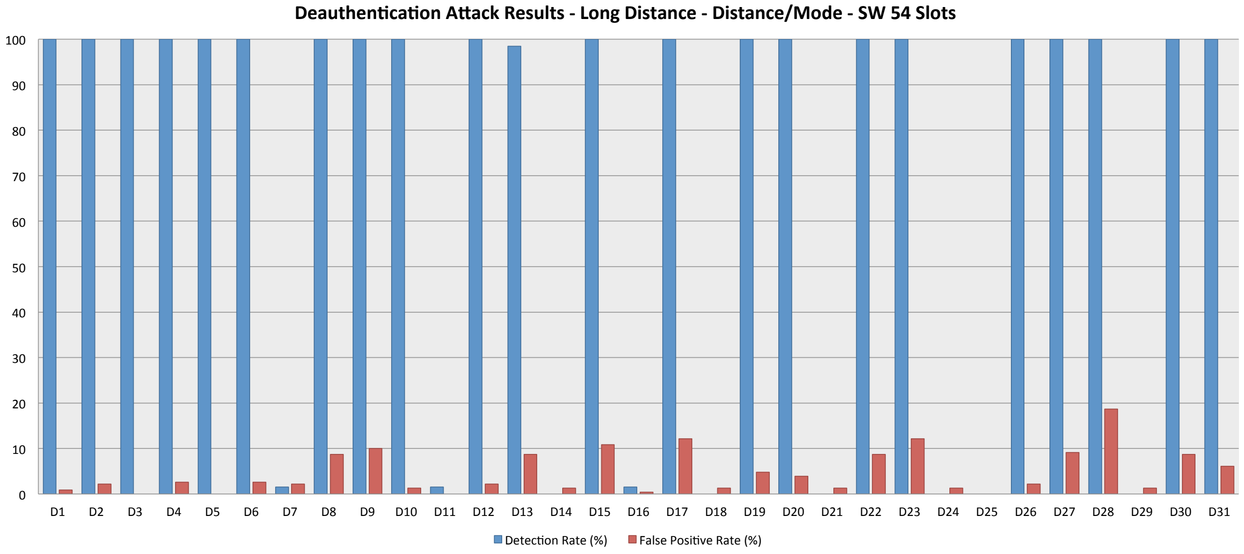


Figure 7.14 Deauthentication Long Distance - 54 Slots - Distance & Mode.

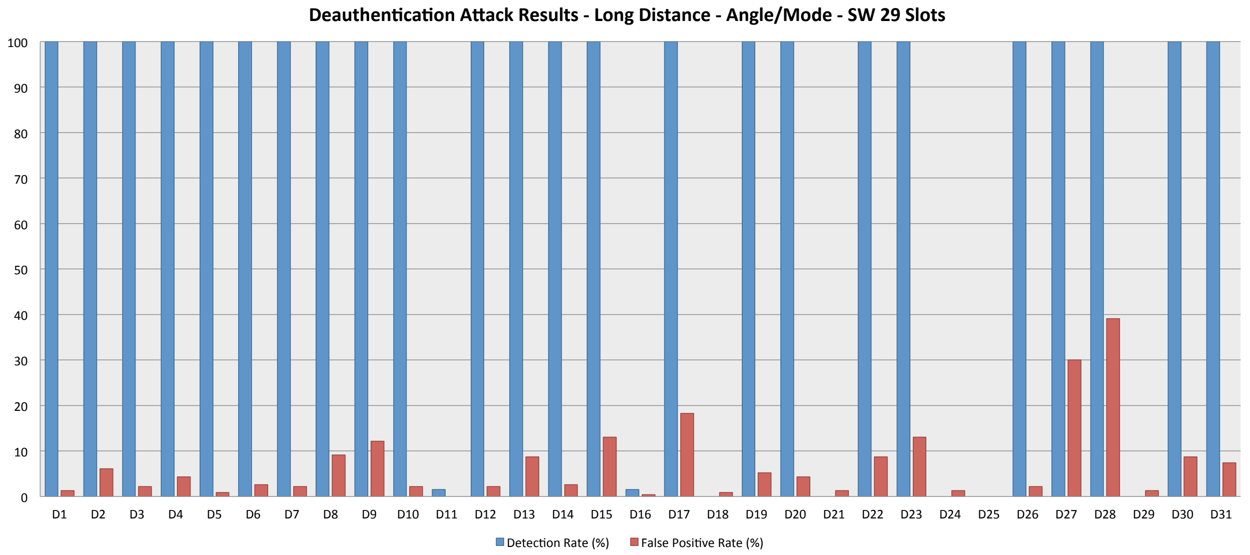


Figure 7.15 Deauthentication Long Distance - 29 Slots - Angle & Mode.

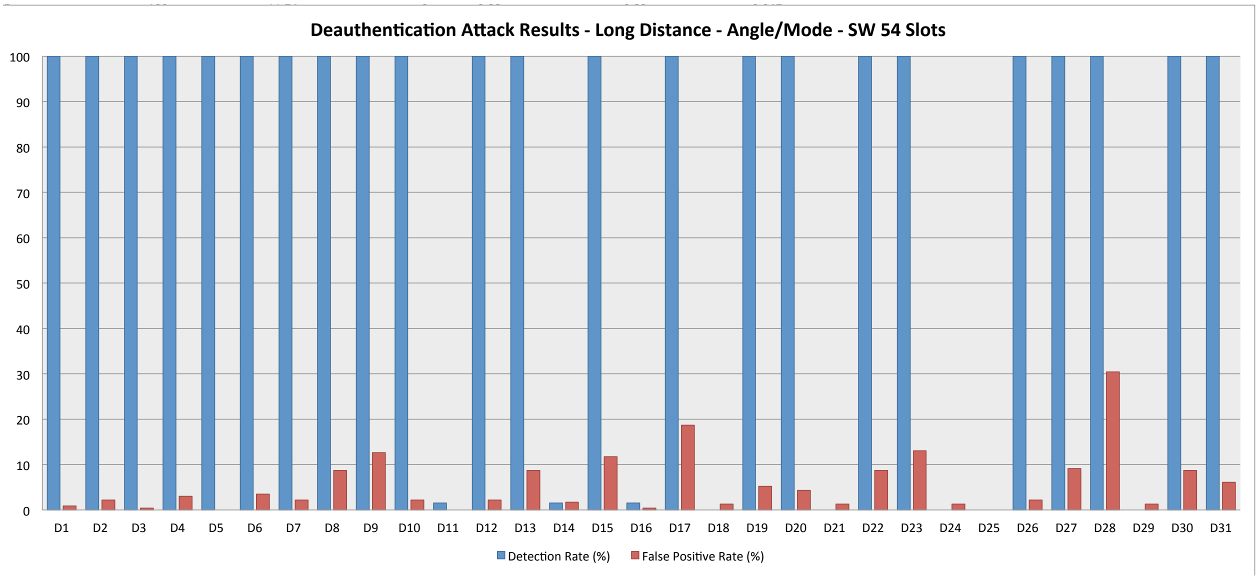


Figure 7.16 Deauthentication Long Distance - 54 Slots - Angle & Mode.

Comparing the $FP_{Rate}$ results produced by the angle-mean and angle-mode configurations, the utilisation of the mode improves the intrusion detection results. In the case of $n = 29$, 11 different metric combinations produce fewer numbers of $FP$ alarms than using the mean, while only 4 of the metric combinations produce higher numbers of $FP$ alarms than using the mean. The remaining 16 metric combinations produce similar number of $FP$ alarms for both configurations. In the case of $n = 54$, 9 different metrics combinations produce fewer numbers of $FP$ alarms than using the mean, while only 1 of the metric combinations produces higher numbers of $FP$ alarms than using the mean. The remaining 22 metric combinations produce similar number of $FP$ alarms. Hence, based on these results, the utilisation of the mode improves the efficiency of the deauthentication attack detection results. Additionally, using the angle-mean configuration, three particular metrics combinations generated $FP_{Rate}$ results higher than 28%, whilst using angle-mode reduces the number of $FP$ alarms in these particular metrics combinations. The $FP_{Rate}$ results in D17 drops from 28.7% to 18.7%, D23 drops from 40.87% to 13.04%, and D28 drops from 69.13% to 30.43%.

The presented results for the experiment using the deauthentication Long distance dataset indicate that the configuration of the methodology that produces the best results overall is the utilisation of the Euclidean distance along with the mode to establish the reference of normality (distance-mode), when the length value of the sliding window is $n \geq 54$. For this $n$ value, the required processing time per frame ranges between 45μsec and 71μsec. Nonetheless, any of the other three system configurations are able to generate highly accurate detection results.

#### 7.3.1.2 Deauthentication Attack Results – Short Distance

The multi-layer results for the deauthentication *Short Distance* dataset experiments are presented in this section. The detection of spoofed frames using all methodology configurations is completely perfect in terms of $DR$, generating 100% $DR$ for $3 \leq n$. In terms of $FP_{Rate}$, the configuration distance-mean always generates $FP_{Rate}$ results lower than 3% for any sliding window length. The best $FP_{Rate}$ results are generated when the sliding window is between 58 and 72 slots long, $[58 \leq n \leq 72]$, (0% $FP_{Rate}$), and after the sliding window reaches 80 slots, $80 \leq n$, the $FP_{Rate}$ is constantly 0.98%.

The angle-mean configuration generates $FP_{Rate}$ results slightly higher than the previous case. Perfect detection (0% $FP_{Rate}$) is never reached. Only after the sliding window length reaches 54 slots, $54 \leq n$, is the $FP_{Rate}$ is reduced to 0.98%. The best $FP_{Rate}$ results are generated when the sliding window is $[66 \leq n \leq 75]$, which generate 0.49% $FP_{Rate}$. Using the distance-mode configuration, 0% $FP_{Rate}$ is reached when the sliding window is $[60 \leq n \leq 76]$, and for any sliding window length larger than 90 slots, $90 \leq n$. For any other sliding window length, the detection system does not generate a $FP_{Rate}$ higher than 3%.

These results evidence that utilising the mode generates better detection results than using the mean in terms of $FP_{Rate}$. Similarly, the $FP_{Rate}$ results generated using the configuration angle-mode also improve on the results generated utilising the angle-mean. The system generates 0% $FP_{Rate}$ when $90 \leq n$. For any other length value of the sliding window, the detection system generates $FP_{Rate}$ results higher than 4% only in one case, $n = 84$.

In order to shown which of the four different assessed configurations is the most appropriate, Figure 7.17 represents a close comparison of the $FP_{Rate}$ results of the detection systems for the four different assessed methodology configurations. The utilisation of the distance-mode configuration again produces the best results overall. The value of the sliding window length that best results produces is $60 \leq n \leq 76$. Table VII.IV shows the results for each of the four cases when $n = 60$. For that particular sliding window length, the results generated by the distance and either the mean or the mode, are identical.

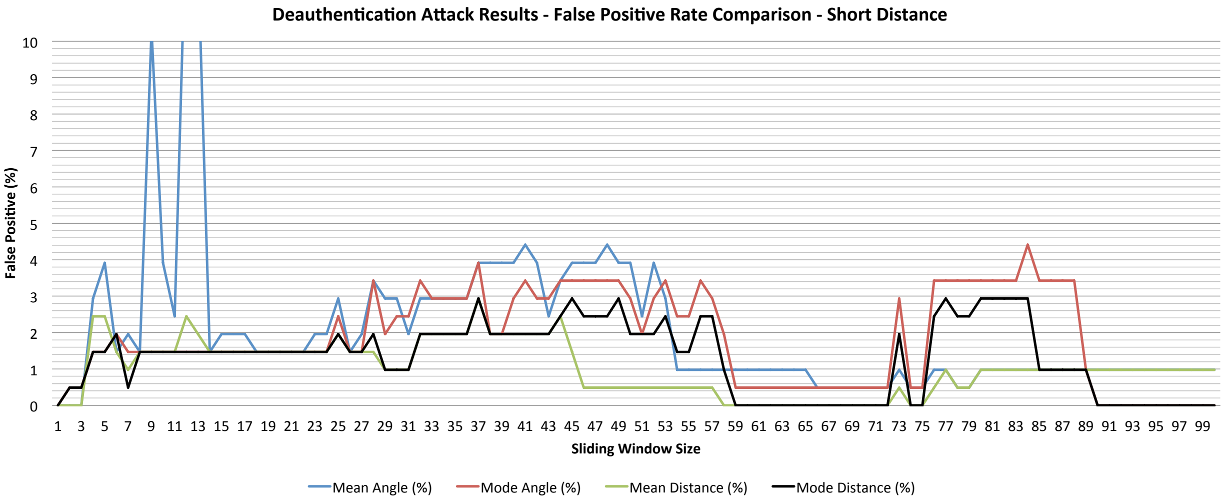


Figure 7.17 Deauthentication Short Distance - $FP_{Rate}$ Results Comparison.

The average processing time required by the detection system to provide a final decision is shown in Figure 7.18. The average processing time for the four different assessed configurations, when sliding window length is $n = 60$, is between 55μsec and 77μsec, being the distance-mode the configuration that longest processing time requires. As can be seen, all the assessed configurations require almost the same processing time, and follow a similar increasing trend. Similar to the long distance experiments, the intrusion detection process can fairly be implemented in real time because the average interarrival time between two consecutive frames is 49msec.

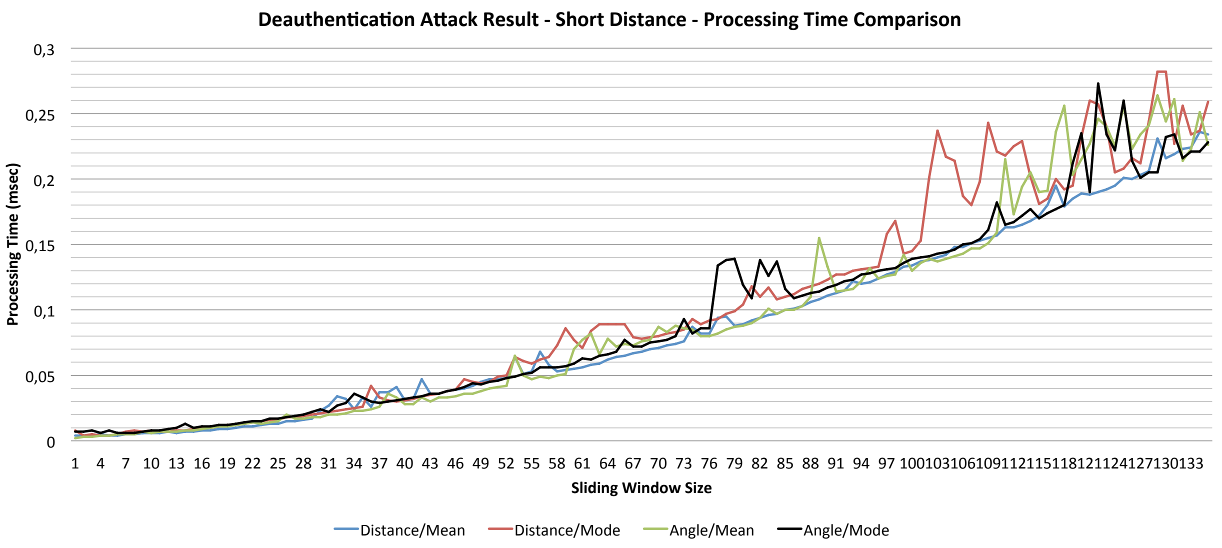


Figure 7.18 Deauthentication Short Distance - Per Frame Detection Analysis Processing Time.

TABLE VII.IV. DEAUTHENTICATION SHORT DISTANCE - 5 METRICS RESULTS - 60 SLOTS

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time\ (\mu sec)$ |
|---|---|---|---|---|---|---|---|
| Distance // Mean | 100 | 0 | 0 | 1 | 1 | 1 | 55 |
| Distance // Mode | 100 | 0 | 0 | 1 | 1 | 1 | 77 |
| Angle//Mean | 100 | 0.98 | 0 | 0.99 | 0.969 | 0.98 | 70 |
| Angle//Mode | 100 | 0.49 | 0 | 0.99 | 0.984 | 0.99 | 59 |

The multi-layer results for the deauthentication *Short Distance* dataset present more inconsistent $FP_{Rate}$ results than the multi-layer results for the deauthentication *Long Distance* dataset. Also, the sliding window length should be larger for the short distance deauthentication attack experiments than the long distance experiments to generate the best intrusion detection results. Hence, the average required processing time would also increase. This is a direct effect of the changes in the metric characteristics generated by the modification in the wireless network topology. It is also very important to highlight the fact that the $DR$ results remain unaltered from one set of experiments to the other, despite the modification in the network topology.

The intrusion detection results generated when using fewer numbers of metrics are shown next. Figures 7.19 – 7.22 represent the $FP_{Rate}$ results of the metric combinations D2, D3, D4, D5, and D6, when the system uses the configurations distance-mean, distance-mode, angle-mean and angle-mode, respectively. Similar to the long distance experiments, only one metric combination outperforms the overall results of D1 for the four assessed methodology configurations. This is the D5 combination of $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$. The four methodology configurations generate similar $DR$ results. These results are constantly 100% when the sliding window is $[8 \leq n \leq 118]$. In contrast to the results for D1, the intrusion detection effectiveness drastically drops after $n$ reaches 119 slots long.

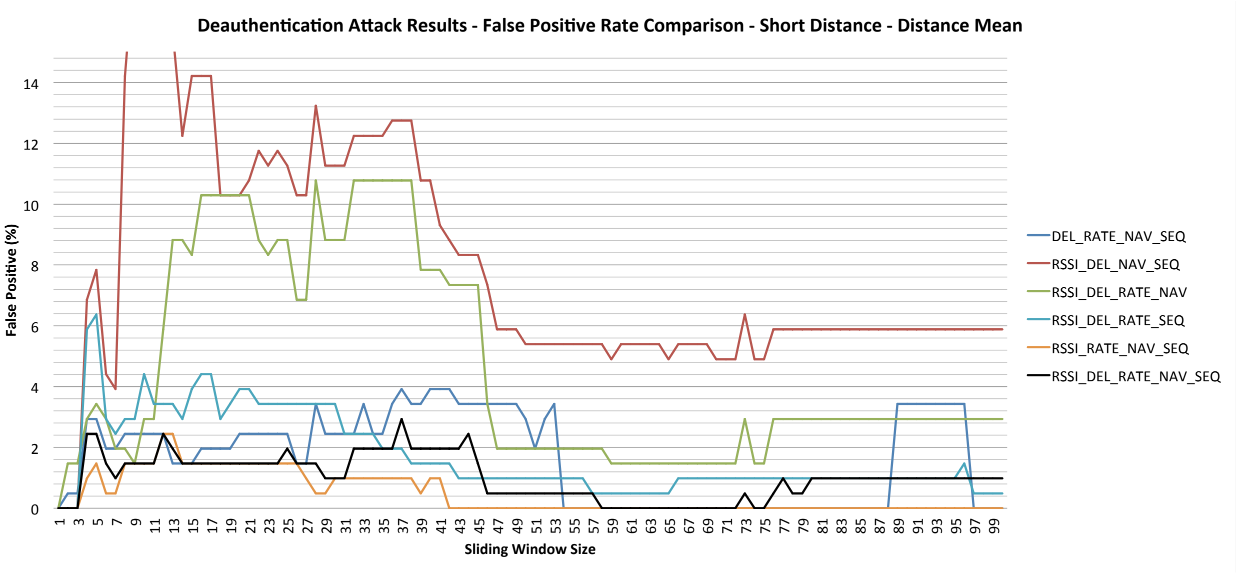


Figure 7.19 Deauthentication Short Distance - $FP_{Rate}$ 4 Metrics - Distance & Mean.

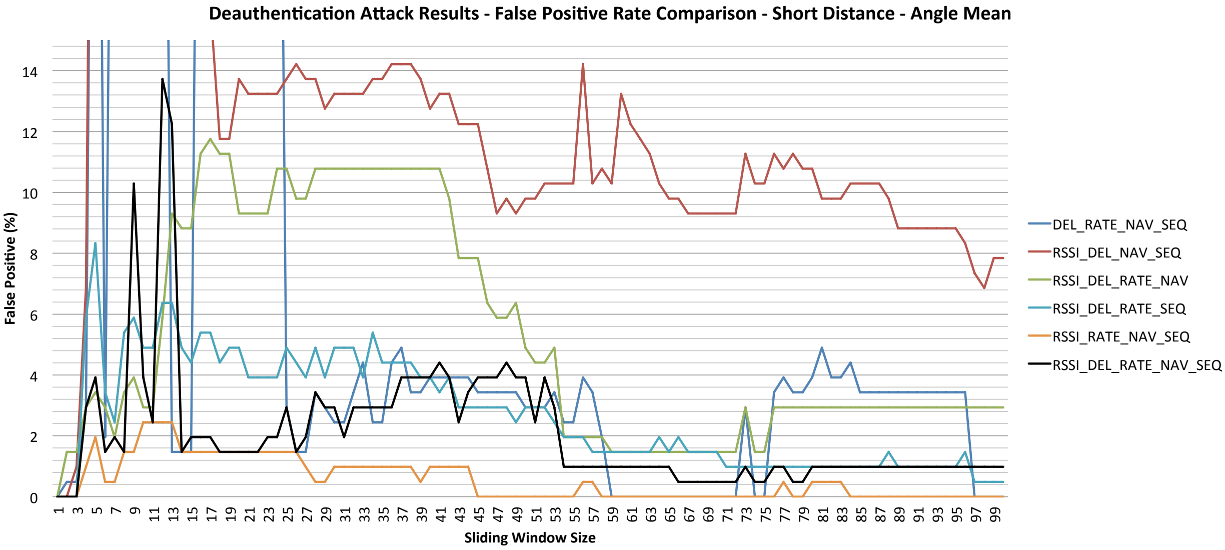


Figure 7.20 Deauthentication Short Distance - $FP_{Rate}$ 4 Metrics - Angle & Mean.

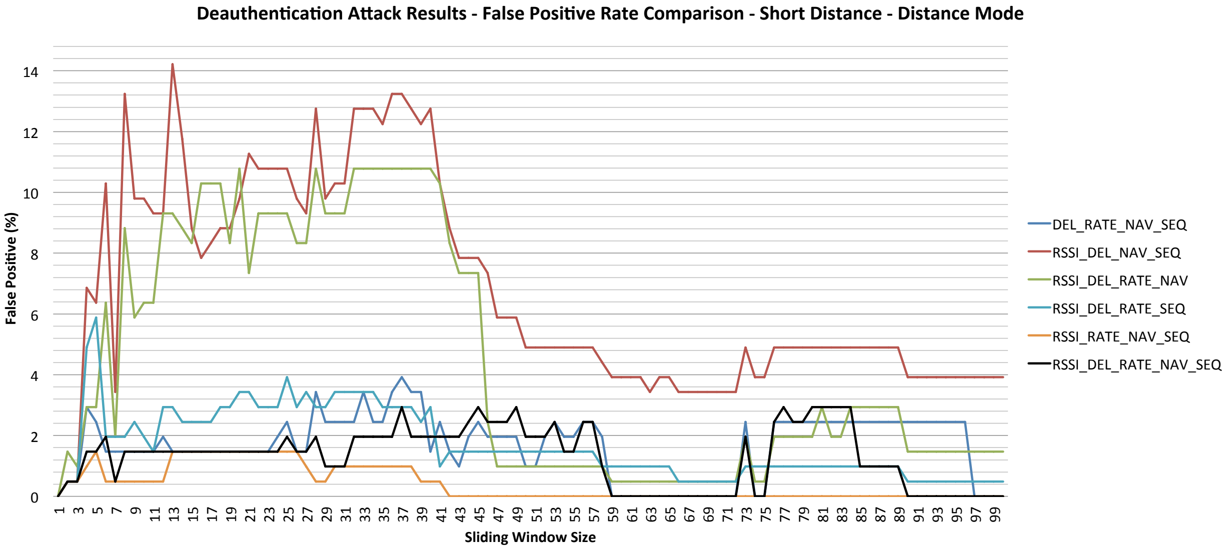


Figure 7.21 Deauthentication Short Distance -$FP_{Rate}$ 4 Metrics - Distance & Mode.

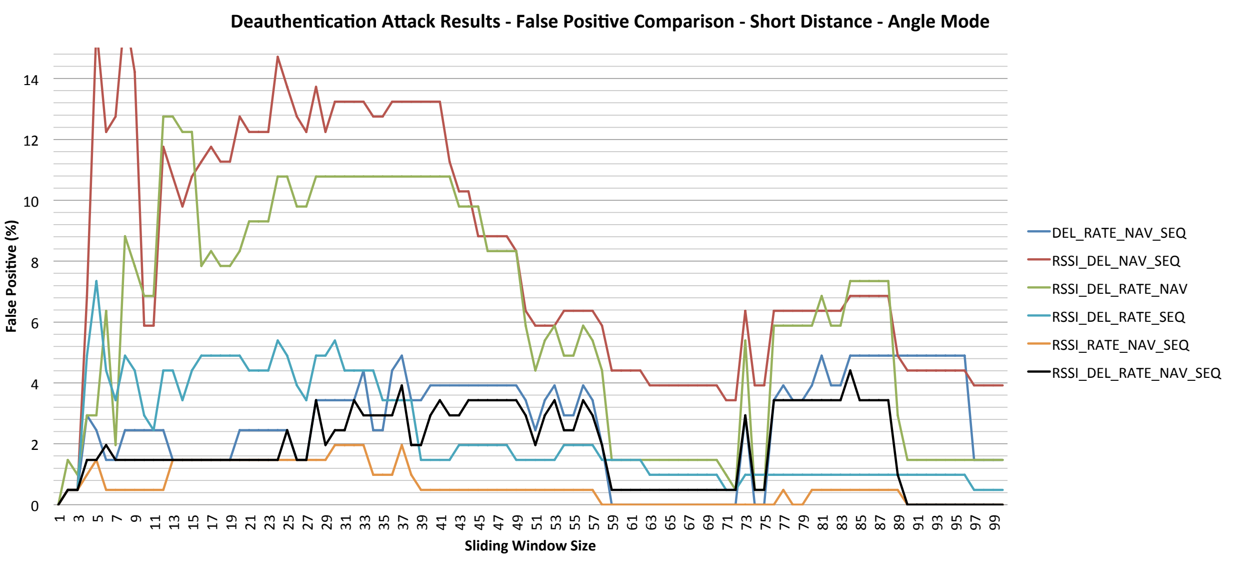


Figure 7.22 Deauthentication Short Distance - $FP_{Rate}$ 4 Metrics - Angle & Mode.

Figure 7.23 represents the $FP_{Rate}$ results of D5 for the four different assessed methodology configurations. The detection system does not generate $FP_{Rate}$ results higher than 2.5% for any $n$ value. The experiments using the distance-mode configuration produce the best result overall. Specifically, when the sliding window is $42 \leq n$, the system generates 0% $FP_{Rate}$. For this sliding window length, the average processing time to produce perfect detection is 35µsec. The experiment using distance-mean also generates 0% $FP_{Rate}$ when $42 \leq n$. However, using the distance-mean configuration, the system generates higher number of $FP$ alarms when the sliding window $[3 \leq n \leq 15]$. In this case, the average processing time is 30µsec. The experiments that use the angle produce slightly higher numbers of $FP$ alarms than using the distance.

Table VII.V shows a comparison of all the methodology configurations for the two different sliding window lengths. The top four rows of the table show the results when $n = 42$, which produces the best results for the metric combination D5. The bottom four rows of the table show the results when $n = 60$, which is the length value of the sliding window that produces the best results for the metric combination D1.

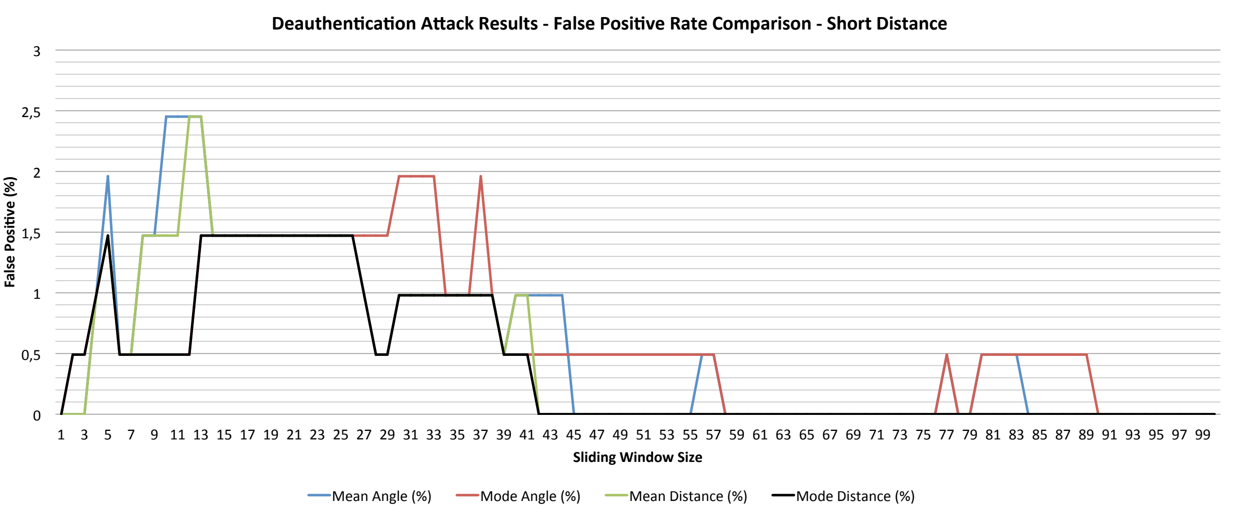


Figure 7.23 Deauthentication Short Distance - $FP_{Rate}$ Results Comparison - $RSSI\ INJ_{Rate}\ NAV\ SEQ_{Dif}$.

TABLE VII.V. Deauthentication Long Distance Results - 4 Metrics - $RSSI\ INJ_{Rate}\ NAV\ SEQ_{Dif}$ - 42 // 60 slots.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time$ (μsec |
|---|---|---|---|---|---|---|---|
| Distance // Mean | 100 | 0 | 0 | 1 | 1 | 1 | 30 |
| Distance // Mode | 100 | 0 | 0 | 1 | 1 | 1 | 35 |
| Angle//Mean | 100 | 0.98 | 0 | 0.99 | 0.969 | 0.98 | 33 |
| Angle//Mode | 100 | 0.49 | 0 | 0.99 | 0.984 | 0.99 | 35 |
| Distance // Mean | 100 | 0 | 0 | 1 | 1 | 1 | 62 |
| Distance // Mode | 100 | 0 | 0 | 1 | 1 | 1 | 63 |
| Angle//Mean | 100 | 0 | 0 | 1 | 1 | 1 | 52 |
| Angle//Mode | 100 | 0 | 0 | 1 | 1 | 1 | 58 |

Figures 7.24 and 7.25 show the $DR$ and $FP_{Rate}$ results for the configuration distance-mean using all the possible combinations, using $n = 42$ and $n = 60$. The Y-axis of the graphs represents the percentage of $DR$ and $FP_{Rate}$. The X-axis of the graphs represents the different metrics combinations indexes. Similar to the long distance experiments, there exist a particular number of metric combinations that generate extremely poor $DR$ results, generating 0% $DR$ in some cases. These metric combinations are D7, D11, D14, D16, D18, D21, D24, D25, and D29. Again, the $INJ_{Rate}$ is the principal cause of these poor $DR$ results.

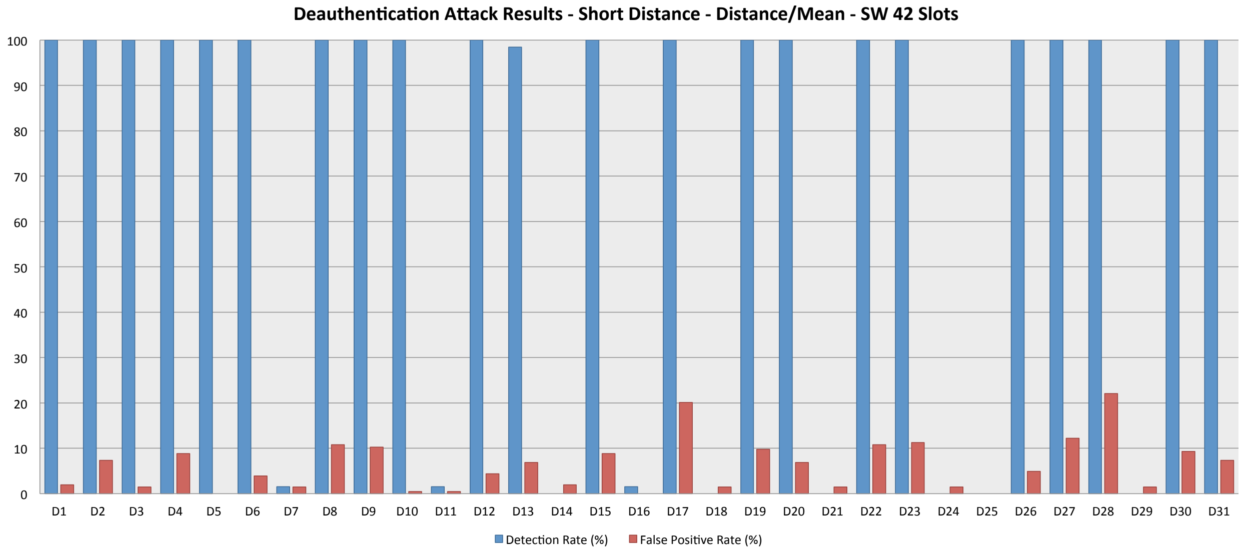


Figure 7.24 Deauthentication Short Distance - 42 Slots - Distance & Mean.

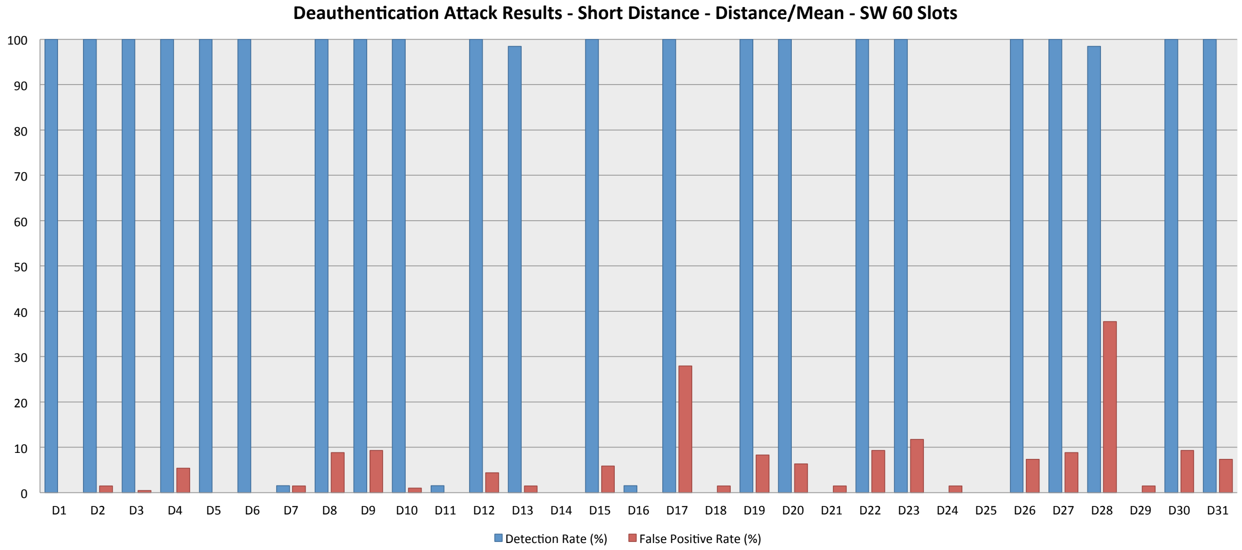


Figure 7.25 Deauthentication Short Distance - 60 Slots - Distance & Mean.

Comparing these two graphs, the presented $DR$ results of both cases remain unchanged after increasing the sliding window length. There is a small $DR$ drop for the single metric $\Delta Time$ (D28). In terms of $FP_{Rate}$, there are 15 metric combinations that produce $FP_{Rate}$ results higher than 5%, in the case of $n = 42$. Seven out of these 15 metrics combinations (D8, D9, D17, D22, D23, D27, and D28) produce $FP_{Rate}$ results higher than 10%. In the case of $n = 60$, there are 14 metric combinations that produce $FP_{Rate}$ results higher than 5%, and only 3 out of these 14 metric combinations produce $FP_{Rate}$ results higher than 10%. These are D17, D23, and D28. In two particular cases (D17 and D28), increasing the length value of the sliding window makes the number of $FP$ alarms drastically increase from 20.1% to 27.94% and 22.06% to 37.75%, respectively. However, 16 different metric combinations reduce the $FP_{Rate}$ results by increasing the sliding window length.

The $DR$ and $FP_{Rate}$ results for the configuration angle-mean are presented in Figures 7.26 and 7.27, respectively. In both cases, the metric combinations D11, D14, D16, D18, D21, D24, D25, and D29 generate extremely poor $DR$ results. In the case of $n = 42$, there are 17 metric combinations that produce $FP_{Rate}$ results higher than 5%, and ten out of these 17 metric combinations produce $FP_{Rate}$ results higher than 10%. In the case of $n = 60$, there are 14 metric combinations that produce $FP_{Rate}$ results higher than 5%. Nine out of these 14 metric combinations produce $FP_{Rate}$ results higher than 10%. In particular, D8, D15, D17, D22, D23, and D28, overpass 25% $FP_{Rate}$, two of these metric combinations overpass 50% $FP_{Rate}$. One particular metric that is included in all these metric combination is the $\Delta Time$. These results show that increasing the sliding window length, degrades the intrusion detection results when the angle-mean configuration is used. In general, 11 metric combinations reduce the number of $FP$ alarms and 9 cases increase the number of $FP$ alarms, after increasing the length value of the sliding window. Comparing the detection results generated by the angle-mean against the distance-mean configuration, the utilisation of the angle makes the system to produce higher number of $FP$ alarms for this dataset.

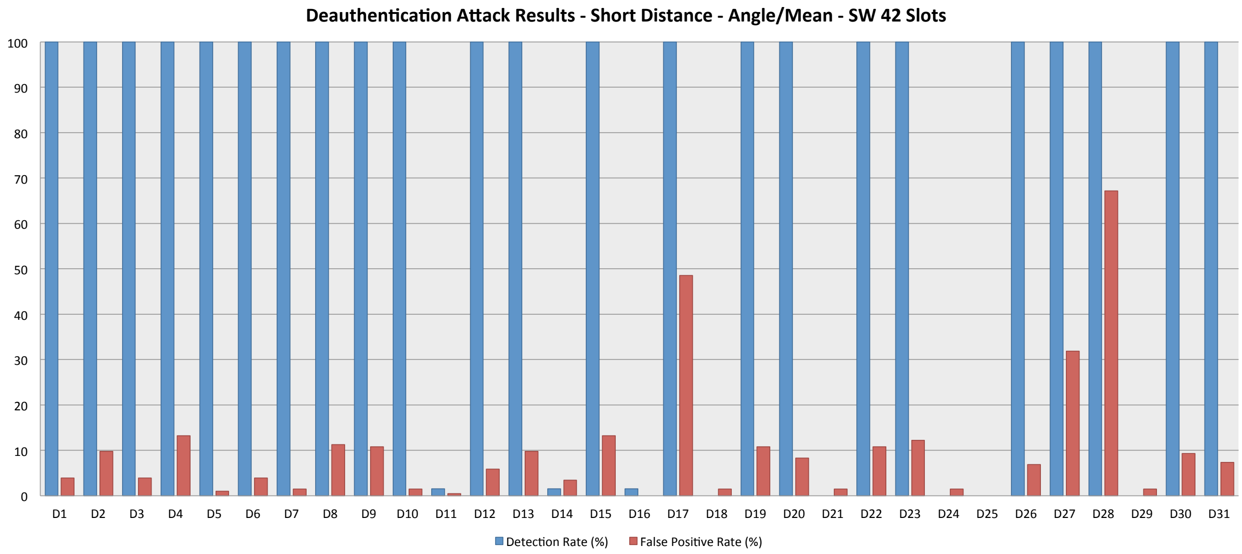


Figure 7.26 Deauthentication Short Distance - 42 Slots - Angle & Mean.

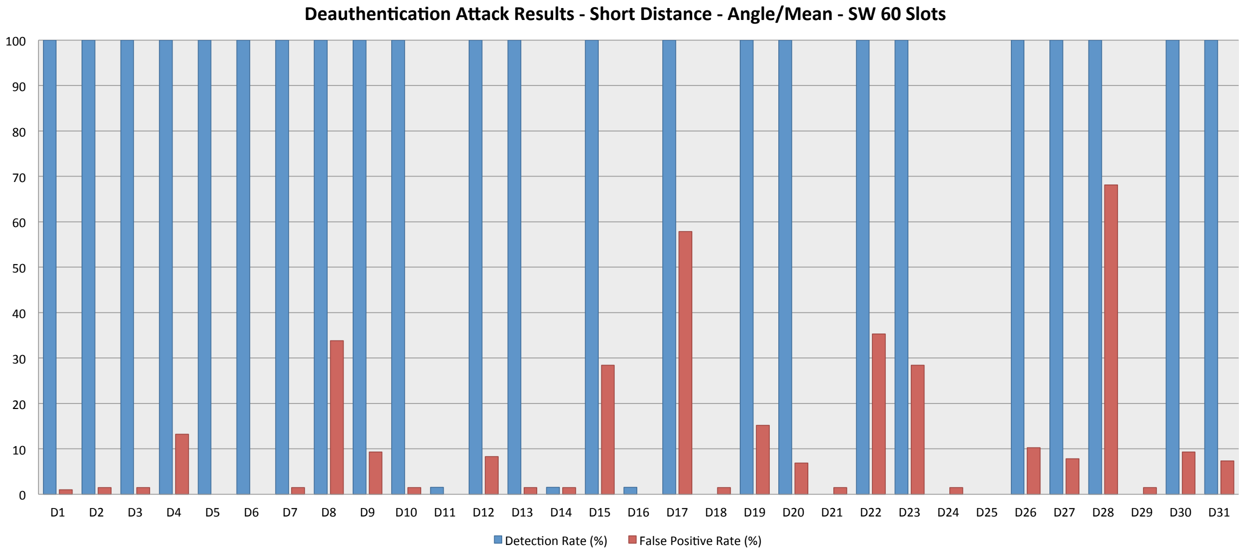


Figure 7.27 Deauthentication Short Distance - 60 Slots - Angle & Mean.

Figure 7.28 shows the $DR$ and $FP_{Rate}$ results for the distance-mode configuration, when the sliding window length is $n = 42$. There are 15 metric combinations that produce $FP_{Rate}$ results higher than 5%. Six out of these 15 metric combinations (D8, D17, D22, D23, D27, and D28) produce $FP_{Rate}$ results higher than 10%. Comparing these results against the results generated by the distance-mean configuration, 7 metric combinations improve the distance-mean results while 5 metric combinations generate higher number of $FP$ alarms. Figure 7.29 shows the $DR$ and $FP_{Rate}$ results for the configuration distance-mode, when the sliding window length is $n = 60$. Eleven of the metric combinations produce $FP_{Rate}$ results higher than 5%, and only 4 out of these 11 metric combinations produce $FP_{Rate}$ results higher than 10%. These are D17, D23, D27, and D28. Again, increasing the length value of the sliding window reduces the number of $FP$ alarms. In total, 18 metric combinations reduce the number of $FP$ alarms and the number of $FP$ alarms generated by 13 metric combinations reaming unchanged. None of the metric combinations increase the $FP_{Rate}$ results.

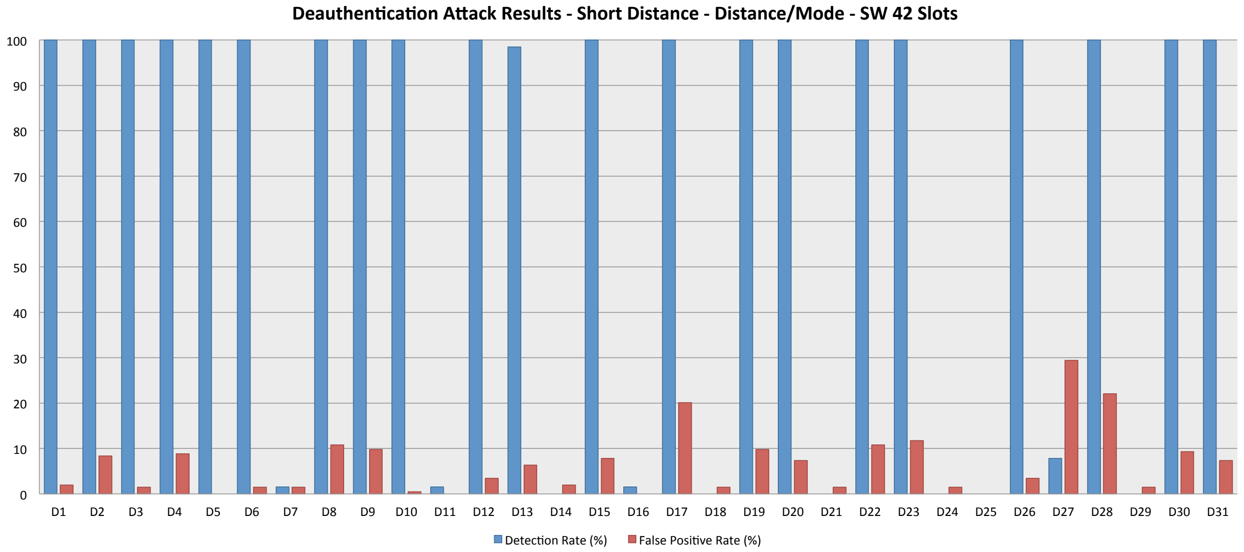


Figure 7.28 Deauthentication Short Distance - 42 Slots - Distance & Mode.

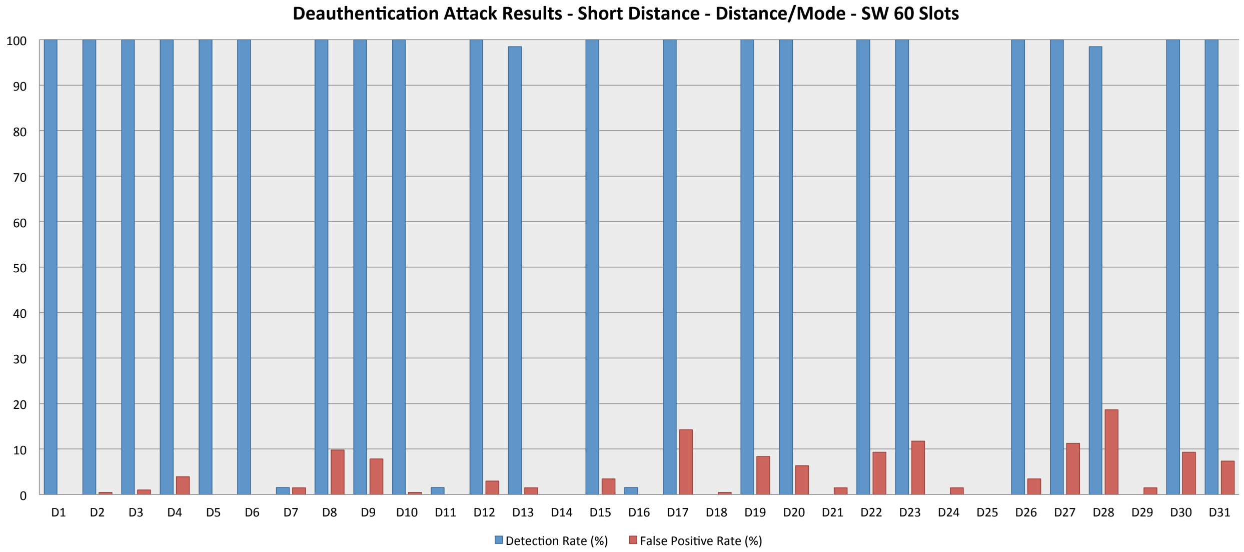


Figure 7.29 Deauthentication Short Distance - 60 Slots - Distance & Mode.

The utilisation of the distance-mode configuration reduces the number of $FP$ alarms compared to the utilisation of the mean, for the two assessed sliding window lengths. In the case of $n = 60$, the $FP_{Rate}$ results generated by the distance-mode configuration outperform the results by the distance-mean configuration. Ten of the metric combinations produce fewer numbers of $FP$ alarms, and 18 of the metric combinations produce similar number of $FP$ alarms than the same metric combination for the case of distance-mean. On the other hand, only 3 cases generate higher numbers of $FP$ alarms. The reduction of the number of $FP$ alarms of the short distance experiments utilising the mode instead of the mean for the cases in which more metrics are combined, is not as noticeable as the long distance deauthentication attack experiment. Although very small, there is improvement in the efficiency of the detection results when the mode is used.

Finally, Figure 7.30 shows the $DR$ and $FP_{Rate}$ results for the configuration angle-mode, using $n = 42$, and Figure 7.31 shows the same results for the configuration angle-mode, when the sliding window length is $n = 60$. In the case of $n = 42$, there are 15 metric combinations that produce $FP_{Rate}$ results higher than 5%, and 11 out of these 15 metric combinations produce $FP_{Rate}$ results higher than 10%. Three of these metric combinations overpass 30% of $FP_{Rate}$. In the case of $n = 60$, there are 11 metric combinations that produce $FP_{Rate}$ results higher than 5%, and 5 out of these 11 metric combinations produce $FP_{Rate}$ results higher than 10%. Only the metric combination $\Delta Time$ (D28) overpasses 30% of $FP_{Rate}$. Once more, increasing the length value of the sliding window reduces the number of $FP$ alarms. In total, 18 metric combinations reduce the number of $FP$ alarms and none metric combination increased the number of $FP$ alarms, after increasing the sliding window length.

Comparing the results generated by the angle-mode configuration against the results generated by the angle-mean configuration, 11 metric combinations improve the angle-mean results while 5 metric combinations generate higher number of $FP$ alarms, in the case of $n = 42$. The $FP_{Rate}$ results generated by the remaining 14 metric combinations are unchanged. In the case of $n = 60$, 14 metric combinations improve the angle-mean results while 2 metric combinations generate higher number of $FP$ alarms. The $FP_{Rate}$ results generated by the remaining 15 metric combinations are unchanged.

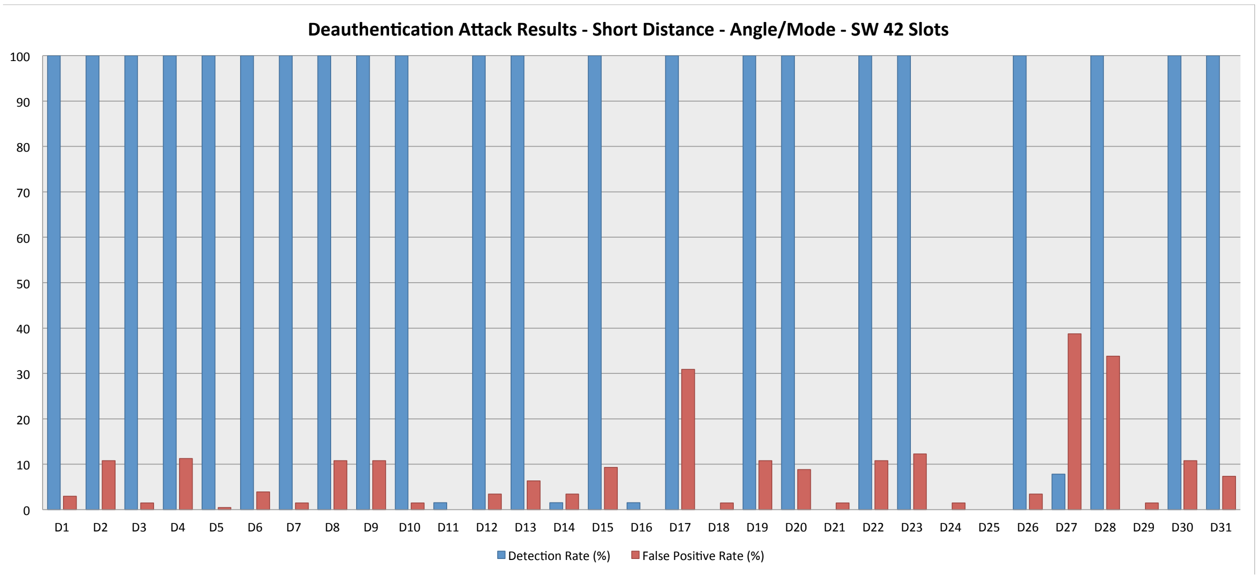


Figure 7.30 Deauthentication Short Distance - 42 Slots - Angle & Mode.

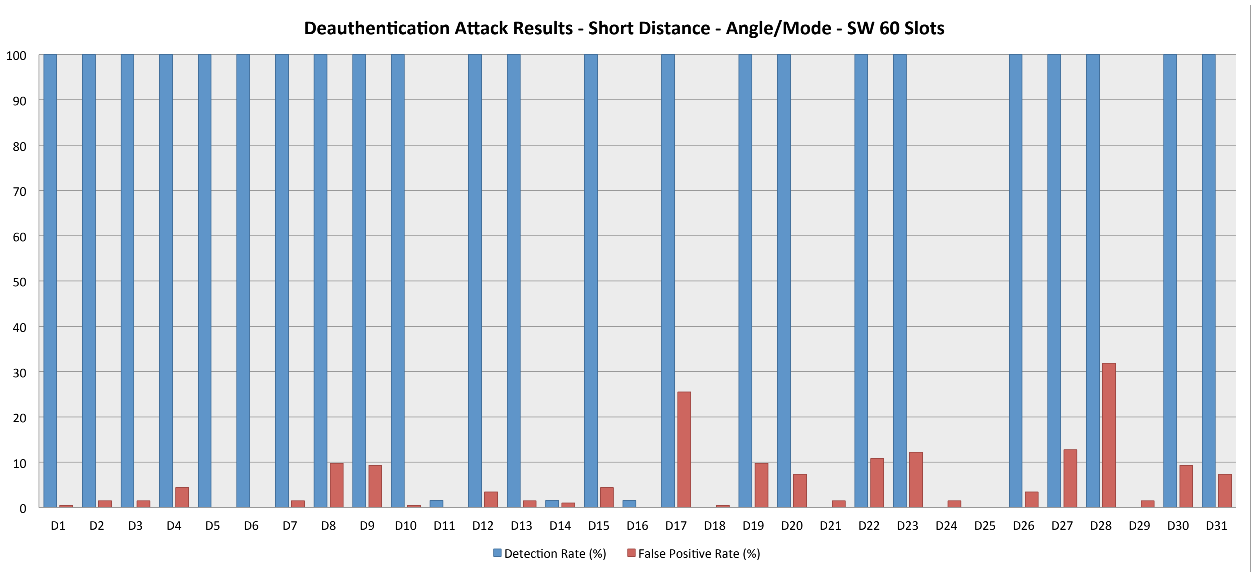


Figure 7.31 Deauthentication Short Distance - 60 Slots - Angle & Mode.

The utilisation of the mode as reference of normality generally improves the $FP_{Rate}$ results of the detection system, for the deauthentication *Short Distance* dataset experiment. On the other hand, comparing the $FP_{Rate}$ results produced by the angle-mode against distance-mode configuration, the utilisation of the angle to assign the belief in $Attack$ reduces the effectiveness of the detection system. Most of the metric combinations generate higher number of $FP$ alarms than the distance-mode results, when the sliding window length is $n \geq 60$. For this $n$ value, the required processing time per frame ranges between 52µsec and 63µsec.

## 7.3.2 Airpwn Attack Experiments

Airpwn is another attack that has been implemented to evaluate the proposed methodology. Two different versions of the same attack have been implemented. Similar to the deauthentication attack analysis, the experiment results have been evaluated using all possible combination of metrics. Six different metrics have been selected. Therefore, the same wireless network dataset has been evaluated 63 times. Additionally, for each combination of metrics, the results are plot using a sequentially increasing length value of the sliding window. The length value $n$ varies from one single slot to 200 slots. In addition, the four possible methodology configurations have been evaluated for each of the metrics combinations.

All these configurations have been evaluated using the four different Airpwn datasets. The first dataset only contains non-malicious traffic instances. The second, $Attack01$, and third dataset, $Attack02$, contain both normal and malicious network traffic instances. Finally, the fourth dataset, $Mixed\ Attack$ contains normal traffic instances, as well as malicious network traffic instances generated by both Airpwn attack versions. The testbed for the detection of the deauthentication attack is the generic testbed described in Chapter 6. All the devices are located in a stationary geographical location, being the attacker is placed 1.5 metres away from the victim.

### 7.3.2.1 Airpwn Attack Experiments Results

The multi-layer results for the Airpwn attack experiments, using the six considered metrics, are presented in the Figures 7.32 – 7.37. Figure 7.32 represents intrusion detection results for the multi-layer approach on the completely non-malicious dataset. This figure represents the $FP_{Rate}$ for the four methodology configurations, modifying the length value of the sliding window. As can be seen, none of the intrusion detection results overpasses 0.45% $FP_{Rate}$. The two methodology configurations that make use of the distance generate 0% $FP_{Rate}$ for sliding window length $31 \leq n$, whilst the two methodology configurations that make use of the angle generate 0% $FP_{Rate}$ for the sliding window length $54 \leq n$.

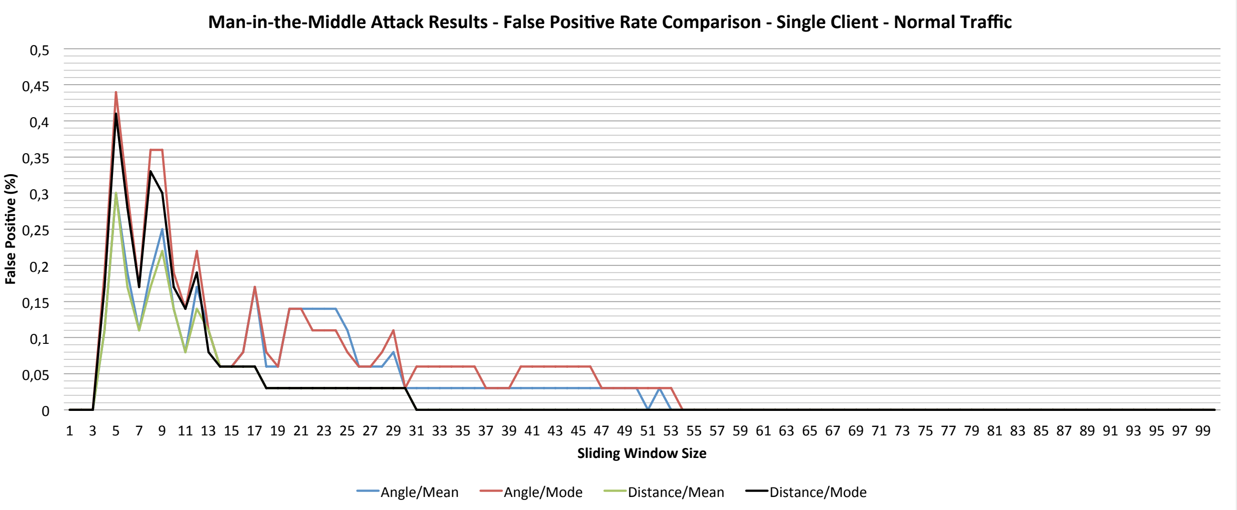


Figure 7.32 Airpwn Attack - Normal Traffic Dataset - 6 metrics - $FP_{Rate}$ Results Comparison.

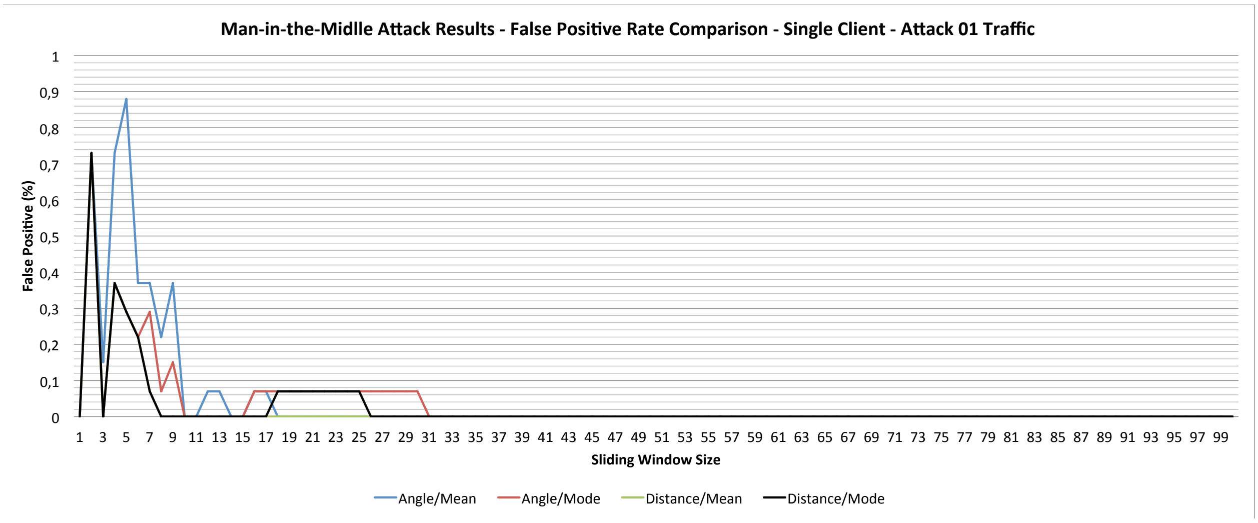


Figure 7.33 Airpwn Attack - $Attack01$ Dataset - 6 metrics - $FP_{Rate}$ Results Comparison.

Figure 7.33 represents the detection results for the multi-layer approach, using the six considered metrics, when the $Attack01$ dataset is used. The $DR$ results are not shown in this graph, but these are constantly 100% for any of the four methodology configurations when $4 \leq n$. In terms of $FP_{Rate}$, the detection system using the distance-mean configuration generates the best results overall. Nonetheless, the difference with the other three methodology configuration variations is very small. Perfect detection is generated after the sliding window length reaches 9 slots long for the two methodology configurations that make use of the distance. All the metric configurations generate 0% $FP_{Rate}$ for the sliding window length $31 \leq n$. None of the detection results overpass 0.45% $FP_{Rate}$ for any sliding window length.

Figures 7.34 and 7.35 represent the $DR$ and $FP_{Rate}$ results for the multi-layer approach when the $Attack02$ dataset is used, respectively. In terms of $DR$ results, the four methodology configurations produce 100% $DR$ for $17 \leq n$. For this particular dataset, the angle-mean configuration produces the best $DR$ results overall. Especially when the sliding window length is relatively small. While the other three configurations drop the effectiveness of the detection system for $[6 \leq n \leq 7]$, the $DR$ results of the angle-mean configuration remain above 85%. In addition, this configuration outperforms the other three for $[7 \leq n \leq 17]$. In terms of $FP_{Rate}$, none of the four methodology configurations achieve 0% $FP_{Rate}$. The two configurations that make use of the Euclidean distance to assign the belief in $Attack$ generate 0.01% $FP_{Rate}$ after the sliding window length reaches 89 slots long. The other two configurations using the angle generate 0.03% $FP_{Rate}$ for $89 \leq n$.

Figures 7.36 and 7.37 represent the $DR$ and $FP_{Rate}$ results when the $Mixed$ $Attack$ dataset is used. The $DR$ results generated by all the configurations are very similar to each other. All the methodology configurations produce 100% of DR for $8 \leq n$. In terms of $FP_{Rate}$, none of the four methodology configurations achieve 0% of $FP_{Rate}$ for $158 \geq n$. The distance-mean configuration, which produces the best results overall, generate 0.01% of $FP_{Rate}$ after the sliding window length reaches 91 slots long. None of the two configurations that use the Euclidean distance overpasses 0.05% of $FP_{Rate}$ for $31 \leq n$. The other two configurations generate slightly higher $FP_{Rate}$

results than the two configurations using the distance. Nonetheless, none of the configurations can be considered a good option for sliding window length $n \geq 10$. From the presented multi-layer results for the Airpwn attack experiments, the difference between the $DR$ and $FP_{Rate}$ results generated with each of the methodology configurations is very small. None of the configurations substantially outperforms any of the other three.

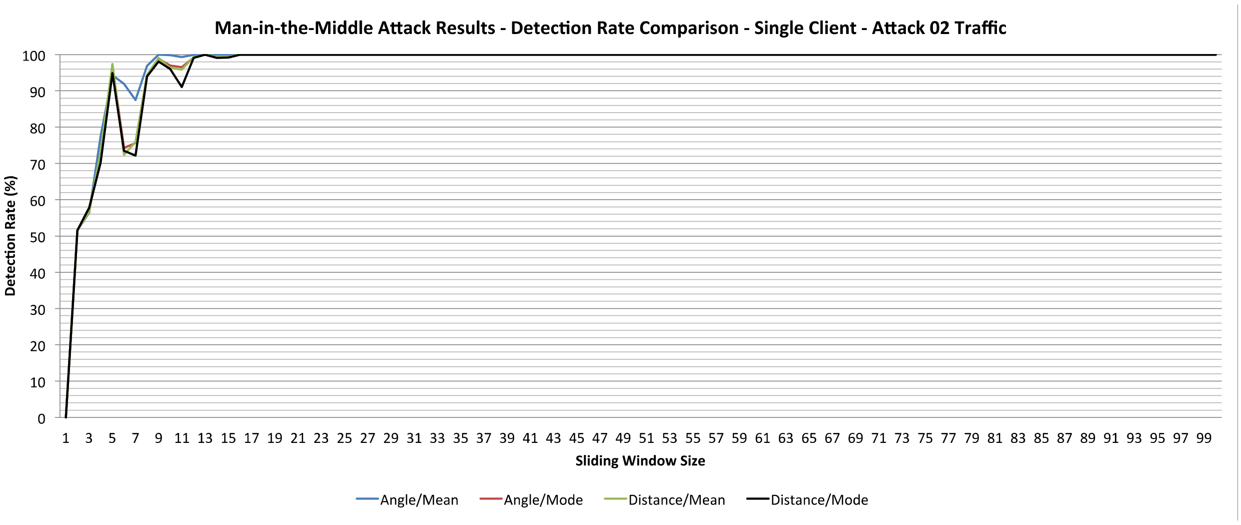


Figure 7.34 Airpwn Attack - $Attack02$ Dataset - 6 metrics - $DR$ Results Comparison.

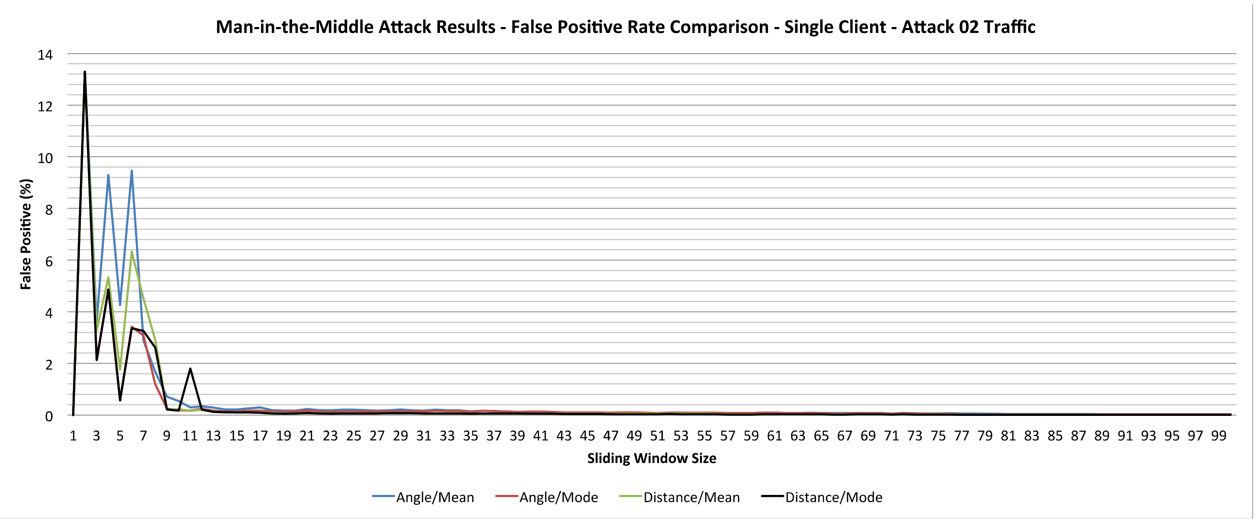


Figure 7.35 Airpwn Attack - $Attack02$ Dataset - 6 metrics - $FP_{Rate}$ Results Comparison.

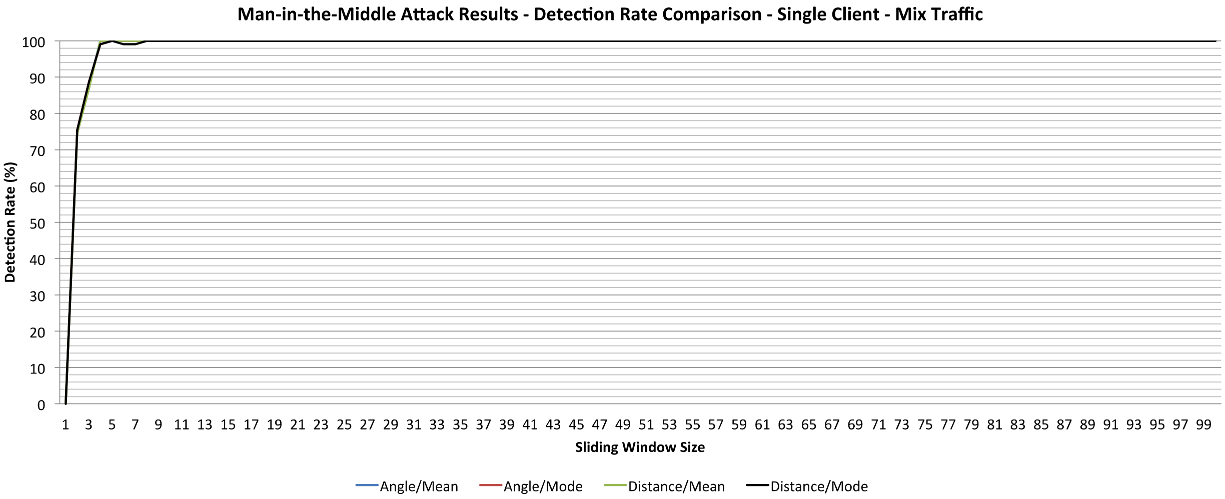


Figure 7.36 Airpwn Attack - $Mixed\ Attack$ Dataset - 6 metrics - $DR$ Results Comparison.

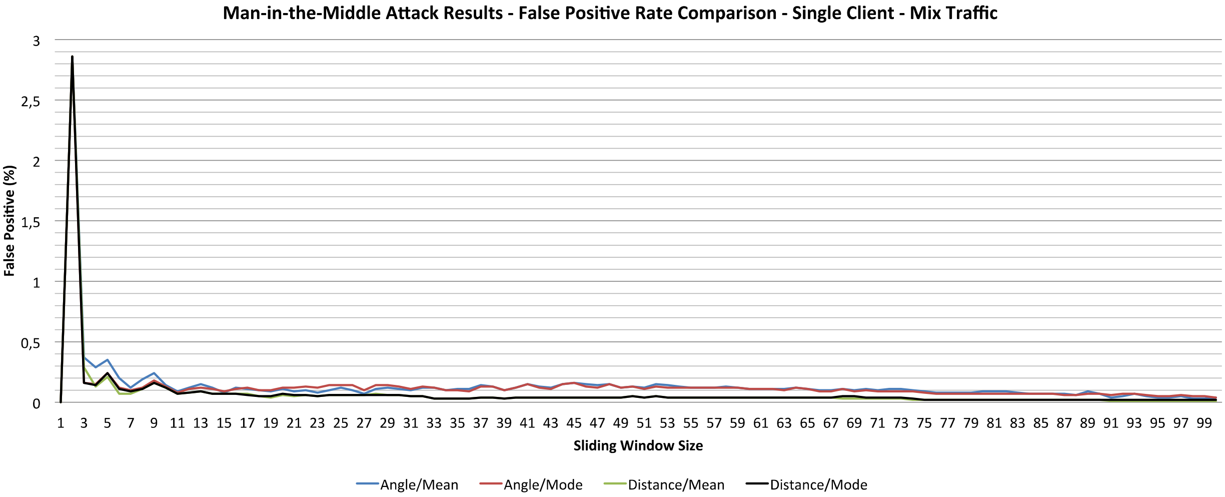


Figure 7.37 Airpwn Attack - $Mixed\ Attack$ Dataset - 6 metrics - $FP_{Rate}$ Results Comparison.

Figure 7.38 shows the average processing time required by the detection system to provide a final decision about the nature of each analysed frame, for the Normal experiment dataset. The sliding window length $n = 31$ has been selected as the smallest value that produces the best results overall for any of the methodology configurations that use the Euclidean distance. On the other hand, the length value of the sliding window $n = 54$ is the smallest length value that produces the best results overall for the methodology configurations that use the angle. For $n = 31$, the average processing time for the normal dataset is between 22μsec and 30μsec. Similarly, if

$n = 54$ was selected, the average processing time is between 52μsec and 72μsec. Since the average interarrival time between two consecutive frames is 6.11msec, the intrusion detection process can fairly be implemented in real time. Figure 7.39 shows the average processing time required by the detection system to provide a final decision about the nature of each analysed frame, for the $Attack01$ dataset. For $n = 31$, the average processing time for the normal dataset is between 20μsec and 39μsec. Similarly, for $n = 54$ the average processing time is between 48μsec and 59μsec. The average interarrival time between two consecutive frames is 10.1msec.

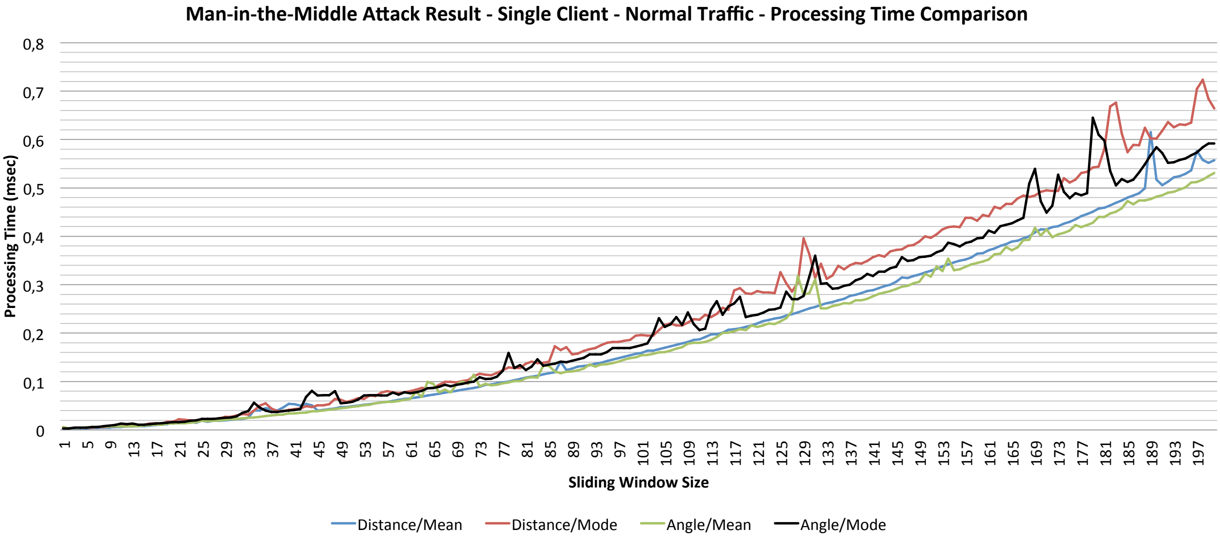


Figure 7.38 Airpwn $Normal$ - Per Frame Detection Analysis Processing Time.

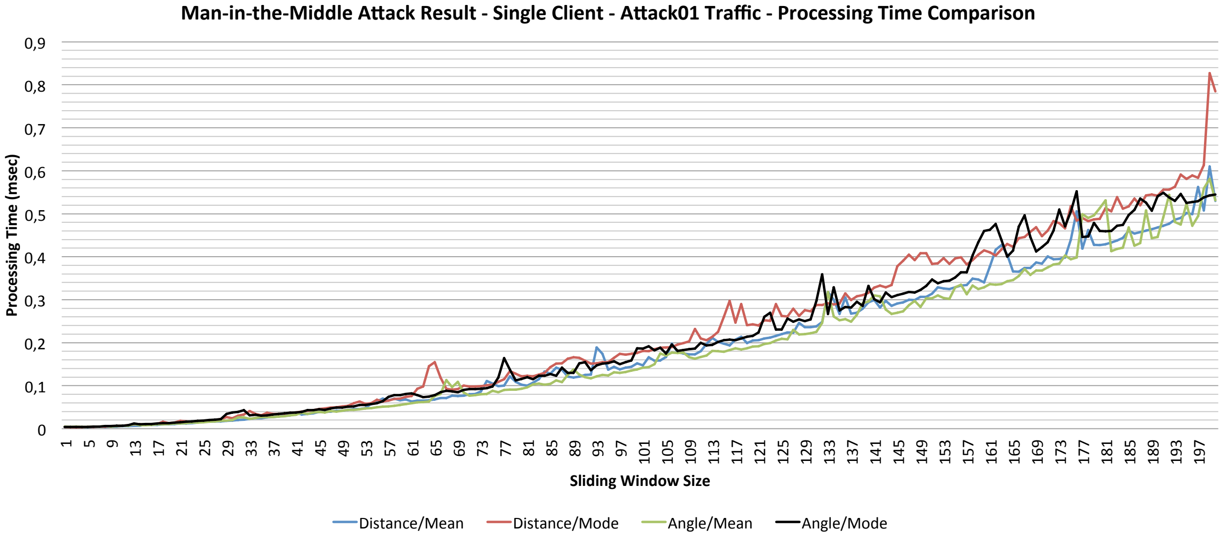


Figure 7.39 Airpwn $Attack01$ - Per Frame Detection Analysis Processing Time.

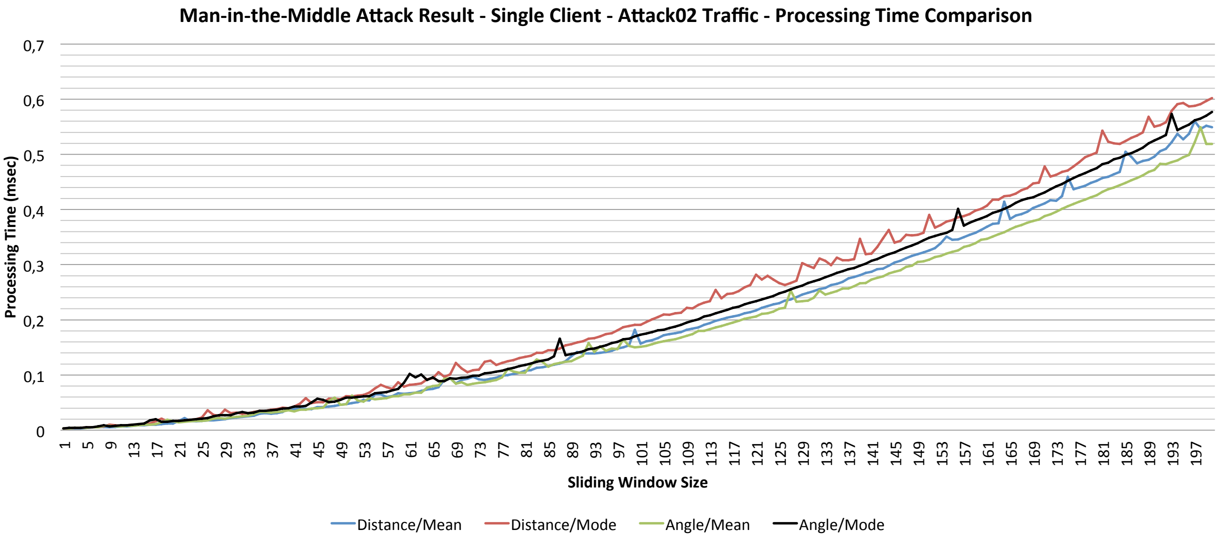


Figure 7.40 Airpwn $Attack02$ - Per Frame Detection Analysis Processing Time.

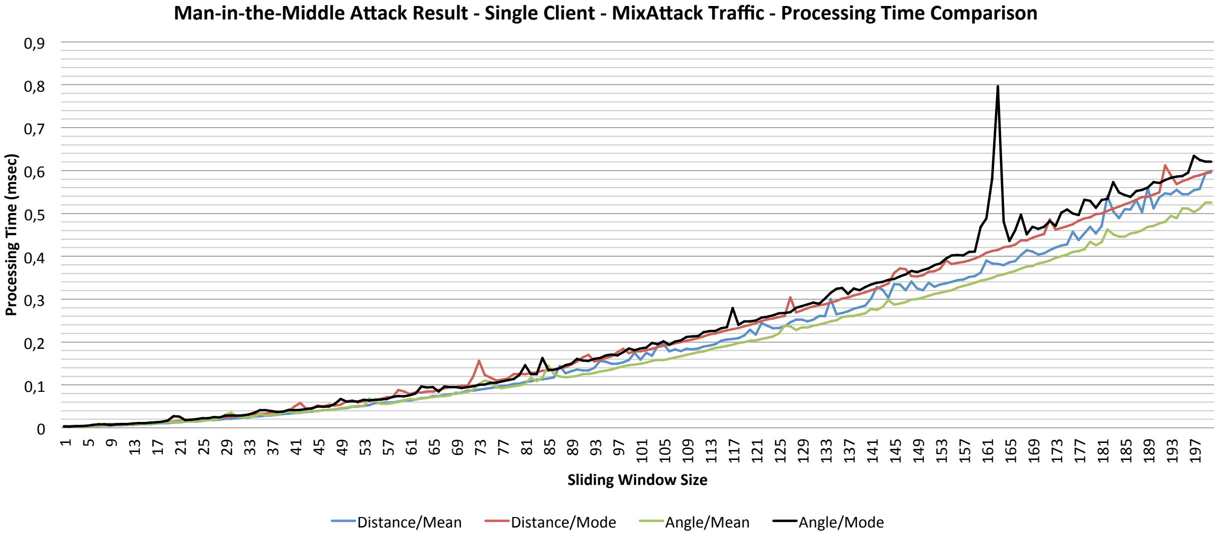


Figure 7.41 Airpwn $Mixed\ Attack$ - Per Frame Detection Processing Time.

Figure 7.40 shows the average processing time for the $Attack02$ dataset. For $n = 31$, the average processing time for the normal dataset is between 23µsec and 32µsec. Similarly, for $n = 54$ the average processing time is between 54µsec and 68µsec. The average interarrival time between two consecutive frames for this dataset is 2.37msec. Figure 7.41 shows the average processing time for the $Mixed\ Attack$ dataset. For $n = 31$, the average processing time for the normal dataset is between 22µsec and 28µsec. Similarly, for $n = 54$ the average processing time is between 53µsec and 68µsec. The average interarrival time between two consecutive frames for this dataset is 2.18msec. The processing time results show that the intrusion detection process can be implemented in real time for all the Airpwn experiment datasets for the selected sliding window lengths.

As a method to compare the four different assessed configurations, the following tables show all evaluation parameters described in Section 7.1, for each of the four configurations when the length value of the sliding window is $n = 31$ and $n = 54$. Table VII.VI corresponds to the Normal dataset; Tables VII.VII and VII.VIII correspond to the datasets $Attack01$ and $Attack02$, respectively; and Table VII.IX corresponds to the $Mixed\ Attack$ dataset.

In order to verify whether the combination six metrics outperforms the combination of fewer numbers of metrics, Figures 7.42 - 7.57 represent the $FP_{Rate}$ results of any possible combination of five metrics for the different datasets when the system uses the methodology configurations distance-mean, distance-mode, angle-mean and angle-mode, respectively. The detection results in terms of $DR$ are not shown but these are always 100% for any case when $16 \leq n$, with the only exception of $TTL - \Delta Time - INJ_{Rate} - NAV - SEQ_{Dif}$, which generate worse $DR$ than the 6 metrics combinations. In terms of $FP_{Rate}$, the four metric combinations $RSSI - \Delta Time - INJ_{Rate} - NAV - SEQ_{Dif}$, $RSSI - TTL - \Delta Time - INJ_{Rate} - NAV$, $RSSI - TTL - INJ_{Rate} - NAV - SEQ_{Dif}$ and $TTL - \Delta Time - INJ_{Rate} - NAV - SEQ_{Dif}$ outperform the intrusion detection results of the 6 metrics combinations, for $53 \geq n$. For $54 \leq n$, the combination of 6 metrics produce the best results overall for all the datasets. From all these results, the methodology configuration distance-mean is the one that produces the best results.

TABLE VII.VI. Results - 6 Metrics Normal Traffic - 31 // 54 slots.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time$ (μsec |
|---|---|---|---|---|---|---|---|
| Distance//Mean | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 22 |
| Distance//Mode | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 30 |
| Angle//Mean | $n/a$ | 0.03 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 23 |
| Angle//Mode | $n/a$ | 0.06 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 26 |
| Distance//Mean | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 53 |
| Distance//Mode | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 71 |
| Angle /Mean | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 52 |
| Angle//Mode | $n/a$ | 0 | $n/a$ | $n/a$ | $n/a$ | $n/a$ | 72 |

TABLE VII.VII. Results - 6 Metrics Attack01 Traffic - 31 // 54 slots.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time$ (μsec |
|---|---|---|---|---|---|---|---|
| Distance//Mean | 100 | 0 | 0 | 1 | 1 | 1 | 20 |
| Distance//Mode | 100 | 0 | 0 | 1 | 1 | 1 | 30 |
| Angle//Mean | 100 | 0 | 0 | 1 | 1 | 1 | 26 |
| Angle//Mode | 100 | 0 | 0 | 1 | 1 | 1 | 39 |
| Distance//Mean | 100 | 0 | 0 | 1 | 1 | 1 | 58 |
| Distance//Mode | 100 | 0 | 0 | 1 | 1 | 1 | 59 |
| Angle /Mean | 100 | 0 | 0 | 1 | 1 | 1 | 48 |
| Angle//Mode | 100 | 0 | 0 | 1 | 1 | 1 | 57 |

TABLE VII.VIII. RESULTS - 6 METRICS ATTACK02 TRAFFIC - 31 // 54 SLOTS.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time\ (\mu sec$ |
|---|---|---|---|---|---|---|---|
| Distance//Mean | 100 | 0.06 | 0 | 1 | 0.992 | 1 | 21 |
| Distance//Mode | 100 | 0.06 | 0 | 1 | 0.992 | 1 | 32 |
| Angle//Mean | 100 | 0.17 | 0 | 1 | 0.975 | 0.99 | 24 |
| Angle//Mode | 100 | 0.14 | 0 | 1 | 0.98 | 0.99 | 31 |
| Distance//Mean | 100 | 0.05 | 0 | 1 | 0.993 | 1 | 54 |
| Distance//Mode | 100 | 0.03 | 0 | 1 | 0.995 | 1 | 68 |
| Angle /Mean | 100 | 0.08 | 0 | 1 | 0.988 | 0.99 | 58 |
| Angle//Mode | 100 | 0.09 | 0 | 1 | 0.987 | 0.99 | 62 |

TABLE VII.IX. RESULTS - 6 METRICS MIXED ATTACK TRAFFIC - 31 // 54 SLOTS.

| Configuration | $DR$ | $FP_{Rate}$ | $FN_{Rate}$ | $OSR$ | $Precision$ | $FScore$ | $Time\ (\mu sec$ |
|---|---|---|---|---|---|---|---|
| Distance//Mean | 100 | 0.05 | 0 | 1 | 0.95 | 0.97 | 22 |
| Distance//Mode | 100 | 0.05 | 0 | 1 | 0.95 | 0.97 | 27 |
| Angle//Mean | 100 | 0.1 | 0 | 1 | 0.904 | 0.95 | 27 |
| Angle//Mode | 100 | 0.11 | 0 | 1 | 0.897 | 0.95 | 28 |
| Distance//Mean | 100 | 0.04 | 0 | 1 | 0.957 | 0.98 | 53 |
| Distance//Mode | 100 | 0.04 | 0 | 1 | 0.957 | 0.98 | 62 |
| Angle /Mean | 100 | 0.13 | 0 | 1 | 0.876 | 0.93 | 68 |
| Angle//Mode | 100 | 0.12 | 0 | 1 | 0.883 | 0.94 | 64 |

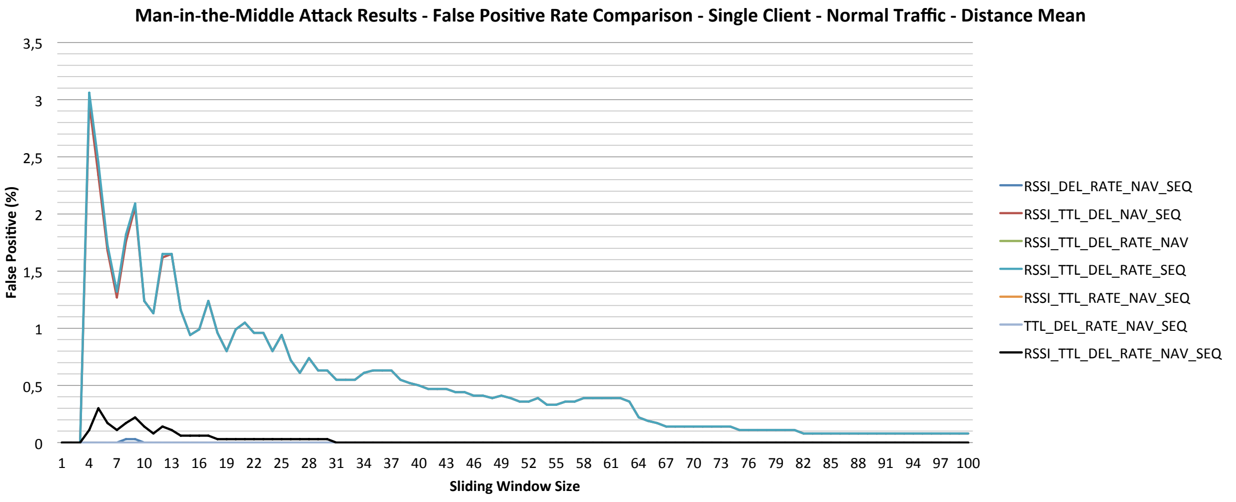


Figure 7.42 Airpwn *Normal* - 5 Metrics - Distance & Mean.

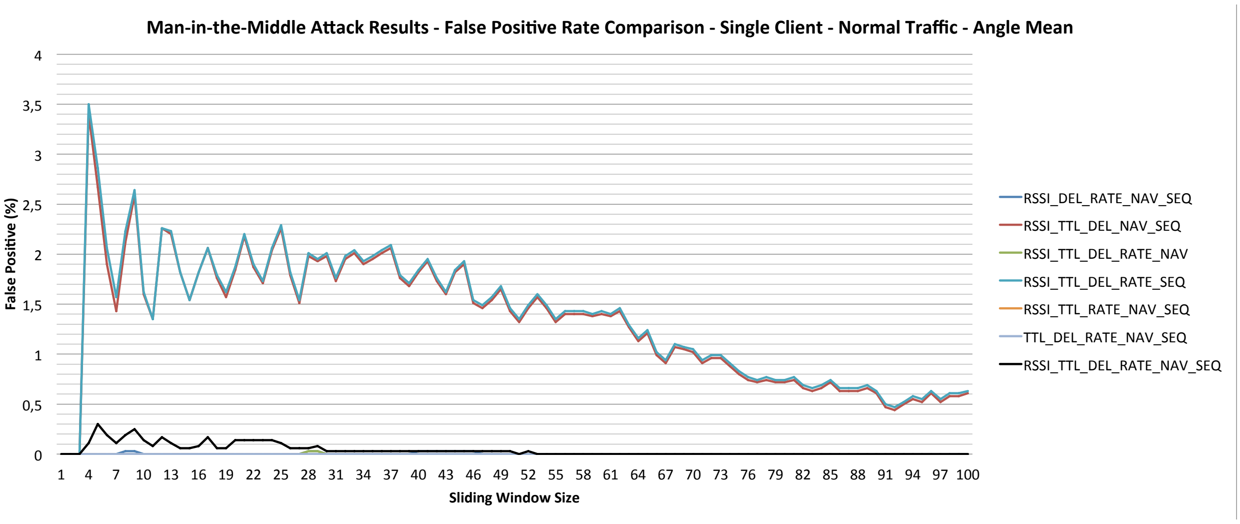


Figure 7.43 Airpwn *Normal* - 5 Metrics - Angle & Mean.

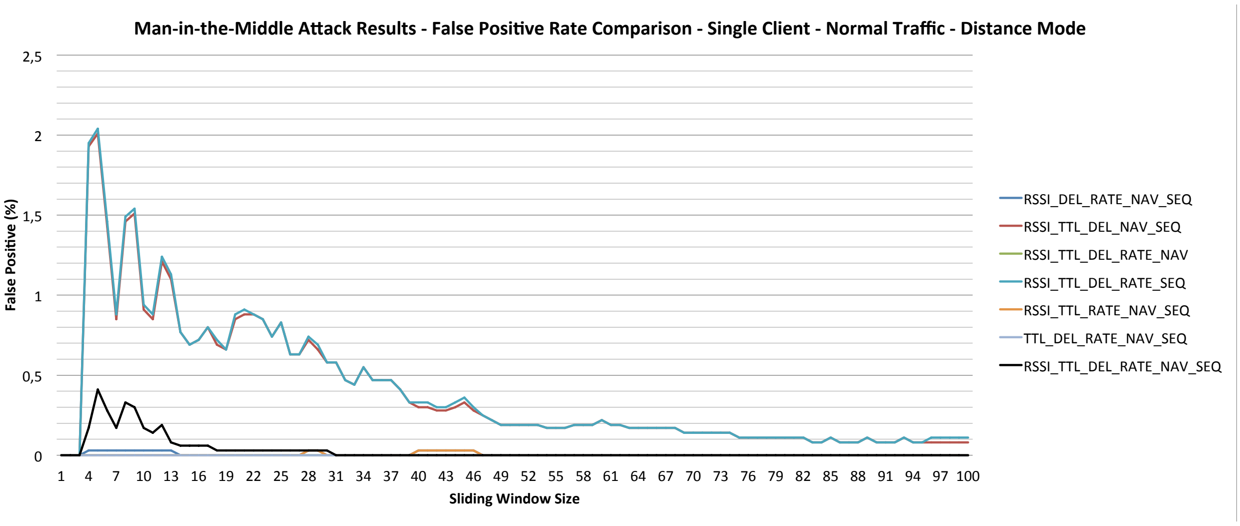


Figure 7.44 Airpwn *Normal* - 5 Metrics - Distance & Mode.

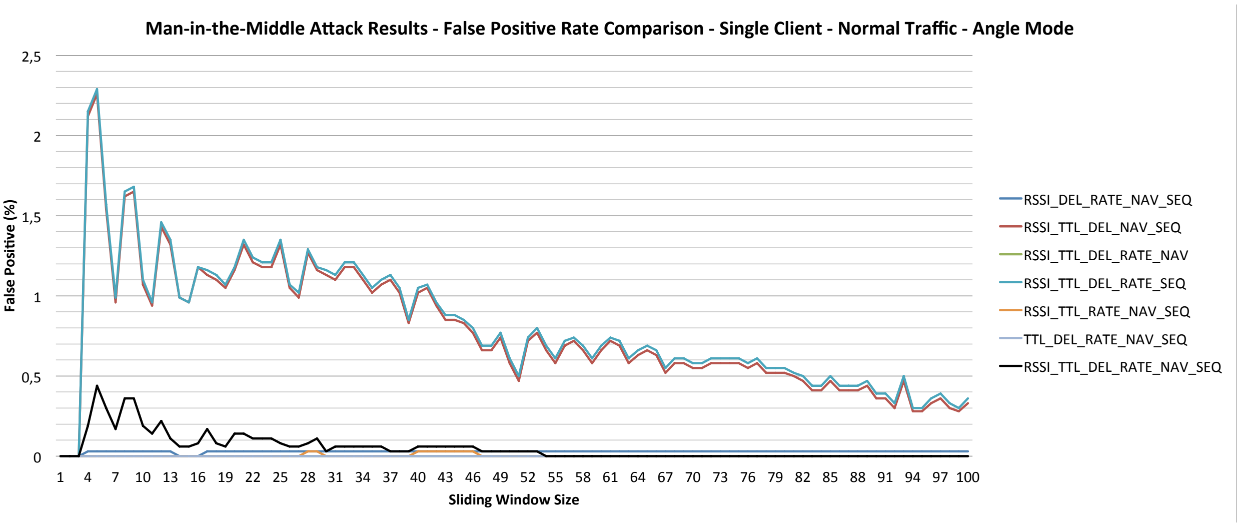


Figure 7.45 Airpwn *Normal* - 5 Metrics - Angle & Mode.

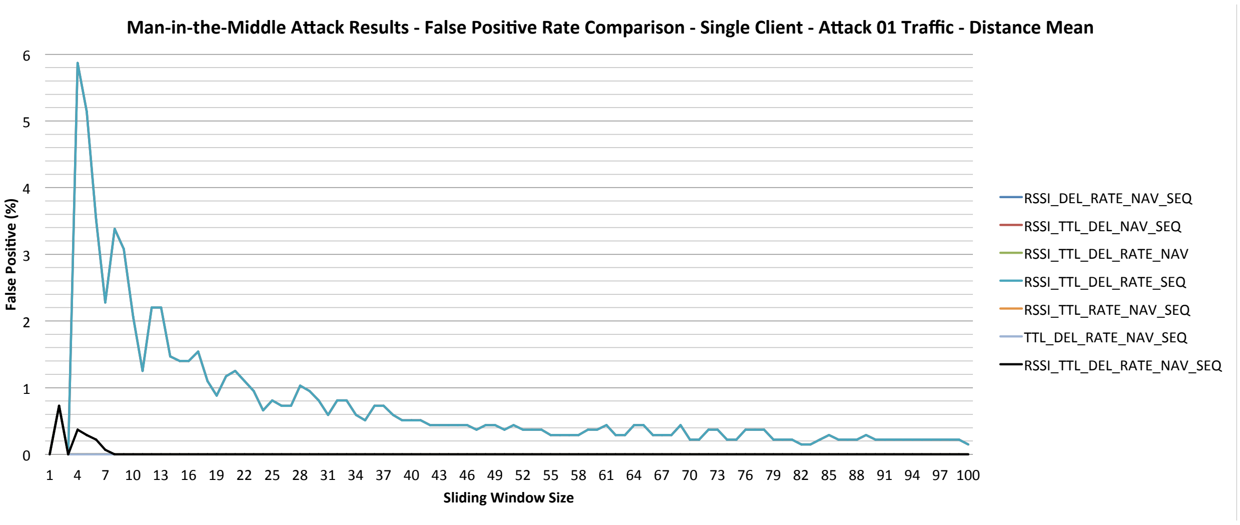


Figure 7.46 Airpwn $Attack01$ - 5 Metrics - Distance & Mean.

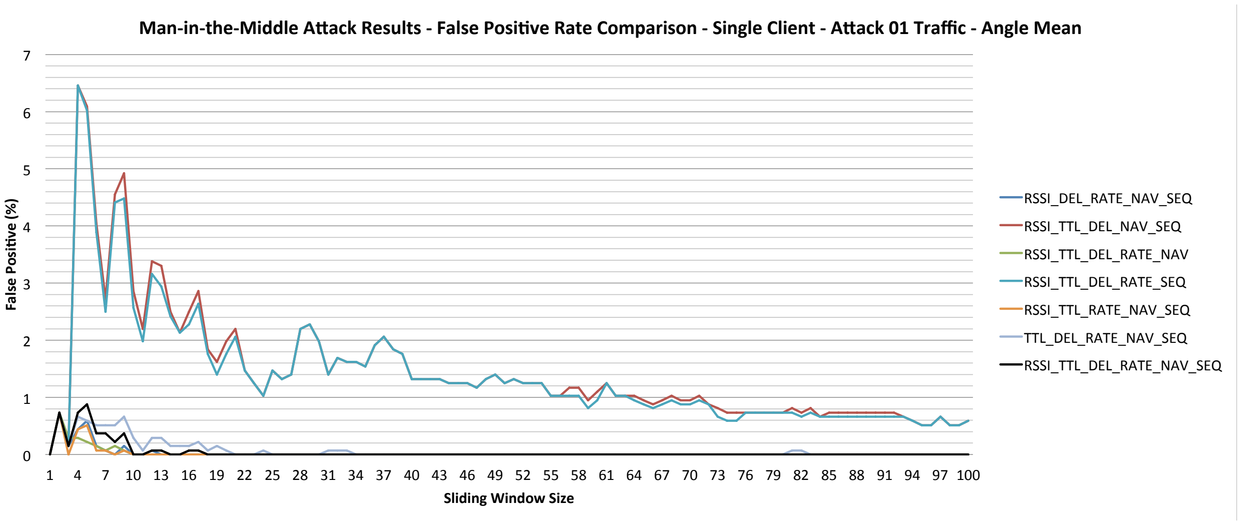


Figure 7.47 Airpwn $Attack01$ - 5 Metrics - Angle & Mean.

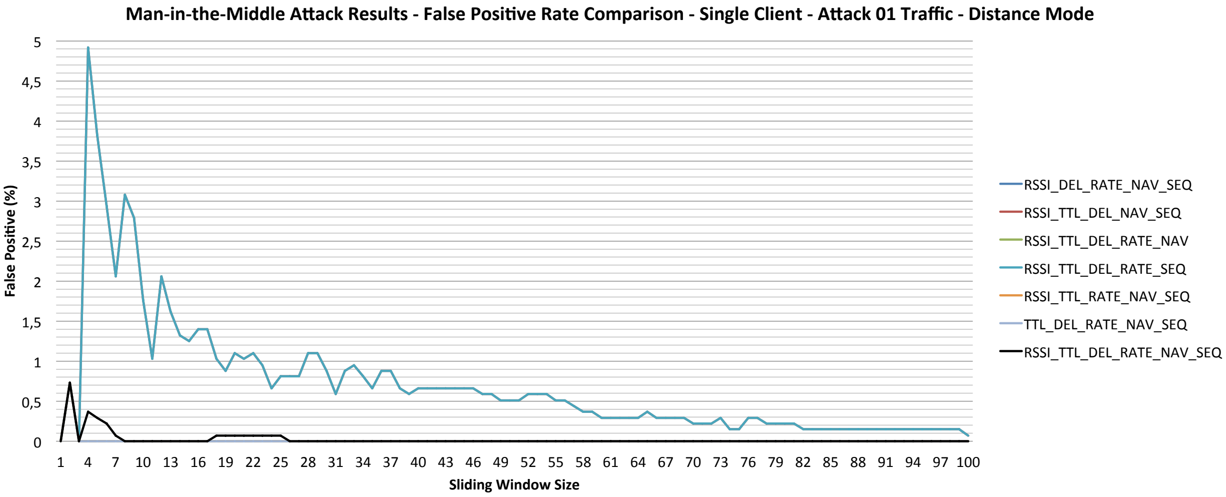


Figure 7.48 Airpwn $Attack01$ - 5 Metrics - Distance & Mode.

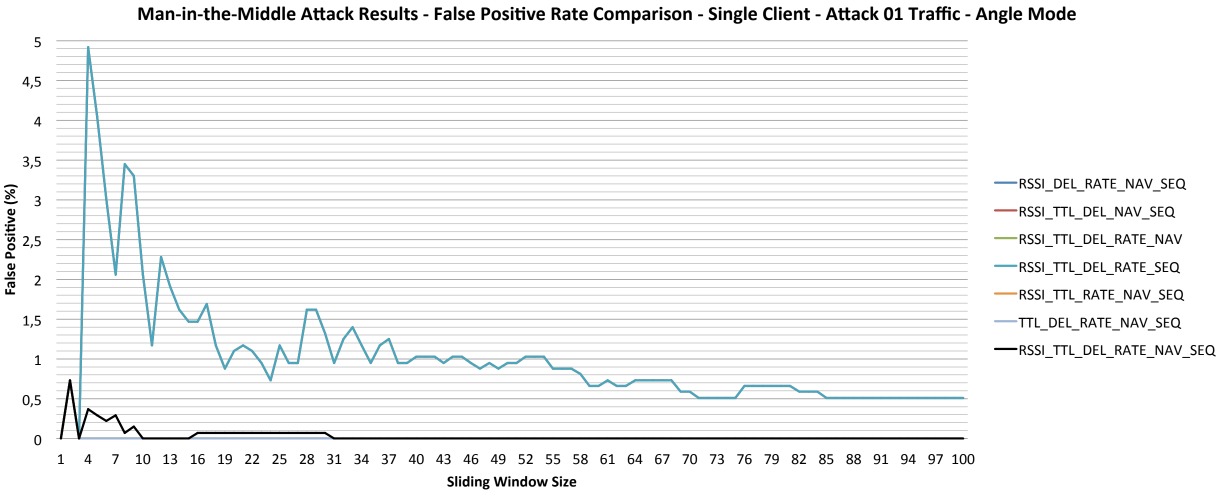


Figure 7.49 Airpwn $Attack01$ - 5 Metrics - Angle & Mode.

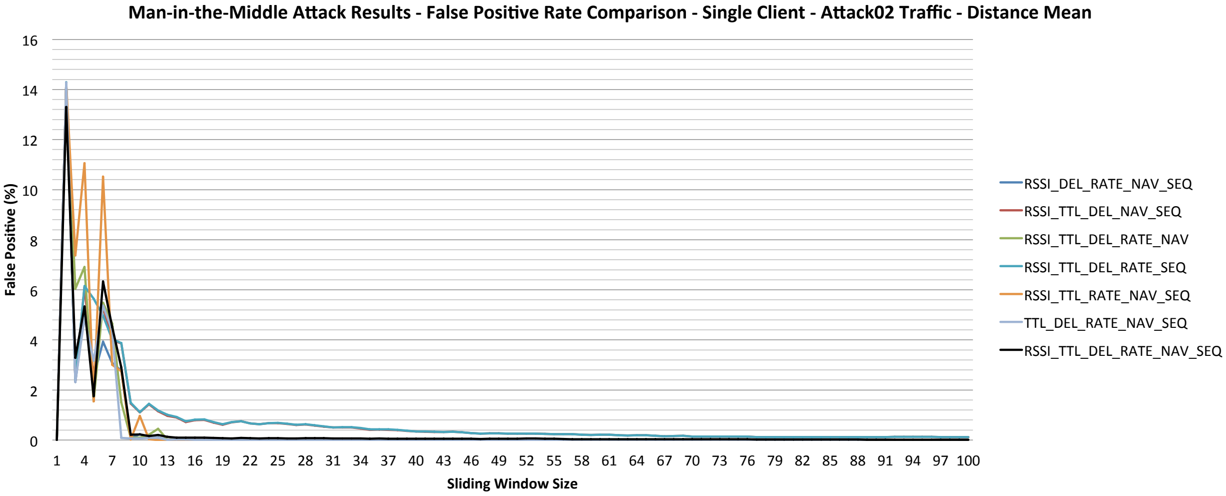


Figure 7.50 Airpwn $Attack02$ - 5 Metrics - Distance & Mean.

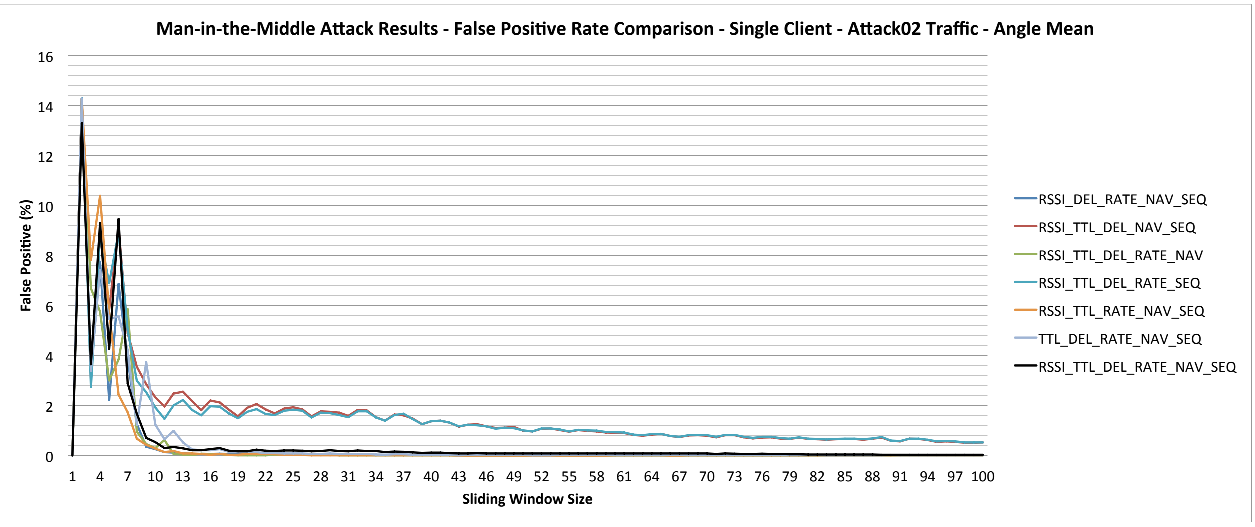


Figure 7.51 Airpwn $Attack02$ - 5 Metrics - Angle & Mean.

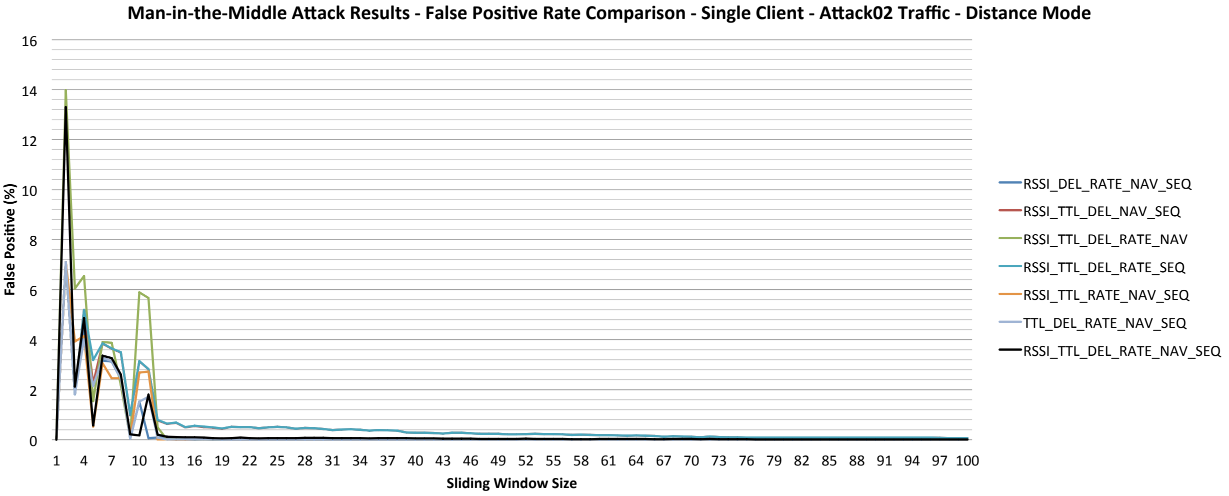


Figure 7.52 Airpwn $Attack02$ - 5 Metrics - Distance & Mode.

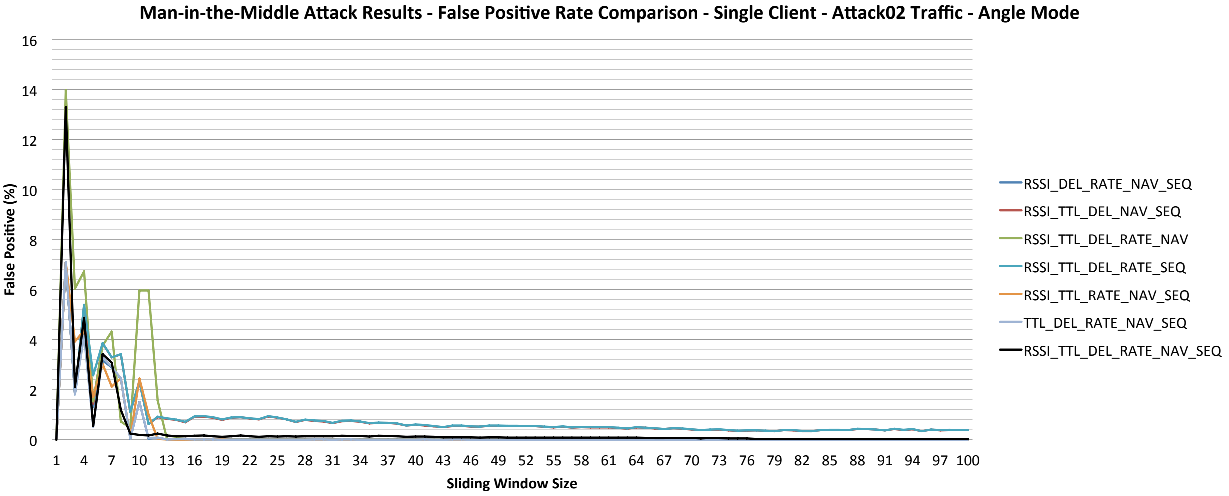


Figure 7.53 Airpwn $Attack02$ - 5 Metrics - Angle & Mode.

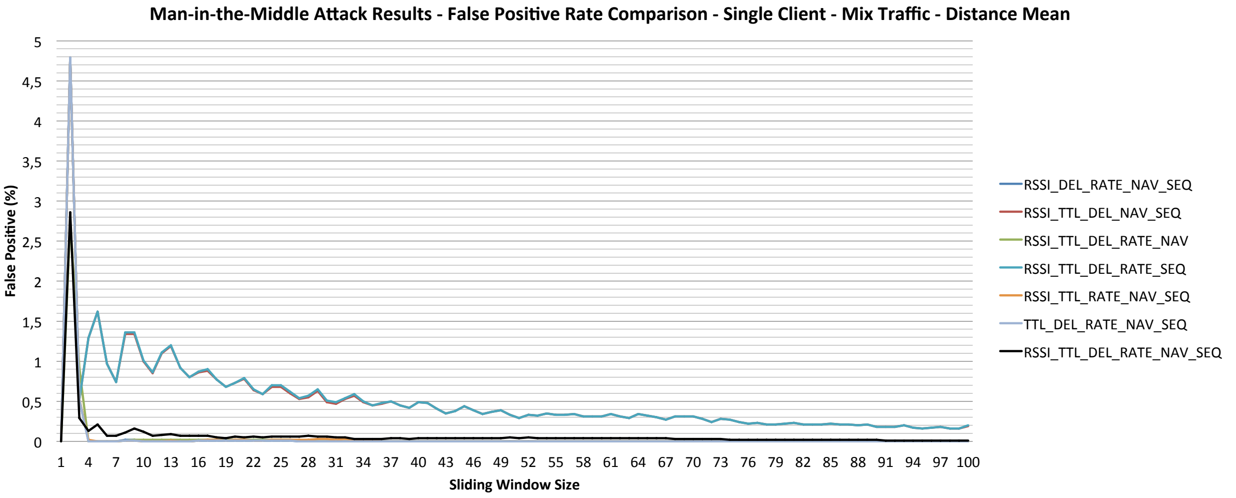


Figure 7.54 Airpwn *Mixed Attack* - 5 Metrics - Distance & Mean.

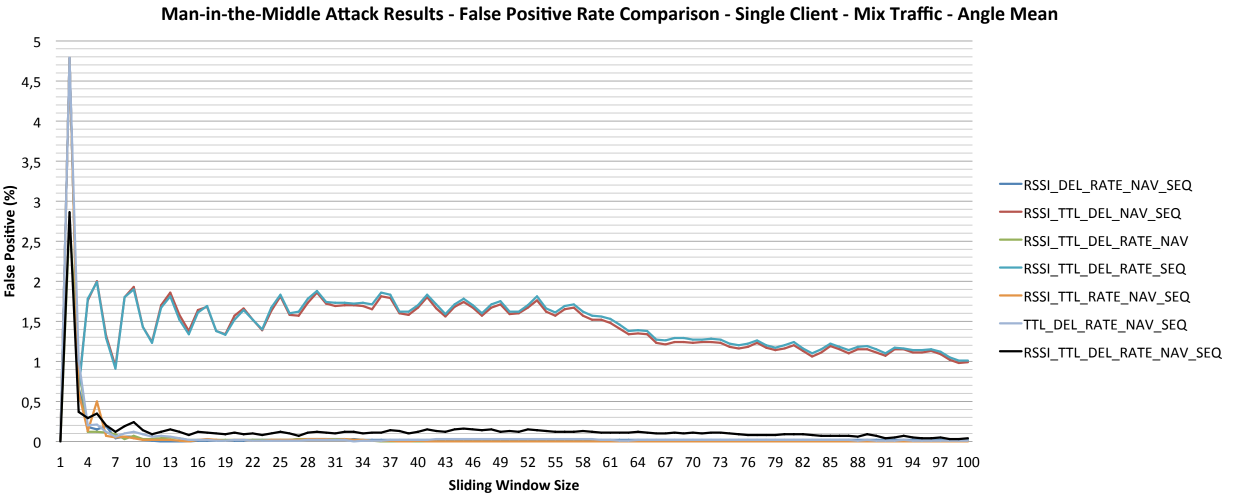


Figure 7.55 Airpwn *Mixed Attack* - 5 Metrics - Angle & Mean.

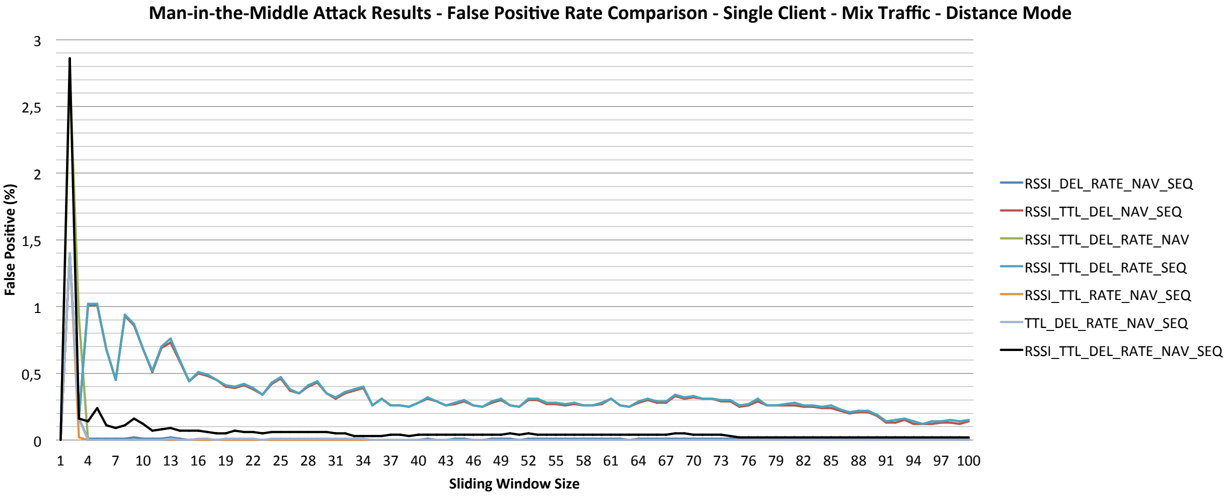


Figure 7.56 Airpwn *Mixed Attack* - 5 Metrics - Distance & Mode.

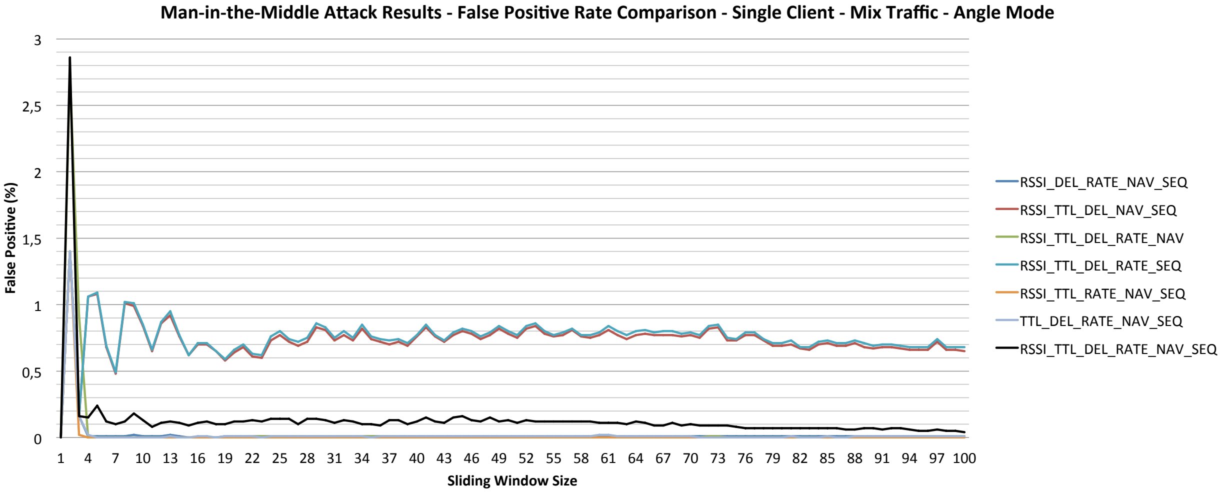


Figure 7.57 Airpwn *Mixed Attack* - 5 Metrics - Angle & Mode.

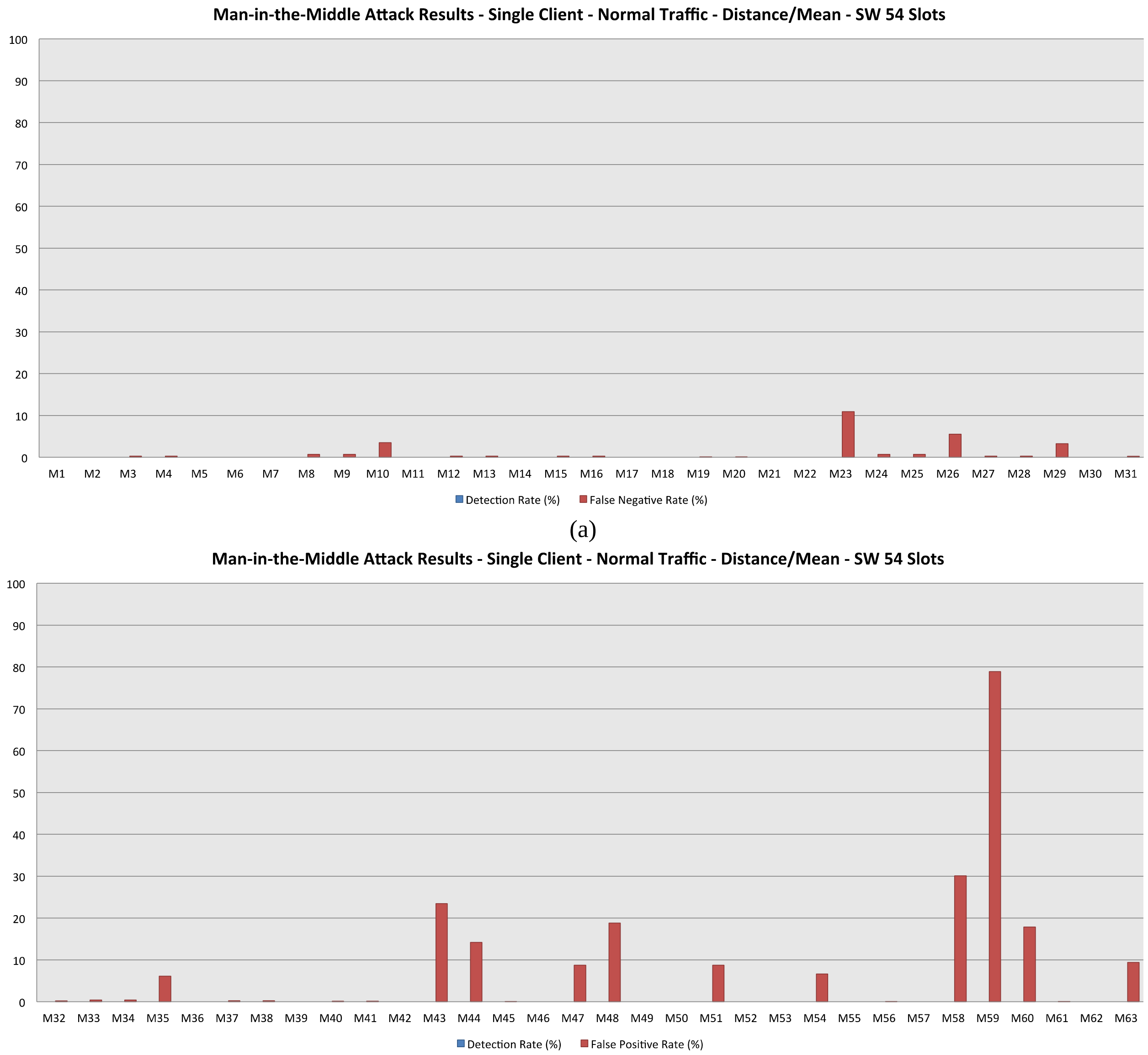
Since the best intrusion detection results from the Airpwn attack datasets are generated using the configuration distance-mean, Figures 7.58 – 7.61 present the $DR$ and $FP_{Rate}$ results for the Airpwn attack experiments using only this particular configuration, when the sliding window length is $n = 54$. The different metrics combinations have been categorised in the indexes represented in the Table VII.X. The index M1 referrers to the set that combines all the considered metrics, and the index M63 referrers to a single metric set. Therefore, the best results are to be expected from the test index M1. These figures help to understand the effect that increasing the number of metrics used to implement the intrusion detection. The Y-axis of the graphs represents the percentage of $DR$ and $FP_{Rate}$. The X-axis of the graphs represents the metric combination indexes shown in Table VII.X.

TABLE VII.X. INDEXES OF THE USED METRICS IN AIRPWN ATTACK.

| Index-Metrics | Index-Metrics | Index-Metrics | Index-Metrics |
|---|---|---|---|
| M1 - $RSSI$ - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M17 - $RSSI$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M33 - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ | M49 - $TTL$ - $INJ_{Rate}$ |
| M2 - $RSSI$ - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | M18 - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | M34 - $TTL$ - $\Delta Time$ - $NAV$ | M50 - $TTL$ - $NAV$ |
| M3 - $RSSI$ - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M19 - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M35 - $TTL$ - $\Delta Time$ - $SEQ_{Dif}$ | M51 - $TTL$ - $SEQ_{Dif}$ |
| M4 - $RSSI$ - $TTL$ - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | M20 - $TTL$ - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | M36 - $TTL$ - $INJ_{Rate}$ - $NAV$ | M52 - $\Delta Time$ - $INJ_{Rate}$ |
| M5 - $RSSI$ - $TTL$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M21 - $TTL$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M37 - $TTL$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M53 - $\Delta Time$ - $NAV$ |
| M6 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M22 - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M38 - $TTL$ - $NAV$ - $SEQ_{Dif}$ | M54 - $\Delta Time$ - $SEQ_{Dif}$ |
| M7 - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M23 - $RSSI$ - $TTL$ - $\Delta Time$ | M39 - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | M55 - $INJ_{Rate}$ - $NAV$ |

| | | | |
|---|---|---|---|
| M8 - $RSSI$ - $TTL$ - $\Delta Time$ - $INJ_{Rate}$ | M24 - $RSSI$ - $TTL$ - $INJ_{Rate}$ | M40 - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M56 - $INJ_{Rate}$ - $SEQ_{Dif}$ |
| M9 - $RSSI$ - $TTL$ - $\Delta Time$ - $NAV$ | M25 - $RSSI$ - $TTL$ - $NAV$ | M41 - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | M57 - $NAV$ - $SEQ_{Dif}$ |
| M10 - $RSSI$ - $TTL$ - $\Delta Time$ - $SEQ_{Dif}$ | M26 - $RSSI$ - $TTL$ - $SEQ_{Dif}$ | M42 - $INJ_{Rate}$ - $NAV$ - $SEQ_{Dif}$ | M58 - $RSSI$ |
| M11 - $RSSI$ - $TTL$ - $INJ_{Rate}$ - $NAV$ | M27 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ | M43 - $RSSI$ - $TTL$ | M59 - $TTL$ |
| M12 - $RSSI$ - $TTL$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M28 - $RSSI$ - $\Delta Time$ - $NAV$ | M44 - $RSSI$ - $\Delta Time$ | M60 - $\Delta Time$ |
| M13 - $RSSI$ - $TTL$ - $NAV$ - $SEQ_{Dif}$ | M29 - $RSSI$ - $\Delta Time$ - $SEQ_{Dif}$ | M45 - $RSSI$ - $INJ_{Rate}$ | M61 - $INJ_{Rate}$ |
| M14 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $NAV$ | M30 - $RSSI$ - $INJ_{Rate}$ - $NAV$ | M46 - $RSSI$ - $NAV$ | M62 - $NAV$ |
| M15 - $RSSI$ - $\Delta Time$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M31 - $RSSI$ - $INJ_{Rate}$ - $SEQ_{Dif}$ | M47 - $RSSI$ - $SEQ_{Dif}$ | M63 - $SEQ_{Dif}$ |
| M16 - $RSSI$ - $\Delta Time$ - $NAV$ - $SEQ_{Dif}$ | M32 - $RSSI$ - $NAV$ - $SEQ_{Dif}$ | M48 - $TTL$ - $\Delta Time$ | |

Figures 7.58.a and 7.58.b present the $DR$ and $FP_{Rate}$ results for the configuration distance-mean for the Normal traffic dataset. Since this dataset only contains non-malicious information, the $DR$ for all the metric combinations is 0%. In terms of $FP_{Rate}$, there are 13 metric combinations that produce $FP_{Rate}$ results higher than 5%, in the case of $n = 54$. Seven out of these 13 metric combinations (M23, M43, M44, M48, M58, M59, and M60) produce $FP_{Rate}$ results higher than 10%. The test index M1, six metrics, produces the best results that could be achieved, 0% $FP_{Rate}$. Single metric (M58) $RSSI$ and (M59) $TTL$, are the two metrics that produce the worst $FP_{Rate}$ results. Apart from (M60) $\Delta Time$, all the metric combinations that overpass 10% $FP_{Rate}$ contain either $RSSI$ or $TTL$. It is also appreciable that the number of $FP_{Rate}$ results is increased when fewer metrics are combined.

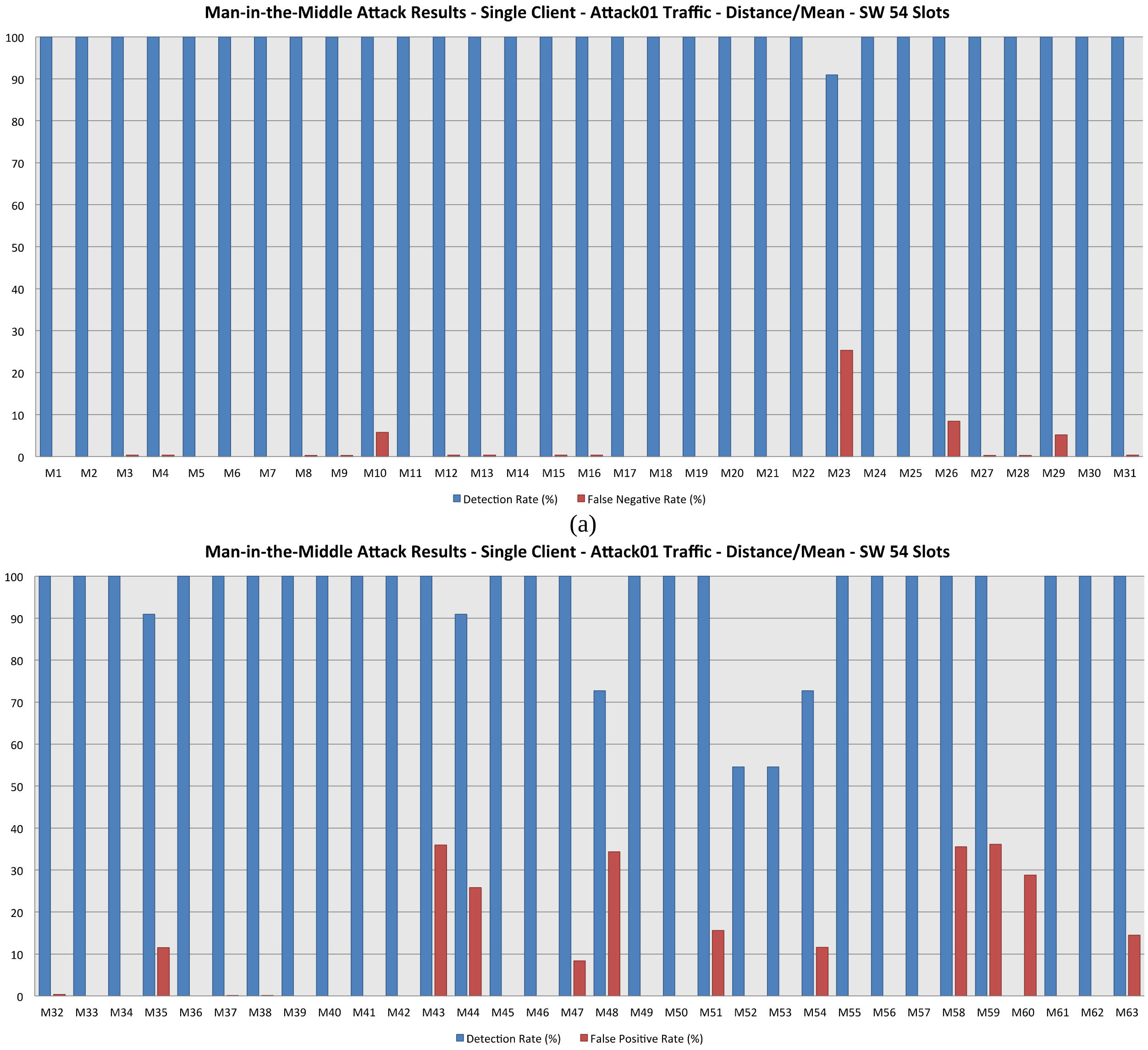


(a)

(b)

Figure 7.58 Airpwn Normal Traffic - 54 Slots - Distance & Mean.

The $DR$ and $FP_{Rate}$ results for the distance-mean configuration and the $Attack01$ dataset, when the sliding window length is $n = 54$, are presented in Figures 7.59.a and 7.59.b, respectively. In terms of $DR$, all of the metric combinations using four, five and six metric generate 100% $DR$. The particular single metric that substantially reduces the $DR$ results is $\varDelta Time$. Single metric (M60) $\varDelta Time$ is the only metric that produces 0% $DR$ results. Additionally, the only metric that is in all the metric combinations that do not produce 100% $DR$ results is $\varDelta Time$. These are M23, M35, M44, M47, M52, M53, M54, and M60. In terms of $FP_{Rate}$, there are 15 metric combinations that produce $FP_{Rate}$ results higher than 5%, in the case of $n = 54$. Seven out of these 15 metric combinations (M23, M43, M44, M48, M58, M59, and M60) produce $FP_{Rate}$ results higher than 25%, reaching single metric (M59) $TTL$ up to 36.13% $FP_{Rate}$. Using this dataset, the test index M1, six metrics, once more produces the best results that could be achieved, 0% $FP_{Rate}$. Similar to the previous results using the Normal traffic dataset, at least one of the metrics $\Delta Time$, $RSSI$ or $TTL$, is in the metric combinations that generate that overpass 25% $FP_{Rate}$.



(a)

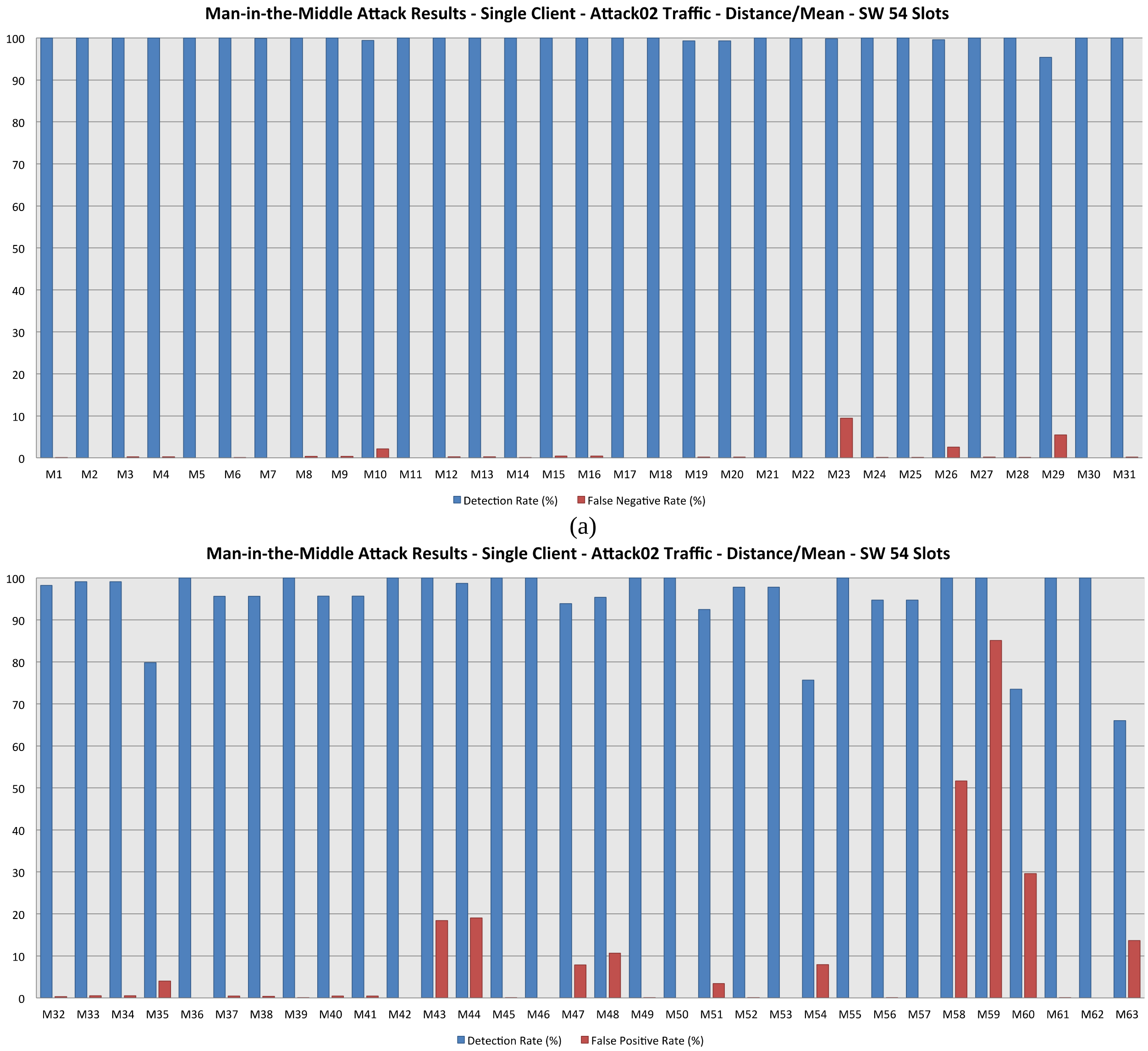
(b)

Figure 7.59 Airpwn $Attack01$ Traffic - 54 Slots - Distance & Mean.

Figures 7.60.a and 7.60.b present the $DR$ and $FP_{Rate}$ results for the configuration distance-mean for the Airpwn $Attack02$ traffic dataset and $n = 54$. In terms of $DR$, a lower number of metric combinations produce 100% $DR$ than using the Airpwn $Attack02$ traffic dataset. However, only four metric combinations generate $DR$ results lower than 90%. These are M35, M54, M60, and M63. Either $\Delta Time$ or $SEQ_{Dif}$ is included in these four metric combinations. The test index M1, six metrics, produces 100% $DR$, the best achievable results. Additionally, none of the metric combinations using four, five and six metric generate $DR$ lower than 99%. In terms of $FP_{Rate}$, eleven metric combinations produce $FP_{Rate}$ results higher than 5%. Three of these metric combinations (M58, M59, and M60) produce $FP_{Rate}$ results higher than 25%, reaching single metric (M58) $RSSI$ up to 51.68% $FP_{Rate}$ and (M59) $TTL$ up to 85.11% $FP_{Rate}$. In this case, the test index M1, six metrics, produces 0.05% $FP_{Rate}$. Among the metric combinations using 5 metrics, only M2 and M5 generate completely perfect detection. Once more, it is clear that the combined use of four and above metrics generates better detection results overall than the combining lower number of metrics.

Finally, Figures 7.61.a and 7.61.b present the $DR$ and $FP_{Rate}$ results for the configuration distance-mean for the Airpwn $Mixed\ Attack$ traffic dataset and $n = 54$. In terms of $DR$, five metric combinations generate $DR$ results lower than 90%. These are M48, M51, M54, M60, and M63. Again, either $\Delta Time$ or $SEQ_{Dif}$ is included in these five metric combinations. The combination of six metrics (M1) produces 100% $DR$, the best achievable results. Additionally, none of the metric combinations using four, five and six metric generate $DR$ lower than 90%. In terms of $FP_{Rate}$, 13 metric combinations produce $FP_{Rate}$ results higher than 5%. Five of these metric combinations (M43, M44, M58, M59, and M60) produce $FP_{Rate}$ results higher than 25%, reaching single metric (M58) $RSSI$ up to 37.26% $FP_{Rate}$ and (M59) $TTL$ up to 94.69% $FP_{Rate}$. Using the $Mixed\ Attack$ dataset M1 produces 0.04% $FP_{Rate}$. Once more, it is clear that the combined use of four and above metrics generates better detection results overall than the combining lower number of metrics.

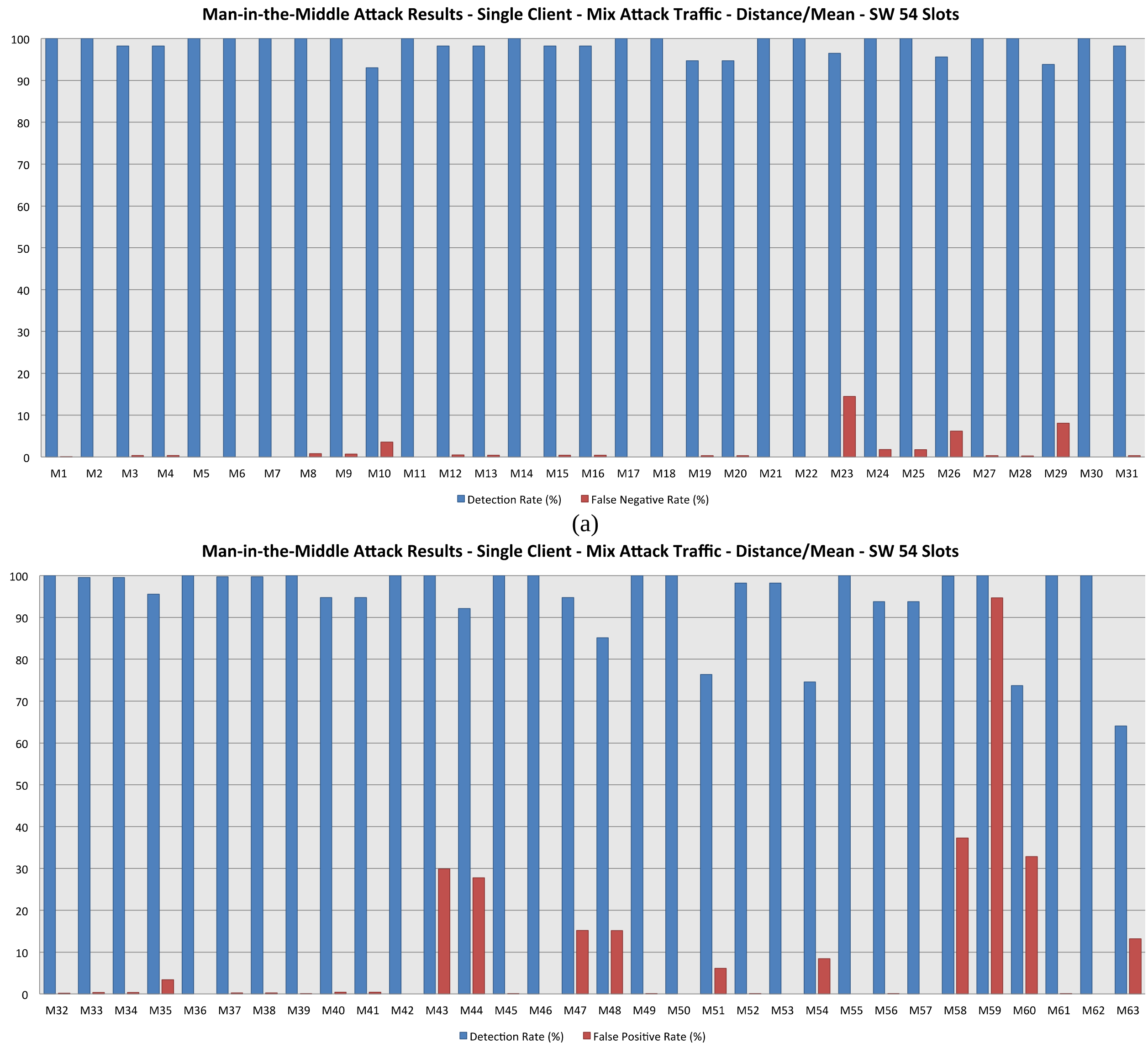
The presented results using all the network traffic dataset while implementing Airpwn attack indicate that the configuration of the methodology that produces the best results overall is the utilisation of the Euclidean distance along with the mean to establish the reference of normality, when the length value of the sliding window is $n \geq 54$. For this $n$ value, the required processing time per frame ranges between 53μsec and 68μsec. The use of the angle methodology to define the belief in $Attack$ generally increases the number of $FP$ alarms. The only benefit of using the angle methodology appears in Airpwn $Attack02$ dataset, for $[7 \leq n \leq 17]$, in which the $DR$ results of the rest of configurations drops.



(a)

(b)

Figure 7.60 Airpwn $Attack02$ Traffic - 54 Slots - Distance & Mean.


Man-in-the-Middle Attack Results - Single Client - Mix Attack Traffic - Distance/Mean - SW 54 Slots

(a)

Man-in-the-Middle Attack Results - Single Client - Mix Attack Traffic - Distance/Mean - SW 54 Slots

(b)

Figure 7.61 Airpwn *Mixed Attack* Traffic - 54 Slots - Distance & Mean.

### 7.3.3 Malicious Frames Within Initial Sliding Window

The first time the proposed detection methodology starts the process of filling the initial sliding window, an attacker could alter the reference of normality injecting malicious frames into the initial sliding window. Nonetheless, even if there exist high proportion of malicious frames within the initial sliding window, the proposed methodology produces good detection results.

In order to calculate the maximum number of malicious frames that could be included into the initial sliding window before the detection performance drops, the detection process has been repeated several times introducing one additional malicious frame for each of the iterations. The results of these experiments are based on the percentage of $DR$ results, the percentage of malicious frames within the initial sliding window, and the sliding window length. The results for these experiments were initially presented in [129]. The deauthentication *Short Distance* dataset and all the Airpwn attack datasets are used in these experiments. For the deauthentication attack, only the four metrics $RSSI - INJ_{Rate} - \Delta Time - NAV$ were used with the angle-mean configuration, whilst the four metrics $RSSI - INJ_{Rate} - TTL - NAV$ were used for Airpwn attack.

Figures 7.62 – 7.64 show the multi-layer results using the Airpwn datasets $Attack01$, $Attack02$ and $Mixed\ Attack$, respectively. There exists an evident consistency in the detection results of all these experiments. For any sliding window length larger than 12 frames, $n \geq 12$, the detection system produces perfect detection with up to 43% of malicious frames within the initial sliding window. A higher percentage of malicious frames causes the detection accuracy to drastically drop. This behaviour is constantly repeated for all the Airpwn attack datasets.

Figure 7.65 shows the multi-layer results using the deauthentication *Short Distance* dataset. The detection system produces perfect detection with up to 20% of malicious frames within the initial sliding window, if the length of the sliding window is between 31 and 90 frames, $[31 \leq n \leq 90]$. If the sliding window length is $n \leq 90$, the system produces 100% $DR$ including up to 13% of malicious frames within the initial sliding window. If the sliding window length is $n \geq 31$, the system is unable to produce higher than 95% $DR$ including any amount of malicious frames within the initial sliding window. In contrast to the Airpwn attacks results, the DR results gradually drop along with the percentage of malicious frames within the initial sliding window.

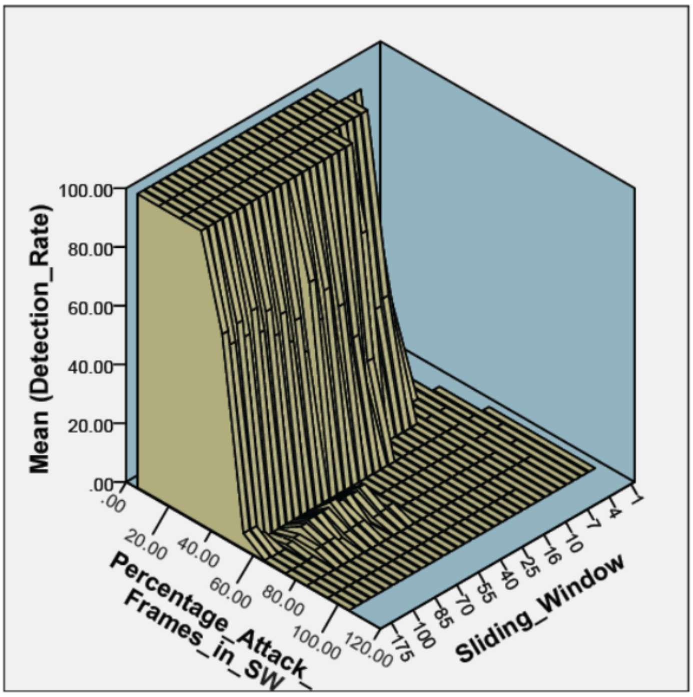


Figure 7.62 Percentage of Malicious Frames in The Initial Sliding Window – Airpwn $Attack01$ Dataset.

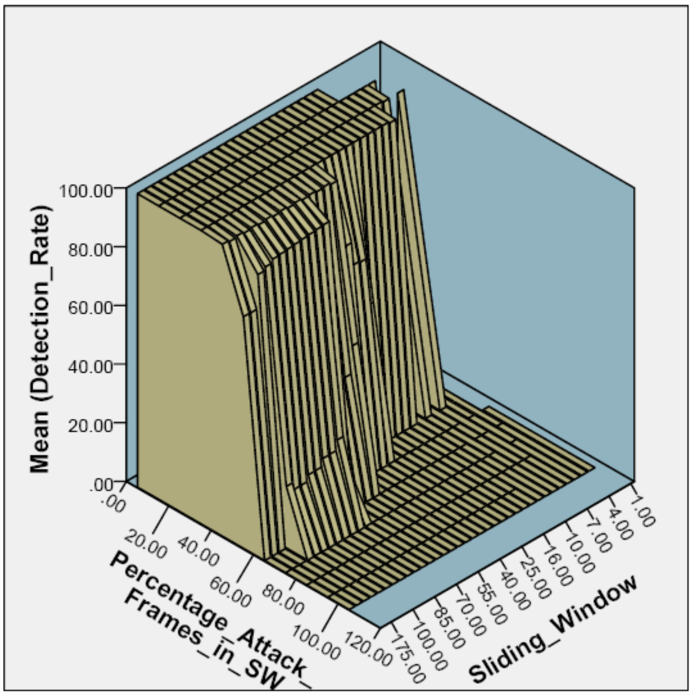


Figure 7.63 Percentage of Malicious Frames in The Initial Sliding Window – Airpwn $Attack02$ Dataset.

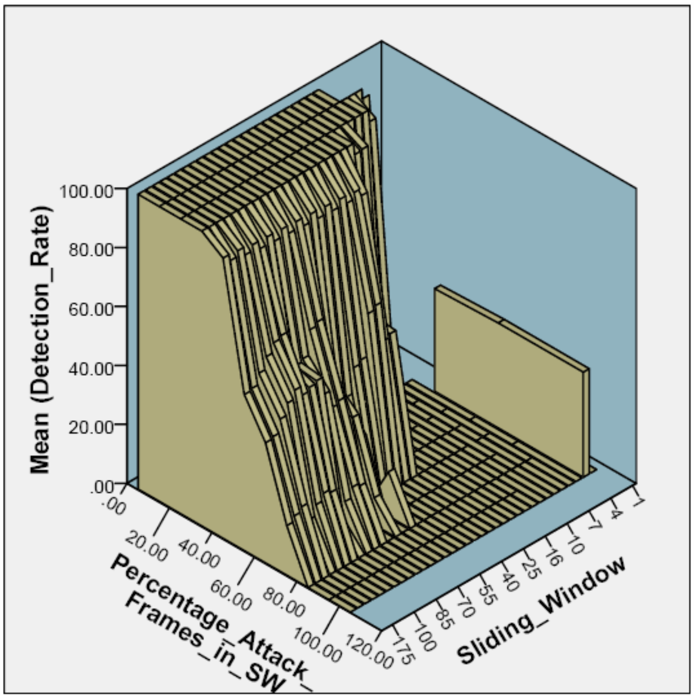


Figure 7.64 Percentage of Malicious Frames in The Initial Sliding Window – Airpwn *Mixed Attack* Dataset.

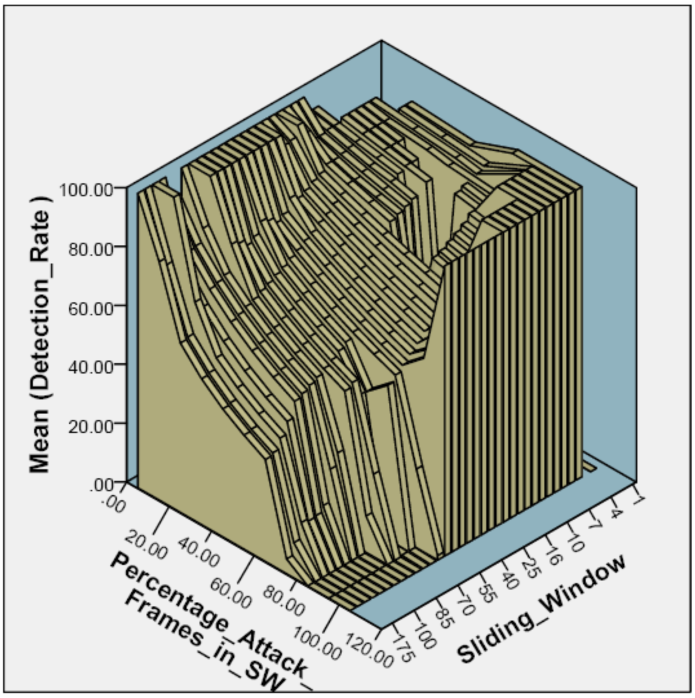


Figure 7.65 Percentage of Malicious Frames in The Initial Sliding Window – Deauthentication Attack – Short Distance Dataset

## 7.4 Summary

This chapter has presented a thorough description of the results that prove effectiveness of the unsupervised anomaly based IDS presented in this thesis. The results have been generated by analysing all the considered datasets using the four proposed IDS configurations. The approach followed to demonstrate the effectiveness of the detection system is to compare the detection results generated using the multi-layer approach against the same methodology utilising different sets of metrics.

A fundamental part of the evaluation experiments has been to assess the performance of the detection methodology using each of the system configurations. It is important to find the methodology that generates the best results. For the *Long Distance* experiments, the combination of metrics that produces the best detection results is $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$. Not including $\Delta Time$ in the detection process improves the final detection results. For $n = 29$, the combined use of all the considered metrics produces $DR$ 100% and $FP_{Rate}$ 0.87%, whereas the $FP_{Rate}$ is constantly reduced to 0.43% after the sliding window reaches $68 \leq n$. For $n = 29$, the use of the 4 metrics produces 100% $DR$ and 0.43% $FP_{Rate}$. However, for $54 \leq n$, the 4 metrics produces $FP_{Rate}$ 0%. For this particular sliding window length, the combined use of the five metrics still produces $DR$ 100% and $FP_{Rate}$ 0.87%. These results have been achieved using the distance-mode configuration, which has proven to outperform the detection results of the other three system configurations for this dataset. These results also show that increasing the length value of the sliding window reduces the number of $FP$ alarms. For both set of metrics and these sliding window lengths, the system produces the detection results in an average time of 20μsec, when $n = 29$, and 76μsec when $n = 68$. Since the average interarrival time between two consecutive frames is 55msec, the intrusion detection can be implemented in real time.

Similar to the *Long Distance* experiments, the metric combination $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$ outperforms the overall results of the combined use of the metrics five for the *Short Distance* dataset. Again, the distance-mode configuration outperforms the detection results of the other three system configurations for this dataset. For this dataset and this system configuration, the combined use of all the considered metrics produces $DR$ 100% and $FP$ 0% when $n = 60$. However, the metric combination $RSSI - INJ_{Rate} - NAV - SEQ_{Dif}$ generates $DR$ 100% and $FP_{Rate}$ 0% when $n = 42$. For $n = 42$, the use of all the considered metrics produces $DR$ 100% and $FP_{Rate}$ 2%. For both set of metrics, 5 metrics and 4 metrics, the system produces the detection results in an average time of 77µsec and 63µsec, respectively, when $n = 60$. Since the average interarrival time between two consecutive frames is 49msec, the intrusion detection can be implemented in real time.

For the $Normal$ dataset, the combination of all the available metrics produces the best detection results. For $n = 31$, the combined use of the six considered metrics produces $FP_{Rate}$ 0%, using the system configurations distance-mode and distance-mean. For this set of metrics and this sliding window length, the average processing time for the normal dataset is between 22µsec and 30µsec. Since the average interarrival time between two consecutive frames is 6.11msec, the intrusion detection process can fairly be implemented in real time.

For the $Attack01$ dataset, the combined use of the six considered metrics produces $DR$ 100% and $FP_{Rate}$ 0%, using the system configuration distance-mean and $n = 9$. This configuration has proven to outperform the detection results of the other three system configurations for this dataset. Nonetheless, the difference with the other three methodology configuration variations is very small. Only after the sliding window length reaches 31 slots long, all the metric configurations generate $FP_{Rate}$ 0%. Once more, the combined use of all the considered metrics produces the best detection results. The average time to process the $Attack01$ dataset, using all the considered metrics ranges between 20µsec and 39µsec, when $n = 31$. Since the average interarrival time between two consecutive frames is 10.1msec, the intrusion detection process can also be implemented in real time.

For the $Attack02$ dataset, all the system configurations produce $DR$ 100% for $17 \leq n$, but none of the four system configurations achieve 0% $FP_{Rate}$. The two configurations that make use of the distance generate 0.01% $FP_{Rate}$ after the sliding window length reaches 89 slots long. The other two configurations using the angle generate 0.03% $FP_{Rate}$ for $89 \leq n$. The different between the results for this dataset and the rest is that the angle-mean configuration produces the best $DR$ results overall when the sliding window length is $[7 \leq n \leq 17]$. For $17 > n$, the system configuration that generates the best results is distance-mean. Similar to all the Airpwn datasets, the combined use of all the considered metrics produces the best detection results. The average interarrival time between two consecutive frames for this dataset is 2.37msec, whereas the average time required to process the dataset ranges between 23µsec and 32µsec.

For the $Mixed\ Attack$ dataset, all the methodology configurations produce $DR$ 100% when $8 \leq n$, but none of the four methodology configurations achieve $FP_{Rate}$ 0% for $158 \geq n$. The distance-mean is the configuration that produces the best results overall, generating $FP_{Rate}$ 0.01% after the sliding window length reaches 91 slots long. For $31 \leq n$, none of the two configurations that use the distance overpasses $FP_{Rate}$ 0.05%. The average time to process the dataset ranges between 22µsec and 28µsec, whereas the average interarrival time between two consecutive frames is 2.18msec. Hence, the intrusion detection process can also be implemented in real time.

To summary up, the presented results have proven the effectiveness of the unsupervised anomaly based IDS presented in this thesis. Also, among all the system configurations, the use of the Euclidean distance makes the system to generate the best results. In the case of the deauthentication attack datasets, the mode is the best reference of normality. Whereas, the mean is the best reference of normality, in the case of the $Normal$ and the Airpwn attack datasets. In addition, the results demonstrate that using the multi-layer approach generates the best detection results overall. This is the case when analysing the $Normal$ and the Airpwn attack datasets. For this datasets, the multi-layer approach generates better results than the same methodology utilising different sets of metrics. For the deauthentication attack datasets, there is one particular set of metrics that outperform the multi-layer approach. Nonetheless, the detection results generated by the multi-layer approach are still highly accurate. Similarly, for the $Normal$ and the Airpwn attack datasets, there are cases in which the attack can be detected by using the information only from the single metric $INJ_{Rate}$. However, using solely this single metric to detect deauthentication attack would be highly ineffective. Therefore, for all the considered datasets, the combination of information from all the metrics produces the best results overall.

The sliding window length that generates the best detection results has also been evaluated. All the datasets have been analysed multiple times varying the length of the sliding window, and the results have been compared. In the case of the deauthentication experiments, the minimum sliding window length that generates the best results is $n = 60$. For the *Normal* and all the Airpwn datasets, the minimum sliding window length that generates the best results is $n = 31$. From all the results, there is not a particular sliding window length that is appropriate for all the evaluated datasets. However, based on the results presented in this chapter, it is estimated that a sliding window length approximate to 60 slots long may generate good detection results, regardless of the attacks been analysed. The results presented have also proved that the proposed methodology is capable of being implemented in real time. Whilst the average time to process the datasets is of a few tens of μsec, the average interarrival time between two consecutive frames is a few msec. Therefore, the intrusion detection process can be fairly implemented in real time.

In addition, the presented results have been used also to evaluate the maximum number of malicious frames that could be included within the initial sliding window before the accuracy of the detection results were affected. The results have shown that, for the Airpwn attack datasets and sliding window length larger than 12 frames, $12 \geq n$, the detection system produces $DR$ 100% and $FP_{Rate}$ 0%, comprising up to 43% of malicious frames within the initial sliding window. A higher percentage of malicious frames makes the detection accuracy to drop drastically. On the other hand, for the deauthentication attack dataset and sliding window length is between 31 and 90 frames, $[31 \leq n \leq 90]$, the detection system produces $DR$ 100% and $FP_{Rate}$ 0%, comprising up to 20% of malicious frames within the initial sliding window. When the sliding window is $90 > n$, the initial sliding window can comprise 13% of malicious frames to generate $DR$ 100% and $FP_{Rate}$ 0%.

# Chapter 8

## Conclusion and Future Work

### 8.1 Conclusions

Wireless Networks based on the IEEE 802.11 standard have experienced a tremendous growth in popularity over the last decade. Unfortunately, these networks present vulnerabilities exploitable by cyber-attackers [51]. The IEEE 802.11 standard has proposed different security protocols, establishing traffic encryption and integrity protection to the network infrastructure, as well as avoiding unauthorised access to the wireless networks. However, all these security protocols are vulnerable to decryption analysis processes [49] [118]. Wireless networks cannot rely on these security protocols to protect the content of the communications. Therefore, the design of secure and reliable wireless networks presents a major challenge to security system designers. Any effort to provide an extra level of protection to a network has become an issue of critical importance. This thesis has tackled the insecurity of wireless networks and has proposed a novel security system able to detect wireless-specific attacks.

There exist different types of wireless-specific attack that can compromise wireless networks. Initially, this thesis has presented an overview of the wireless-specific attacks that more commonly compromise wireless networks. The purpose of this overview was to find, if possible, a common implementation pattern, which could help to identify a common detection or countermeasure mechanism against these attacks. One of the countermeasures most commonly proposed in the literature is access control filters or MAC address filtering [50] [67]. However, it is easy to realise that the success of these mechanisms relies on the unequivocally identification of the attacker. Unfortunately, the MAC address in the frames is not a reliable approach to assess the real identity of the wireless device. An attacker can easily implement techniques of MAC address spoofing, and masquerade itself behind a fake MAC address. The efforts to provide reliable security should focus on the identification of the device that transmits the frames. The solution proposed in this thesis is to use, not only the MAC address, but also multiple other metrics from the wireless device or the wireless communication to infer the real identity of these devices. The higher the number of parameters or metrics used to identify the real identity of the wireless devices, the higher the probability to identify any attempt of an attacker to masquerade itself behind the spoofed identity of a legitimate wireless device.

This thesis presents a security system that makes use of metrics from multiple layers of observation to produce a collective decision on whether an attack is taking place. Although there are cases in which IDSs that utilise the information from a single metric give positive detection results, the combined use of multiple metrics from the same or different protocol layers commonly outperform the detection of the single-layer IDSs. Among different methods, the D-S theory has been chosen in this thesis as the data fusion technique to combine the evidences from the different layers. Despite been proven as a powerful and efficient technique, the major challenge for applying D-S theory in IDS is to automatically determine the BPA values, based on the information extracted from the network measurements.

The most important contribution of this thesis is a novel BPA methodology able to automatically adapt its probabilities assignment to the current characteristics of the wireless network, without requiring manual intervention from an IDS administrator. The current methods that perform the BPA process do not dynamically adapt to the measured characteristics of the monitored environment. The proposed approach is computationally simple, scalable and could easily applicable to other wireless technologies. The novel automatic, unsupervised and self-adaptive BPA methodology developed for this thesis is composed of three different and independent statistical techniques, able of generating predictions regarding the presence of attacks, based on historical parameters. Only two of these techniques work independently in order to meet the requirement of the independency of the beliefs. The third technique is based on the outcome beliefs of the other two. Despite being rather simplistic, and dissimilar to well-known information management techniques, the combined utilisation of the three techniques applied to the different wireless network metrics produce a highly accurate detection system. The detection system is able to generate $DR$ 100% and $FP_{Rate}$ 0% using the considered combination of metrics. The security system has been written in the C language, which provides great flexibility to be easily adapted or integrated to other security implementations.

The proposed security system has been evaluated with a reduced number of datasets and attacks. The effectiveness of the IDS proposed in this thesis will be evaluated using five different wireless network datasets, generated with two particular wireless-specific attacks (i.e. Airpwn and deauthentication attacks), as well as a dataset with non-malicious information. There might be a concern about whether the number of experiments is large enough to prove the efficiency of this security mechanism. Whilst evaluating all the existing wireless-specific attacks would be the most appropriate decision to assure that the IDSs can identify all these attacks, research wise, evaluating all these attacks would be impractical. Nonetheless, the attacks implemented in this thesis are an adequate sample to showcase the efficiency of the IDSs.

The proposed approach only requires a lightweight process for generating a baseline profile of normal utilisation, in order to generate high intrusion detection accuracy and low number of false alarms. A sliding window scheme is used to manage the information, implement the training process and construct an accurate statistical reference of normal behaviour. Only a reduced number of frames are required to generate a reference of normal wireless network behaviour. The best intrusion detection results for the evaluated datasets are achieved collecting as little as 60 frames. Four different configurations of the proposed BPA methodology have been described. Although there are differences in the final detection results, all the configurations provide very precise detection results. The sliding window length also influences the final detection results, as well as the time required to implement the intrusion detection analysis. The results have proven that the system is able to implement the intrusion detection in real time. For all the evaluated datasets, the processing time required to analyse each frame ranges between 45μsec and 71μsec.

The presented results have proven the effectiveness of the detection system presented in this thesis, generating $DR$ 100% and $FP_{Rate}$ 0% for most of the results. This performance is produced even if there exist a high proportion of malicious frames within the initial sliding window. These are, up to 43% of malicious frames within the initial sliding window using all the Airpwn datasets, and up to 20% of malicious frames within the initial sliding window using the deauthentication dataset. The used wireless-specific attacks have demonstrated the robustness of the detection approach irrespectively of the nature of the launched attack. To the best of the author knowledge, this is the first methodology able to efficiently identify Airpwn attack.

## 8.2 Future Work

Multiple research avenues still remain open in the field of wireless network intrusion detection. In the future, we would like to continue exploring some of these research avenues and enhancing the capabilities of the proposed IDS framework.

One of these future researches is to implement the proposed framework in additional type of wireless communications. In Chapter 1 an early attempt to evaluate the proposed framework on a WiMAX network was discussed. This approach was not feasible for a number of reasons previously explained. However, the interest for security of this type of wireless communication technology is still very high. The need for further security methods is not restricted only to WiFi or WiMAX. Other wireless communication technologies such as LTE, WSNs, Personal Area Networks (PANs) or Bluetooth would also require additional level of protection against attacks. Therefore, evaluating the effectiveness of the proposed framework in all these wireless communication technologies would be desirable.

Another future research area is to provide strong resilience capabilities to the wireless networks against attacks, and make transparent to the user the presence of these attacks. In Chapter 4 were briefly introduced the concepts of IPSs and ITSs, which allow the system administrator to specify a set of reactions to be implemented in case of detecting an attack. In an early attempt to provide resilience capabilities to the proposed IDS framework, this was configured to shut the web browser of the user down in case an Airpwn attack was detected. Although this might seem a very crude defence action, malicious and spurious information is then never displayed in the web browser to the user. This action is a proof of concept, and an early attempt to respond to the attack. Other actions, such as alerting the user, changing the security configuration or disconnecting the wireless communication may not be effective. One of the defence actions for future research is to discard the frames considered malicious. This is a defence approach totally transparent to the users, which would keep the network correctly working, despite the presence of attack. In [70], researchers have started investigating this type of prevention approach.

Providing active security mechanisms to IDSs is another future research that needs to be explored. The research presented in this thesis has obviated the fact that the IDS could also be compromised. However, similar to the wireless networks, IDSs are also targeted by attackers. This is a concern that has been previously arisen in [55] [139]. According to [36], these systems perform poorly defending themselves when they are targeted by attackers.

Other future research includes the automatic generation of labelled datasets and the automatic selection of metrics. Whereas these are two different research avenues, both are dependent on one another. In Chapter 5, the concept of feature selection was introduced, along with the multiple drawbacks of current feature selection techniques. These techniques work as supervised implementations. These techniques work only if the records in the analysed datasets have been previously labelled. One possible research avenue to tackle this issue is to propose a feature selection technique that could be implemented when unsupervised. These techniques would not require previously labelled datasets in order to operate. Another possible research avenue to tackle this issue is to propose a technique that could correctly label the instances in the analysed dataset automatically. This automatically labelled dataset could then be sent to a supervised feature selection technique for further analysis.

Another further research avenue would be adding the proposed framework in a suite of tools with additional security mechanisms, or integrating it in a more complete and robust security system. There always exists the option of collaborating with other researching groups to extend the security capabilities of the proposed system, or working with any company who wanted to include the proposed framework in any commercial tool.

## 8.3 Publications Part of This Thesis

- Konstantinos G. Kyriakopoulos, Francisco J. Aparicio-Navarro and David J. Parish, “Manual and automatic assigned thresholds in multi-layer data fusion intrusion detection system for 802.11 attacks,” in IET Information Security, vol.8, no.1, pp. 42-50, 2014.
- Konstantinos G. Kyriakopoulos, Francisco J. Aparicio-Navarro and David J. Parish, “Detecting misbehaviour in WiFi using multi-layer metric data fusion,” in Proc. of the IEEE International Workshop on Measurements and Networking Proceedings, M&N 2013, pp. 155-160, 2013.
- Francisco J. Aparicio-Navarro, Konstantinos G. Kyriakopoulos and David J. Parish, “An automatic and self-adaptive multi-layer data fusion system for WiFi attack detection,” in International Journal of Internet Technology and Secured Transactions, Inderscience, vol.5, no.1, pp. 42-62, 2013.
- Francisco J. Aparicio-Navarro, Konstantinos G. Kyriakopoulos and David J. Parish, “A multi-layer data fusion system for Wi-Fi attack detection using automatic belief assignment,” in Proc. of the Proceedings of World Congress on Internet Security, WorldCIS 2012, pp. 45-50, 2012.
- Francisco J. Aparicio-Navarro, Konstantinos G. Kyriakopoulos and David J. Parish, “An on-line wireless attack detection system using multi-layer data fusion,” in Proc. of the IEEE International Workshop on Measurements and Networking Proceedings, M&N 2011, pp. 1-5, 2011.
- Konstantinos G. Kyriakopoulos, Francisco J. Aparicio-Navarro and David J. Parish, “Fusing multi-layer metrics for detecting security attacks in 802.11 networks,” in Proc. of the Wireless Telecommunications Symposium, WTS 2011, pp. 1-6, 2011.
- Francisco J. Aparicio-Navarro and David J. Parish, “Misbehaviour metrics in WiMAX networks under attack,” in Proc. of the 11th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, PGNET 2010, pp. 1-6, 2010.

# References

[1] Ericsson, “Ericsson Mobility Report: On the Pulse of the Networked Society”. 2012.

[2] R. Webb, "Mobile operators ramp WiFi deployments for data offload, seek to integrate WiFi," Infonetics Research, 2012. Available: http://www.infonetics.com/pr/2012/Carrier-WiFi-Offload-and-Hotspot-Strategies-Survey-Highlights.asp (Access Date: 24 Feb, 2014).

[3] R. Webb, "Carrier WiFi equipment market exploding to $2.1 billion by 2016," Infonetics Research, 2012. Available: http://www.infonetics.com/pr/2012/Carrier-WiFi-Equipment-Market-Highlights.asp (Access Date: 24 Feb, 2014).

[4] Deloitte, "Wi-Fi leads the way for UK smartphone owners," Available: http://www.deloitte.com/view/en_GB/uk/news/news-releases/39069a1aa63b8310VgnVCM1000001a56f00aRCRD.htm (Access Date: 24 Feb, 2014).

[5] H. Yang, F. Ricciato, S. Lu, and L. Zhang, "Securing a wireless world," in IEEE, vol.94, no.2, pp. 442-454, 2006.

[6] Financial Times Website, "Security: Risk to data is underestimated," Available: http://www.ft.com/cms/s/0/1c997086-7121-11e2-9b5c-00144feab49a.html#axzz2LuSHoAat (Access Date: 24 Feb, 2014).

[7] J. Dixon, "Wireless intrusion detection systems including response," Available: http://www.infosecwriters.com/text_resources/pdf/Wireless_IDS_JDixon.pdf (Access Date: 24 Feb, 2014).

[8] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless LAN monitoring and its applications," in Proc. of the 3rd ACM Workshop on Wireless Security, pp. 70-79, 2004.

[9] K. El-Khatib, "Impact of feature reduction on the efficiency of wireless intrusion detection systems," in IEEE Transactions on Parallel and Distributed Systems, vol.21, no.8, pp. 1143-1149, 2010.

[10] R. S. Gill, "Intrusion detection techniques in wireless local area networks," PhD dissertation, Queensland University of Technology, June 2009.

[11] Symantec, "Internet Security Threat Report: 2011 Trends Volume 17," Available: http://www.symantec.com/about/news/resources/press_kits/detail.jsp?pkid=threat_report_17 (Access Date: 24 Feb, 2014).

[12] D. Welch, and S. Lathrop, "Wireless security threat taxonomy," In IEEE Systems, Man and Cybernetics Society, Information Assurance Workshop, pp. 76-83, 2003.

[13] M. A. Maloof, "Machine learning and data mining for computer security, " Springer-Verlag London Limited, 2006.

[14] C. Thomas, and N. Balakrishnan, "Selection of intrusion detection system threshold bounds for effective sensor fusion," in Proc. of the International Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, pp. 657007-657007, 2007.

[15] M. Xue, and C. Zhu, "Applied research on data mining algorithm in network intrusion detection," in Proc. of the International Joint Conference on Artificial Intelligence, JCAI 2009, pp. 275-277, 2009.

[16] Y. Liao, V. R. Vemuri, and A. Pasos, "Adaptive anomaly detection with evolving connectionist systems," in Journal of Network and Computer Applications, vol.30, no.1, pp. 60-80, 2007.

[17] G. Combs, "TShark - The Wireshark Network Analyser," Available: http://www.wire shark.org/docs/man-pages/tshark.html (Access Date: 24 Feb, 2014).

[18] Airpwn Available: http://airpwn.sourceforge.net/Airpwn.html.

[19] V. Jacobsen, C. Leres, and S. McCanne, "Tcpdump/libpcap," 2005. Available: http:// www.tcpdump.org (Access Date: 24 Feb, 2014).

[20] Dr. K. G. Kyriakopoulos personal website, Measurement Data, Available: http://home pages.lboro.ac.uk/~elkk/Site/Testbed_data.html (Access Date: 24 Feb, 2014).

[21] R. R. Yager, "Aggregating non-independent Dempster–Shafer belief structures," in Proc. of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2008, vol.1, pp. 289-297. 2008.

[22] M. F. Triola, "Bayes' Theorem," Elementary Statistics 11th edition, Addison Wesley, 2010.

[23] C. Shih, and G. Kochanski, "Bayes' Theorem," 2006. Available: http://kochanski.org/ gpk/teaching/0401Oxford/Bayes.pdf (Access Date: 24 Feb, 2014).

[24] R. Perdisci, G. Giacinto, and F. Roli, "Alarm clustering for intrusion detection systems in computer networks," in Engineering Applications of Artificial Intelligence, vol.19, no.4, pp. 429-438. 2006.

[25] BBC News Website, "Unauthorised YouTube adverts exposed by security firm," Available: http://www.bbc.co.uk/news/technology-23696805 (Access Date: 24 Feb, 2014).

[26] Forbes Website, "The creepy ad firm that's charging top brands for YouTube ads that aren't supposed to exist," Available: http://www.forbes.com/sites/alexkonrad/ 2013/08/13/charging-top-brands-for-made-up-youtube-ads/ (Access Date: 24 Feb, 2014).

[27] Financial Times Website, "Software that hijacks visits to YouTube uncovered," Available: http://www.ft.com/cms/s/0/1b6edec2-03a1-11e3-980a-00144feab7de.html#axzz2bv m0HBJX (Access Date: 24 Feb, 2014).

[28] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, "Knowing your enemy: understanding and detecting malicious web advertising," in Proc. of the ACM Conference on Computer and Communications Security, CCS 2012, pp. 674-686, 2012.

[29] T. M. Khoshgoftaar, S. V. Nath, S. Zhong, and N. Seliya, "Intrusion detection in wireless networks using clustering techniques with expert analysis," in Proc. of the Fourth International Conference on Machine Learning and Applications, pp. 6, 2005.

[30] G. Shafer, "A Mathematical Theory of Evidence. Princeton University Press," Princeton university press, vol.1, 1976.

[31] F. J. Aparicio-Navarro, K. G. Kyriakopoulos, and D. J. Parish, "A multi-layer data fusion system for Wi-Fi attack detection using automatic belief assignment," in Proc. of the Proceedings of World Congress on Internet Security, WorldCIS 2012, pp. 45-50, 2012.

[32] K. G. Kyriakopoulos, F. J. Aparicio-Navarro, and D. J. Parish, "Manual and automatic assigned thresholds in multi-layer data fusion intrusion detection system for 802.11 attacks," in IET Information Security, vol.8, no.1, pp. 42-50, 2014.

[33] K. G. Kyriakopoulos, F. J. Aparicio-Navarro, and D. J. Parish, "Fusing multi-layer metrics for detecting security attacks in 802.11 networks," in Proc. of the Wireless Telecommunications Symposium, WTS 2011, pp. 1-6, 2011.

[34] C. Rottondi, and G. Verticale, "Using packet interarrival times for Internet traffic classification," in Proc. of the IEEE Latin-American Conference on Communications, LATINCOM 2011, pp. 1-6, 2011.

[35] P. Varga, and G. Kún, "Utilizing higher order statistics of packet interarrival times for bottleneck detection," in Proc. of the Workshop on End-to-End Monitoring Techniques and Services, 2005, pp. 152-163, 2005.

[36] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," in Computers & Security, vol.28, no.1, pp. 18-28, 2009.

[37] G. Thamilarasu, S. Mishra, and R. Sridhar, "A cross-layer approach to detect jamming attacks in wireless ad hoc networks," in Proc. of the Military Communications Conference, MILCOM 2006, pp. 1-7, 2006.

[38] X. Wang, J. S. Wong, F. Stanley, and S. Basu, "Cross-layer based anomaly detection in wireless mesh networks," in Proc. of the 9th Annual International Symposium on Applications and the Internet, SAINT 2009, pp. 9-15, 2009.

[39] C. Siaterlis, and B. Maglaris, "Towards multisensor data fusion for DoS detection," in Proc. of the ACM Symposium on Applied Computing, SAC 2004, pp. 439-446, 2004.

[40] V. Chatzigiannakis, G. Androulidakis, K. Pelechrinis, S. Papavassiliou, and V. Maglaris, "Data fusion algorithms for network anomaly detection: classification and evaluation," in Proc. of the 3rd International Conference on Networking and Services, ICNS 2007, pp. 50-50, 2007.

[41] A. G. Fragkiadakis, V. A. Siris, and A. P. Traganitis, "Effective and robust detection of jamming attacks," in Proc. of the Future Network and Mobile Summit, pp. 1-8, 2010.

[42] D. Yu, and D. Frincke, "Alert confidence fusion in intrusion detection systems with extended Dempster-Shafer theory," in Proc. of the 43rd ACM Annual Southeast regional conference, vol.2, pp. 142-147, 2005.

[43] K. G. Kyriakopoulos, F. J. Aparicio-Navarro, and D. J. Parish. "Detecting misbehaviour in WiFi using multi-layer metric data fusion," in Proc. of the IEEE International Workshop on Measurements and Networking Proceedings, M&N 2013, pp. 155-160, 2013.

[44] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala, "An accurate technique for measuring the wireless side of wireless networks." in Proc. of the Workshop on Wireless Traffic Measurements and Modeling, pp. 13-18, 2005.

[45] MIT Lincoln Laboratory - DARPA intrusion detection evaluation dataset, Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/ (Access Date: 24 Feb, 2014).

[46] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of DARPA dataset for intrusion detection system evaluation," SPIE Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, vol.6973, pp. 1-8, 2008.

[47] B. R. Mathon, M. M. Ozbek, and G. F. Pinder, "Dempster–Shafer theory applied to uncertainty surrounding permeability," in Journal of Mathematical Geosciences, vol.42, no.3, pp. 293-307, 2010.

[48] M. Verleysen, and D. François, "The curse of dimensionality in data mining and time series prediction," Computational Intelligence and Bioinspired Systems, in Proc. of the 8th International Work-Conference on Artificial Neural Networks IWANN 2005, pp. 758-770, 2005.

[49] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in Proc. of the 7th Annual International Conference on Mobile Computing and Networking, pp. 180-189, 2001.

[50] W. A. Arbaugh, N. Shankar□, and Y. C. J. Wan, "Your 80211 wireless network has no clothes," in IEEE Wireless Communications, vol.9, no.6, pp. 44-51, 2002.

[51] J. Bellardo, and S. Savage, "802.11 Denial-of-service attacks real vulnerabilities and practical solutions," Available: http://cseweb.ucsd.edu/~savage/papers/UsenixSec 03.pdf (Access Date: 24 Feb, 2014).

[52] M. D. Aime, G. Calandriello, and A. Lioy, "Dependability in wireless networks: Can we rely on WiFi?," in IEEE Security & Privacy, vol.5, no.1, pp. 23-29, 2007.

[53] M. Shin, J. Ma, A. Mishra, and W. A. Arbaugh, "Wireless network security and interworking," in IEEE vol.94, no.2, pp. 455-466, 2006.

[54] H. Nguyen, K. Franke, and S. Petrovic, "Improving effectiveness of intrusion detection by correlation feature selection," in Proc. of the International Conference on Availability, Reliability, and Security, ARES 2010, pp. 17-24, 2010.

[55] K. Scarfone, and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800.94, 2007.

[56] Y. Bai, and H. Kobayashi, "Intrusion detection systems: Technology and development," in Proc. of the 17th International Conference on Advanced Information Networking and Applications, AINA 2003, pp. 710-715, 2003.

[57] C. A. Catania, and C. García Garino, "Automatic network intrusion detection: Current techniques and open issues." in Computers & Electrical Engineering, vol.38, no.5, pp. 1062-1072, 2012.

[58] H. T. Elshoush, and I. M. Osman, "Alert correlation in collaborative intelligent intrusion detection systems—A survey," in Applied Soft Computing vol.11, no.7, pp. 4349-4365, 2011.

[59] K. Bicakci, and B. Tavli, "Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks," in Computer Standards & Interfaces, vol.31, no.5, pp. 931-941, 2009.

[60] S. Khan, K.-K. Loo, T. Naeem, and M. A. Khan, "Denial of service attacks and challenges in broadband wireless networks," in IJCSNS International Journal of Computer Science and Network Security, vol.8, no.7, pp. 1-6, 2008.

[61] M. Malekzadeh, A. Azim, A. Ghani, J. Desa, and S. Subramaniam, "Empirical analysis of virtual carrier sense flooding attacks over wireless local area network," in Journal of Computer Science, vol.5, no.3, pp. 1-7, 2009.

[62] B. Chen, and V. Muthukkumarasamy, "Denial of service attacks against 802.11 DCF," in Proc. of the IADIS International Conference: Applied Computing, pp. 1-5, 2006.

[63] D. J. Thuente, and M. Acharya, "Intelligent jamming in wireless networks with applications to 802.11b and other networks," in Proc. of the IEEE Military Communications Conference, MILCOM 2006, vol.6, pp. 1-7, 2006.

[64] M. Acharya, T. Sharma, D. Thuente, and D. Sizemore, "Intelligent jamming in 802.11b wireless networks," in Proc. of the Conference OPNETWORK-2004, pp. 1-10, 2004.

[65] M. Acharya, and D. Thuente, "Intelligent jamming attacks, counterattacks and (counter)2 attacks in 802.11b wireless networks," in Proc. of the Conference OPNETWORK-2005, pp. 1-12, 2005.

[66] OPNET Modeler, OPNET Technologies Inc. Available: http://www.opnet.com (Access Date: 24 Feb, 2014).

[67] C. Liu, and J. Yu, "A solution to WLAN authentication and association DoS attacks," in IAENG International Journal of Computer Science, vol.34, no.1, pp. 1-6, 2007.

[68] F. Guo, and T.-c. Chiueh, "Sequence number-based MAC address spoof detection," in Proc. of the 8th International Conference on Recent Advances in Intrusion Detection, RAID 2005, pp. 309-329, 2005.

[69] Q. Li, and W. Trappe, "Relationship-based detection of spoofing-related anomalous traffic in ad hoc networks," in Proc. of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON 2006, vol.1, pp. 50-59, 2006.

[70] Y. Zhang, and S. Sampalli, "Client-based intrusion prevention system for 802.11 wireless LANs," in Proc. of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2010, pp. 100-107, 2010.

[71] Y. Sheng, T. Keren, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in Proc. of the 27th IEEE Conference on Computer Communications, INFOCOM 2008, pp. 2441-2449, 2008.

[72] D. Madory, "New methods of spoof detection in 802.11b wireless networking," Master dissertation, Dartmouth College, Hanover, New Hampshire, June 2006.

[73] K. Zeng, K. Govindan, D. Wu, and P. Mohapatra, "Identity-based attack detection in mobile wireless networks," in Proc. of the IEEE Conference on Computer Communications, INFOCOM 2011, pp. 1880-1888, 2011.

[74] D. B. Faria, and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in Proc. of the 5th ACM Workshop on Wireless security, WiSe 2006, pp. 43-52, 2006.

[75] Q. Chen, and U. Aickelin, "Anomaly detection using the Dempster-Shafer method," in Proc. of the International Conference on Data Mining, DMIN 2006, pp. 232-240, 2006.

[76] G. Chandrasekaran, J.-A. Francisco, V. Ganapathy, M. Gruteser, and W. Trappe, "Detecting identity spoofs in IEEE 802.11e wireless networks," in Proc. of the IEEE Global Telecommunications Conference, GLOBECOM 2009, pp. 1-6, 2009.

[77] G. A. Marin, "Network security basics," IEEE Security & Privacy, vol.3, no.6, pp. 68-72, 2005.

[78] D. Bolzoni, "Revisiting anomaly-based network intrusion detection systems," University of Twente, 2009.

[79] W. Stallings, "Network security essentials: Applications and Standard," Prentice Hall, 2003.

[80] C. Thomas, and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," in IEEE Transactions on Information Forensics and Security, vol.4, no.3, pp. 542-551, 2009.

[81] D. Hongbo, "Data mining techniques and applications: An introduction," Course Technology Cengage Learning, 2010.

[82] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots," in Proc. of the 2nd International Conference on Mobile systems, Applications, and Services, MobiSYS 2004, pp. 84-97, 2004.

[83] Y. Sheng, K. Tan, U. Deshpande, B. Vance, H. Yin, C. McDonald, T. Henderson, G. Chen, D. Kotz, A. Campbell, and J. Wright, "MAP: A scalable measurement infrastructure for securing 802.11 wireless networks," Available: http://citeseerx.ist.psu.edu/viewdoc /download?doi=10.1.1.83.4763&rep=rep1&type=pdf (Access Date: 24 Feb, 2014).

[84] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in Proc. of the IEEE Symposium on Security and Privacy, SP 1999, pp. 120-132, 1999.

[85] T. N. Phyu, "Survey of classification techniques in data mining," in Proc. of the International MultiConference of Engineers and Computer Scientists, IMECS 2009, vol.1, pp. 18-20, 2009.

[86] D. Joo, T. Hong, and I. Han, "The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors," in Expert Systems with Applications vol.25, no.1, pp. 69-75, 2003.

[87] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in Applications of Data Mining in Computer Security, pp. 77-101. Springer US, 2002.

[88] C. Song, and K. Ma, "Design of intrusion detection system based on data mining algorithm," in Proc. of the International Conference on Signal Processing Systems, ICSPS 2009, pp. 370-373, 2009.

[89] L. Bo, and J. Dong-Dong, "The research of intrusion detection model based on clustering analysis," in Proc. of the International Conference on Computer and Communications Security, CCS 2009, pp. 24-27, 2009.

[90] W. Lee, and S. J. Stolfo, "Data mining approaches for intrusion detection," in Proc. of the 7th USENIX Security Symposium, pp. 79-94, 1998.

[91] A. M. Nambiar, A. Vijayan, and A. Nandakumar, "Wireless intrusion detection based on different clustering approaches," in Proc. of the 1st Amrita ACM-W Celebration on Women in Computing in India, pp. 42, 2010.

[92] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," in Proc. of the 43rd annual Southeast regional conference, vol.2, pp. 136-14, 2005.

[93] A. Tamilarasan, S. Mukkamala, A. H. Sung, and K. Yendrapalli, "Feature ranking and selection for intrusion detection using artificial neural networks and statistical methods," in Proc. of the International Joint Conference on Neural Networks, IJCNN 2006, pp. 4754-4761, 2006.

[94] A. H. Sung, and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in Proc. of the International Symposium on Applications and the Internet, SAINT 2003, pp. 209-216, 2003.

[95] W. Wang, X. Zhang, S. Gombault, and S. J. Knapskog, "Attribute normalization in network intrusion detection." in Proc. of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks, ISPAN 2009, pp. 448-453, 2009.

[96] G. Frazier, and R. Gray, "Cross-layer anomaly correlation and response selection," in Proc. of the Military Communications Conference, MILCOM 2010, pp. 405-410, 2010.

[97] D. Xhemali, C. J. Hinde, and R. G. Stone, "Naïve Bayes vs. decision trees vs. neural networks in the classification of training web pages," In International Journal of Computer Science Issues - IJCSI, vol.7, no.4, 2010.

[98] W. Hu, J. Li, and Q. Gao, "Intrusion detection engine based on dempster-shafer's theory of evidence," in Proc. of the International Conference on Communications, Circuits and Systems Proceedings, vol.3, pp. 1627-1631, 2006.

[99] J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in Advances in Neural Information Processing Systems, vol.10, 1998.

[100] S. Zhong, T. M. Khoshgoftaar, and S. V. Nath, "A clustering approach to wireless network intrusion detection," in Proc. of the 17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2005, pp. 7-13, 2005.

[101] Y. Zhang, and W. Lee, "Intrusion detection in wireless ad-hoc networks," in Proc. of the 6th ACM Annual International Conference on Mobile Computing and Networking, MobiCom 200, pp. 275-283, 2000.

[102] T. M. Chen, and V. Venkataramanan, "Dempster-Shafer theory for intrusion detection in ad hoc networks," in IEEE Internet Computing, vol.9, no.6 pp. 35-41, 2005.

[103] G. Thamilarasu, and R. Sridhar, "Exploring cross-layer techniques for security: Challenges and opportunities in wireless networks," in Proc. of the Military Communications Conference, MILCOM 2007, pp. 1-6. 2007.

[104] J. Yu, Y. V. Ramana Reddy, S. Selliah, S. Reddy, V. Bharadwaj, and S. Kankanahalli, "TRINETR: An architecture for collaborative intrusion detection and knowledge-based alert evaluation," in Advanced Engineering Informatics vol.19, no.2, pp. 93-101, 2005.

[105] C. Thomas, "Performance enhancement of intrusion detection systems using advances in sensor fusion," in Proc. of the 11th International Conference on Information Fusion, FUSION 2008, pp. 1-7, 2008.

[106] C. Katar, "Combining multiple techniques for intrusion detection," in International Journal of Computer Science and Network Security, vol.6, no.2B, pp. 208-218, 2006.

[107] C. R. Parikh, M. J. Pont, and N. Barrie Jones, "Application of Dempster–Shafer theory in condition monitoring applications: A case study," in Journal of Pattern Recognition Letters, vol.22, no.6, pp. 777-785, 2001.

[108] J. R. Boston, "A signal detection system based on Dempster-Shafer theory and comparison to fuzzy detection," in IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.30, no.1, pp. 45-51, 2000.

[109] S. Zander, A. Grenville, and P. Branch, "An empirical evaluation of IP Time To Live covert channels," in Proc. of the 15th IEEE International Conference on Networks, ICON 2007, pp. 42-47, 2007.

[110] S. Sheen, and R. Rajesh, "Network intrusion detection using feature selection and decision tree classifier," in Proc. of the IEEE Region 10 Conference, TENCON 2008, pp. 1-4, 2008.

[111] IEEE 802.11 Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Available: http://standards.ieee.org/about/ get/802/802.11.html (Access Date: 24 Feb, 2014).

[112] G. Matthew, "802.11 wireless networks: The definitive guide," O'Reilly Media, Inc., 2005.

[113] M. Malekzadeh, A. Azim, A. Ghani, J. Desa, and S. Subramaniam, "An experimental evaluation of DoS attack and its impact on throughput of IEEE 802.11 wireless networks," in IJCSNS International Journal of Computer Science and Network Security, vol.8, pp. 1-5, 2008.

[114] A. Vartak, S. Ahmad, and K. N. Gopinath, "An experimental evaluation of over-the-air (OTA) wireless intrusion prevention techniques," in Proc. of the 2nd International Conference on Communication Systems Software and Middleware, COMSWARE 2007, pp. 1-7, 2007.

[115] A. E. Earle, "Wireless security handbook," CRC Press, 2006.

[116] D. Papini, "An Anomaly based wireless intrusion detection system," PhD dissertation, Technical University of Denmark, 2008.

[117] M. Ohta, Y. Kanda, K. Fukuda, and T. Sugawara, "Analysis of spoofed IP traffic using Time-to-Live and identification fields in IP headers," in Proc. of the IEEE Workshops of International Conference on Advanced Information Networking and Applications, WAINA 2011, pp. 355-361, 2011.

[118] AirTight Networks "WPA2 Hole196 vulnerability," Available: http://www.slideshare .net/AirTightWIPS/wpa2-hole196vulnerabilityfa-qs (Access Date: 24 Feb, 2014).

[119] R. S. Gill, J. Smith, and A. J. Clark, "Specification-based intrusion detection in WLANs," pp. 141-152, 2006.

[120] V. R. Vemuri, "Enhancing computer security with smart technology," CRC Press, 2006.

[121] T. d’Otreppe, "Aircrack-ng," Available: http://www.aircrack-ng.org/ (Access Date: 24 Feb, 2014).

[122] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in Proc. of the 6th ACM international symposium on Mobile ad hoc Networking and Computing, pp. 46-57, 2005.

[123] C. Pöpper, M. Strasser, and S. Čapkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques," in IEEE Journal on Selected Areas in Communications, vol.28, no.5, pp. 703-715, 2010.

[124] K. Nasr, A. A. E. Kalam, and C. Fraboul, "A holistic methodology for evaluating wireless Intrusion Detection Systems," in Proc. of the 5th International Conference on Network and System Security, NSS 2011, pp. 9-16, 2011.

[125] J. Timofte, "Wireless intrusion prevention systems," in Revista Informatica Economica, vol.47, pp. 129-132, 2008.

[126] H. Labiod, "Wi-Fi, Bluetooth, Zigbee and WiMAX," Springer London, Limited. 2007.

[127] C.-C. Tuan, Y.-C. Wu, W.-S. Chang, and W.-T. Huang, "Fault tolerance by quartile method in wireless sensor and actor networks," in Proc. of the International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2010, pp. 758-763, 2010.

[128] J. Malinen, "Host AP Driver," 2007. Available: http://hostap.epitest.fi/hostapd/ (Access Date: 24 Feb, 2014).

[129] F. J. Aparicio-Navarro, K. G. Kyriakopoulos and D. J. Parish, "An automatic and self-adaptive multi-layer data fusion system for WiFi attack detection, " in International Journal of Internet Technology and Secured Transactions, Inderscience, vol.5, no.1, pp. 42-62, 2013.

[130] R. Zhang, and G. Zhi, "Using KPCA feature selection and fusion for intrusion detection," in Proc. of the Sixth International Conference on Natural Computation, ICNC 2010, vol.2, pp. 981-985, 2010.

[131] P. E. Veríssimo, N. F. Neves, and M. P. Correia, "Intrusion-tolerant architectures: Concepts and design," in Architecting Dependable Systems, vol.2677, pp. 3-36. Springer Berlin Heidelberg, 2003.

[132] R. Zhang, D. Qian, C. Ba, W. Wu, and X. Guo, "Multi-agent based intrusion detection architecture," in Proc. of the International Conference on Computer Networks and Mobile Computing, ICCNMC 2001, pp. 494-501, 2001.

[133] Y. Zhang, and W. Lee, "Intrusion detection in wireless ad-hoc networks," in Proc. of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom 2000, pp. 275-283, 2000.

[134] P. Sandford, D. J. Parish, and J. M. Sandford, "Understanding increasing traffic levels for internet abuse detection," in Security Journal vol.20, no.2, pp. 63-76, 2007.

[135] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in Image Analysis and Processing, ICIAP 2005, pp. 50-57. Springer Berlin Heidelberg, 2005.

[136] S. F. Yusufovna, "Integrating intrusion detection system and data mining." in Proc. of the International Symposium on Ubiquitous Multimedia Computing, UMC 2008, pp. 256-259, 2008.

[137] Y. Fu, J. He, and G. Li, "A distributed intrusion detection scheme for mobile ad hoc networks," in Proc. of the 31st Annual International Computer Software and Applications Conference, COMPSAC 2007, vol.2, pp. 75-80, 2007.

[138] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," in Computer Communications, vol.34, no.18, pp. 2227-2235, 2011.

[139] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," in Information Sciences, vol.239, pp. 201-225, 2013.

[140] L. Portnoy, "Intrusion detection with unlabeled data using clustering," in Proc. of the Proceedings of ACM CSS Workshop on Data Mining Applied to Security, DMSA 2001, pp. 1-14, 2001.

[141] S. X. Wu, and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," in Applied Soft Computing vol.10, no.1, pp. 1-35, 2010.

[142] C. Dartigue, H. I. Jang, and W. Zeng, "A new data-mining based approach for network intrusion detection," in Proc. of the Seventh Annual Communication Networks and Services Research Conference, CNSR 2009, pp. 372-377, 2009.

[143] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "A comprehensive approach to intrusion detection alert correlation," IEEE Transactions on Dependable and Secure Computing, vol.1, no.3, pp. 146-169, 2004.

[144] V. Marinova-Boncheva, "Applying a data mining method for intrusion detection," in Proc. of the International Conference on Computer Systems and Technologies, CompSysTech 2007, pp. 1-6, 2007.

[145] C. Zhang, G. Zhang, and S. Sun, "A mixed unsupervised clustering-based intrusion detection model," in Proc. of the 3rd International Conference on Genetic and Evolutionary Computing, WGEC 2009, pp. 426-428, 2009.

[146] F. Gargiulo, C. Mazzariello, and C. Sansone, "Automatically building datasets of labeled IP traffic traces: A self-training approach," in Applied Soft Computing, vol.12, no.6, pp. 1640-1649, 2012.

[147] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," in Computers & Security, vol.31, no.3, pp. 357-374, 2012.

[148] M. V. Mahoney, and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," In Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, vol.2820, pp. 220-237. Springer Berlin Heidelberg, 2003.

[149] I. Ruthven, and M. Lalmas, "Using Dempster-Shafer's theory of evidence to combine aspects of information use," in Journal of Intelligent Information Systems, vol.19, no.3, pp. 267-301, 2002.

[150] J. Parker, A. Patwardhan, and A. Joshi, "Cross-layer analysis for detecting wireless misbehavior." in Proc. of the IEEE Consumer Communications and Networking Conference, CCNC 2006, pp. 6-9, 2006.

[151] A. P. F. Chan, W. W. Y. Ng, D. S. Yeung, and E. C. C. Tsang, "Comparison of different fusion approaches for network intrusion detection using ensemble of RBFNN," in Proc. of the International Conference on Machine Learning and Cybernetics, vol.6, pp. 3846-3851, 2005.

[152] SPSS, IBM. "IBM SPSS Statistics Base 20," Available: http://www-01.ibm.com/software/analytics/spss/products/statistics/ (Access Date: 24 Feb, 2014).

[153] A. Bryman, and D. Cramer, "Quantitative data analysis with SPSS 14, 15 and 16: A guide for social scientists," Routledge, 2009.