ANOMALY DETECTION TECHNIQUES USING ENTROPY SPACE CHARACTERIZATION

by

RAÚL HERNÁNDEZ AQUINO

Monterrey, N.L., December 2011
ANOMALY DETECTION TECHNIQUES USING ENTROPY SPACE CHARACTERIZATION

by

RAÚL HERNÁNDEZ AQUINO

Thesis

Submitted to the School of Engineering and Information Technology at the

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey in partial fulfillment of the requirements for the degree of

Master of Science in

ELECTRONIC ENGINEERING

MAJOR IN TELECOMMUNICATIONS

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Monterrey, N.L., December 2011
Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

School of Engineering and Information Technology

Thesis committee members recommend the present thesis of RAÚL HERNÁNDEZ AQUINO, Eng. to be accepted in partial fulfillment of the requirements for the degree of Master of Science in ELECTRONIC ENGINEERING MAJOR IN TELECOMMUNICATIONS.

Thesis Committee

César Vargas Rosales, Ph.D.
Thesis advisor

Jorge Carlos Mex Perera, Ph.D.
Examiner

Gerardo Antonio Castañón Ávila, Ph.D.
Examiner

Gerardo Antonio Castañón Ávila, Ph.D.
Director of the Graduate Program in the School of Engineering and Information Technology

December 2011
Abstract

Since the nature of the communication networks is to provide the means to establish a correct information interchange between users, there are vulnerabilities that can be exploited with malicious intentions. The above can be observed in the fact that the amount and diversity of attacks that are reported every year is increasing, and the economic losses and information leaks are very important issues that bring out the significance of the network security subject.

As the communication networks increase in both size and scalability, more mechanisms are needed to overcome the issue of network security. Firewalls and anti-virus and Intrusion Detection Systems (IDS) were developed with the objective to provide a higher level of network security, however, they can only offer limited protection as the amount and sophistication of the attacks has increased over the years.

A new generation of IDS based on the concept of entropy has emerged over the last years facing the security network issue. The entropy, used as a measurement of the disorder of certain traffic features fits in a large scale with the characterization of traffic, as malicious activities affect the typical behavior of a network.

In this thesis, five methods based on the concept of entropy for anomaly detection were developed and analyzed. The entropy concept was used to create entropy space models by examining the intrinsic features (source IP, destination IP, source port and destination port) of the IP packet headers. With the information gathered from the intrinsic features, 3-D entropy spaces were created (and 1-D entropy spaces for one of the methods) for each of the intrinsic feature, so at the end, 4 entropy spaces were obtained, each one describing each intrinsic feature. The five methods proposed were: Excess points method, which makes use of the Bayes classifier rule to generate decision regions for benign and anomalous traffic distances to a benign centroid, Continuous entropy method, which also uses the Bayes classifier rule but applied to 1-D entropy spaces, Statistical method, which makes use of statistics to determine an anomaly behavior, Centroids method, which bases its decision to declare traffic as benign or anomalous in the cumulative centroid of previous time windows and, the $K$-nearest neighbor entropy method that uses a combination of Principal Component Analysis (PCA) and the $K$-nearest neighbor classifier to create patterns of benign and anomalous traffic.

Each method showed advantages and drawbacks that were analyzed. Finally, a scheme which uses the combination of the methods, with the objective of strengthen their benefits and minimize their disadvantages was proposed. Good results were obtained in the detection of anomalies with the methods proposed after being tested with different types of attacks.
Dedication

To my beloved parents for giving me the privilege of living and for being with me in every step of my life. There is nothing I could have done in my life without your love and support. There are no words to describe how grateful I am and will always be to you.

To my sister for showing me the unconditional love and loyalty that only a sister has. Thank you for loving me and showing me your care.

To my grandfather “Papatayo”. Thank you for your advices and life example, and thank you too for teaching me the importance of the 51%.
Acknowledgments

I thank God for giving me the opportunity to live and for his enormous love that has been shown by providing me with many blessings such as my family and friends.

I thank Instituto Tecnologico y de Estudios Superiores de Monterrey for having financed my master’s studies and for making possible for me to receive a higher level education. I thank the chair of wireless networks an mobility for the facilities provided for my thesis and specially to doctor César Vargas Rosales for his support as my thesis advisor.

I thank my parents for being my guidance and example of excellence in both academic and life aspects. I also thank my sister for her support in every decision that i have taken in my life.

I thank my family for the love that they have given to me trough all these years. Thank you aunt Elvia, grandmother Ema, cousin Gibrán, uncles Armando and Luz and cousins Mariluz and Mariana, grandmother Ina and grandfathers “papaúl” and “papatayo”.

I thank all of my friends for all the love that they have given to me, and for all the memories from which i found true motivation to carry on in difficult times. Thank you Mario for being everything one man can ask for in a friend, for continuing showing me that for a true friend the love is beyond anything. Thank you for being not only someone to turn to in any situation but also for becoming my pride. Thank you Boni for also giving me the example of excellence as a friend, for never letting me down and for simply being there showing me how noble a human being can be. You two are my brothers as you know.

I thank all of the people that i have met and that have changed my life in so many ways, thank you for all your support and love that you have given me. Thank you so much Indy, Serafin, Chito, Christian, Fer L., Fer R., Raúl, Flor, Pedito, Rafa, Chela, Víctor, Violeta, Edson. Thank you for everything, you will always be in my heart.
Contents

1 Introduction .................................................. 1
  1.1 Motivation .................................................. 3
  1.2 Problem statement .......................................... 3
  1.3 Objectives .................................................. 4
    1.3.1 Particular objectives ................................ 4
  1.4 Contributions .............................................. 4
  1.5 Work organization ......................................... 5

2 Fundamentals .................................................. 6
  2.1 Intrusion Detection Systems theory ....................... 6
  2.2 Entropy based A-NIDS ...................................... 9
  2.3 Attacks’ description ....................................... 10
    2.3.1 Blaster worm .......................................... 10
    2.3.2 Sasser worm .......................................... 11
    2.3.3 Ps .................................................. 12
    2.3.4 Sshtrojan ............................................ 12
    2.3.5 Portsweep ............................................ 12
    2.3.6 Ipsweep .............................................. 12
    2.3.7 DDoS1 ................................................ 13
    2.3.8 DDoS2 ................................................ 13
  2.4 K-nearest neighbor algorithm ............................. 14

3 Model .......................................................... 17
  3.1 Entropy Space modeling .................................... 17
  3.2 Excess points method ..................................... 21
    3.2.1 Training stage ........................................ 21
    3.2.2 Detection stage ....................................... 27
  3.3 Continuous entropy method ................................ 29
    3.3.1 Training stage ........................................ 31
    3.3.2 Detection stage ....................................... 35
  3.4 Statistical method ......................................... 36
  3.5 Centroids method .......................................... 38
    3.5.1 Training stage ........................................ 38
    3.5.2 Detection stage ....................................... 41
List of Figures

2.1 Multiprototype pattern with two classes .......................... 15
3.1 Entropy spaces for the 4 cluster keys ................................. 20
3.2 Entropy space identification of $H^1$ for a Blaster worm attack .. 24
3.3 Probability density function estimation for Src IP benign traffic ... 26
3.4 Excess points method diagram ........................................ 28
3.5 Entropy space identification of $H^2$ for a ssh trojan attack ....... 30
3.6 $R_b$ entropy parameter of each cluster key .......................... 33
3.7 Entropy parameter pdf of each cluster key ........................... 34
3.8 Continuous entropy method diagram .................................. 36
3.9 Centroid convergence for 4 different attacks ......................... 40
3.10 Centroids method diagram .............................................. 43
3.11 Representative 2-D space patterns of benign ($\tilde{Z}^r$) and anomalous ($\tilde{Z}^a$) traffic for all the cluster keys ......................... 47
3.12 K-nearest neighbor entropy method diagram ......................... 49
4.1 Probability comparison for a Blaster worm attack for the excess point method ......................................................... 52
4.2 Probability comparison for a Sasser worm attack for the excess point method ......................................................... 53
4.3 Probability comparison for a Blaster worm attack for the continuous entropy method ...................................................... 55
4.4 Probability comparison for a Sasser worm attack for the continuous entropy method ...................................................... 56
4.5 Centroids of 2 cluster keys .............................................. 60
4.6 Decision regions for the $H^1$ cluster key ............................... 63
4.7 Decision regions for the $H^2$ cluster key ............................... 64
4.8 Decision regions for the $H^3$ cluster key ............................... 64
4.9 Decision regions for the $H^4$ cluster key ............................... 65
4.10 Anomaly detection examples of the 4 cluster key entropy spaces ... 67
4.11 Ultimate detection methodology ........................................ 69
A.1 Excess points graphical statistics of benign points $H^1$ ............ 74
A.2 Excess points graphical statistics of benign points $H^2$ ............ 75
A.3 Excess points graphical statistics of benign points $H^3$ ............ 76
A.4 Excess points graphical statistics of benign points $H^4$ ................. 77
A.5 Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^1$ ................................................................. 78
A.6 Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^3$ ................................................................. 79
A.7 Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^4$ ................................................................. 80
A.8 Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^1$ ................................................................. 81
A.9 Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^2$ ................................................................. 82
A.10 Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^3$ ................................................................. 83
A.11 Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^4$ ................................................................. 84
A.12 Continuous entropy graphical statistics of benign points $H^1$ ............ 85
A.13 Continuous entropy graphical statistics of benign points $H^2$ .......... 86
A.14 Continuous entropy graphical statistics of benign points $H^3$ .......... 87
A.15 Continuous entropy graphical statistics of benign points $H^4$ .......... 88
A.16 Continuous entropy graphical statistics of Blaster worm distances to the benign centroid for $H^2$ ................................................................. 89
A.17 Continuous entropy graphical statistics of Blaster worm distances to the benign centroid for $H^4$ ................................................................. 90
A.18 Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^1$ ................................................................. 91
A.19 Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^2$ ................................................................. 92
A.20 Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^3$ ................................................................. 93
# List of Tables

<table>
<thead>
<tr>
<th>Section</th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Cartesian plane mapping for the 3-D entropy spaces</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Identification of the attacks for the Excess points method</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>Statistics for the benign points distances to the benign centroid ((D^e_k)) for the Excess points entropy method</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>Statistics for the benign points differences to the benign centroid for the Continuous entropy method</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>Likelihood ratios of the Blaster and Sasser worms</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Likelihood ratios of the Blaster and Sasser worms</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Statistics for the excess points distances to the benign centroid of a Sasser worm attack</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>Statistics for the excess points distances to the benign centroid of a Blaster worm attack</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>Statistics for the continuous entropy distances to the benign centroid of a Sasser worm attack</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>Statistics for the excess points distances to the benign centroid of a Blaster worm attack</td>
<td>58</td>
</tr>
<tr>
<td>4.7</td>
<td>Results of the benign and malicious traffic centroids for the destination IP cluster key</td>
<td>59</td>
</tr>
<tr>
<td>4.8</td>
<td>Results of the benign and malicious traffic centroids for the destination Port cluster key</td>
<td>61</td>
</tr>
<tr>
<td>4.9</td>
<td>Results for the (H^1) cluster key anomaly detection for the centroids method</td>
<td>61</td>
</tr>
<tr>
<td>4.10</td>
<td>Results for the (H^2) cluster key anomaly detection for the centroids method</td>
<td>61</td>
</tr>
<tr>
<td>4.11</td>
<td>Results for the (H^3) cluster key anomaly detection for the centroids method</td>
<td>62</td>
</tr>
<tr>
<td>4.12</td>
<td>Results for the (H^4) cluster key anomaly detection for the centroids method</td>
<td>62</td>
</tr>
<tr>
<td>4.13</td>
<td>Results for the (H^1) cluster key anomaly detection</td>
<td>65</td>
</tr>
<tr>
<td>4.14</td>
<td>Results for the (H^2) cluster key anomaly detection</td>
<td>66</td>
</tr>
<tr>
<td>4.15</td>
<td>Results for the (H^3) cluster key anomaly detection</td>
<td>66</td>
</tr>
<tr>
<td>4.16</td>
<td>Results for the (H^4) cluster key anomaly detection</td>
<td>66</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The telecommunication network industry has grown up since its early origins because of the need to be communicated in a big scale and as fast as possible. The importance of communications rests in the concept of information that can be carried from one location to another. The power of information is vital in today’s networked society as it can be translated into development of the one who holds that information.

There has been a great number of advances in technologies and protocols used in networks for both wired and wireless communications aiming to provide higher rates, high availability and achieving low costs for those features. With the evolution of the technologies and protocols, a variety of services that can be more time and cost-effective have been deployed. The use of network oriented services has led to different types of transactions carried out over the network ([1]). In many of those services, the information that is sent to a given destination is personal or sensitive, and therefore there is a need to guarantee the security of the data. Moreover, as the number of networks has exponentially increased, the number of transactions and communications between different network clients is also increasing in big scale and the network security has become a very important issue.

The typical approach of network security based on information technology (IT) involves 3 principal goals described in [1] and [2]:

- Confidentiality. The property of protecting the content of information from all users other than those intended by the legal owner of the information.
- Integrity. The property of protecting information from alteration by unauthorized users.
- Availability. The property of protecting information from non-authorized temporary or permanent withholding of information.

The mechanisms and technologies intended for security purposes in a network are designed attempting to achieve those goals. However, the massive quantity of
networks and the growth of Internet has derived into an increased connectivity, that is many times exploited by malicious intruders that carry out different and (more and more) sophisticated attacks to the integrity of a given network. A series of different types of attacks are reported each year that range from Denial of Service attacks (DoS), viruses, worms, trojan horses, sweeps on ports and/or IP addresses searching for a vulnerability, etc.

Depending on the network in which the attack is deployed, the consequences of that attack being successful in harming the integrity of the system in different aspects can go from lost of vital or personal information to great amount of economic losses. Among the many cases of attacks that succeeded in their attempts is the Code Red worm, which in 2001 infected over 359,000 Internet hosts in less than 14 hours. Another example is the Slammer worm that propagated to over 75,000 hosts in less than 30 minutes as it is mentioned in [1].

The current mechanisms that aim to provide the security needed for the network in which they are deployed can only offer a limited protection against the variety and complexity of new attacks’ threats. The use of preventive mechanisms such as firewalls, anti-virus of authentication protocols help to increase the security of the networks, however their use is not sufficient enough to provide the high levels of security needed. Therefore, the use of Intrusion Detection Systems (IDS) is becoming a practice more and more common in today’s networks administrations.

Many of the current Intrusion Detection Systems schemes are still under development and many others have been proposed to overcome the problem that arises whenever an attacker is successful in its attempt to damage or break into a given network. Among the principal categories of IDS, the anomaly-based types offer good levels of security as they report an alarm whenever a behavior in a network is considered as not normal.

A new generation of anomaly-based IDS that use the concept of entropy has been gaining momentum because of its characteristic of using the entropy as a measurement of disorder in a random variable or process, in this case that process is characterized by traffic’s features. It has been proven that by analyzing the traffic features such as IP packets’ ports and addresses can give a more realistic model of the current traffic behavior than the use of only volumes of that traffic. By analyzing the IP addresses of the incoming packets, information about the usual flow of data in a network can be inferred, and by gathering information about the ports common services utilized can be obtained.

In this thesis, the use of entropy is applied to the intrinsic features of traffic gathered from a network segment following the line of research introduced in [3] and [4] by creating entropy spaces that contain information about the entropy of each of the features acting as a pivot and with the other 3 corresponding to free dimensions.
1.1 Motivation

With the basis of the entropy spaces, 5 methods are proposed aiming to characterize the benign traffic behavior as “Internet Noise” and generate a methodology to detect anomalous behaviors with the entropy information.

The use of entropy as a criterion to provide a characterization of traffic showed good results as attacks’ anomalous behaviors were successfully observed. The methods investigated make use of distances, centroids, statistics and locations of the entropy points found as parameters to judge whether there is an anomalous behavior or not in the current traffic. Each model presents advantages and drawbacks, so an implementation considering the methods described is proposed with the objective to magnify the benefits of each one and minimizing their disadvantages.

1.1 Motivation

Given the fact that the telecommunication networks’ aim is precisely to provide an information exchange between hosts, an increasing opportunity area is found to be the security in networks.

Since their conception, the telecommunication networks have been target of different attacks with diverse objectives aiming to exploit diverse network vulnerabilities with a variety of purposes that can go from leisure to malicious intentions. That is why new mechanisms and systems are studied and proposed aiming to mitigate and help to increase the confiability and security in the present and future networks. Some of those mechanisms can be of preventive type (such as creating secure protocols or authentication) or reactive type (such as network monitoring and the reaction against threats found). The preventive mechanisms help in a big scale to increase the security in networks, however, the networks are still very vulnerable to attacks because of its nature of sharing information, and that is why the reactive mechanisms such as Intrusion Detection Systems (IDS) have great relevance.

The Intrusion Detection Systems have two main categories: those known as signature based and those referred as anomaly based. The main advantage of the anomaly based systems resides in the flexibility that they exhibit in the presence of new (previously unknown) attacks as they report an alert of alarm whenever a current traffic behavior is found to be unusual or suspicious. Given the flexible nature of the anomaly based IDS, a proposed research on this topic is the goal of this thesis project.

1.2 Problem statement

Although an important number of systems’ methods have been proposed (some of them already in the market), to provide security in communication networks, they have not been able to offer very high confiability given the fact that the number
and sophistication of attacks has greatly increased over the last years. Each year, financial losses and information leaks are reported as a consequence of malicious attacks or intrusions. The attackers’ objectives range from universities, companies and other public institutions to personal computers. With the exponential increase in the number of communication networks and systems (the wireless communications area is a clear example), the quantity of users and equipments that could potentially be victims of an attack is also widely increased, so there is a need for new systems that could guarantee a high security level in the available networks.

1.3 Objectives

The objective of the present work is, in a general sense, to develop a methodology based on stochastic and statistical modeling of traffic for an Intrusion Detection System using entropy spaces that could provide trustful results in the anomalous Internet traffic behavior.

1.3.1 Particular objectives

- Generate entropy spaces out of the intrinsic features (source IP, destination IP, source port and destination port) of the packets analyzed using the concept of entropy.

- Create models for the concept of “Internet noise” based on a statistical and stochastic analysis of the entropy spaces created for the Internet packets.

- Generate metrics from which traffic can be classified as normal or anomalous.

- Propose methodologies for anomaly-based Intrusion Detection System based on the models for the Internet noise previously created.

1.4 Contributions

The main contributions of this work are:

- A definition of the Internet noise based on traces of typical traffic gathered from normal working hours.

- The evaluation of different metrics to identify anomalous traffic.

- The definition of the centroid concept in the entropy spaces that provides an evaluation point from which distances of incoming traffic’s entropy can be obtained to determine if there is suspicious behavior.
• The concept of ”Excess points”, from which the anomalous traffic can be classified as malicious according to points exceeding a maximum threshold attained from benign traffic.

• A statistical characterization of the Internet noise based on distance measurements from a benign centroid.

• The $K$-nearest neighbor entropy model formed by the fusion of a pattern recognition technique with the entropy spaces to characterize both normal and anomalous behaviors based on a pattern recognition technique.

• The use of entropy spaces to create methodologies to detect anomalies in a given network segment by using a more practical analysis by means of Shannon entropy.

1.5 Work organization

The organization of this thesis document is the following. In Chapter 2 the background topics related to the main subjects used to obtain the different methodologies for anomaly detection in networks are introduced. The topics covered in this chapter are a brief introduction to the Intrusion Detection Systems architecture and characteristics, a revision of some recent work in IDS based on the concept of entropy to determine theprofiling of traffic, a description of each of the attacks used in this thesis project to test the correct functioning of the methods proposed, and a introduction to the concept of $K$-nearest neighbor classifier rule which is used in to detect patterns in one of the methodologies developed in Chapter 3. In Chapter 3 the description of the 5 methodologies proposed is presented along with relevant information about the concepts used in each of those methods. In Chapter 4 the numerical and qualitative results of the 5 methodologies presented are analyzed once they have been tested with a number of generic attacks. In Chapter 5 the conclusive remarks and future work resulting from the research carried out in this thesis project are presented. Finally, in the Appendix A the graphical results of the Statistical method described in Chapter 3 are presented.
Chapter 2

Fundamentals

In this chapter, the theoretical framework for the analysis carried out is presented. The concepts described here functioned as a baseline to develop the algorithms and methods detailed in the following chapters.

First, an overlook of Intrusion Detection Systems is provided along with their respective characteristics and types. A state of the art revision of the entropy based techniques used for intrusion detection is next explained. A description of the attacks used to evaluate the correct operation of the methods proposed is then introduced. The K-nearest neighbor classifier used for the K-nearest neighbor entropy method derived in Section 3.6 and its principles are also presented.

2.1 Intrusion Detection Systems theory

The Intrusion Detection Systems (IDS) are tools designed with the objective of securing information in communications systems. Since the first works related to this subject appeared, there has been an increasing interest in IDS because as the communication networks have increased in both number and technologies, they are also subject of more and more sophisticated attacks, and therefore there is a tangible need to protect the information. The above is explained in detail in [5], as well as the IDS general characteristics, next described.

Depending on the information source from which an IDS collects its data, the systems can be divided into two main divisions: host based systems or network based systems. The host based IDS are concerned mainly in information related to the host operating systems, while the network based systems acquire information from the network segment in which they are located such as traffic volume, IP addresses, service ports, etc.

Another parameter to classify an IDS is the type of analysis that it carries out. There are two main categories in this matter: signature based or anomaly based systems. The signature based IDS focus on previous information about attacks and
find patterns (signatures) of those attacks in the analyzed data. The anomaly based
IDS attempt to estimate the normal behavior of a network and then generate an alarm
when the current behavior is considered as unusual or anomalous. The signature based
IDS are commonly used in the industry because they provide very good detection
result for specified previously known attacks. The main disadvantage of signature
based attacks rests in the fact that for knew threats, they are not able to detect knew
attacks from which their signatures are not known. On the other hand anomaly-based
IDS are very flexible to adapt to knew threats, as they focus in anomalies, i.e. unusual
behaviors of the traffic. The disadvantage of anomaly based IDS is that they have
higher false positive rates. It is because of the potential advantages that anomaly
based network IDS (A-NIDS) are studied in detail and different approaches have been
proposed in the last years.

In a general way, there are 4 stages in every A-NIDS system:

- Parametrization. In which the observed traffic features are represented in a pre-
established form.

- Training. In which a model for the previous data collected is created.

- Detection. The model obtained is compared to the current traffic’s behavior and
  an alarm is raised if it super passes a given threshold.

Also, depending on the behavioral model derived for the A-NIDS, the systems can
be categorized in 3 main types:

- Statistical based. The A-NIDS creates a model based on statistics, and a stochas-
tic behavior is proposed. For this type of systems there are 3 classes: Univariate
model, Multivariate model and Time Series Model. The main advantage of this
type of IDS is that there is no need to have a previous knowledge on the net-
work segment, as they can estimate the expected behavior of the network out
of observations made. Also, they can provide accurate notification of anomalous
traffic over long time periods. The main drawback of this systems is that they are
susceptible to be trained by the attackers to consider a traffic as normal. Also, its
is very difficult to establish the parameters and metrics for this type of systems
as many optimizations are possible and affecting parameters such as the rate of
true positives also has an effect on others (rates of false negatives for example).

- Knowledge based. The knowledge based systems classify the audit data according
to a set of rules or parameters that have been previously stated by an expert (some
times human). Systems of this type are FSM (Finite State Machine), description
languages and expert systems. The main advantages of this type of systems is
that they are very robust and flexible. The main drawbacks however is, that the
development of high quality knowledge is difficult and time consuming.
• Machine learning based. This type of systems are based on the establishment of a model from which patterns can be classified. In this scheme, the machine can learn from the new information available, so it can adapt to changes in the traffic evolution. The main advantages are then, the flexibility and adaptability but the major drawback is that this systems tend to be highly resource consuming. The principal systems of this type are those based on Bayesian networks, Markov models, Neural networks, fuzzy logic, genetic algorithms and clustering and outlier detection.

Even though there are many IDS techniques proposed, there is a lack of a general test or evaluation method in which one can in a standardized way verify all the aspects related to the proper operation of a given IDS ([6]). Although there has been some efforts to established a complete methodology to evaluate an IDS ([6], [7]), there is not still a recognized evaluation test. However, there are some evaluation indexes that are often used to report and IDS performance related to different aspects of the IDS, such as system function, system performance and system management.

The most basic evaluation indexes related to every IDS system are located among the system function category: False positive rate along with its complementing correct negative rate and false negative rate along with its complementing correct positive rate. The false positive rate is the percentage of false alarms that are reported in the system, while the correct negative rate is the percentage of benign traffic accurately reported as benign. On the other hand the false negative rate is the percentage of events that were reported as benign when they were anomalous, and the correct positive rate is the percentage of anomalous behavior correctly classified as malicious. Then, the false positive rate (FP) is defined by

\[ FP = \frac{N_{FP}}{N_{FP} + N_{CN}} \]  

where \( N_{FP} \) is the number of false positive events, and \( N_{CN} \) is the number of correct negative events. It is clear that the correct negative rate (CN) is simply

\[ CN = 1 - FP = \frac{N_{CN}}{N_{FP} + N_{CN}} \]  

For the false negative rate, we have

\[ FN = \frac{N_{FN}}{N_{FN} + N_{CP}} \]  

where \( N_{FN} \) is the number of false negative events, and \( N_{CP} \) is the number of correct positive events. And for the correct positive rate
With the increase in reported attacks in current networks, new approaches for IDS have been proposed. In recent years, a new generation of IDS based on entropy measurements have been proposed and their approaches are quite varied as it is explained in the next section.

2.2 Entropy based A-NIDS

Among the diverse schemes proposed since the conception of the IDS, in recent years a new generation of systems based on entropy have been introduced and studied. The use of the entropy concept to characterize a network traffic’s behavior has been the aim of many researches given that the entropy captures the essence of the traffic as a random process and its level of uncertainty is described precisely by the entropy. Some of the most recent developments of intrusion detection systems based on the concept of entropy to identify the anomalous behaviors in network segments are described in this section.

In [8] an analysis is carried out extracting the IP and port features and then applying an iterative process in which the less frequent feature’s outcomes according to a threshold are analyzed by obtaining a low bias entropy estimator. The rate of those remnant elements provides a good parameter from which anomalies in a network can be detected.

In [9], a method is proposed in which by taking short term observations of selected features in the network segment’s traffic such as average port, high port, server ports, and peered ports are used. After a series of observations, the average entropy is found for the mentioned features and then, the first and second order time varying statistics are attained for the entropy measurement. The method also uses the commonly used pattern recognition algorithm FLD (Fisher Linear Discriminant) to identify anomalies in the featured statistics.

The use of entropy as a measurement that could help to detect anomalies is reviewed in [10]. The work deals with issues related to the use of entropy as an information theoretic statistic and it also analyzes the use of conditional entropy as a more reliable measure in detecting anomalies.

In [11] the Entropy-based PCA is compared with the Hierarchical heavy hitter algorithm. The aim of the entropy based PCA is to identify uncorrelated changes in the entropies of different packet feature distributions corresponding to different O-D flows. A parameter known as $top_k$ is used to identify the normal and residual
subspaces and the projection of an observation vector onto the residual subspace is used to identify anomalies.

The Shannon entropy is used in [12] to analyze the distribution characteristics of the alerts present in Intrusion Detection Systems. Five attributes are analyzed in the alerts' distribution features: source IP, destination IP, source threat, destination threat and datagram length. Then a method known as Reny cross entropy is applied to fuse the Shannon vector and detect anomalies. The analysis carried out in this work is tested in the Snort Intrusion Detection System.

In [13], an analysis is performed on low-rate Distributed Denial of Service attacks aiming to find an effective method to identify such attacks given the fact that they act in a great scale as normal traffic. In this work two information metrics are proposed: generalized entropy metric and the information metric distance. The generalized entropy metric showed earlier attack detection than the traditional Shannon metric. On the other hand, for the information metric distance proposed showed better results than the commonly used Kullback-Leibler divergence.

In [3] and [4] the use of 3-D entropy spaces by selection is proposed to specify the profiles of a network's traffic. With the use of a data mining process, the IP's an ports features are obtained from traffic flows from a network segment. The modeling of the profiles are then obtained by constructing the spaces of each of the 4 features. For each one of the mentioned clusters there will be a “cluster key” acting as a pivot and 3 free dimensions corresponding to the features other than the cluster key. The works related to the entropy spaces constitute the basis of the featured thesis work as the detection method presented in the next section were developed with the entropy space approach, with some differences explained in detail in Chapter 3.

In the next section, the description of the attacks that were used to test the methodologies proposed in Chapter 3 are presented.

2.3 Attacks’ description

Different attacks have different ways or performing a strike on a given host or network segment. The attacks used to evaluate the methods proposed in this work were chosen as generic classes from which common types of strikes were characterized. A brief description of the attacks used is now presented.

2.3.1 Blaster worm

The description of a Blaster worm attack is provided in [14]. The Blaster worm tries to infect an address based on the local machine’s IP address or a random one and the it attempts to infect continuous IP addresses. There is a 40 % chance that the Blaster
2.3. Attacks' description

The Blaster worm will start at the first address of the "Class C" size subnet (x.x.x.0) and a 60% chance that it will start at a random IP address when a host is infected.

After determining a starting address, the Blaster worm then tries blocks of 20 addresses at a time for hosts with the TCP port 135 (Windows RPC service) open, attempting to connect to each one of the hosts simultaneously. After around 2 seconds have passed, the Blaster worm tries to transmit a payload over the wire to a designated RPC service listening on that port for all the hosts with which a connection was established. If the RPC service is vulnerable to the DCOM buffer overflow, the payload generates a command shell to be bound to the port 444 on the target. The shell will only remain open for one connection and it will be closed after the Blaster worm finishes sending commands.

Once the payload has been issued and a short time interval has passed the Blaster worm assumes that the host is listening on the port 444, and then it sends a command to that host to download a copy of the executable file “msblast.exe” by means of a TFTP. If the download is successful and after waiting around 20 seconds, the TFTP server is shut down and instructions to execute the downloaded file are sent to the host by the worm. Then, the cycle begins once again from the recently infected host. Then each infected host will start to SYN flood windowsupdate.com on port 80.

2.3.2 Sasser worm

The Sasser worm description is found in [15] and it is explained as follows. The Sasser worm takes advantage of the LSA buffer overflow vulnerability to obtain a SYSTEM-level command shell on the victim hosts. Once it is connected to the host, the worm sends a command to download a copy of an executable file via FTP. The first time the worm is installed into a host system, it starts a FTP service on the TCP port 5554 and creates 128 threats that execute in an infinite propagation cycle.

In the propagation loop, the Sasser worm attacks random IP addresses. The worm locates an IP address from the infected host likely to be Internet-routable and it generates a partially or totally random IP address to attempt to infect. For each attempt there is a 52% chance to select a totally random address, a 25% chance that the first 2 octets are taken from the local address, and a 23% that the first octet of the local IP address will be used.

The worm attempts to connect to the TCP port 445 at the generated IP address and if it is successful, the worm sends the LSA exploit and tries to connect to an command shell that must be available on TCP port 9996. If it is successful, the worm issues commands to download and execute an EXE on the infected host.

The description of all the remaining attacks can be found in [16] as they were
obtained from the DARPA dataset available.

2.3.3 Ps

In [16] the description of the ps attack is given. The Ps attack takes advantage of a race condition in the version of ps distributed with Solaris 2.5. The ps attack allows a user to execute an arbitrary code with root user privileges. If the user has access to temporary files, this race condition can then be exploited. Access to temporary files may be obtained if the permissions on the /tmp and /var/tmp directories are set incorrectly. The permissions on the /tmp directory are often reset incorrectly by the system if tmpfs (which is mounting swap as /tmp) is in use.

2.3.4 Sshtrojan

The description of the Sshtrojan is found in [16]. In the SSH Trojan attack the attacker tricks the system administrator to install as an upgrade a trojan version of the SSH program. This trojan version allows anyone with the login previously known to login to the victim, via ssh and without asking for a password. Then, a root privilege shell is spawned for the attacker.

During the setup phase of the attack an email is sent to the system administrator advising that he/she should install a particular compliant version of the ssh server software that they use. The admin then installs the software, thereby replacing the good version of sshd with the trojan version.

During the breakin phase, the attacker uses ssh to login to the victim via ssh as a particular user. The login is successful, does not require a password, and yields a rootshell for the attacker.

2.3.5 Portsweep

The Portsweep reported in [16] is simply identified as a surveillance sweep through many ports to determine which services are supported on a single host.

2.3.6 Ipsweep

The Ipsweep attack reported is explained in [16]. This attack is a surveillance sweep to determine which hosts are listening on a particular network segment. This type of attack is commonly used in attacking hosts as a first search on host that can be vulnerable in a network. A common method was used for this attack, which consists in sending ICMP Ping packets to every possible address within a subnet and observe which hosts respond. The sweeps for the dataset were carried out linearly, quickly and from a single source.
2.3.7 DDoS1
A Distributed Denial of Service gathered from the DARPA dataset has 5 different phases of operation explained in [16].

- In the first phase, the attacker performs a scripted IP sweep of multiple class C subnets. The attacker sends ICMP echo requests in this sweep and waits for ICMP echo-replies to determine which hosts are available.

- In the second stage, the host discovered previously are tested via a ping that makes a rpc request to a host and ask what TCP port number to connect to for the sadmind service. Afterwards the ping connects to the port number received to check if the daemon is listening.

- In the third phase the attacker tries to break into the hosts that it discovered were running the sadmind daemon. The attacker needs to create an entry on the /etc/passwd file and another on the /etc/shadow file. On those files, a new root user name is created as 'hacker2' and its home directory is set to the /tmp folder. If the attack is successful in creating the entries, it moves to the next potential victim.

- In the fourth phase the script of the attack has acquired a list of hosts where the 'hacker2' user has been installed. For each host in that list the attacker performs a telnet login and creates a directory on the victim called /tmp/.mstream/ and uses rcp to copy the server-sol binary into the new directory. The script also installs a .rhosts file in /tmp so they can make a remote shell to startup of binary programs. On the first host in the list, the attacker installs the "master-sol" software, which is the mstream master. Afterwards, the attacker uses rsh to startup first the master, and then the rest of the servers.

- In the last phase, the attacker manually launches the DDOS performed via a telnet on the victim where the master is running. Then, from the victim a telnet is opened to port 6723 (this is the TCP port on which the master listens for connections to its user-interface) of the localhost. Afterwards an attack is launched to a given server present in the list supplied with a 5 second duration. The attack consists in all the servers sending a lot of connection requests to several ports of the victim chosen. All the packets have false and random IP addresses. The attacker then logs out.

2.3.8 DDoS2
The description of this Denial of Service attack is found in [16] where its 5 phases are described as follows.

- The attacker makes use of an HINFO query of the Base’s public dns server. With the information in the DNS server the attacker acquires information about the
hosts which are listed in the servers database and locates victims where the attack will probably work.

- The attacker breaks into the dns server using the sadmin exploit twice, first to add a user in the password file and the other one to create an entry of a user in the shadow-password file.

- The attacker creates a ftp session to the dns server and places a script on it. The attacker also places the mstream server software and the mstream master software.

- The attacker creates a telnet session to the dns server, starts the mstream master software and launches the script that will probe other hosts in the dns records. It then breaks in to those with the desired characteristics.

- The attacker manually launches the DDOS. This is performed via a telnet login to the victim on which the master is running, and then, from the victim, a "telnet" to port 6723 of the localhost. Afterwards an attack is launched to a given server present in the list supplied with a 5 second duration. The attack consists in all the servers sending a lot of connection requests to several ports of the victim chosen. All the packets have false and random IP addresses. The attacker then logs out.

2.4  **K-nearest neighbor algorithm**

The classification of traffic as benign or anomalous using of entropy spaces was found to highly depend on the location of the entropy points in the spaces, so a pattern recognition classifier with which decision regions could be estimated was needed.

There is a great number of classifiers that can be found in literature varying from linear, non-linear, Bayes decision theory based, etc. Among the non-linear classifiers, a subset of classifiers are based on distances to classify a pattern into one class or another. The so called “minimum distance” classifiers present good performance when the patterns of a given class are distributed in the vicinity of one or several vectors that act as representatives for that class. If there are many clusters and therefore vectors representing each class, we are dealing with a multiprototype case as it is explained in [17]. The Figure 2.1 shows an example of such a behavior.

In Figure 2.1 the sets of vectors $X_1$, $X_3$ and $X_5$ correspond to class 1, while the sets of vectors $X_2$ and $X_4$ correspond to class 2.

The K-nearest neighbor classifier is one example of a minimum distance classifier that has been long time studied and its characteristics well described. The K-nearest neighbor classifier is now formally explained. It is detailed in [17], [18] and [19].
If there is a set of m classes \( \{ C_i \}_{i=1}^m \) and a set of n sample patterns \( \{ X_i \}_{i=1}^n \) whose classification is a priori known (training vectors or patterns), for a new incoming pattern \( x \) the K-nearest neighbor classifier rule will classify \( x \) into the class that has the majority of closest vectors among the K nearest. In other words, the K-nearest neighbor algorithm selects the K nearest training vectors to the incoming pattern \( x \), and then assigns \( x \) to the class with the maximum number of closest vectors among the K selected. The \( K-NN \) rule is formally defined as:

\[
g_n(x) = \begin{cases} 
1 & \text{if } \sum_{i=1}^n \omega_{ni} I_{C_i=1} > \sum_{i=1}^n \omega_{ni} I_{C_i=0}, \\
0 & \text{elsewhere,} 
\end{cases}
\]

(2.5)

where \( \omega_{ni} = \frac{1}{K} \) if \( X_i \) is among the K nearest neighbors of \( x \), and \( \omega_{ni} = 0 \) elsewhere. \( I_{\{C_i\}} \) is the indicator function with value \( I_{\{C_i=1\}} = 1 \) if \( X_i \) belongs to class \( i \) and \( I_{\{C_i=0\}} = 1 \) if it does not belong to class \( i \).

If a selection of \( K = 1 \) is used, the algorithm is known as the Nearest neighbor rule. The use of \( K > 1 \) ensures that an incoming pattern \( x \) will be classified according to a majority of training vectors, and in that way the effect of outliers that could be immersed into the area of another class can be avoided. For a \( K-NN \) classifier it is recommended that the value of \( K \) should be odd, to avoid voting ties.

The computational effort for the \( K-NN \) rule depends on the selection of \( K \) and the number of training vectors \( n \). It has been proved ([19]) that the probability of misclassification error probability of the \( K-NN \) rule \( (P_{KNN}) \) is bounded in a two class case by

\[
P_B \leq P_{KNN} \leq P_B + \sqrt{\frac{2P_{NN}}{K}},
\]

(2.6)

where \( P_B \) is the optimal Bayesian error. \( P_{NN} \) is the probability of misclassification for the nearest neighbor rule, which was demonstrated to be bounded by
Thus, by taking a large number of prototype training vectors $N \to \infty$ an higher values of $K$, a performance similar to a Bayesian classifier can be found. However, as the number of prototype vectors increase as well as the value of $K$, this results in a higher computational effort, that is why a compromise is needed for the chosen values taking into account resource considerations.

In Chapter 3, a set of methodologies developed in this work to detect anomaly detection on a number of attacks is introduced. The methods proposed use the concepts that were review in this section to implement the anomaly detection.
Chapter 3

Model

In this Chapter, a description of the methodologies and procedures carried out to obtain intrusion detection in a network segment is explained. The methodologies were developed aiming to work as a support to networks administrators to identify an anomaly occurring in a network segment.

The outline of this Chapter is the following. First, the methodology to obtain the entropy spaces is explained, followed by the description of the models in the chronological order in which they were developed in the pursuit of better results in the anomaly detection. In each section, the methodology for each method is presented, explaining the steps necessary for each approach. The description of the models is divided in two parts corresponding to the training stage and the detection stage of the methodology.

The benign traffic traces used in all the methods described, were gathered from a network segment during normal working hours. The traces of the Blaster and Sasser worm attacks were also obtained from the same segment when the attacks were released. The rest of the attacks' traces were obtain from the highly used DARPA dataset found in [16].

3.1 Entropy Space modeling

The entropy space generation follows in a large degree the procedures proposed in [3] and [4], in which four features obtained from data flow features are used to attain information about the current traffic: Source IP, Destination IP, Source Port and Destination Port, with the difference that the analysis used for the methods derived in this work are obtain in a packet-level, while in the references mentioned, the methods proposed were obtain from a flow-level.

A trace $X$ is obtained from the traffic of a network segment. The trace with complete duration $t$ seconds and $W$ packets is obtained by capturing traffic in $M$ time slots (windows), each with a fixed duration of $t_w$ seconds. Any given window $i$,
will then have a total of \( w_i \) packets (with \( w_i \leq W \)) that correspond to network traffic being reported in that period of time. For every packet of each window, the four header features mentioned are obtained.

Then, for each one of the 4 features, a cluster is defined with a cluster key being assigned to it. The clusters will constitute the entropy spaces denoted \( H_r \) where \( r \) is the cluster key indicator that can take the values \( r = 1, 2, 3, 4 \) for each entropy space. For the traffic to be analyzed in a window \( i \), the entropy spaces then will be \( H_i^r \) and by convention \( H_i^{r-1} \) corresponds to source IP, \( H_i^{r-2} \) corresponds to destination IP, \( H_i^{r-3} \) corresponds to source port and \( H_i^{r-4} \) corresponds to destination port. Also, for every time window an alphabet \( A_r^i = \{a_1^r, a_2^r, ..., a_n^r\} \) is defined as the group of different outcomes that the corresponding cluster key of each entropy space takes for that specific time window. The cluster key is used as a pivot by gathering all the packets that share one of its outcomes (every different value of \( A_r^i \)) as a fixed value, with the other 3 dimensions free to take any value. As an example if an entropy space with source IP cluster key (\( H_1^i \)) is to be obtained from a given time window with packets that contain 5 different source IP’s (\( n=5 \)), the packets will be gathered together into 5 different groups: the fist formed by all the packets that share source IP \( a_1^1 \), the second formed by all the packets that share source IP \( a_2^1 \), and so on. It is clear now that for every time window, groups of packets that share the same cluster key output will be formed regardless of the value that their other 3 features can take (the 3 free dimensions).

Then, for each group sharing the same cluster key an entropy value can be found, for the 3 free dimensions in a time window, i.e., in a time window there will be a total of \( n = |A_r^i| \) entropy values for each of the 3 free dimensions. The definition of entropy as the expected value of the information of a random variable is described in [20], and [21] and it is used to find the entropy spaces. The general entropy formula is given by

\[
H(X) = \sum_{x_i \in X} p(x_i) \log \frac{1}{p(x_i)}. \tag{3.1}
\]

For each value of the alphabet \( (a_c^r) \), there is a set of different outcomes in each of the free dimensions \( S_j^c = \{s_1^j, s_2^j, ..., s_l^j\} \) with \( c=1, 2, ..., n \) being the alphabet outcome indicator, and \( l = |S_j^c| \) outcomes. The probabilities associated with each element of the set \( S_j^c \) can be computed then as

\[
p(s_j^c) = \frac{f_{s_j^c}}{f_{a_c^r}} \tag{3.2}
\]

where \( f_{s_j^c} \) is the frequency observed for the \( s_j^c \) outcome of \( S_j^c \) and \( f_{a_c^r} \) is the frequency of the alphabet outcome \( a_c^r \). The entropy values are found then as
3.1. Entropy Space modeling

The entropy spaces points will be then formed by associating the entropy values found in the three free dimensions \( k = \{1, 2, 3\} \) for each of the \( c = 1, 2, \ldots n \) alphabet outcome with a 3-D coordinate value in an axis of the Cartesian plane

\[
(x_c, y_c, z_c) = (H_{c,k=1}(S^c_j), H_{c,k=2}(S^c_j), H_{c,k=3}(S^c_j)).
\]  \hspace{1cm} (3.4)

The entropy spaces of the time window \( i \) will be formed with all the 3-D entropy points found

\[
H_i^c = \left[ \begin{array}{ccc}
(H_{1,1}(S^c_j), & H_{1,2}(S^c_j), & H_{1,3}(S^c_j)) \\
(H_{2,1}(S^c_j), & H_{2,2}(S^c_j), & H_{2,3}(S^c_j)) \\
\cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots \\
(H_{n,1}(S^c_j), & H_{n,2}(S^c_j), & H_{n,3}(S^c_j))
\end{array} \right].
\]  \hspace{1cm} (3.5)

In Table 3.1, the cluster keys and the entropy values mapping into the Cartesian plane are showed.

<table>
<thead>
<tr>
<th>Cluster key</th>
<th>Free dimensions</th>
<th>X axis</th>
<th>Y axis</th>
<th>Z axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src IP ((r = 1))</td>
<td>Src Prt</td>
<td>Dst Prt</td>
<td>Dst IP</td>
<td></td>
</tr>
<tr>
<td>Dst IP ((r = 2))</td>
<td>Src Prt</td>
<td>Dst Prt</td>
<td>Src IP</td>
<td></td>
</tr>
<tr>
<td>Src Prt ((r = 3))</td>
<td>Src IP</td>
<td>Dst IP</td>
<td>Dst Prt</td>
<td></td>
</tr>
<tr>
<td>Dst Prt ((r = 4))</td>
<td>Src IP</td>
<td>Dst IP</td>
<td>Src Prt</td>
<td></td>
</tr>
</tbody>
</table>

The 4 benign entropy spaces found with the described method for traces obtained from traffic that was know to be attack-free, gathered from a network segment in
3.1. Entropy Space modeling

$M=1189$ windows ($H^r = H^r_1 \cup H^r_2 \ldots \cup H^r_{M=1189}$), with $t_w = 60$ seconds in a 5 day period are showed in figure 3.1. The traffic traces were collected in normal working hours.

![Entropy Spaces](image)

(a) Source IP ($H^1$) entropy space  
(b) Destination IP ($H^2$) entropy space  
(c) Source Port ($H^3$) entropy space  
(d) Destination Port ($H^4$) entropy space

Figure 3.1: Entropy spaces for the 4 cluster keys

The entropy spaces found can then be used to apply different methodologies that are described in the next sections in the chronological order that there were analyzed.
3.2 Excess points method

The excess points method consists in finding a centroid for the benign traffic, and then characterizing it by the distances of the normal traffic points to that benign centroid. After the normal traffic characterization, the anomaly traffic is characterized by taking the anomaly points distances to the benign centroid previously found that exceed a maximum distance found for the benign traffic (excess points). A probability density function can then be found for both the benign and the anomalous traffic distances to the benign centroid with which one can obtain a decision region that will determine if a given window can be declared as having a normal or an anomalous behavior. The steps that describe the excess point method are the following:

3.2.1 Training stage

- **STEP T1.** One must chose a number of days $N$ in which Internet traces in the network segment to be analyzed will be gathered to have reliable information about the normal behavior or the network. There is a need to make a compromise in this matter, as having lots of information (many days of collected data) will have the effect of finding a more accurate overall behavior of the network but it will consume lots of memory and therefore time to analyze the traffic. On the other hand, having few training days for the data will result in a less accurate modeling of the network but will consume little memory which will help to have a faster system. The proposed period to train the data is 1 to 2 weeks. The model will be found then each $N$ days, to adapt to changes in the network segment changes.

- **STEP T2.** A window time of $t_w$ seconds is declared. This period of time will be the basic unit for capturing packets in every trace. Just as in the number of days, there must be a compromise in choosing the window size as it must be sufficiently large to gather enough information about the trace, and it should be sufficiently small to have a faster response when an anomaly appears in the network segment being analyzed. The data captured in this time windows will constitute the actual training data for the IDS. In this method, a window size of $t_w = 60$ seconds was used.

- **STEP T3.** Traffic from the $N$ days is captured in time windows, each one with duration of $t_w$ seconds. A total of $M$ time windows containing traffic over the $N$ days will be obtained. Then a data mining process must be applied to extract the information about the packet headers in each window time. Although there are other parameters that could be exploited to have a different analysis, in this system the features that will be used will be those regarding computer information (IP’s) and service information (ports).
3.2. Excess points method

- **STEP T4.** Using the method described in Section 3.1, the entropy spaces $H^r_i$ must be found for each cluster key, and for every $i$-th window obtained in the $N$ days from which the data was gathered, such as those depicted in Figure 3.1 for $N = 5$ days.

- **STEP T5.** The next step consists in finding the benign centroid $(c^r = (x_{cn}, y_{cn}, z_{cn}))$ from the captured data. The benign centroid will constitute the representative point that will help to differentiate a benign from a malicious traffic. To obtain the centroid of the captured data, a method based on “bins” $b_j$ is proposed.

The method consists in dividing the whole entropy space $H^r_i$ of each cluster key, into $m$ smaller cubes (bins) of side length $s$ entropy units (according to the logarithm base, the units may be different). The value of $s$ must be chosen between certain range, as it must be small enough to preserve the behavioral characteristics of the traces and it must be big enough to reach an appropriate centroid. Once a side length $s$ is chosen, the number of bins in which each entropy space will be divided to is given by

$$m = \left\lceil \frac{\max(x_c)}{s} \right\rceil \left\lceil \frac{\max(y_c)}{s} \right\rceil \left\lceil \frac{\max(z_c)}{s} \right\rceil .$$

(3.6)

After choosing $s$, an average is obtained from all the points that fall into the volume ($s^3$) of each bin found. The average of each bin will function as a representative point for that bin

$$(x_b, y_b, z_b) = \frac{1}{J} \sum_{u=1}^{J} (x_u, y_u, z_u), \quad \text{for } b = 1, 2, \ldots, m,$$

(3.7)

where $J$ is the total number of points that fall within the $s^3$ volume.

Once the representative points for all the bins, in which the space is divided ($m$) have been obtained, an average of all these representative points must be found and it will become the centroid $(c^r_i)$ for each $i$th window

$$c^r_i = (x_i, y_i, z_i) = \frac{1}{m} \sum_{b=1}^{m} (x_b, y_b, z_b), \quad \text{for } i = 1, 2, \ldots, M.$$  

(3.8)

Finally, the benign centroid is found by calculating an average of all the windows’ ($M$) centroids.
3.2. Excess points method

\[ c^r = (x_{cn}, y_{cn}, z_{cn}) = \frac{1}{M} \sum_{i=1}^{M} c^r_i. \] (3.9)

The centroids must be found for each cluster key. In figures 3.2(a) and 3.2(b), it can be seen that the entropy space resulting from a Blaster worm attack is easily distinguished from the benign traffic entropy space as there are many excess points.

The reason for using this procedure (the use of bins) to find the benign centroid follows the fact that the normal behavior of the traffic inside the network analyzed reflects many entropy points which are very near the space borders. Having that many points gathered into specific areas has the effect of pulling the centroid near that area and as a result of that, the centroid found will not be a good representation of the whole extension of the entropy points. With this procedure, a centroid closer to the middle point of the range of the space points in every dimension can be found.

**STEP T6.** Once the benign centroids are found for each cluster key, the distances of all the points \( P \) in each of the entropy spaces \( H^r_c \) to the benign centroid can be found using the Euclidean distance formula

\[
\delta_p = \sqrt{(x_{cn} - x_p)^2 + (y_{cn} - y_p)^2 + (z_{cn} - z_p)^2},
\]

for \( p = 1, 2, ..., P \). (3.10)

The set of benign points’ distances to the benign centroid \( D_b = \{\delta_{b1}, \delta_{b2}, ..., \delta_{bP}\} \) is then obtained.

**STEP T7.** The next step in the excess point method consists in obtaining the anomalous traffic distances from the anomalous points to the benign centroid. The entropy points are obtained for traffic traces known to be anomalous using the same method used for the benign traffic. Then, the malicious distances set \( D_m = \{\delta_{m1}, \delta_{m2}, ..., \delta_{mT}\} \) corresponding to the \( T \) anomaly points distances to the benign centroid.
3.2. Excess points method

Figure 3.2: Entropy space identification of $H^1$ for a Blaster worm attack

(a) Benign traffic entropy space for Src IP

(b) Blaster worm attack traffic entropy space for Src IP
3.2. Excess points method

- **STEP T8.** Having found the distances sets for the benign \( D_b \) and anomalous traffic \( D_m \) entropy points, elements of Bayes decision theory can be used to provide what is known as a Bayesian classifier with which a traffic trace can be classified as having normal or unusual behavior. The rules of the Bayes decision theory, explained in more detail in [17], [19] and [21] are then used.

In this methodology, there are 2 classes defined \( C_i \) with \( i = \{1, 2\} \), corresponding to benign (\( C_1 \)) and anomalous (\( C_2 \)) class. With the event of an input entropy point \((x_p, y_p, z_p)\), its distance to the benign centroid \( x = \|(x_{cn}, y_{cn}, z_{cn}) - (x_p, y_p, z_p)\| \) and the a priori probabilities of the incoming point \( p(x) \) and \( p(C_i) \), the Bayes classifier rule is applied as follows

\[
\text{if } P(C_1|x) > P(C_2|x), \quad x \to C_1, \\
\text{if } P(C_1|x) < P(C_2|x), \quad x \to C_2.
\]  

(3.11)

By using the channel probabilities \( P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)} \), Equation (3.19), can be expressed as

\[
\text{if } P(x|C_1)P(C_1) > P(x|C_2)P(C_2), \quad x \to C_1, \\
\text{if } P(x|C_1)P(C_1) < P(x|C_2)P(C_2), \quad x \to C_2.
\]  

(3.12)

Then, channel probabilities \( P(C_i|x) \) can be associated with probability density functions obtained for the benign an anomalous entropy points distances

\[
P(C_1|x) = f_{D_b}(\delta_b), \\
P(C_2|x) = f_{D_e}(\delta_e),
\]  

(3.13)

where \( f_{D_b}(\delta_b) \) corresponds to the probability density function of the benign distances, and \( f_{D_e}(\delta_e) \) corresponds to the distances of the excess points distances to the benign centroid. The excess points, as it was stated in the initial description of the method, are those points from the malicious distances set that exceed a maximum benign distance threshold \( t_h \).

The probability density function of the benign traffic distances \( f_{D_b} \) must be obtained. The method used for the probability density function estimation was the Kernel smoothing algorithm, which is described in detail in [22]. After applying
3.2. Excess points method

the Kernel Smoothing algorithm the probability density function describing the distances from the benign points to the centroid is found, satisfying

\[ P[\delta_{b1} < D_b < \delta_{b2}] = \int_{\delta_{b1}}^{\delta_{b2}} f_{D_b}(\delta)d\delta_b. \] (3.14)

Figure 3.3(a) shows a histogram of the distances of the normal points to the src IP benign centroid, and 3.3(b) shows a probability density function found with the kernel estimation for the destination IP and also a normal probability density function with the mean and variance of the distance set \( D_b \).

- **STEP T9.** Then, for the anomaly traffic, the excess points distance set \( D_c = \delta_{e1}, \delta_{e2}, \ldots, \delta_{eE} \) is found by taking all the distances from the malicious set \( D_m \) that exceed the maximum benign distance threshold \( t_h \)

\[ D_c = \{\delta_{e1}, \delta_{e2}, \ldots, \delta_{eE}\} \subseteq D_m \mid \delta_{ej} > t_h, \ \forall j. \] (3.15)

To obtain that threshold, a method that consists in obtaining a likelihood ratio (LR) is proposed. The likelihood ratio (LR) is obtained by taking the total number of malicious point’s distances exceeding the maximum benign distance \( (N_e) \) and dividing it by the total number of malicious points
3.2. Excess points method

\[ LR = \frac{N_e}{T}. \] (3.16)

Then, after an empirical analysis, the attained \( LR \) keeps its value if this value is under 0.01, but it is changed to 0.01 if the found ratio exceeds that value.

\[
\begin{align*}
\text{if } LR \leq 0.01 & \rightarrow LR = LR, \\
\text{if } LR > 0.01 & \rightarrow LR = 0.01.
\end{align*}
\] (3.17)

The benign distance threshold is given by the percentile(1 – \( LR \)) of the benign distances probability density function previously obtained.

\[ t_h = P_{1-LR}(f_{De}). \] (3.18)

Next, the probability density function for the excess points distance set \( f_{De} \) can be obtained.

- **STEP T10.** Having obtained both probability density functions for the benign and excess points distances mentioned in Equation (3.13), the Bayes decision rule can be applied to determine if a given entropy point should be classified as benign or anomalous. The system’s training is completed at this stage, and the detection phase is next performed.

### 3.2.2 Detection stage

- **STEP D1.** Captured traces from current traffic need to be obtained. The traces are gathered just as in the training period in windows of \( t_w \) seconds. For every window there will be a captured trace \( Y \).

- **STEP D2.** Information about the trace packet headers’ features is retrieved.

- **STEP D3.** The entropy spaces for the current window are obtained for every cluster key \( H_Y^r \) for \( r=1,2,3,4 \) using the same methodology described in Section 3.1.
• **STEP D4.** A new set of distances from each of the entropy space points to the benign centroid is found: $D_Y = \{\delta_1, \delta_2, ..., \delta_n\}$ for every cluster key.

• **STEP D5.** Every point in the set $D_Y$ can be classified as benign or anomalous by using the Bayes decision Rule considering with the probability density functions found in the training stage

\[
\text{if } f_{D_b}(\delta) > f_{D_c}(\delta), \quad Y \rightarrow \text{benign}, \\
\text{if } f_{D_b}(\delta) < f_{D_c}(\delta), \quad Y \rightarrow \text{anomalous}. \quad (3.19)
\]

In Figure 3.4, the Excess points methodology previously described is summarized.

![Figure 3.4: Excess points method diagram](image-url)
Performing the excess points method results in a very clear differentiation of some attacks, however for other types of attacks it was impossible to find conclusive statistics that could reveal an attack. The above occurred for example in a Sshtrojan attack, in which the complete set of anomalous points were located within the volume of the benign traffic, as depicted in figures 3.5(a) and 3.5(b). In that case, there were no excess points.

In figures 3.5(a) and 3.5(b) it can be seen that the excess points method fails. The later is because, as it was discovered after a series of tests ran on different types of attacks, that some of those attacks showed entropy space points immersed inside the benign traffic area, which means that there were no excess points for those attacks. However it is also shown that most of the attack points are situated in an area where there are only few benign points.

The excess points method showed good results for the attacks related to Sasser and Blaster worms but it was inefficient for other types of attacks where there were no excess points, as it can be seen in Table 3.2.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Total points</th>
<th>Excess points</th>
<th>Attack detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sasser worm</td>
<td>117495</td>
<td>54</td>
<td>✓</td>
</tr>
<tr>
<td>Blaster worm</td>
<td>984</td>
<td>94</td>
<td>✓</td>
</tr>
<tr>
<td>ps</td>
<td>2952</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>236</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>ipsweep</td>
<td>270</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>DDOS</td>
<td>34190</td>
<td>0</td>
<td>x</td>
</tr>
</tbody>
</table>

The results for the probability densities estimations for both the benign and malicious attacks are presented in one against each other in Chapter 4.

In Section 3.3, the continuous entropy method proposed is obtained, seeking for better results in the identification process than those found for the Excess points method.

3.3 Continuous entropy method

The continuous entropy method is very similar to the Excess points method explained in Section 3.2, as a full characterization of the benign and malicious points' distances to a benign centroid is obtained first, and then, those distances are described by means of probability density functions. The main difference between the Excess points method and the Continuous entropy method is that the entropy values are not obtained to
3.3. Continuous entropy method

(a) Benign traffic entropy space for Dst IP

(b) Ssh trojan attack traffic entropy space for Dst IP

Figure 3.5: Entropy space identification of $H^2$ for a ssh trojan attack
3.3. Continuous entropy method

form 3-D entropy spaces. Instead, for each packet header feature, an entropy value is found using the feature’s probabilities, regardless of the other features results. The Continuous entropy method finds a set of entropy points found for continuous time windows (hence the Continuous entropy name). The methodology for the Continuous entropy method is described next.

3.3.1 Training stage

- **STEP T1.** A number of days $N$ in which attack-free Internet traces in the network segment to be analyzed is chosen with the same compromise about the quantity of information needed and the memory consideration mentioned before.

- **STEP T2.** A window time of $t_w$ seconds is declared. Once again, a compromise between the amount of information and the memory restrictions is needed.

- **STEP T3.** A data mining process must be applied to the gathered information from the 4 packet headers’ features in each window time. A total of $M$ windows collected from the $N$ days.

- **STEP T4.** Next in the Continuous entropy method, for each window $i$, the entropy values are found for each feature. The same convention as in the Excess points method is used now for the 1-D entropy spaces of the Continuous entropy method: $H_i^r$, with $r = 1, 2, 3, 4$.

To obtain the 1-D entropy points the formula for the entropy (Equation 3.1) is used. The probabilities for each feature in a given window $i$ are found by defining the alphabet $A_i^r = \{a_1, a_2, ..., a_j, ..., a_n\}$ formed by each different feature value in that window. Once the alphabet for that window is determined, the probabilities $p(a_j)$ of each different feature’s value $a_j$ are obtained dividing the frequency of each alphabet outcome by the total frequency $f_{A_i^r}$ observed in that window

$$p(a_j) = \frac{f_{a_j}}{f_{A_i^r}} = \frac{f_{a_j}}{\sum_{i=1}^{n} f_{a_i}}. \quad (3.20)$$

An entropy value then will be found for each window and for every entropy space $H_i^r = (x_i, y_i, z_i)$.

- **STEP T5.** After obtaining the entropy points over the $M$ time windows, the centroid $c'$, corresponding to the expected value of each entropy space points (the expected entropy value of each feature)
3.3. Continuous entropy method

\[ c^r = \sum_{i=1}^{M} H_i^r. \]  

- **STEP T6.** After obtaining the centroids for each cluster key, a new parameter set \( R_b = \{r_{b1}, r_{b2}, ..., r_{bM}\} \) to be analyzed is obtained by taking the difference between each benign point and its centroid

\[ r_{bi}^r = H_i^r - c^r, \]

for \( i = 1, 2, ..., M. \) (3.22)

Given the fact that the new parameter is obtained by taking the difference (not the distance) of each point to its centroid, the set \( R_b \) will take both, positive and negative values. Figure 3.6 shows the entropy parameters found for each cluster key.

- **STEP T7.** To use the rule of a Bayes classifier stated in Equation 3.19, a probability density function \( (f_{Rb}(r_b)) \) must be obtained for the benign traffic entropy space parameter set \( R_b \). The Kernel smoothing algorithm was used to achieve the above. Figure 3.7 shows the probability density functions found for the benign traffic entropy parameter of each cluster key.

- **STEP T8.** Malicious data traffic is then analyzed and the set \( R_m = \{r_{m1}, r_{m2}, ..., r_{mT}\} \) attained by taking the difference between all the \( T \) malicious points \( Ha_j^r \) and the benign centroid \( (c^r) \).

\[ r_{mj}^r = Ha_j^r - c^r, \]

for \( j = 1, 2, ..., T. \) (3.23)

- **STEP T9.** The concept of exceeding points is also applied in the continuous entropy method. First, the likelihood ratio LR is attained, and then benign threshold \( t_h \). As there will be positive and negative points for the parameter set \( R \), the exceeding parameter points set \( (R_e) \) will be formed by all the parameter points in \( R_m \) with values below a low threshold \( t_{hL} \), or above a high threshold \( t_{hH} \)

\[ R_e = \{r_{e1}, r_{e2}, ..., r_{ej}, ...r_{eE}\} \subseteq R_m \mid r_{ej} < t_{hL} \lor r_{ej} > t_{hH} \forall j. \]  

(3.24)
The Likelihood ratio for the continuous entropy method are found by taking the total number of points \( N_e \) that exceed the maximum benign parameter set value or that are below minimum benign parameter set and dividing it by the total number of malicious points \( T \).

\[
LR = \frac{N_e}{T}, \tag{3.25}
\]
3.3. Continuous entropy method

Just as in the excess point method, the likelihood ratio will remain the same if its value is below 0.01, but if it is above that, its original value will be changed to 0.01, as it is stated in Equation (3.17).

The thresholds are found by taking the percentile \( \left( \frac{LR}{2} \right) \), and \( 1 - \left( \frac{LR}{2} \right) \) respectively.
3.3. Continuous entropy method

\[ t_{hL} = P_{\frac{LR}{2}}(f_{\bar{R}b}), \quad (3.26) \]
\[ t_{hH} = P_{1-\frac{LR}{2}}(f_{\bar{R}b}). \quad (3.27) \]

- **STEP T10.** The probability density function must be found for the anomalous exceeding traffic \( f_{\bar{R}e}(r_e) \). Once again, the Kernel smoothing algorithm was used to accomplish the probability density estimation.

Having the probability density functions for the benign and exceeding anomalous traffic, the method is now properly trained, and the detection stage follows.

3.3.2 Detection stage

- **STEP D1.** A new trace of current traffic are collected every \( t_w \) seconds.

- **STEP D2.** The extraction of the packet headers’ information is applied.

- **STEP D3.** The entropy spaces \( H_Y \) are obtained with the same methodology described in Section 3.1.

- **STEP D4.** The set of differences between the current entropy spaces points \( H_Y \) to the benign centroid is found \( (R_Y = \{r_1, r_2, ..., r_n\} \text{ for } r=1,2,3,4) \).

- **STEP D5.** Each element of \( R_Y \) can be classified then using the Bayesian rule

\[
\begin{align*}
\text{if } f_{Rb}(r) > f_{R_e}(r), & \quad Y \rightarrow \text{benign}, \\
\text{if } f_{Rb}(r) > f_{R_e}(r), & \quad Y \rightarrow \text{anomalous},
\end{align*}
\]

(3.28)

The summarized Continuous entropy methodology is shown in Figure 3.8. The Continuous entropy method also failed in detecting anomalies for other types of attacks different from the Blaster and Sasser worms, as all of the points of those attacks where located within the boundaries of the benign traffic entropy parameter’s maximum and minimum points. The results of the Continuous entropy method showing the probability density functions obtained for benign and anomalous traffic are shown, and their effectiveness is discussed.
In Section 3.4, the Statistical methodology is presented. The Statistical method was proposed as a supporting tool to the other methodologies, seeking to provide a second level of characterization for the traffic.

3.4 Statistical method

The statistical method was proposed with the purpose of enhancing the detection rate effectiveness to the other methodologies proposed. The Statistical method consists in finding a set of statistics such as mean, variance, standard deviation, median, variance mean factor, kurtosis and skewness. Also, for a graphical identification, the use of Q-Q plots, autocorrelation and power spectral density are proposed to help the system administrator to identify an anomalous behavior.
3.4. Statistical method

- After the Excess points method or the Continuous entropy method described in sections 3.2 and 3.3 respectively, has been applied, the statistics of the parameters are evaluated for both benign an anomalous traffic. In the case of the Excess points method, the statistics are found for the benign and the anomalous entropy points distances to the benign centroid. In the case of the Continuous entropy method, the statistics are found for the parameter of the differences between the points to their benign centroid.

- The statistics for the benign points to the benign centroid for the Excess points method are presented in Table 3.3.

Table 3.3: Statistics for the benign points distances to the benign centroid ($D_y^b$) for the Excess points entropy method

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>2.5010</td>
<td>3.5882</td>
<td>2.6177</td>
<td>3.2426</td>
</tr>
<tr>
<td>variance</td>
<td>0.2821</td>
<td>0.1141</td>
<td>0.3115</td>
<td>0.0705</td>
</tr>
<tr>
<td>var/mean</td>
<td>0.1128</td>
<td>0.0318</td>
<td>0.1190</td>
<td>0.0218</td>
</tr>
<tr>
<td>standard deviation</td>
<td>0.5311</td>
<td>0.3377</td>
<td>0.5581</td>
<td>0.2656</td>
</tr>
<tr>
<td>median</td>
<td>2.6791</td>
<td>3.6468</td>
<td>2.6788</td>
<td>3.2851</td>
</tr>
<tr>
<td>kurtosis</td>
<td>10.3060</td>
<td>40.2642</td>
<td>17.1463</td>
<td>68.4854</td>
</tr>
<tr>
<td>skewness</td>
<td>1.3436</td>
<td>-0.2449</td>
<td>1.7064</td>
<td>1.6111</td>
</tr>
<tr>
<td>max value</td>
<td>7.1620</td>
<td>8.2256</td>
<td>8.0080</td>
<td>7.9762</td>
</tr>
<tr>
<td>min value</td>
<td>0.2675</td>
<td>0.1322</td>
<td>0.0679</td>
<td>0.1883</td>
</tr>
</tbody>
</table>

- The statistics found for the benign differences to the benign centroid ($R_y^b$) found for the Continuous entropy method are showed in Table 3.4 for the 4 features (cluster keys).

In Chapter 4, the resulting statistics for the anomalous traffic are compared with the values found for the benign traffic for both the Excess point method and the Continuous entropy method.

In Appendix A the graphics of histogram, Q-Q- plot, autocorrelation and power spectral density from the benign and malicious points’ distances to the benign centroid found for both, the Excess points and the Continuous entropy methods are presented.

In the next Section, the Centroids method is explained. The Centroids method was derived from the previous analysis aiming to find a more robust result in the anomaly traffic detection by taking the cumulative centroid of a number of windows to obtain information about the behavioral evolution of a trace.
3.5 Centroids method

The Centroid method was derived given the fact that, results from experiments showed that taking the cumulative centroid of a number of windows, the benign traffic converged to a point that was clearly distant enough from anomalous traffic’s centroid obtained in the same way. Thus, by finding the cumulative centroid and combining that information with the current distance of the current window centroid, a better detection is attained. Also, the centroids method provides an adaptive approach that lies in the fact that information about current and previous traffic states is combined to generate a detection. The methodology for the Centroids method is described next.

### 3.5.1 Training stage

- **STEP T1.** A number of days $N$ in which attack-free Internet traces in the network segment to be analyzed is collected having in mind the compromise between the quantity of information that can be obtained and the memory aspects.

- **STEP T2.** A window time of $t_w$ seconds is declared with sufficient length to gather enough behavioral information and not very large so a faster detection can be accomplished.

- **STEP T3.** The data mining process is applied to the gathered information from the 4 packet headers’ features in each window time. A total of $M$ windows collected from the $N$ days.

Table 3.4: Statistics for the benign points differences to the benign centroid for the Continuous entropy method

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>-0.1797 e-14</td>
<td>-0.5604 e-14</td>
<td>-0.5051 e-14</td>
<td>0.0176 e-14</td>
</tr>
<tr>
<td>variance</td>
<td>1.6928</td>
<td>0.8668</td>
<td>1.0026</td>
<td>1.6524</td>
</tr>
<tr>
<td>var/mean</td>
<td>-0.9419 e15</td>
<td>-0.1547 e15</td>
<td>-0.1985 e15</td>
<td>9.3732 e15</td>
</tr>
<tr>
<td>standard deviation</td>
<td>1.3011</td>
<td>0.9310</td>
<td>1.0013</td>
<td>1.2855</td>
</tr>
<tr>
<td>median</td>
<td>0.2949</td>
<td>0.0038</td>
<td>0.1690</td>
<td>0.1817</td>
</tr>
<tr>
<td>kurtosis</td>
<td>4.7187</td>
<td>5.3113</td>
<td>6.6737</td>
<td>6.0060</td>
</tr>
<tr>
<td>skewness</td>
<td>-1.2959</td>
<td>-1.0077</td>
<td>-1.6161</td>
<td>-1.4486</td>
</tr>
<tr>
<td>max value</td>
<td>2.2392</td>
<td>1.9128</td>
<td>2.0711</td>
<td>2.3777</td>
</tr>
<tr>
<td>min value</td>
<td>-4.5029</td>
<td>-3.8954</td>
<td>-4.2430</td>
<td>-5.1966</td>
</tr>
</tbody>
</table>
3.5. Centroids method

- **STEP T4.** Then, for each window $i$, the entropy values are found for each feature. The same convention as in the Excess points and Continuous entropy methods method is used entropy spaces of the Centroids entropy method: $H_i^r$, with $r = 1, 2, 3, 4$. Using the method described in Section 3.1, the entropy spaces $H_i^r$ must be found for each cluster key, and for every $i-th$ window obtained in the $N$ days from which the data was collected.

- **STEP T5.** The benign centroid must be attained. First, a benign centroid $(c_i^r = (x_i, y_i, z_i))$ for each of the $M$ window is obtained by using the bins method described in Section 3.2 for the Excess points method.

Then, the benign centroid $c^r = (x_{cn}, y_{cn}, z_{cn})$ is found by taking the average of all the windows’ centroids.

$$c^r = (x_{cn}, y_{cn}, z_{cn}) = \frac{1}{M} \sum_{i=1}^{M} c_i^r. \tag{3.29}$$

After a series of simulations, it was found that the centroid of the benign traffic converges to a point that is clearly distinguishable from the point to which the attacks’ centroid converge, therefore the classification by means of cumulative centroids over a set of windows is proposed. The results of centroids locations are presented in Chapter 4. Figure 3.9 shows 4 different examples of the convergence of the cumulative centroid for different attacks.

Next, the training and the detection stages are merged together, as information about previous data (obtained from the training stage) is needed for the detection stage in the current traffic and also, information about the current traffic is used to find parameters needed to train the data for later windows anomalies’ detection.

- **STEP T6.** When a centroid is found for a new time window in the detection stage (STEP D4) there will be a cumulative centroid $c_{cum}^r = (x, y, z)$ obtained by averaging the previous windows centroids. The cumulative centroid then for the $j - th$ window analyzed in the current traffic will be

$$c_{cum}^r = \frac{1}{j} \sum_{k=1}^{j} c_k^r. \tag{3.30}$$
3.5. Centroids method

Figure 3.9: Centroid convergence for 4 different attacks

- **STEP T7.** The distance \( \delta_{cum}^r \) of the cumulative centroid in that window to the benign centroid \( c^r \) needs to be computed as

\[
\delta_{cum}^r = \| (x_{cn}, y_{cn}, z_{cn}) - (x, y, z) \|,
\]

\[
= \sqrt{(x_{cn} - x)^2 + (y_{cn} - y)^2 + (z_{cn} - z)^2}. \tag{3.31}
\]
The distance from the cumulative centroid to the benign centroid will be the parameter to be evaluated to determine if a window is to be classified as normal or anomalous, depending if it is below or above a certain threshold, next described.

**STEP T8.** A sliding window algorithm is proposed to storage information about previous centroids and distances. The sliding window size \((S_w)\) will be used to obtain an average of the last \(S_w\) windows’ centroids distance to the benign centroid. That average \((\hat{p})\) of the last windows distances is then found with

\[
\hat{p} = \frac{1}{S_w} \sum_{k=j-(S_w-1)}^{j} \delta_{Y_k},
\]

(3.32)

The steps T6 to T8 are then carried out for every new incoming traffic window.

### 3.5.2 Detection stage

**STEP D1.** For each \(j\) window from the current traffic in the network, a new trace \(Y_j = (x_j, y_j, z_j)\) is obtained.

**STEP D2.** A data mining process is carried out to obtain the intrinsic features of the packets headers for the new trace.

**STEP D3.** The space entropy for that window \((H_{Y_j})\) is obtained by using the methodology described in Section 3.1.

**STEP D4.** The centroid for that window is obtained by using the bins method described in section 3.2.

**STEP D5.** The current window distance \((\delta_{Y_j})\) to the benign centroid needs to be attained as

\[
\delta_{Y_j} = \|(x_{cn}, y_{cn}, z_{cn}) - (x_j, y_j, z_j)\|,
\]

\[
= \sqrt{(x_{cn} - x_j)^2 + (y_{cn} - y_j)^2 + (z_{cn} - z_j)^2}.
\]

(3.33)
3.6. *K*-nearest neighbor entropy method

**STEP D6.** The threshold for the Centroids method was derived empirically taking into account not only the distance of the current window \((\delta_{Y_j})\), but also the average of the previous windows \((\bar{p})\) aiming to avoid the presence of too many false positives resulting from sporadic benign traces having distant current windows distances to the benign centroid. Good results were observed when the factor given by the current distance had a weight of twice the factor given by the average of the previous \(S_w\) windows distances to the benign centroid. The threshold obtained was then

\[
t_h = \frac{1}{2} \left( \delta_{Y_j} + \frac{1}{2} \bar{p} \right).
\]  

(3.34)

Finally, a trace is classified as "benign" if the distance of its cumulative centroid distance \((\delta_{cum})\) to the benign centroid is below or equal the threshold found, and it is classified as "anomalous" if its cumulative centroid distance to the benign centroid is above the threshold

\[
\begin{align*}
\text{if } \delta_{cum} < t_h, & \quad Y \rightarrow \text{benign}, \\
\text{if } \delta_{cum} > t_h, & \quad Y \rightarrow \text{anomalous}.
\end{align*}
\]  

(3.35)

The summarized steps for the Centroids method are presented in Figure 3.10.

As it was found in the results for the Centroids method, presented in Chapter 4, the location of the entropy points presents better results than its distance to a benign point when it is used as a parameter from which the traffic trace classification is to be carried out. To use the location of the entropy space points instead of the distance, the *K*-nearest neighbor method that uses a pattern recognition technique is proposed and its description is presented in the next Section.

### 3.6 *K*-nearest neighbor entropy method

The use of the distance as a measure of the traffic anomaly behavior is not accurate enough for some types of attacks in which the malicious points are located in a close vicinity of the benign traffic. Also, for some types of attacks, their entropy space points are located within a radius even closer than the benign traffic points to the benign centroid. However, in most of the cases there was a clear visual differentiation between the benign traffic points location and the malicious one, as the later was situated in volumes where there were few (or none) benign traffic entropy points. Thus, a pattern recognition approach is proposed, in which not the distance, but the position in the entropy space will be the parameter to judge the entropy points of a given trace and to determine if they correspond to a benign or anomalous behavior.
The $K$-nearest neighbor classifier was proposed, because of its easy implementation and its properties to classify patterns where there are some outliers of one class immersed in the other as it was explained in Chapter 2. The steps of the $K$-nearest neighbor method are explained next.

### 3.6.1 Training stage

- **STEP T1.** A number of days $N$ is analyzed for attack-free Internet traces in the network segment. The traces are collected over the $N$ taking into account the compromise between the quantity of information that can be obtained and the memory aspects.
• **STEP T2.** The traces will be obtained once again in windows of \( t_w \) seconds with sufficient length to gather enough information about the current traffic and not to big in order to accomplish a faster detection.

• **STEP T3.** With a data mining process the packets contained in each window \( i \) are analyzed to obtain the headers’ feature information. A total of \( M \) windows collected in the \( N \) days.

• **STEP T4.** The entropy spaces as explained in Section 3.1 are obtained for each window \( i \). The same convention as in the Excess points and Continuous entropy methods method is used entropy spaces of the \( K \)-nearest neighbor method: \( H^r_i \), with \( r = 1, 2, 3, 4 \) for each cluster key.

• **STEP T5.** As the amount of points will be very large given the fact that there must be a great quantity of data collected to make a proper modeling, a significant sample of the data is needed to avoid long waiting times or resource consuming processes. The proposed method to extract a significant sample from the entropy spaces is the same that was used in the Excess points algorithm to find the benign centroid: the whole space is divided into \( m \) bins, from which an average of all the points that fall within the volume of each bin is obtained. In this case however, the quantity of bins needs to be bigger as having a large number will guarantee a significant sample for each entropy space. The suggested size length for each bin is \( s = 0.5 \) bits. The total number of bins in which the entropy space \( H^r_i \) divided to \( (m) \) is found using Equation 3.6. Each bin then will have a representative point \( b_j(H^r_i) = (x_j, y_j, z_j) \), and the whole set of representative bins for each the entropy space in a window \( (H^r_i) \) will form an estimated entropy space

\[
\hat{H}^r_i = \begin{bmatrix}
    b_1(H^r_i) \\
    b_2(H^r_i) \\
    \vdots \\
    b_m(H^r_i)
\end{bmatrix}
\tag{3.36}
\]

By gathering the entropy spaces \( H^r_i \) of all the windows \( M \), a representative sample of the whole entropy space is obtained \( \hat{H}^r \).
3.6. *K-nearest neighbor entropy method*

The entropy spaces $\hat{H}^r \in \mathbb{R}^3$ are mapped into a 2-D $\hat{Z}^r$ space by means of a dimension reduction technique

$$\hat{H}^r = \begin{bmatrix} \hat{H}^r_1 \\ \hat{H}^r_2 \\ \vdots \\ \hat{H}^r_M \end{bmatrix}.$$

The proposed technique is the widely used Principal Component Analysis (PCA) given that it has been long time studied. The PCA is formally described in [23]. The PCA was applied to reduce the complexity of the decision regions into a 2-D space and it showed good results because of its characteristic to maximize the variance of the transformed data and therefore it is a well established method commonly used as a helpful tool for data extraction in Pattern recognition systems, as it is explained in [24] and [25].

**STEP T6.** The new 2-D entropy spaces found ($\hat{Z}^r$) will be used to train the K-nearest algorithm for pattern recognition.

**STEP T7.** The new 2-D entropy spaces found ($\hat{Z}^r$) will be used to train the K-nearest algorithm for pattern recognition.

**STEP T8.** Next, the same process for finding the representative sample carried out for the benign traffic must be performed with traces of malicious or anomalous traffic behavior. The entropy spaces for the anomalous traffic $Ha^r_i$ are attained.

**STEP T9.** The estimated entropy space ($Ha^r_i$) is found using the bins method.

**STEP T10.** The next step consists in applying PCA to find a mapping from the 3-D space $Ha^r_i$ into a 2-D space $Za^r_i$. 
3.6. **K-nearest neighbor entropy method**

- **STEP T11.** The last step of the training stage consists in obtaining the representative sample $Z_{a_1^r}$, by gathering the 2-D spaces $Z_{a_1^r}$ of all the windows.

- **STEP T10.** The representative entropy spaces for both benign and anomalous traffic will function as the training dataset for the $K$-nearest neighbor classifier. As it was explained in Chapter 2, the $K$-nearest algorithm’s processing time depends on the $K$ number of points (vectors) closer to the dataset to be classified, so smaller values of $K$ will be less time consuming. A value for $K > 1$ was proposed to avoid the effect of outliers in the misclassification. Also, an uneven number was needed to avoid the ambiguity that would be present for an even value of $K$ in hypothetical the case of a tie in the number of closest neighbors to a point to be classified. A value of $K = 3$ is recommended as it complies with the above being the smallest uneven number above 1.

The training stage of the data is complete at this point, and then the detection stage starts. Figure 3.11 shows the representative spaces mapped into 2-D ($Z^r$).

### 3.6.2 Detection stage

- **STEP D1.** A new trace $Y$ for a window of time $t_w$ is obtained.

- **STEP D2.** A data mining process is used to extract the intrinsic features of the packets headers’ intrinsic features.

- **STEP D3.** The entropy spaces for the current window ($H_Y$) are obtained following the procedure explained in Section 3.1.

- **STEP D4.** Dividing the space into $m$ bins, and averaging the points that fall within the volume of each bin, a representative point in each bin is found. The entropy space of the new trace will be formed by all the representative points in the bins.
3.6. *K*-nearest neighbor entropy method

Figure 3.11: Representative 2-D space patterns of benign ($\tilde{Z}^r$) and anomalous ($\tilde{Z}^a$) traffic for all the cluster keys

\[
\tilde{H}^r_Y = \begin{bmatrix}
b_1(H_Y^r) \\
b_2(H_Y^r) \\
\vdots \\
b_m(H_Y^r)
\end{bmatrix} 
\]  
(3.39)
• **STEP D5.** PCA is carried out on the entropy space $H^e_Y \in \mathbb{R}^3$ to obtained the entropy space mapping $Z^e_Y \in \mathbb{R}^2$.

• **STEP D6.** Each point in $Z^e_Y$ is fed to the $K$-nearest neighbor classifier, and the classification of that point $(y_l)$ is attained with the use of Equation (2.5) as

\[
\text{if } g_1(y_l) = \sum_{i=1}^{n} \omega_l I_{\{C_1=1\}} > g_2(y_l) = \sum_{i=1}^{n} \omega_l I_{\{C_2=1\}}, y_l \rightarrow \text{benign,}
\]

\[
\text{if } g_1(y_l) = \sum_{i=1}^{n} \omega_l I_{\{C_1=1\}} < g_2(y_l) = \sum_{i=1}^{n} \omega_l I_{\{C_2=1\}}, y_l \rightarrow \text{anomalous.} \quad (3.40)
\]

where $\omega_l = \frac{1}{K}$ if $Z_l^e$ is among the $K$-nearest neighbors of $Y$, and $\omega_l = 0$ elsewhere. $I_{\{C=c\}}$ is the Indicator function of the set of classes $C$, being defined as $C = 1$ for benign traffic and $C = 2$ for anomalous traffic.

• **STEP D7.** After classifying each point in the trace $Y$, a decision for the whole trace is made by establishing a threshold $t_k$ that corresponds to the percentage of the total number points in $Z^e_Y$ that were classified as anomalous. The value of $t_k$ that showed good results after a series of tests was $t_k = 0.5$ (50% of anomalous points in that window). This value can be decreased to raise the percentage of true positives, but the rate of false positives will also increase.

The classification on the whole window for the trace $Y$ is obtained by

\[
\text{if } \frac{N_a}{N_T} \geq t_k, \quad Y \rightarrow \text{benign,}
\]

\[
\text{if } \frac{N_a}{N_T} < t_k, \quad Y \rightarrow \text{anomalous.} \quad (3.41)
\]

where $N_a$ is the total of points classified as anomalous in that window and $N_T$ is the total of points in that window.

The summarized $K$-nearest neighbor entropy method is presented in Figure 3.12.
3.6. *K-nearest neighbor entropy method*

Figure 3.12: K-nearest neighbor entropy method diagram
Chapter 4

Results

In this chapter, we present the results from the series of simulations from the different methodologies that were derived in order to obtain better responses in the anomaly detection of a network segment that were described in detail in Chapter 3, are presented.

The scenario from which the benign traces were obtained was a class C IP network divided into four subnets. There were 100 hosts running Windows XP SP2 mainly. Two routers connect the subnets with 10 Ethernet switches and 18 IEEE 802.11 b/g wireless access points. The data rate of the core network is 100 Mbps as it was described in [26]. The traces were obtained over a 5 day period ($N = 5$) in typical working hours to characterize the benign traffic.

Traces from 8 different types of attacks were obtained to train and test the methods described in Chapter 3. The Blaster and Sasser worms' traces were obtained from the same segment as those of the benign traffic described above, by releasing the worms in a sector of the network that was left vulnerable on purpose with 10 not patched hosts. The traces of the ps, sshtrojan, portsweep, DDoS1 and DDoS2 were gathered from the DARPA dataset that can be found in [16]. The description of each attack was presented in Chapter 2.

Depending on the working environment in which an Intrusion Detection System is to be deployed, the emphasis in determining the parameters needed can reside in different evaluation metrics. In this thesis, the selection of the parameters after a series of tests was carried out by establishing a compromise in finding the highest detection rate (True positive rate) with a low false alarm rate (False positive rate).

The window-level true positive, false positive, true negative and false negative traces presented later in this chapter were obtained, after training the methods, by running simulations in which benign traffic windows were applied followed by the introduction of attacks' traces in a given window. The use of this procedure infers stationarity of the traffic, given the fact that the benign traces gathered were used
both for training and testing purposes. Another limitation of this procedure resides in the fact that there is not an exactly real continuous behavior on a network when an attack is released in the given segment, but its advantage is that the traffic was obtained directly from a real network segment, instead of a simulated one with viability in testing different attacks.

The results are divided in subsections for each of the four methods pursuing an accurate detection of the anomalies corroborating the weaknesses and strengths of each. All of the methods were tested with time windows of $t_w = 60$ seconds.

### 4.1 Excess points method

The benign centroid for the Excess points method was found using the bins method with a bin’s side length of $s = 1.5$ bits which showed good results for the centroid’s location in the entropy space. The likelihood ratio (LR) for the Blaster and Sasser worms where excess points were found are presented in Table 4.1, where a value of 0 means that there were no excess points found for a given cluster key.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Cluster key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SRC IP</td>
</tr>
<tr>
<td>Blaster</td>
<td>0.01</td>
</tr>
<tr>
<td>Sasser</td>
<td>0.0034</td>
</tr>
</tbody>
</table>

In Figure 4.1 the results of the probability density functions found for all the parameters of a Blaster worm attack are shown, while in Figure 4.2 the results for all the parameters of a Sasser worm attack are shown. As it is depicted in Figure 4.1 for 3 out of the 4 cluster keys (having Dst IP as an exception, i.e., there were no excess points for this parameter), there is a clear detection of a Blaster attack, as the probability density functions are easily distinguished. In Figure 4.2 for the four cluster keys there is a clear distinction between both probability density functions and therefore, the attack is detected. The probability density function fit found with the kernel smoothing method could be replaced by a normal distribution for both the benign and the anomalous traffic to simplify the analysis, as is has been long time studied and the probabilities of error are well established for that type of distribution. The normal distribution can be used given that the goal of effectively identify a distinction between two classes of traffic behaviors is accomplished.

As it was mention in Chapter 3, for other types of attacks, the detection with the excess point method was ineffective and other methods were required to indicate
4.1. Excess points method

Figure 4.1: Probability comparison for a Blaster worm attack for the excess point method

(a) Probability density function comparison for source IP ($H^1$) cluster key

(b) Probability density function comparison for source Port ($H^3$) cluster key

(c) Probability density function comparison for destination Port ($H^4$) cluster key

Figure 4.1: Probability comparison for a Blaster worm attack for the excess point method
4.1. Excess points method

Figure 4.2: Probability comparison for a Sasser worm attack for the excess point method

(a) Probability density function comparison for source IP ($H^1$) cluster key
(b) Probability density function comparison for destination IP ($H^2$) cluster key

(c) Probability density function comparison for source Port ($H^3$) cluster key
(d) Probability density function comparison for destination Port ($H^4$) cluster key

Figure 4.2: Probability comparison for a Sasser worm attack for the excess point method
a malicious behavior on the current traffic as it is shown in the next sections. The reason for which the methodology of the excess points method was inefficient for other types of attacks is because some of them exhibit a behavior that derive into entropy points very near the benign centroid and therefore, the amount of anomaly points that are located outside the boundary provided by the benign behavior (i.e., the excess points) is very small or zero in some cases and the profiling for that traffic becomes impossible.

4.2 Continuous entropy method

The likelihood ratios found for the Continuous entropy method are presented in Table 4.2 for the Blaster and Sasser worm attacks given the fact that points exceeding the maximum benign distance limits were found only for those attacks. A value of LR=0 means that there were no points exceeding the benign limit for the specified cluster key.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Cluster key</th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blaster</td>
<td>SRC IP</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Sasser</td>
<td>SRC PRT</td>
<td>0.01</td>
<td>0.01</td>
<td>0.001</td>
<td>0</td>
</tr>
</tbody>
</table>

In figures 4.3 and 4.4, the resulting probability density functions found using the kernel smoothing algorithm are presented for the cluster keys in which there was an evidence of anomalous traffic for the Blaster and Sasser worms respectively.

Even though the continuous method exhibited a difference in 3 of the 4 parameters while in the excess points method there was a difference in the four parameters, with the continuous entropy method there was a clearer difference in the malicious traces compared with the benign traffic. This difference follows the fact that many exceeding points were outside the maximum and minimum benign limits. In the case of source Prt and destination IP all the malicious points were located outside the benign boundaries, and in the source IP, almost sixty percent of the malicious points were outside the those boundaries.

Once again, with this method, the amount of cluster keys that correctly showed a considerable difference in the statistics found were lesser than those found with the excess points method (2 and 3, respectively), however, in the case of the destination Port, the amount of points that exceeded the boundaries of the benign traffic were about 23%, and in the case of destination IP cluster key, this percentage had a 94.8% of exceeding points.
4.3. Statistical method

Figure 4.3: Probability comparison for a Blaster worm attack for the continuous entropy method

The continuous entropy method was then successful for worm type attacks, in which the excess points are very far from the boundaries provided by the benign traffic and specially from the origin of the entropy space, but it was inefficient for other types of attacks, where the malicious entropy points were located within the benign limits. It is then, a similar behavior and performance as the one that the excess point method exhibited.

4.3 Statistical method

As explained in Chapter 3, the statistics used to describe the benign and the anomalous traffic were found as a support to either the Excess points or the Continuous entropy method previously proposed. The resulting statistics then were obtained from the excess points method and the continuous entropy method.

The results of the Statistical method for the excess points method are presented The table 4.3 for a Sasser worm attack and in Table 4.4 for a Blaster worm attack, in both tables a value “NA” means that there were no excess points for the given cluster key. The Statistical method showed better results when there was a separation between the traffic involving excess points in the malign traces than finding statistics with the whole malicious traces without distinction.
As it can be seen from tables 4.3 and 4.4, there is an obvious difference between the means of the distances of the excess points to the benign centroid than those of the benign points, which makes it very easy to classify as an attack and to obtain probability density functions that could be easily separated as it is presented in the excess points method. It is also clear that the variance (and therefore the standard deviation) is greater in both cases in comparison with that found for the benign traffic. On the other hand, the kurtosis for both worm attacks presented a lower value than that found for the attack-free traces.
4.3. Statistical method

Table 4.3: Statistics for the excess points distances to the benign centroid of a Sasser worm attack

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>10.8493</td>
<td>12.4663</td>
<td>9.6546</td>
<td>13.1288</td>
</tr>
<tr>
<td>variance</td>
<td>1.9601</td>
<td>0.9355</td>
<td>2.2193</td>
<td>4.0618</td>
</tr>
<tr>
<td>var/mean</td>
<td>0.1807</td>
<td>0.0750</td>
<td>0.2299</td>
<td>0.3094</td>
</tr>
<tr>
<td>standard deviation</td>
<td>1.4000</td>
<td>0.9672</td>
<td>1.4897</td>
<td>2.0154</td>
</tr>
<tr>
<td>median</td>
<td>10.6322</td>
<td>12.5096</td>
<td>9.2956</td>
<td>13.8304</td>
</tr>
<tr>
<td>kurtosis</td>
<td>5.5150</td>
<td>13.3894</td>
<td>5.1796</td>
<td>6.7731</td>
</tr>
<tr>
<td>skewness</td>
<td>1.2611</td>
<td>-2.9183</td>
<td>1.4504</td>
<td>-2.3128</td>
</tr>
<tr>
<td>max value</td>
<td>15.0354</td>
<td>13.6660</td>
<td>14.5563</td>
<td>14.3502</td>
</tr>
<tr>
<td>min value</td>
<td>7.4721</td>
<td>8.1822</td>
<td>7.0719</td>
<td>6.7130</td>
</tr>
</tbody>
</table>

Table 4.4: Statistics for the excess points distances to the benign centroid of a Blaster worm attack

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>11.4716</td>
<td>12.5705</td>
<td>NA</td>
<td>13.3053</td>
</tr>
<tr>
<td>variance</td>
<td>3.1084</td>
<td>8.0899</td>
<td>NA</td>
<td>1.3278</td>
</tr>
<tr>
<td>var/mean</td>
<td>0.2710</td>
<td>0.6436</td>
<td>NA</td>
<td>0.0998</td>
</tr>
<tr>
<td>standard deviation</td>
<td>2.8443</td>
<td>1.7631</td>
<td>NA</td>
<td>1.1523</td>
</tr>
<tr>
<td>median</td>
<td>11.0119</td>
<td>14.4438</td>
<td>NA</td>
<td>13.6309</td>
</tr>
<tr>
<td>kurtosis</td>
<td>4.9100</td>
<td>1.5130</td>
<td>NA</td>
<td>8.9401</td>
</tr>
<tr>
<td>skewness</td>
<td>0.5043</td>
<td>-0.4921</td>
<td>NA</td>
<td>-2.4970</td>
</tr>
<tr>
<td>max value</td>
<td>15.5087</td>
<td>15.1129</td>
<td>NA</td>
<td>14.1054</td>
</tr>
<tr>
<td>min value</td>
<td>4.7724</td>
<td>7.5758</td>
<td>NA</td>
<td>9.0274</td>
</tr>
</tbody>
</table>

In Table 4.5 the resulting statistics of the distances for the continuous entropy method from a Sasser worm attack to the benign mean are presented for the points that exceed the maximum or, in this case also, the minimum threshold parameters of the benign traces. In Table 4.6 the resulting statistics of the continuous entropy method for a blaster worm attack are showed. A value “NA” means that there were no excess points for the given cluster key.

The statistics method showed better results when used with excess points, however, in other types of attacks the entropy found for the distances to the benign centroid was not exceeding the limit provided by the maximum value of the benign traffic and therefore an analysis of the statistics was impossible for those cases. The statistics method was then helpful, specially for Blaster and Sasser worm attacks but it was inefficient for other types of attacks. The above applies not only for the excess points method, but also for the continuous entropy method.
4.4 Centroids method

For the Centroids method, the benign centroid was found using the bins method explained in section 3.5, using a side length of \( s = 1.5 \) bits as good positioning of the resulting centroid was found with that longitude. Also, a sliding window size of \( S_w = 10 \) was used as it provided good results for the detection. A value of \( S_w = 10 \) means that the distances of the last 10 windows to the benign centroid are used to obtain the value \( \hat{p} \).

In Figure 4.5, the centroid location of each attack, as well as the benign traffic are showed for destination IP and destination Port cluster keys. It can easily be seen that event though in some cases the distances of the malicious traces are not very far from the benign centroid, there is a clear difference in the points to which each centroid will converge in comparison with the benign centroid.

The centroids method showed better results for other attacks besides the Blaster

---

Table 4.5: Statistics for the continuous entropy distances to the benign centroid of a Sasser worm attack

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>2.6613</td>
<td>4.1447</td>
<td>5.2976</td>
<td>NA</td>
</tr>
<tr>
<td>variance</td>
<td>0.1207</td>
<td>0.2734</td>
<td>0.4737</td>
<td>NA</td>
</tr>
<tr>
<td>var/mean</td>
<td>0.0453</td>
<td>0.0660</td>
<td>0.0894</td>
<td>NA</td>
</tr>
<tr>
<td>standard deviation</td>
<td>0.3474</td>
<td>0.5228</td>
<td>0.6882</td>
<td>NA</td>
</tr>
<tr>
<td>median</td>
<td>2.5936</td>
<td>4.2981</td>
<td>5.4876</td>
<td>NA</td>
</tr>
<tr>
<td>kurtosis</td>
<td>2.9355</td>
<td>7.1294</td>
<td>12.7986</td>
<td>NA</td>
</tr>
<tr>
<td>skewness</td>
<td>0.7803</td>
<td>-2.0270</td>
<td>-3.1930</td>
<td>NA</td>
</tr>
<tr>
<td>max value</td>
<td>3.5208</td>
<td>4.6511</td>
<td>5.7714</td>
<td>NA</td>
</tr>
<tr>
<td>min value</td>
<td>2.2218</td>
<td>2.4541</td>
<td>2.4303</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 4.6: Statistics for the excess points distances to the benign centroid of a Blaster worm attack

<table>
<thead>
<tr>
<th></th>
<th>SRC IP</th>
<th>SRC PRT</th>
<th>DST IP</th>
<th>DST PRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>NA</td>
<td>NA</td>
<td>9.7374</td>
<td>2.6838</td>
</tr>
<tr>
<td>variance</td>
<td>NA</td>
<td>NA</td>
<td>0.8712</td>
<td>0.3092</td>
</tr>
<tr>
<td>var/mean</td>
<td>NA</td>
<td>NA</td>
<td>0.0895</td>
<td>0.1152</td>
</tr>
<tr>
<td>standard deviation</td>
<td>NA</td>
<td>NA</td>
<td>0.9334</td>
<td>0.5560</td>
</tr>
<tr>
<td>median</td>
<td>NA</td>
<td>NA</td>
<td>10.1790</td>
<td>2.4625</td>
</tr>
<tr>
<td>kurtosis</td>
<td>NA</td>
<td>NA</td>
<td>9.4194</td>
<td>3.8524</td>
</tr>
<tr>
<td>skewness</td>
<td>NA</td>
<td>NA</td>
<td>-2.6344</td>
<td>1.5052</td>
</tr>
<tr>
<td>max value</td>
<td>NA</td>
<td>NA</td>
<td>10.2119</td>
<td>3.9761</td>
</tr>
<tr>
<td>min value</td>
<td>NA</td>
<td>NA</td>
<td>6.2354</td>
<td>2.2648</td>
</tr>
</tbody>
</table>
4.4. **Centroids method**

and Sasser worms, so it was more efficient than the excess points, continuous entropy and statistics methods.

From Figure 4.5(a) it can be seen that, for the destination IP cluster key, the majority of the attacks centroids are gathered in a lower position than the benign centroid. The Sasser attack on the other hand, is located in a higher position than the benign centroid and it is the more distant centroid from the benign one. In Figure 4.5(b) the centroids of most of the attacks for the destination Port cluster key are located nearer to the benign centroid than in other cases, however there is still a distinguishing difference between them and the benign traces centroid. The Sasser attack centroid was again the more distant point from the benign traffic.

In table 4.7 the results from all the attacks centroids points in comparison with the benign centroid for the destination IP ($H^2$) cluster key are presented.

Table 4.7: Results of the benign and malicious traffic centroids for the destination IP cluster key

<table>
<thead>
<tr>
<th>Centroid</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$z_i$</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benign</td>
<td>0.941510</td>
<td>2.026277</td>
<td>1.477684</td>
<td>0.0</td>
</tr>
<tr>
<td>DDOS1</td>
<td>1.526179</td>
<td>1.399646</td>
<td>0.594104</td>
<td>1.230942</td>
</tr>
<tr>
<td>DDOS2</td>
<td>1.477257</td>
<td>1.593130</td>
<td>0.601160</td>
<td>1.114870</td>
</tr>
<tr>
<td>IP sweep</td>
<td>0.563558</td>
<td>0.485492</td>
<td>0.109995</td>
<td>2.094622</td>
</tr>
<tr>
<td>Port sweep</td>
<td>0.952650</td>
<td>1.028166</td>
<td>0.776394</td>
<td>1.219900</td>
</tr>
<tr>
<td>ps</td>
<td>1.129699</td>
<td>1.167899</td>
<td>0.582944</td>
<td>1.254108</td>
</tr>
<tr>
<td>ssh trojan</td>
<td>0.880402</td>
<td>0.911831</td>
<td>0.805155</td>
<td>1.303080</td>
</tr>
<tr>
<td>Blaster</td>
<td>0.824996</td>
<td>1.071298</td>
<td>1.098246</td>
<td>1.034182</td>
</tr>
<tr>
<td>Sasser</td>
<td>0.732402</td>
<td>3.699243</td>
<td>4.018475</td>
<td>3.049288</td>
</tr>
</tbody>
</table>

It can be seen from table 4.7 that for all the attacks, their centroids converge to a quite different position than the benign centroid with the Blaster worm attack being the nearest one and the Sasser worm attack being the more distant.

The cluster key in which the attacks were located closer to the benign centroid was destination Port ($H^4$), and this results are presented in the table 4.8:

In table 4.8 it can be seen that, for two denial of service attacks, the centroids are close to the benign centroid and therefore they are the most difficult to detect for the destination Port cluster key. However, for other destination keys the distances from the benign centroid of the denial of service attacks are more distant, and therefore the recognition of a present attack is effective.

The true positives, false negatives, true negatives and false positives for the centroids method are presented in tables 4.9, 4.10, 4.11 and 4.12.
4.4. Centroids method

(a) Centroids of the dst IP entropy space

(b) Centroids of the dst Prt entropy space

Figure 4.5: Centroids of 2 cluster keys
4.4. **Centroids method**

Table 4.8: Results of the benign and malicious traffic centroids for the destination Port cluster key

<table>
<thead>
<tr>
<th>Centroid</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$z_i$</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benign</td>
<td>1.010705</td>
<td>1.748925</td>
<td>2.590627</td>
<td>0.0</td>
</tr>
<tr>
<td>DDOS1</td>
<td>1.067170</td>
<td>1.195875</td>
<td>2.422671</td>
<td>0.580743</td>
</tr>
<tr>
<td>DDOS2</td>
<td>1.029880</td>
<td>1.300418</td>
<td>2.548454</td>
<td>0.450893</td>
</tr>
<tr>
<td>IPsweep</td>
<td>0.106889</td>
<td>0.137001</td>
<td>0.789704</td>
<td>2.580408</td>
</tr>
<tr>
<td>Portsweep</td>
<td>1.191231</td>
<td>1.243038</td>
<td>1.713883</td>
<td>1.028198</td>
</tr>
<tr>
<td>ps</td>
<td>0.773585</td>
<td>0.903474</td>
<td>1.937513</td>
<td>1.094336</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>1.266078</td>
<td>1.337038</td>
<td>1.588066</td>
<td>1.113552</td>
</tr>
<tr>
<td>Blaster</td>
<td>0.727969</td>
<td>5.478430</td>
<td>3.088570</td>
<td>3.773207</td>
</tr>
<tr>
<td>Sasser</td>
<td>1.514468</td>
<td>3.387656</td>
<td>2.990648</td>
<td>1.760464</td>
</tr>
</tbody>
</table>

Table 4.9: Results for the $H^1$ cluster key anomaly detection for the centroids method

<table>
<thead>
<tr>
<th>Attack</th>
<th>True Positive</th>
<th>False Negative</th>
<th>True Negative</th>
<th>False Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDOS1</td>
<td>55.38 %</td>
<td>44.62 %</td>
<td>91.08 %</td>
<td>8.92 %</td>
</tr>
<tr>
<td>DDOS2</td>
<td>34.95 %</td>
<td>65.05 %</td>
<td>91.84 %</td>
<td>8.16 %</td>
</tr>
<tr>
<td>IPsweep</td>
<td>93.18 %</td>
<td>6.82 %</td>
<td>89.99 %</td>
<td>10.01 %</td>
</tr>
<tr>
<td>Portsweep</td>
<td>33.33 %</td>
<td>66.67 %</td>
<td>92.26 %</td>
<td>7.74 %</td>
</tr>
<tr>
<td>ps</td>
<td>62 %</td>
<td>38 %</td>
<td>91.84 %</td>
<td>8.16 %</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>62.5 %</td>
<td>37.5 %</td>
<td>92.35 %</td>
<td>7.65 %</td>
</tr>
<tr>
<td>Blaster</td>
<td>94.87 %</td>
<td>5.13 %</td>
<td>85.95 %</td>
<td>14.05 %</td>
</tr>
<tr>
<td>Sasser</td>
<td>96.97 %</td>
<td>3.03 %</td>
<td>88.73 %</td>
<td>11.27 %</td>
</tr>
</tbody>
</table>

Table 4.10: Results for the $H^2$ cluster key anomaly detection for the centroids method

<table>
<thead>
<tr>
<th>Attack</th>
<th>True Positive</th>
<th>False Negative</th>
<th>True Negative</th>
<th>False Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDOS1</td>
<td>94.87 %</td>
<td>5.13 %</td>
<td>80.99 %</td>
<td>19.01 %</td>
</tr>
<tr>
<td>DDOS2</td>
<td>84.47 %</td>
<td>15.53 %</td>
<td>82.76 %</td>
<td>17.24 %</td>
</tr>
<tr>
<td>IPsweep</td>
<td>97.73 %</td>
<td>2.27 %</td>
<td>84.1 %</td>
<td>15.9 %</td>
</tr>
<tr>
<td>Portsweep</td>
<td>76.19 %</td>
<td>23.81 %</td>
<td>84.86 %</td>
<td>15.14 %</td>
</tr>
<tr>
<td>ps</td>
<td>90 %</td>
<td>10 %</td>
<td>80.24 %</td>
<td>19.76 %</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>62.5 %</td>
<td>37.5 %</td>
<td>85.2 %</td>
<td>14.8 %</td>
</tr>
<tr>
<td>Blaster</td>
<td>66.67 %</td>
<td>33.33 %</td>
<td>84.52 %</td>
<td>15.48 %</td>
</tr>
<tr>
<td>Sasser</td>
<td>100 %</td>
<td>0 %</td>
<td>84.69 %</td>
<td>15.31 %</td>
</tr>
</tbody>
</table>

In tables 4.9, 4.10, 4.11 and 4.12, the results show how the centroids method can achieve a correct detection on many of the attacks. However, one important aspect
4.4. **Centroids method**

Table 4.11: Results for the $H^3$ cluster key anomaly detection for the centroids method

<table>
<thead>
<tr>
<th>Attack</th>
<th>True Positive</th>
<th>False Negative</th>
<th>True Negative</th>
<th>False Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDOS1</td>
<td>10.26 %</td>
<td>89.74 %</td>
<td>84.86 %</td>
<td>15.14 %</td>
</tr>
<tr>
<td>DDOS2</td>
<td>14.56 %</td>
<td>85.44 %</td>
<td>85.2 %</td>
<td>14.8 %</td>
</tr>
<tr>
<td>IPsweep</td>
<td>97.73 %</td>
<td>2.27 %</td>
<td>84.61 %</td>
<td>15.39 %</td>
</tr>
<tr>
<td>Portsweep</td>
<td>42.86 %</td>
<td>57.14 %</td>
<td>85.11 %</td>
<td>14.89 %</td>
</tr>
<tr>
<td>ps</td>
<td>62 %</td>
<td>38 %</td>
<td>83.94 %</td>
<td>16.06 %</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>50 %</td>
<td>50 %</td>
<td>85.11 %</td>
<td>14.89 %</td>
</tr>
<tr>
<td>Blaster</td>
<td>94.87 %</td>
<td>5.13 %</td>
<td>83.26 %</td>
<td>16.74 %</td>
</tr>
<tr>
<td>Sasser</td>
<td>69.7 %</td>
<td>30.3 %</td>
<td>84.19 %</td>
<td>15.81 %</td>
</tr>
</tbody>
</table>

Table 4.12: Results for the $H^4$ cluster key anomaly detection for the centroids method

<table>
<thead>
<tr>
<th>Attack</th>
<th>True Positive</th>
<th>False Negative</th>
<th>True Negative</th>
<th>False Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDOS1</td>
<td>3.08 %</td>
<td>96.92 %</td>
<td>82.84 %</td>
<td>17.16 %</td>
</tr>
<tr>
<td>DDOS2</td>
<td>8.74 %</td>
<td>91.26 %</td>
<td>83.01 %</td>
<td>16.99 %</td>
</tr>
<tr>
<td>IPsweep</td>
<td>97.73 %</td>
<td>2.27 %</td>
<td>81.83 %</td>
<td>18.17 %</td>
</tr>
<tr>
<td>Portsweep</td>
<td>38.1 %</td>
<td>61.9 %</td>
<td>82.59 %</td>
<td>17.41 %</td>
</tr>
<tr>
<td>ps</td>
<td>56 %</td>
<td>44 %</td>
<td>82.34 %</td>
<td>17.66 %</td>
</tr>
<tr>
<td>sshtrojan</td>
<td>0 %</td>
<td>100 %</td>
<td>82.51 %</td>
<td>17.49 %</td>
</tr>
<tr>
<td>Blaster</td>
<td>89.74 %</td>
<td>10.26 %</td>
<td>81.08 %</td>
<td>18.92 %</td>
</tr>
<tr>
<td>Sasser</td>
<td>78.79 %</td>
<td>21.21 %</td>
<td>86.71 %</td>
<td>13.29 %</td>
</tr>
</tbody>
</table>

that arises is the percentage of true negative. A compromise between possible level of detection that the IDS can obtain (true positive percentage) with this method and the proportion of the true negative rate is needed. In this case, the lower limit for the method was proposed to be about 80%. The reason for which the compromise between the two true rates is required follows the fact that the centroid method takes under consideration only the information about the distance from the benign centroid from the lectures of the traffic entropy being received.

The weakness of the centroids method was present in the fact that some series of benign traffic with a distant location from the benign centroid were treated as anomalous, and some malicious traffic entropies close to the benign centroid were treated as benign. This is because the method focuses only on distances and not in location of the attacks. In the next section, the $K$-nearest neighbor entropy method’s results are presented. For this method, the location of the entropy points was the parameter used to detect anomalies.
4.5 K-Nearest neighbor entropy method

The bin length used for the K-Nearest neighbor entropy method was smaller than the one used for the Excess points and the Continuous entropy method given the fact that, as it was explained in Chapter 3, there is a need to find a representative sample of all the entropy space points. The value for the bin side length employed was $s = 0.5$ bits. The value of the $K$ nearest neighbors was selected to comply with the requirements explained in Chapter 3. First an odd value was needed to avoid ties when detecting the majority of closer neighbors to an incoming point. Also, choosing a greater value for $K$ derives into a more resource consuming result (i.e., a bigger classification time). A value of $K = 3$ was selected after a series of tests as it conforms with the requirements previously mentioned.

As it was stated in Chapter 3, the method with the highest efficiency in detecting anomalies of different types was the Pattern recognition method, using the $K$-nearest neighbor algorithm. The Figures 4.6, 4.7, 4.8 and 4.9 show the 2-D decision regions for the 4 cluster keys as they were found using the $K$-nearest neighbor pattern recognition classifier rule after applying the PCA on the training data and using a value $K = 3$ that was selected as it represents a low computational effort with a good decision region definition for the entropy points obtained.

![Decision regions](image)

(a) Decision regions with benign and anomalous entropy points for the source IP ($H^1$) entropy space

(b) Decision regions for the source IP ($H^1$) entropy space

Figure 4.6: Decision regions for the $H^1$ cluster key

In tables 4.13, 4.14, 4.15 and 4.16 the results of the pattern recognition method for each cluster key are presented as well as the percentage of true positives, false positives, true negatives and false negatives, found at a window level.
4.5. K-Nearest neighbor entropy method

In tables 4.13, 4.14, 4.15 and 4.16 the results show that there is a correct detection in the presence of diverse attack types. The output of the IDS demonstrated to be highly efficient, as a high detection true positive percentage was found in at least one of the cluster key results for the majority of the attacks. Also, the percentage of true negatives was found to be very high, as the minimum was 97.65 % found in the source Port cluster key ($H^3$). The attack with the minimum detection rate observed
was an IPSweep with the highest detection percentage (37.21 %) found with the $H^2$ cluster key. However, it is important to mention that the true positive, false negative, true negative and false positive rates were reported at a window-level. This means that even when most of the windows in for which the network segment was under this attack were classified as benign, there were alerts located in the time where the attack was released and in a visual way it is easy to detect an anomaly in that period of time, as the alarms where showed very near one from another, in contrast with the false positives that normally appear as a occasional (with many time windows between one and the other). Therefore, there was a detection in the whole malicious trace for all the attacks being considered in the experiments. The true negative and false positive percentages for this method maintain the same value for all of the simulations, because
for the $K$-nearest neighbor method, the parameter to be evaluated is the location of the entropy points at the current time (the decision is not based in previous results as in the centroids method). Therefore, the results will be the same regardless of the windows in which the attack was released, which is an important feature of this method.
4.5. *K*-Nearest neighbor entropy method

In Figure 4.10 the anomaly detection for different types of attacks and different cluster keys is showed. The Figure presents the detection level that would be the output of the IDS. The output is treated as a 0 when the alarm level is low, that is, when the current traffic can be classified as "normal" and the output is treated as a 1 when the current traffic is classified as "anomalous".

![Anomaly detection examples of the 4 cluster key entropy spaces](image)

(a) Detection for $H^1$ cluster key with a Blaster worm attack
(b) Detection for $H^2$ cluster key with the Distributed Denial of Service 1 attack
(c) Detection for $H^3$ cluster key with the portsweep attack
(d) Detection for $H^4$ cluster key with the Distributed Denial of Service 2 attack

Figure 4.10: Anomaly detection examples of the 4 cluster key entropy spaces

In Figure 4.10(a), the presence of a Blaster worm attack is shown. The attack was released in the 1000th window, and it is very clear that the amount of false
positives is very low. The found true negative rate is 99.66 %. In Figure 4.10(b), a Distributed Denial of Service was released in the 200th window with the 90.26 % detection of the windows being under attack and the 99.07 % of the benign windows being classified as normal. In Figure 4.10(c), the detection of a PortswEEP attack released in the 1000th time window is presented. The detection rate for this attack in a window-level was 90.48 % with a 97.65 % true negative percentage. Figure 4.10(d) presents the detection of a Distribute Denial of Service attack (different from that in Figure 4.10(a)) with a 79.61 % window detection rate and 98.15 % true negative percentage. It is worthwhile mentioning that, as it was stated above, the detection rates are presented in a window-level, so even when the rates previously referred are not very high in some cases, there is still a clear distinction of an anomalous traffic as the alarms occur in time windows very close one from another. In the case of false positives, they are often found as sporadic behaviors, as an alarm usually can occur at one time window but other alarms arise in time windows far from that one.

As it is shown in Figure 4.10, the IDS has high efficiency in reporting an alarm when threats form diverse traffic attacks are present and it presents also a good performance of misclassification, as it reports an alert when the traffic is normal in very few cases only. The $K$-nearest neighbor method exhibited the best true positive, false negative, true negative and false positive rates.

Having exposed the results from each of the methods analyzed, some conclusions about the best methodology to achieve an efficient anomalous detection can be deduced. By combining the use of the methods analyzed aiming to integrate the advantages of each and minimizing their drawbacks, an improvement on the anomaly detection rate can be achieved. The proposed combination of the methods is the following:

- First, apply the Excess points and the Continuous entropy method in parallel. Once the probabilistic model for both of them has been obtained, they provide the fastest response as any incoming entropy point can be classified based on a probability density function. As they can detect anomalies that exceed by much the average behavior, such as worm type virus, in which the corresponding entropy points are very distant from the benign ones, a fast anomaly detection is desirable.

- Next, apply the Statistical model next. For both the Excess points and the Continuous entropy method, a second level support can be obtained by obtaining statistics from which anomalies can be observed.

- Finally, apply in parallel the Centroids and the $K$-nearest neighbor entropy models. To detect anomalies that may not be reflected into points very distant from the benign entropy points, this two methods showed better results. The Centroids method helps to detect a series of anomalies in the normal entropy points that have been present in some previous widows, while the $K$-nearest entropy method helps by determining unusual patterns in the current window.
The ultimate methodology proposed by combining the methods studied in this work is shown in Figure 4.11.

Requirements for Anomaly Based Network Intrusion Detection Systems (A-NIDS) to be considered as a well working performance are to obtain a False Positive Rate of about $FP = 1\%$, with very high True Positive Rates ($TP$). Works related to A-NIDS have reported high data rates with low False Positive Rates, such as the ones reported in [27] of $TP = 98\%$ with $FP = 1\%$, or even a $TP$ of about $99\%$ with as small as $0.04\%$ of FP for some attacks reported in [28]. However, the use of many features are required to provide the accurate detection in some cases and for others faster detection rates are required. In this thesis, only the extraction of the 4 intrinsic features of an IP packet header is used and the methods developed follow a practical approach to find anomalies in a network segment. The detection rates found in this thesis were presented at a window-level, which means that if the True positive rates are taken at a trace-level, there was an evidence of an anomaly for all of the attacks analyzed with the $K$-nearest neighbor entropy method and with the smaller False Positive Rate of $FP = 0.34\%$. By combining the methods as shown in Figure 4.11 higher detection rates and a compensation on false positive rates can be accomplished. Also, the time windows of $t_w = 60$ seconds can provide faster detection than other methods and can establish the basis of a future online approach.

In the next chapter, the conclusions and future work related to this thesis project are presented.
Chapter 5

Conclusions and future work

In this chapter, the conclusive remarks of the work carried out are presented as well as some aspects regarding helpful future work in this matter.

5.1 Conclusions

In this work network anomalies methodologies based on entropy spaces were analyzed and their characteristics were evaluated to propose a method to detect anomalies in a network segment.

Different types of attacks showed behaviors that can be reflected in an entropy space. Depending of the type of attack, the resulting entropy space can be quite far from the benign centroid that is found with the method presented (this is the case of the two types of worms attacks analyzed), or in other cases it can be near that centroid, however in many cases, most of the entropy space for the malicious traffic is located outside the volume in which the benign traffic is enclosed and therefore a recognition of an anomalous traffic present in a network segment can be possible.

The excess points method exhibited good results when there is an attack in the network segment with a very drastic change in the entropy points, i.e. in attacks in which the entropy points are located very distant from the benign points. For those attacks, a detection can be found by finding a probability density function with which the decision regions from a communication theory can be obtained.

The statistics of the benign and the malicious traffic does not differ in a great scale, as there are anomalous points located outside the boundaries provided by the benign traffic of the training data, but some of them are immersed within the volume contained, and therefore, in a big scale, the statistics found converge to similar values and therefore the information that one can obtain from the statistics only is not conclusive enough to prove that there is an anomalous trace present.
The continuous entropy method also turned out to be a reliable detection algorithm only when the traffic being evaluated contained an attack with high changes in the uncertainty (entropy) of the expected behavior. In a more discrete attack, with more distributed attack over time, the continuous entropy method is inefficient because most (if not all) of the malicious entropy points fall within the benign boundaries provided by the training data.

The distance of the malicious entropy points to a benign centroid presents a good reference on when an attack is currently affecting a network segment, however, it does not provide a sufficient parameter with which a good detection rates an low false alarms could be achieved. This is because the entropy points found for the benign traffic are located in many cases very distant from the centroid found. Also, in some cases the malicious entropy points are located in a position even closer to the benign centroid than some benign points, which makes it harder for the method to detect a correct anomaly and reduce the false positive rates at the same time.

The location of the points of an entropy space showed to be a more trustful parameter to evaluate the presence of an anomaly because, many of the attacks' entropy points are located in a position close to the benign traffic, but they are present specially in areas in which the benign traffic points have lower probability to be located. The use of pattern recognition techniques used in the \(K\)-nearest method showed better results than the other methods because with the classifier used effectively captures the location of the current points to correctly detect a threat in the system.

A definition of the Internet noise was obtained by means of its characterization in entropy spaces that can be trained to obtain good results in the anomaly detection whenever a suspicious traffic is currently present in a network segment. The system is based in the uncertainty measure (entropy) as the Internet traffic is characterized as a stochastic model.

## 5.2 Future work

To optimize the algorithms used. By creating a parallelization of the methods, a better resource efficiency is achievable which in turn derives into better processing times and therefore faster detection. The later can be achieved if the functions required for some methods are used in parallel, obtaining results that can be used for more than one method instead of treating each method as independent.

In order to correctly define the influence in the location of the entropy space for a given attack, the method proposed in this thesis (\(K\)-nearest neighbor) creates a classification of the entropy points received in a time window. Another method that could be proposed for a future work, is to modeling the entropy points as
vectors with a magnitude and a phase. The properties that could be exploited out of this method is that, according to a training data pattern, a good classification of the data could be achieved by finding the phase and magnitude of the vector and then, carry out an analysis in which, the magnitude of the vector will be a parameter to evaluate the distance to a benign point but also the position of the entropy point will be obtain by its phase and will become another parameter to judge if a trace.

In the case of the centroids method, the main weakness of the method found was that using the distance as the parameter to judge the behavior of a trace was a limited approach because of the fact that some malicious points were even closer to the benign traffic’s centroid found that some of the benign entropy points. A method that could help to make more efficient the use of the distance parameter is to use more than one centroid (clusters) to classify the benign traffic. With more than one cluster being define to characterize the benign traffic, a better differentiation on the malicious points could be achieved given the fact that, points that were once close to the benign centroid, now are more distant from their closest benign centroid in comparison with the benign points around that location.

The $K$-nearest neighbor method showed good results because it takes into account the location of the attacks’ entropy points instead of their distance, but by combining the $K$-nearest neighbor method with a measure of the distance can help to raise the effectiveness of the method by finding a weighting algorithm for both the position and the distance of the entropies being analyzed.
Appendix A

Graphical results

In this appendix, the graphics results from the Statistical method analyzed are presented. The appendix is divided into two sections corresponding to the statistical results of the Excess points and the Centroids method described in Chapter 3.

A.1 Statistical with Excess points method

In figures A.1, A.2, A.3 and A.4 the graphics of histogram, Q-Q plot, autocorrelation, and power spectral density of the benign points distances to the benign centroid for all the cluster keys are presented. In figures A.5, A.6 and A.7 the same statistics now for the Blaster worm points to the benign centroid for all the cluster keys are presented. Figures A.8, A.9, A.10 and A.11 the graphical results of the Sasser worm points to the benign centroid for all the cluster keys are presented.

A.2 Continuous entropy method

In figures A.12, A.13, A.14 and A.15 the graphics of histogram, Q-Q plot, autocorrelation, and power spectral density of the benign points distances to the benign centroid for all the cluster keys of the Continuous entropy method. Figures A.16 and A.17 the graphical statistics of the Blaster worm points to the benign centroid for all the cluster keys are shown. The graphical statistics of the Sasser worm points to the benign centroid for all the cluster keys are found in figures A.18, A.19 and A.20.
A.2. Continuous entropy method

Figure A.1: Excess points graphical statistics of benign points $H^1$

(a) $H^1$ histogram of distances to the benign centroid

(b) $H^1$ Q-Q plot of distances to the benign centroid

(c) $H^1$ Sample autocorrelation of distances to the benign centroid

(d) $H^1$ Power spectral density of distances to the benign centroid
Figure A.2: Excess points graphical statistics of benign points $H^2$
A.2. Continuous entropy method

Figure A.3: Excess points graphical statistics of benign points $H^3$
A.2. Continuous entropy method

Figure A.4: Excess points graphical statistics of benign points $H^4$

(a) $H^4$ histogram of distances to the benign centroid

(b) $H^4$ Q-Q plot of distances to the benign centroid

(c) $H^4$ Sample autocorrelation of distances to the benign centroid

(d) $H^4$ Power spectral density of distances to the benign centroid

Figure A.4: Excess points graphical statistics of benign points $H^4$
A.2. Continuous entropy method

![Figure A.5](image)

(a) $H^1$ histogram of distances to the benign centroid
(b) $H^1$ Q-Q plot of distances to the benign centroid
(c) $H^1$ Sample autocorrelation of distances to the benign centroid
(d) $H^1$ Power spectral density of distances to the benign centroid

Figure A.5: Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^1$
A.2. Continuous entropy method

Figure A.6: Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^3$
A.2. Continuous entropy method

Figure A.7: Excess points graphical statistics of Blaster worm distances to the benign centroid for $H^4$
A.2. Continuous entropy method

Figure A.8: Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^1$

(a) $H^1$ histogram of distances to the benign centroid

(b) $H^1$ Q-Q plot of distances to the benign centroid

(c) $H^1$ Sample autocorrelation of distances to the benign centroid

(d) $H^1$ Power spectral density of distances to the benign centroid
A.2. Continuous entropy method

Figure A.9: Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^2$
A.2. Continuous entropy method

Figure A.10: Excess points graphical statistics of Sasser worm distances to the benign centroid for $H^3$
A.2. Continuous entropy method

Figure A.11: Excess points graphical statistics of Sasser worm distances to the benign centroid for $H_4^4$
A.2. Continuous entropy method

Figure A.12: Continuous entropy graphical statistics of benign points $H^1$
A.2. Continuous entropy method

Figure A.13: Continuous entropy graphical statistics of benign points $H^2$
A.2. Continuous entropy method

Figure A.14: Continuous entropy graphical statistics of benign points $H^3$
A.2. Continuous entropy method

(c) H^4 Sample autocorrelation of distances to the benign centroid

(d) H^4 Power spectral density of distances to the benign centroid

Figure A.15: Continuous entropy graphical statistics of benign points H^4
A.2. Continuous entropy method

Figure A.16: Continuous entropy graphical statistics of Blaster worm distances to the benign centroid for $H^2$
Figure A.17: Continuous entropy graphical statistics of Blaster worm distances to the benign centroid for $H^4$
A.2. Continuous entropy method

Figure A.18: Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^1$
A.2. Continuous entropy method

Figure A.19: Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^2$
A.2. Continuous entropy method

Figure A.20: Continuous entropy graphical statistics of Sasser worm distances to the benign centroid for $H^3$
Bibliography


IEEE International Conference on Cyber Technology in Automation, Control, and

by using new information metrics,” IEEE Transactions on Forensics and Security,
vol. 6, no. 2, pp. 426–437, June 2011.


AD20040501.html

data

Structural, Neural and Fuzzy Logic Approaches, ser. Series in machine perception


& Sons, 2006.


[22] W. L. Martinez and A. R. Martinez, Computational Statistics Handbook with

Springer, 2002, no. XXIX.

pattern recognition applications in cancer diagnosis,” SOCPAR ’09, pp. 448–453,
December 2009.

