UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA


# TESIS DOCTORAL

## Reconocimiento de anomalías para la detección de instrucciones en nuevos escenarios de monitorización

## Anomaly recognition for intrusion detection on emergent monitoring environments

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Jorge Maestre Vidal

Directores

Luis Javier García Villalba
Ana Lucila Sandoval Orozco

Madrid
Ed. electrónica 2019

# Reconocimiento de Anomalías para la Detección de Intrusiones en Nuevos Escenarios de Monitorización

# Anomaly Recognition for Intrusion Detection on Emergent Monitoring Environments


Thesis by

**Jorge Maestre Vidal**

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

**Luis Javier García Villalba**
**Ana Lucila Sandoval Orozco**

Facultad de Informática
Universidad Complutense de Madrid

Madrid, 2018

# Anomaly Recognition for Intrusion Detection on Emergent Monitoring Environments


Thesis by

**Jorge Maestre Vidal**

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

**Luis Javier García Villalba**
**Ana Lucila Sandoval Orozco**

Facultad de Informática
Universidad Complutense de Madrid

Madrid, 2018

# Reconocimiento de Anomalías para la Detección de Intrusiones en Nuevos Escenarios de Monitorización


**TESIS DOCTORAL**

*Memoria presentada para obtener el título de*
*Doctor por la Universidad Complutense de Madrid*
*en el Programa de Doctorado en Ingeniería Informática*

**Jorge Maestre Vidal**

*Directores*

**Luis Javier García Villalba**
**Ana Lucila Sandoval Orozco**

Facultad de Informática
Universidad Complutense de Madrid

Madrid, 2018

Tesis Doctoral presentada por el doctorando Jorge Maestre Vidal en la Facultad de Informática de la Universidad Complutense de Madrid para la obtención del título de Doctor por la Universidad Complutense de Madrid en el Programa de Doctorado en Ingeniería Informática.

*Título*:
**Reconocimiento de Anomalías para la Detección de Intrusiones en Nuevos Escenarios de Monitorización**

*Doctorando*:
**Jorge Maestre Vidal** (jmaestre@ucm.es)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, España

*Directores*:
**Luis Javier García Villalba** (javiergv@fdi.ucm.es)
**Ana Lucila Sandoval Orozco** (asandoval@fdi.ucm.es)

Esta tesis doctoral ha sido realizada dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-ICT-2014-2/671672-SELFNET).

Dissertation submitted by Jorge Maestre Vidal to the *Facultad de Informática* of the *Universidad Complutense de Madrid* in Partial Fulfillment of the Requirements for the Degree of *Doctor por la Universidad Complutense de Madrid en el Programa de Doctorado en Ingeniería Informática.*

*Title*:
**Anomaly Recognition for Intrusion Detection on Emergent Monitoring Environments**

*PhD Student*:
**Jorge Maestre Vidal** (jmaestre@ucm.es)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain

*Advisor*:
**Luis Javier García Villalba** (javiergv@fdi.ucm.es)
**Ana Lucila Sandoval Orozco** (asandoval@fdi.ucm.es)

This work has been done within the Group of Analysis, Security and Systems (GASS, `http://gass.ucm.es/`), Research Group 910623 from the Universidad Complutense de Madrid (UCM) as part of the activities of the research project funded by the European Commission Horizon 2020 Programme under Grant Agreement number H2020-ICT-2014-2/671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks).

*La presente tesis está dedicada a mi familia:*
*Mi madre - María del Carmen Vidal Valtuille*
*Mi padre - José María Maestre Vega*
*Mi hermano - Marcos Maestre Vidal*
*Mi hermano - Diego Maestre Vidal*

# Agradecimientos

A mi madre María del Carmen.
*Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser quien soy, por todo el amor y apoyo incondicional que me brinda día a día.*

A mi padre José María.
*Por los ejemplos de perseverancia, esfuerzo y constancia que lo caracterizan, por el valor que me ha inculcado para salir adelante, por transmitirme la pasión de aprender y tratar de comprender el entorno que me rodea.*

A mis hermanos Diego y Marcos.
*Por estar conmigo y apoyarme siempre, por los buenos ratos compartidos, por haber aprendido juntos a seguir el camino correcto.*

A mi compañera Yaiza.
*Por quererme, por soportarme y por darme la energía necesaria en los peores momentos. Por estar conmigo en aquellos momentos en que el estudio y el trabajo ocuparon todo mi tiempo y esfuerzo.*

A mis amigos Marco, Lorena y Leonardo.
*Por nuestro apoyo mutuo, por los buenos momentos que hemos compartido y por los que faltan por compartir.*

Esta tesis doctoral ha sido realizada dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-ICT-2014-2/671672-SELFNET).

# Índice General

# Índice de Figuras

# Índice de Tablas

# Lista de Acrónimos

| | |
|---|---|
| AD | *Anomaly Diagnosis* |
| APAP | *Advancer Payload Analyzer Preprocessor* |
| APC | *Antigen-Presenting Cell* |
| AIS | *Artificial Immune System* |
| ANN | *Artificial Neural Network* |
| ARIMA | *Auto Regressive Integrated Moving Average* |
| AUC | *Area Under the Curve* |
| BF | *Bloom Filter* |
| CaaS | *Cybercrime-as-a-Service* |
| CBF | *Countering Bloom Filter* |
| CIDF | *Common Intrusion Detection Framework* |
| CIDS | *Collaborative Intrusion Detection Systems* |
| COBIT | *Control Objectives for Information and Related Technologies* |
| CVE | *Common Vulnerabilities and Exposures* |
| CVSS | *Common Vulnerability Scoring System* |
| DARPA | *Defense Advanced Research Projects Agency* |
| DBSCAN | *Density-Based Clustering Based on Connected Regions with High Density* |
| DCA | *Dendidric Cell Algorithm* |
| DDOS | *Distributed Denial of Service Attacks* |
| DIDS | *Distributed Intrusion Detection Systems* |
| DOS | *Denial of Service* |

| | |
|---|---|
| DPI | *Deep Packet Inspection* |
| DTW | *Dynamic Time Warping* |
| ECM | *Eigen Co-occurrence Matrix* |
| ELM | *Extreme Learning Machines* |
| EM | *Expectation-Maximization* |
| ENISA | *European Union Agency for Network and Information Security* |
| FLS | *Fuzzy Logic System* |
| FN | *False Negatives* |
| FP | *False Positives* |
| GA | *Genetic Algorithm* |
| GIDO | *General Intrusion Detection Object* |
| GMM | *Gaussian Mixture Model* |
| HIDS | *Host-based Intrusion Detection System* |
| HMM | *Hidden Markov Model* |
| IDMEF | *Intrusion Detection Message Exchange Format* |
| IDS | *Intrusion Detection Systems* |
| IDWG | *Intrusion Detection Working Group* |
| IETF | *International Engineering Task Force* |
| IOT | *Internet of Things* |
| IPS | *Intrusion Prevention System* |
| ISO | *International Organization for Standarization* |
| ITIL | *Information Technology Infrastructure Library* |
| LCS | *Longest Common Subsequence* |
| LDOF | *Local Distance-based Outlier Factor* |
| LOF | *Local Outlier Factor* |
| MANET | *Mobile Ad Hoc Networks* |
| MIDS | *Mixed Intrusion Detection Systems* |
| MMS | *Multimedia Messaging Service* |

| | |
|---|---|
| NBA | *Network Behavioral Analysis System* |
| NBSC | *Never-Before-Seen Command* |
| ND | *Nature Diagnosis* |
| NFV | *Network Function Virtualization* |
| NFV&O | *NFV Management and Orchestration* |
| NIDS | *Network-based Intrusion Detection System* |
| NK | *Natural Killer Cell* |
| NL | *Nested Loop Algorithm* |
| NTP | *Network Time Protocol* |
| OVAL | *Open Vulnerability and Assessment Lenguage* |
| P2P | *Peer-to-Peer* |
| PCA | *Principal Component Analysis* |
| PCAP | *Application Package File Format* |
| PHMM | *Profile Hidden Markov Model* |
| PIT | *Passive IP Traceback* |
| PMS | *Packet Management Service* |
| PPS | *Probabily Proportional to Size* |
| QOE | *Quality of Experience* |
| QOS | *Quality of Service* |
| RAT | *Remote Access Tool* |
| SDN | *Software Definer Networking* |
| ROC | *Receiver Operating Characteristic* |
| ROF | *Resolution-based Outlier Factor* |
| SOM | *Self-Organizing Map* |
| SON | *Self-Organizing Network* |
| SRM | *Structural Risk Minimization* |
| SSE | *Sum of Squared Errors* |
| SVM | *Support Vector Machine* |

| | |
|---|---|
| TN | *True Negatives* |
| TP | *True Positives* |
| XSS | *Cross-Site Scripting* |
| WIDS | *Wireless-based Intrusion Detection System* |

## Resumen

La protección de la información y el ciberespacio se ha convertido en un aspecto esencial en el soporte que garantiza el avance hacia los principales desafíos que plantean la sociedad de la información y las nuevas tecnologías. Pero a pesar del progreso en esta área, la eficacia de los ataques dirigidos contra sistemas de la información ha aumentado drásticamente en los últimos años. Esto es debido a diferentes motivos: en primer lugar, cada vez más usuarios hacen uso de tecnologías de la información para llevar a cabo actividades que involucren el intercambio de datos sensibles. Por otro lado, los atacantes cada vez disponen de una mayor cantidad de medios para la ejecución de intentos de intrusión. Finalmente, es de especial relevancia la evolución de los escenarios de monitorización. Este hecho es propiciado por el avance tecnológico, dando lugar a sistemas de cómputo mucho más complejos, con mayor capacidad de procesamiento y que son capaces de manejar información masiva proporcionada por fuentes de diferente naturaleza.

Por lo tanto, los nuevos sistemas defensivos deben hacer frente al análisis de una mayor cantidad de información, cuyas características son mucho más variables y heterogéneas. A este problema se le añaden los desafíos que ya planteaban los sistemas de detección de intrusiones convencionales, como la evolución de las técnicas intrusivas de evasión, o los nuevos retos de la sociedad de la información, como su accesibilidad universal o la salvaguarda de la privacidad de los usuarios. La consecuencia directa de estos cambios es que los resultados obtenidos al evaluar los sistemas de detección de intrusiones convencionales no presentan coherencia con los resultados obtenidos al operar sobre entornos de monitorización reales, ya sujetos a las características imbuidas por las nuevas tecnologías.

Las diferentes organizaciones para la seguridad de la información se han hecho eco de este problema, alertando de la necesidad de disponer de herramientas de detección mucho más eficaces, y adaptadas a las nuevas amenazas. Con el fin de contribuir al estudio de sus causas y plantear soluciones, el trabajo de investigación realizado se centra en una de las estrategias de detección de intrusiones con mayor impacto en la última década: el reconocimiento de anomalías. Las propuestas que implementan estas técnicas analizan comportamientos discordantes observados en el entorno de monitorización, y asumen que, si alguna de estas observaciones difiere de manera significativa de su modo de uso habitual, es indicadora de una posible intrusión. A lo largo de esta tesis se ha llevado a cabo una revisión de los esquemas de detección de anomalías más relevantes, analizando sus características y las consecuencias de su operatividad sobre escenarios actuales. A continuación se han estudiado diferentes casos de uso, para los cuales se ha llevado a cabo la identificación de las dificultades inherentes al despliegue de estos sistemas, indagándose en sus causas y el cómo han sido afrontados por la comunidad investigadora. Finalmente se han introducido nuevos principios de diseño, metodologías y estrategias de detección capaces de mitigar estos problemas.



**Palabras clave**: Inteligencia Artificial, Reconocimiento de Anomalías, Sistemas de Detección de Intrusiones.

## Abstract

The security on information and cyberspace has become a fundamental component of the support that guarantees progress towards the main challenges posed by the information society and the new technologies. But despite progress in this research field, the effectiveness of the attacks against information systems has increased dramatically in recent years. This is due to different reasons: firstly, more and more users make use of information technologies to carry out activities that involve exchanges of sensitive data. On the other hand, attackers dispose an increasable amount of means for executing intrusion attempts. Finally, is of particular relevance the evolution of the protected environment, which is fostered by technological advances, hence giving rise to much more sophisticated computer systems, with greater processing capacity and which are able to handle massive information provided by sources of varying nature.

Therefore, the new defensive schemes must address the analysis of a greater amount of data, which much more variable and heterogeneous characteristics. The problem is further compounded by the challenges already posed by the conventional intrusion detection systems, among them the evolution of the evasion techniques, or issues related with the new goals of the information society, such as bringing universal accessibility to the protected environments or safeguarding their privacy policies. A direct consequence of these difficulties is that the results obtained when evaluating conventional intrusion detection systems are not consistent with the results obtained when operating in real monitoring environments, which are already subject to the characteristics imbued by the emergent scenarios.

The different organizations for information security have echoed this problem, warning that it is needed to develop much more effective detection tools, adapted to this context. In order to contribute to the study of their causes and propose solutions, our research focuses on one of the intrusion detection strategies with greatest relevance at the last decade: anomaly recognition. The different approaches that implement these techniques are based on the analysis of discordant behaviors observed at the monitoring environment, and they assume that if any of these observations differ significantly from their habitual usage mode, it is indicative of a possible intrusive action. Throughout this thesis an in-depth review of the most important anomaly detection strategies, analyzing their characteristics and the consequences derived from operating on recent monitoring environments, has been carried out. In addition, the study of several use cases in order to identify they particular challenges and risks inherent in their deployment on these scenarios is performed, which includes the analysis of their causes and how they have been addressed by the research community. Finally, novel design principles, methodologies and detection strategies to mitigate these inconveniences have been introduced.

**Keywords**: Anomaly Recognition, Artificial Intelligence, Intrusion Detection Systems.

# Parte I

# Resumen de la investigación en inglés

# Chapter 1

# Introduction

Intrusion Detection Systems (IDS) are defensive elements that monitor and analyze events on information systems for signs of potentially malicious activities. In their early stages they were based on recognition of patters related with previously known threats, which the research community termed signature-based intrusion detection [LLL13]. But the rapid proliferation of new attack strategies, as well as the appearance of tens of thousands of new threats, led to the need of developing detection methods capable of identifying novel intrusive actions. Among them it is worth noting for their relevance in the bibliography, the anomaly-based intrusion detection systems. They implement discordant behavior analysis techniques and assume that, if any of the performed observations significantly differ from the usual and legitimate usage of the protected environment, they indicate potential intrusion processes. This methodology played a crucial role in the defensive schemes of the last decade [BBK14]. Consequently, there is currently a wide variety of proposals that implement its basis. Anomaly-based intrusion detection modus operandi is usually oriented towards modeling the legitimate activities observed in the protected environment; if the studied samples differ representatively from the previously built models, they are tagged as anomalous, hence notifying the incident to the security administrators or the decision-making components.

## 1.1 Research problem

But despite progress in the intrusion detection field, the effectiveness of the attacks against the information systems has increased dramatically in the past few years. The different organizations for information security have echoed this problem, hence alerting about the need for developing more effective defensive tools, which must be adapted to the new threats but also to the recent monitoring environments. For example, the European Union Agency for Network and Information Security (ENISA) highlighted the increase in damage caused by these attacks, which is mainly related to an enhancement in their ability to go unnoticed, and the exploitation of new services or vulnerabilities [ENI15]. Other sources, such as the different Emergency Response Teams (CERT), also coincide with this observation. Among them the European division (CERT-EU) periodically alerts

about novel malware specimens, concluding that in general terms, there is a growing tendency to cause a greater impact on the victim systems and escalating their propagation capabilities to more specialized scenarios [CE18]. From the private sector, information security organizations such as Symantec [Sym16], ISACA [ISA15] or McAfee Labs [Lab16] also noticed their evolution. In their incident reports agreed on the alarming increase in the number of never seen before threats and the growth of the losses they cause.

In general terms, they all coincide in that this is due to different reasons: firstly, more and more users relay on information technologies to carry out activities that involve exchanging sensitive data, such as electronic commerce, remote healthcare or accomplish administrative procedures. It is often that with this purpose, users apply technologies with which they are not completely familiar, that make them more susceptible to become victims of frauds, scams or identity thefts. On the other hand, attackers increasingly dispose a greater amount of means for launching intrusion attempts. This has led to the emergence of a new business model, popularly known as Cybercrime-as-a-Service (CaaS). Each merchant in CaaS offers a range of products that support the intrusion capabilities of the attacker, among them zero-day vulnerabilities, bulletprof, tracking, fake/stolen credentials or money laundering. Usually each salesperson specializes in the development of a type of potentially malicious technology, having to offer more efficient products in order to increase their possibilities to occupy a market space between their competitors. One of the most dangerous characteristics of CaaS is that it has a feedback relationship with the appearance of new threats [Man13]; for example, when the exploitation of a malware specimen ceases to be profitable, its developers offer for sale the original source code. This results in the appearance of new strains that strengthen the vulnerable aspects of the original program, and which may include additional functionalities. As the European Police Office (Europol) warned, behind most of the attacks in recent years are large criminal organizations capable of funding the initial investments and get the most benefit from contracted CaaS, whose mitigation implies the need to organize and coordinate the defense mechanisms deployed by the different agencies and organizations for information security [Eur16].

Another key circumstance to be taken into account is the evolution of the monitoring environments. This fact is favored by the technological advances, giving rise to much more complex computer systems, with greater processing capabilities and that are able to handle information provided by a vast variety of sources of different nature, being good examples of these progresses the fifth generation networks (5G), virtualization, ubiquitous computing, cloud computing or the Internet of Things (IoT). Therefore, the new defensive deployments must address the problem of analyzing a greater amount of information, which have much more variable and heterogeneous characteristics. This problem is compounded by the challenges already posed by the conventional intrusion detection systems, as is the case of the adaptation of the intrusive techniques or dealing with the new challenges on the information society, among them guarantee universal accessibility or safeguard user privacy. The direct consequence of these innovations is that the results obtained when evaluating traditional IDS through methodologies functionally standardized by the research community are not consistent with the results obtained

when operating in real monitoring environments, which are already subject to constraints inherited from technological advances and legal restrictions [Zim14, BSMT14, MVK+15]. This tendency has acquired protagonist in the recent publications, leading the research community to question the effectiveness of the predominant methodologies and design principles [HSB+12].
In order to contribute to the study of the causes and solutions derived from the adaptation of the anomaly recognition for intrusion detection on the emergent monitoring environments, this thesis reviews their principal approaches and difficulties, proposing new design principles, strategies and detection schemes. From the obtained results it is possible to observe that the adaptation of the conventional detection methods is possible, even in spite of the difficulties that the new scenarios pose. With the purpose of facilitating the understanding of the performed research, the following sections summarize its main motivations, objectives, contributions and the organization of the rest of the document.

## 1.2 Motivation

Nowadays anomaly-based intrusion detection becomes an indispensable element of the defensive schemes that try to ensure information security. Unlike signature-based methods, these systems require the acquisition of specific knowledge about the monitoring scenarios on which they are deployed. This allows elaborating images of the characteristics of their habitual mode of use and hence identifying discordant behaviors based on them. However, advances in information technologies have led to significant changes in these scenarios, which entail the need to deal with very different data and their modeling based on novel features. Consequently, with the purpose of introducing new threats, attackers have also been able to take advantage of these changes, which at present are able to go unnoticed across traditional intrusion detection schemes.
Another important consequence is the lack of coherence between the results obtained by intrusion detection systems when validated by methodologies functionally standardized by the research community, and those in real monitoring environments. In other words, it is often not feasible to deploy anomaly-based detection systems on recent monitoring scenarios, even if in the past they demonstrated impressive behaviors on obsolete evaluation frameworks (i.e. hit rates close to 100% and false positive rates close to 0%). There are different reasons that lead to this situation, among them high battery consumption, inability to operate in real-time on large volumes of data, tendency to issue more false positives or vulnerability against evasion techniques. The main motivations of this research are the study of the circumstances leading to this situation and the proposal of new methodologies capable of adapting the essential ideas of the traditional anomaly-based intrusion detection systems to the emergent monitoring scenarios, in this way demonstrating that they remain effective, and that they are still crucial elements to guarantee information security.

## 1.3 Goals

Bearing in mind the state of the art about information security and how the recent monitoring environments implement novel technologies, the research described throughout this document raises a main objective: the study of anomaly-based intrusion detection systems when operating on emergent monitoring scenarios. With this purpose three fundamental tasks have been proposed. The first one is the in-depth review of the most relevant anomaly recognition schemes in the bibliography, in this way analyzing their principal features and the consequences of them operating on recent scenarios. The second task is to identify the difficulties and risks inherent in these use cases, which entails investigating their causes and how the research community has dealt with them. Finally, new design principles, methodologies and detection strategies capable of reducing the impact of their deployment on these environments are introduced. The effectiveness of the proposals when operating in some of their most common applications is demonstrated. In order to achieve the main goal of this research and fulfill each one of the aforementioned tasks, the following activities have been carried out in the course of the investigation:

1. Identification of the different types of threats, accuracy assessment of the different proposals for their mitigation, and selection of the uses cases on which they must be studied.

2. Extraction and evaluation of the most common features of the techniques applied for evading defensive systems, as well as analysis of the difficulties posed by the selected monitoring scenarios.

3. Selection of a set of reference anomaly-based intrusion detection systems bearing in mind their popularity and effectiveness on experimental scenarios.

4. Development of methods/frameworks for attack recognition considering the most representative aspect of the emergent monitoring scenarios.

5. Implementation of evasion techniques against anomaly-based intrusion detection systems and emulation of the difficulties on emergent monitoring scenarios.

6. Evaluation of the accuracy of the proposed strategies when identifying threats in recent monitoring environments, and discussion of the obtained results based on the effectiveness of the conventional intrusion detection systems.

7. Evaluation of the behavior of the proposed solutions at the selected use cases.

8. Conclusions and proposition of future research steps.

## 1.4 Contributions

The main contributions of the performed research are organized as illustrated in Figure 1.1, where a central block of contents sets the grounds for the introduced proposals and five

use cases are studied: recognition of insiders by masquerade detection, payload analysis on communication networks, malware identification on ubiquitous computing, mitigation of denial of service attacks on fifth generation networks and correlation of alerts reported by payload-based malware detection systems.

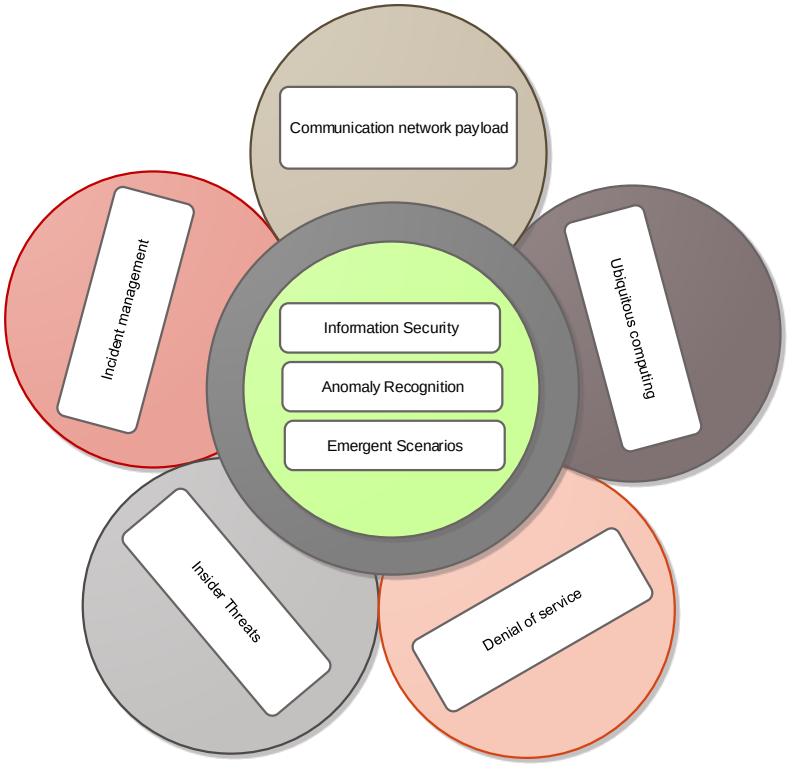



Figure 1.1: Principal contributions as blocks of contents.

The contents on the central block are almost theoretical and review the principles of information security, intrusion detection, anomaly recognition and the challenges on the recent monitoring scenarios. Based on this study, in [MVSOGV15a] a survey about the deployment of intrusion detection systems on datacenters is proposed. On the other hand, the contributions related to defense against internal attackers begin with [MVSOGV14], which proposes a bio-inspired strategy for identifying masquerade attacks. In order to gain robustness against imitation-based evasion methods the aforementioned strategy is refined and optimized in [MVSOGV16].

The contributions related to malware recognition in networks have their origin in [GVSOMV15], where a detection method based on analyzing traffic payload is proposed. This strategy is reviewed and improved in [GVMVSO17], where it adapts to the heterogeneity of recent networks by adopting some of the strategies introduced by the PAYL sensor family. In order to further improve its behavior by reducing its false positive rate, a correlation strategy based on quantitative criteria is proposed [MVSOGV15b]. It is the first research included in the alert correlation block of contents, where [MVSOGV15c, MVSOGV17] were added later as a framework for much more generic purpose incident management.

To date, contributions in anomaly-based intrusion detection on ubiquitous computing

are concentrated in [MVSMGV18], where a method for identifying malicious contents in applications, based on the study of their booting activities is proposed. An extended version of this approach was published in [MVSMGV18]. Finally, an adaptive artificial immune network for mitigating DoS flooding attacks on fifth generation networks is introduced in [MVSOGV18, MVSOGV15d].

## 1.5 ORGANIZATION

This thesis is organized as follows:

**Chapter 1** introduces the problem of anomaly-based intrusion detection adaptation to new monitoring scenarios by reviewing several aspects of the performed research: motivation, goals, contribution and the present organization of the rest of the document.

**Chapter 2** delves into intrusion detection systems and the impact of their deployment on classical monitoring scenarios. With this purpose, the following topics are studied: characteristics of the security on information technologies, general purpose architectures for IDS, their main features and taxonomies, and how the incidents they report are managed.

**Chapter 3** reviews every aspect of anomaly recognition that must be taken into account for understanding the classical intrusion detection approaches and the introduction of new proposals. These include the definition of the term anomaly and its scope, the different types of anomalies, the mechanisms for acquiring knowledge for anomaly recognition, the distances and similarity measures most used with this purpose, the impact of the non-stationary monitoring environments in this process, metrics and methodologies for their evaluation.

**Chapter 4** studies the major challenges on anomaly recognition emphasizing those directly related to the emergent monitoring scenarios. This chapter also describes the use cases in which the research done is focused, and the advances of the research community to accommodate the conventional detection strategies to their more specific characteristics.

**Chapter 5** is centered on the problems posed by adversarial attacks in anomaly-based intrusion detection on local environments. For their mitigation, a novel detection method and a suspicious event processing scheme are proposed. Their effectiveness is demonstrated by extensive experimental results and their comparison with the proposals in the bibliography.

**Chapter 6** explores the problem of anomaly-based intrusion detection when analyzing traffic from current communication networks. In particular, it focuses at strategies based on the inspection of packet payload. As alternative, a novel method capable of precisely operating in this context that presents consistency with the results obtained when it is evaluated by functional standards is introduced.

**Chapter 7** addresses the difficulties involves in managing the incidents reported by anomaly-based intrusion detection systems. In order to facilitate its adaptation to the new challenges, a framework for alert correlation on anomalous payload-based intrusion detection is proposed.

**Chapter 8** introduces a novel strategy inspired by the immune processes of living beings

for detection, mitigation and identification of the source of flooding denial of service attacks at fifth generation networks.
Finally, **Chapter 9** brings together conclusions and proposals for future work derived from the performed research.
This document also contains an annex describing the most refereed anomaly recognition methods. With the purpose of facilitating the understanding of the rest of the document, it emphasizes the strategies that are mentioned in previous chapters, as well as their different taxonomies. The following four major families of techniques are thoroughly reviewed: modeling-based, proximity-based, clustering, and statistical hypotheses.

# Chapter 2

# Contributions of the research

This chapter summarizes the main contributions and/or findings of each chapter that describes the performed research.

## 2.1 Chapter 2: Security and intrusion detection systems

The protection of the information and the cyberspace became essential for progressing towards bring effective solutions to the main challenges posed by the information society and the emergent technologies. The intrusion detection systems, as part of the set of tools necessary for its safeguarding, become increasingly important at the different security schemes and defensive strategies. Since their deployment and organization depend on such models, identifying their main characteristics goes hand in hand with the circumstances, policies and limitations that predetermine their elaboration. In order to introduce the reader to the context in which their design, implementation, deployment and evaluation are framed, this chapter introduces the Intrusion Detection Systems (IDS), and the impact of their deployment on current monitoring environments.

Their proper effectiveness requires to analyze in depth the scenarios to be protected, as well as to be able to identify and evaluate the threats to which they may potentially be subjected. Therefore, a greater specialization in this area of research and applications, representatively improves the effectiveness of the defensive deployments. As a result, and with a view of optimizing tasks related with protect organizations and their corporate objectives, information security tends to implement models, standards, and methods for the assessment of their potential risks and threats. This drives the design, configuration and implementation of intrusion detection systems to be heavily influenced to the coverage of the security model for which they operate, being classical and well-known schemes like the ISO/IEC 27001:2013 series or NIST-SP800 reviewed thorough this chapter. With the purpose of facilitate the reader understanding of the existing solutions, also the Common Intrusion Detection Framework (CIDF) is detailed, which components somehow are taken into consideration in recent contributions.

From the standardization of the general-purpose intrusion detection schemes to the present, the research community has published a great number of proposals related to

their improvement and optimization. This has been propitiated by the rapid evolution and adaptation of the attacks to the new technologies, the increase in their sophistication, growth in variety and the ease provided by some of the recent offensive tools, which allow perpetrating threats without the attackers presenting a profile with advanced knowledge about information security. With the purpose of expedite tasks related to the identification and deployment of those defensive strategies which may best behave in each use case, the research community has tried to organize all this knowledge through taxonomies and ontologies. Some of these classifications have existed for more than a decade, but despite their seniority, they became a very important reference for new researchers. The performed revision of the bibliography allowed to identify three common elements in the different taxonomies: detection strategy, monitoring scenario and architecture; which establish the greatest differences between publications. In particular, two main methods for intrusion detection are distinguished: signature recognition and anomaly recognition. They are often combined to take advantage of their benefits and minimize their negative impact on the protected system. In the bibliography, it is also common to consider other more specific categories, such as WIDS (Wireless-based Intrusion Detection Systems), NBA (Network Behavior Analysis systems) or MIDS (Mixed Intrusion Detection Systems systems). However, within the framework of the performed research they are considered subsets of HIDS (Host-based Intrusion Detection Systems) and NIDS (Network-based Intrusion Detection Systems) approaches, so the chapter focuses on these two main groups and their possible combinations. Finally, the architecture of IDS is influenced by the characteristics of the deployment scenario and the information it must consider. Its components could be grouped in a single point, or expanded throughout the monitoring environment, which entails the need to propose a more sophisticate design and to define communication processes between them. It worth to highlight the simplicity of centralized designs versus the complexity and adaptability on the collaborative Intrusion Detection Systems (CIDS). With a view to introduce the reader to these design paradigms, the three architectures most adopted in the bibliography are described: centralized, distributed and hierarchical.

## 2.2 CHAPTER 3: ANOMALY RECOGNITION

The problem of identifying anomalies has been object of study for decades, where instead of the term “anomaly”, equivalent expressions like “discordant observation” were considered. Consequently, the word anomaly has been replaced by equivalent concepts over the years, being also referred as outliers, isolated parts, exceptions, aberrations, surprises, peculiarities or polluting elements. The use of each of these tags usually varied according to the domain in which it has been used, just as it happened with their definition. In order to familiarize the reader with this concept, at this chapter some of the most popular anomaly definitions in the bibliography are collected, as well as the areas of research strongly related to the anomaly concept and their identification, which have risen to discussions about whether they should be considered as subtopics of the outlier detection field, among them noise elimination and accommodation, novelty detection, identification

of change points or trend discovery.
Bearing in mind the current disagreement of the research community on how it defines the term anomaly and its scope, it is difficult to generalize a classification that facilitates their distinction. From the different ontologies presented in the bibliography, this chapter merges some of the reviewed classifications, thus stablishing three large sets of outliers: point anomalies, contextual anomalies and collective anomalies. Accordingly, point categories bring together all those particular instances of discordance with respect to the rest of the monitored data. Bearing in mind the contextual anomaly (also referred to as conditional anomaly), a data instance is anomalous in a specific context, but not otherwise; but when a set of observations presents divergence regarding the expected behavior, they are termed as collective anomalies.
The nature of the data that anomaly recognition methods analyze for defining what is normal and what is discordant, restraints the selection of the strategies to be implemented and affects their behavior. In the scope of intrusion detection, the greater distinction can be identified from sample labeling, where normal is legitimate, and anomalous is potentially malicious, being these the most frequent classifications in the previous publications. Bearing this in mind, two essential observation necessary to understand the different strategies for acquiring knowledge and to decide the most appropriate for each use case are considered: firstly, it must be taken into account that labeled samples are very difficult to obtain, a situation that can be aggravated if the sensor operates in real-time, and/or on monitoring environments where it is not possible to extract all the data; on the other hand, there is no way of collecting malicious samples that cover every possible attack, so the system is potentially vulnerable to unknown threats. Anomaly detection methods are often classified based on the nature of the reference samples, which correspond to the principal paradigms on machine learning: supervised, semi-supervised, unsupervised, reinforcement, transduction and multi-task learning; which are reviewed thorough the chapter. On the other hand, anomaly detection systems often must compare the perception of "normal" elements made during their knowledge acquisition stage with the observations being analyzed. Therefore, the characteristics of the distances and measures of similarity that are taken into consideration affect directly to their effectiveness.
Finally, it is important to highlight that most of the machine learning and datamining methods, especially those that are oriented to anomaly recognition, assume the premise that the collections of reference data present stationary distributions. They also assume that the information they analyze is gathered from a monitoring environment of static characteristics, a situation that is not always satisfied in their deployment at real use cases. This can lead to unrealistic and unpredictable results. With the aim of introducing the reader to the challenges posed by these varying scenarios, the chapter describes the causes that lead to representative changes in the monitored data, as well as their consequences on anomaly recognition and the different strategies developed for their mitigation. In addition, a collection of well-known evaluation methodologies and effectiveness indicators applied on anomaly recognition systems is reviewed.

## 2.3 Chapter 4: Challenges and emergent communication environments

The evolution of the monitoring scenarios has been propitiated by technological advances. This has led to the emergence of much more sophisticated computing systems, with greater processing capabilities and which are able to handle the information provided by a wide variety of sources. Consequently, the new proposals in the field of anomaly recognition must deal with a greater amount of information of much more heterogeneous characteristics when operating on these monitoring environments. This problem is compounded by challenges already addressed by the traditional outlier detection schemes, such as ability of certain attackers to evade them, or the difficulties on satisfy the new agreements of the information society, such as universal accessibility or safeguarding privacy of users. This chapter identifies five illustrative emergent scenarios where anomaly recognition plays a major role.

### 2.3.1 Masquerade detection

A pair of well-known insider attackers have usually widely studied in the bibliography: masqueraders and traitors. Every legitimate user trying to gain privileges for accessing restricted assets belongs to the traitor group. On the other hand, masqueraders are unauthorized external users that are somehow able to impersonate authorized users. The in-depth review of the literature allows deducing that, with the exception of masquerade detection approaches based on biometrics, the bulk of the studied publications focused on analyzing the behavior of users on the protected environment and modeling their habitual and legitimate activities. This facilitates the implementation of anomaly recognition looking for malicious events. In general terms, the major concerns of the research community lie at improving the hit rate of the sensors, reducing the number of false positives reported, and more recently in strengthening them against imitation-based evasion methods.

### 2.3.2 Payload-based malware detection on communication networks

In the last decade, the anomaly-based detection of malware on communication networks through traffic payload analysis became an essential measurement for new specimen identification. Consequently, nowadays there is wide variety of proposals that adopt this paradigm. Their modus operandi usually lays on modeling habitual activities performed by legitimate users at the monitored environment. When the payload of the analyzed traffic differs representatively from the models it is considered anomalous, hence reporting the possible threats. After an in-depth study of the literature, it is remarkable the observed need to propose solutions that are easy to adapt to the heterogeneity and the volume of information inherent in these emergent monitoring environments. It also required the development of robust strategies against evasion methods, which must address the problem of malware obfuscation.

### 2.3.3 Alert correlation

Another important aspect to consider when deploying IDS is how they will manage the alerts issued. Under normal conditions they tend to report thousands of events in short periods of time. From the viewpoint of a human operator, their analysis is unfeasible if there are not mechanisms that facilitate their classification. On the other hand, when the response process is automated, the huge amount of reports can dramatically affect the quality of service of the monitored environments, as well as the effectiveness of their analysis. The management of the alerts issued by IDS poses an interesting scenario, where the information to be analyzed is not directly acquired from the monitoring environment, but from reports received from intrusion detection systems deployed on the different security perimeters. According to the bibliography, this is a research area mainly focused on complementing the information provided by sensors, hence facilitating decision-making tasks and the deployment of more effective countermeasures. However, and despite the extensive bibliography reviewed during the performed research, no proposals were found specifically focused on anomaly-based sensors.

### 2.3.4 Denial of Service attacks

By definition, Denial of Service (DoS) has the objective of disabling computer systems or networks. In recent years, the number of incidents related with these threats reported by the various organizations for cyber defense shows an alarming growth. After reviewing the state of the art related with the defense against DoS attacks, it is important to emphasize the importance of anomaly recognition in tasks involved at their detection, as well as the insistence of the research community on improving their effectiveness. This motivated the publication of a large number of approaches, which are usually separated into detection, mitigation, prevention and identification of the origin of the malicious actions. However, few of them meet all the requirements to be effective in real use cases, emphasizing the needs of high true positive rates, unrepresentative false positive rates, low consumption of computational resources and real-time performance. In addition, it is noteworthy that proposals in the literature seldom consider advantages of the new trends on networking. It is expected that the emerging networks, taking the example of 5G, increasingly move towards self-management.

### 2.3.5 Malware in mobile devices

Over recent years a significant growth in the popularity of mobile devices was observed. This places smartphones directly in the line of fire for cyber criminals. The revision of the bibliography leads to highlight different aspects concerning the defense against malware on mobile technologies. The most relevant is the impact of the resource limitation on ubiquitous technologies. This is because most anomaly modeling and recognition strategies are computationally costly, either in terms of performance or memory consumption. This prevents analytical algorithms from taking advantage of their full potential, forcing the restriction of the selection of fit parameters, and therefore their effectiveness. In some publications, this is solved by deriving part of the data processing tasks to external services.

Another aspect to be noted is the tendency to implement sandboxes; detection methods based on static traits and metadata allow discarding a significant part of the malware, but the most evasive specimens are detected at runtime, which makes sandboxing an accurate solution. But the use of this strategy involves an additional penalization in terms of resource consumption, which may have impact on other capabilities of the detection strategy.

## 2.4 CHAPTER 5: MASQUERADE DETECTION ROBUST AGAINST MIMICRY

The proposal described throughout this chapter is focused on enhancing the effectiveness of the conventional masquerade detection strategies based on the analysis of action sequences by considering the anomaly-based pattern recognition paradigm. The monitored behaviors are modeled and classified according to local sequence alignment algorithms. The labels of the analyzed samples are validated on the statistical non-parametric U-test proposed by Mann-Whitney. Through this it is possible to refine the labeling of sequences to avoid making hasty decisions when their nature is not sufficiently clear. In order to strengthen their effectiveness against mimicry attacks, the analysis of the monitored sequences is performed in concurrency. This involves partitioning long sequences with two purposes: making subsequences of small intrusions more visible and analyzing new sequences when suspicious situations occur, such as the execution of never before seen commands or the discovery of potentially harmful activities. Its main goals are increasing their hit rate, reducing their false positive rate and improving their straightening against imitation-based adversarial attacks.
For the evaluation of the approach, the SEA dataset published by Schonlau et al. has been implemented. This decision is based on that despite the controversy that involves its use, most of the previous similar approaches considered it. Thus, it is possible to compare the obtained statistics with other approaches based on the action sequence analysis The obtained results were compared with those of the previous proposals. Both, (1) the best base sequencing and (2) the best parallel sequencing configuration were considered. Bearing this in mind the following items are highlighted:

1. The results in terms of accuracy are located among the most effective proposals. In this way, it is demonstrated that the introduced approach is intrinsically a good alternative.

2. The best base sequencing setting (1) is more accurate when analyzing the original samples of SEA (TPR=0.98, FPR=0.007) than the best parallel sequencing implementation (2). In particular, it presents the lowest FPR of the proposals to compare, and the third-highest TPR. This is mainly due to the incorporation of the refinement step by applying the online verification method based on U-test. Nevertheless, the experimentation proved its low capacity for detecting mimicry attacks. In particular, its TPR is 0.568. These are similar results to the best proposals in the bibliography.

3. The best parallel sequencing configuration (2) has demonstrated to be more sensitive when analyzing legitimate activities. It retains a high TPR, but this time the FPR is higher than most of the previous works. Although this is it the case, it presents a great ability to identify mimicry attacks, with TPR=0.872. This makes it a more robust alternative, which is specially required in certain use cases.

Therefore, two configurations which facilitate the adaptation of the proposed method to different uses cases have been deducted. The first of them (1) is recommended for protecting organizations which contain assets with moderate values and that must provide high QoE (Quality of Experience). In this case the hit rate when detecting masqueraders is high and the false positive rate is unrepresentative, so that users/operators are not continuously bothered with the need to manage false notifications. In this setting, the consumption of computing resources considers a single line of processing, which optimizes performance and connects the memory and processing consumption with the implementation of the local sequence algorithm. Notwithstanding, this configuration is more vulnerable to mimicry attacks. On the other hand, the second implementation (2) has proved very effective against evasion attempts based on imitation. But this has been achieved at the expense of increasing the consumption of computing resources and the false positive rate. The acquired robustness is necessary in use cases where critical or sensitive information is protected.

## 2.5 Chapter 6: Malware detection by traffic paylaod analysis

In the last decade, the malware detection on communication networks through anomaly-based statistical analysis of traffic payload has become an essential countermeasure against the rapid proliferation of the recent specimens. But its deployment has been the subject of controversy on part of the research community, opening the debate on its possible consequences when operating on current networks. In order to contribute to the adaptation of these technologies to the new monitoring scenarios, this chapter introduces a new intrusion detection method for communication networks based on anomaly recognition, which is able to operate effectively on both functional standardized evaluation frameworks and real monitoring environments. The proposal brings together all the advantages of signature-based detection, because it is deployed as a preprocessing module of the well-known NIDS Snort; in addition, it performs anomaly recognition, which is achieved by identifying patterns that represent the usual and legitimate network usage. It is based on the well-known modification of PAYL termed ANAGRRAM, and like most of the members within the PAYL family, it considers two main processing steps: training and detection: at the training phase, a statistical model of the legitimate traffic is created through Bloom filters applying the N-gram technique. By comparing this model with that of several known attacks it is possible to create a rule set that allows detecting the anomalies present in the traffic of the protected network. On the other hand, at the detection stage the traffic to be analyzed is compared with the legitimate traffic model

generated at training, according to the rules stated at the previous level. This allows determining if the observed payload has potentially harmful contents.
The NIDS related with malware detection tend to be evaluated based on two criteria. Firstly, in order to facilitate their comparison with previous publications they adopt the functional standard DARPA'99. But this dataset is obsolete and subject of controversy, so their effectiveness is also evaluated according to custom but real traffic. In the case of the proposal a dataset of real traffic has been collected from traffic samples provided by the datacenter of the University Complutense of Madrid (UCM).
The proposal performed with similar accuracy to the best configurations in the bibliography when dealing with DARPA'99. In particular, it is one of three proposals with 100% accuracy in the recognition of true attacks (as is the case of ANAGRAM and AnPDPP) and its false positive rate is very close to 0.15%, only outperformed by ANAGRAM, but with the inherent benefits of the introduced design (scalability, rule refinement, etc.). Despite the good results obtained, the results on DARPA'99 are not scalable to recent networks, principally because that their heterogeneity leads to a significant increase in the false positive rate of the detectors, so a second evaluation stage was required.
The average of the obtained results on real traffic demonstrated 94.75% hit rate and 0.8075% false positive rate. As expected, these results represent a slightly decrease in the detection capabilities of the sensor regarding DARPA'99, which is mainly due to the grown in heterogeneity of the monitoring scenario. However, the deployment of the proposal remains viable on new communication networks, and it is easily upgradeable by varying the rule sets. In general terms, the performed experimentation demonstrated that the proposal provides greater consistency when operating in real scenarios than most of the PAYL family members refereed in the bibliography. Therefore, the proposed adaptation of Bloom filters and N-gram to the payload-based malware detection problem, as well as the introduced smoothing of K-values and strategies for rule generation, have serve to improve the effectiveness of the PAYL family to this emergent monitoring scenario.

## 2.6 CHAPTER 7: FRAMEWORK FOR ALERT CORRELATION AT ANOMALY BASED NIDS

Intrusion detection based on identifying anomalies typically emits a large amount of reports about the malicious activities monitored; hence information gathered is difficult to manage. In this chapter, an alert correlation system capable of dealing with this problem is introduced. The work carried out has focused on the study of a particular family of sensors, namely those which analyze the payload of network traffic looking for malware. Unlike conventional approaches, the information provided by the network packet headers is not taken into account. Instead, the proposed strategy considers the payload of the monitored traffic and the characteristics of the models built during the training of such detectors, in this way supporting the general-purpose incident management tools. The approach is motivated by the need to complement the NIDS APAP (*Advanced Payload Analyzer Preprocessor*), which was previously described at Chapter 6. Note that as an

initial attempt of improving the alert management capabilities of the proposed sensor, techniques similar of those published in the bibliography were considered. But they are grounded on studying information mainly provided by packet headers and traffic flows, such as IP addresses, ports, protocols, services, duration of the communications, etc. which completely ignored the information analyzed by APAP: packet payload, models involved in decision-making, or triggered detection rules.

Bearing this in mind, two analysis components are distinguished: Anomaly Diagnosis components (AD) and Nature Diagnosis components (ND). AD components perform correlation by analyzing the degree of anomaly on the samples which triggered the emission of alerts. The main idea of this analysis is that the larger the difference of the payload on detected packets regarding the legitimate payload model, the greater the possibility of containing real threats. On the other hand, ND components correlate incidences by taking their nature as reference point. Both criteria are important, and before deploying countermeasures, they must be studied together. This is because in certain situations, to consider only one of them can lead to the implementation of insufficient or disproportionate actions. For example, AD components could be sure that an incidence is a real threat. But if its nature presents low risk, it is not advisable to give it high priority treatment. In the opposite case, an ND component could report highly dangerous content within a packet. But if it has low degree of abnormality, there is a high chance that it leads to the issuance of a false positive, situation that must be taken into account by operators.

The effectiveness of the proposal has been demonstrated in a real use case, when it is deployed for complementing APAP. It has been evaluated by analyzing alerts reported when monitoring traffic on the subnet of the faculty of Computer Science, at the University Complutense of Madrid (UCM), were promising results were demonstrated. In particular, the AD component was able to successfully filter about 95.7% of the false positives with 1% maximum system overload. The precision obtained when comparing alerts correlated by the ND component at packet level demonstrated 99.512% hit rate with 2.3% maximum system overload. Finally, when evaluating the ND component at sequence-level, it was important to bear in mind that it provided non-deterministic results. As with the previous component, all alerts that have not participated in training steps are analyzed. 100% of them have been correlated successfully in some of the possible natures proposed. Though successes have been distributed between the first two more probable options proposed, being a 75.124% in the most likely and 24.876% in the second most probable. This is the desired behavior because it sanitizes grouping errors and poses other alternatives. But in this case, the performance was worse (22.88% maximum overhead).

## 2.7 Chapter 8: DDoS mitigation at emergent communication networks

Denial of service attacks pose a threat in constant growth. This is mainly due to their tendency to gain in sophistication, ease of implementation, obfuscation and the recent improvements in occultation of fingerprints. On the other hand, progress towards self-organizing networks and the different techniques involved in their development, such as software-defined networking, network-function virtualization, artificial intelligence or cloud computing, facilitates the design of new defensive strategies, more complete, consistent and able to adapt the defensive deployment to the current status of the network. In order to contribute to their development, in this chapter the use of artificial immune systems to mitigate denial of service attacks is proposed. The approach is based on building networks of distributed sensors suited to the requirements of the monitored environment, where different actors assume the various roles of the biological immune systems. Its success depends mainly on two types of agents spread along the protected network: $H$ detectors ($D_H$) and $A$ detectors ($D_A$). $H$ are involved in the innate immune response and at the adaptive response. Consequently, they are capable of recognizing and blocking new attacks, as well as cooperating in the construction of the immunological memory. On the other hand, $D_A$ have the ability to detect and mitigate the attacks previously identified by $D_H$ assuming a very important role in the adaptive response.

As in biological systems, the innate immunity on the approach is the first line of the defense strategy. It aims to identify and mitigate new threats and protect $H$ detectors of disablement by flooding. The process of innate immunity requires maintaining activated $D_H$ agents along the protected network. These agents implement VNFs as IDS that monitor the entire traffic flowing through them looking for suspicious anomalies. Therefore, detected attacks must present certain evident characteristics related to considerable fluctuations in the analyzed traffic distribution. Once a threat is identified, the mitigation measures consist mainly on the adoption of directives that restrict the communications with nodes, ports or services involved in the attack vector. The innate response provides quick and efficient countermeasures, requiring no communication with the Orchestrator prior to their launch. By recognition and elimination of pathogens before they enter into the system, the proposal innate response emulates the behavior of the immune system of human beings. This is because it acts in the same way as the various external physical barriers or cells, and without specificity. In addition, it should be noted how agents involved in this task act coincide with those of most conventional Intrusion Prevention Systems (IPS), i.e. IDS with the ability to apply basic countermeasures.

The adaptive response is the next defensive step in the proposal. It is triggered every time a $D_H$ agent recognizes a new threat, which implies that they must hold at least a short memory capable of storing their latest decisions. In this context, determining when an attack is Non-Seen-Before (NSB) implies it is not presence in the immunological memory. Once the adaptive reaction is released, the $D_H$ that identified the attack sends activation signals to the $D_A$ agents in close proximity via the control/management plane of the Orchestrator. Then the activated $D_A$ agents instantiate VNFs that analyze traffic flowing

through them. Unlike $D_H$, their detection engines increase restrictiveness in proportion to the flood of the attack, usually acting stricter than $D_H$. In this way, it is prevented that the division of the attack flow reach the victim by alternative routes, hence assuming that when it is split, it becomes less noisy and hence more difficult to be detected. In order to prevent this measure resulting in a substantial increase in the false positive rate, specificity is taken into account. To ensure specificity, they are only able to apply countermeasures against the threat that has activated them, which imply that a specific VNF with analysis capabilities is instantiated for each discovered threat. Therefore, they can only act against several attacks if they have been activated to mitigate each of them. While the threat persists, the immune response remains activated. If it is no longer visible, a quarantine period is activated. The quarantine is interrupted only upon detection of replicas of the intrusion (implying back to the previous state), or when the countdown expires. The network segments covered by a set of $D_A$ agents that are active against a specific threat and coordinated by the same $D_H$ sensor, are their quarantine region.
The proposal was evaluated in the grounds of KDD'99, CAIDA'07 and DDoSIM/UCM traffic, proven its effectiveness when compared when previous publications. To evaluate the effectiveness of the deployment of the AIS, a simulator capable of generating traffic distributions and different networks with different locations of $D_H$ and $D_A$ has been implemented. This is because none of the functional standards for the evaluation of similar systems provides a complete knowledge of the organization of various networks. From them the following findings should be highlighted: 1) the proposed AIS poses a significant improvement in the cases where conventional solutions have more difficulties to operate adequately. 2) the higher the power of the attacks, the greater the noise caused, and therefore the threats are easier to detect. 3) When the traffic density is low, the proposed strategy is very accurate, as is the case of the conventional schemes. However, when congestion is high, especially above 0.7, the hit rate decreases, and the false positive rate increases. In the same way as in the previous tests, this problem is reduced by the activation of the adaptive response, thus outperforming conventional solutions. 4) the larger the protected network, more relevant is the improvement obtained with this proposal.

# Chapter 3

# Conclusions and future work

## 3.1 Conclusions

Intrusion detection based on anomaly recognition became a fundamental element of the current information security strategies. However, advances on information technology, as well as the emergent of new black-market models, have led to the evolution of the conventional monitoring scenarios. Consequently, most of the classic detection strategies turned obsolete, being necessary their upgrading and evaluation according to the demand of the new information security stage. Throughout the research performed in the context of this thesis, we have deepened the causes that brought to the loss of effectiveness of the emergent technologies. With this motivation, a comprehensive study of the bases for their design and implementation has been accomplished, which required revising norms, frameworks, models, strategies, paradigms and their scope from the point of view of security management. The concept anomaly, its most habitual interpretations, recognition methods and how it is interpreted and adapted to the different circumstances have also been reviewed. This empowered the identification of a series of common difficulties in operating at emergent monitoring scenarios, which will undoubtedly be part of the main challenges to be addressed by the research community in the coming years. Among them high false positive rates in heterogeneous monitoring environments, lack of an universal approach to anomaly recognition problem, strengthening against evasion methods, adaptability to scenarios with variable characteristics, difficulty when proposing/finding evaluation methodologies and adequate reference dataset, discrepancy in the choice of distances and similarity measures, and soaring computing resource consumption; the latter being a particularly important issue in ubiquitous computing. With the purpose of more clearly illustrating these problems, the emphasis has been placed on five use cases that significantly represent their impact: detection of masquerade attacks and their obfuscation by imitation; anomaly-based malware recognition on the payload of communication networks; incident management and correlation; mitigation of flooding-based denial of service attacks in new generation networks; and malware recognition on mobile devices. All of them have been studied in detail, and in the case of the first four topics, novel approaches have been introduced.

The first proposal has as main objective to improve the conventional masquerade attack detection methods. To this end, different techniques are applied to optimize their accuracy, reducing the false positive rate, facilitating real-time event analysis, and gaining robustness against imitation-based attacks. It is based on the local sequence analysis of actions performed by users of the protected system, construction of behavioral profiles and their comparison looking for anomalies. The proposal has been designed bearing in mind the challenges in providing strengthening against mimicry attacks. To this end, a novel strategy based on analyzing the extracted information in concurrency is introduced, which distinguishes two data processing layers: base and parallel sequencing. The first is continuous throughout the decision stage and analyzes every monitored action, and the second initiates new sequences to be analyzed when suspicious events are identified. The reduction of the search space in the new sequences facilitates the identification of the actions perpetrated by masqueraders disguised in actions that try to imitate the legitimate behavior models.

As a second proposal, a novel network-based intrusion detection system for recognition of unknown threats (zero-day attacks) has been introduced. This is done through a detailed statistical analysis of the binary contents of the payload. Its information processing tasks involve data extraction by the N-gram methodology and its management by Bloom filter structures. The approach is based on the family of detectors inspired by PAYL, and it aims on strengthening two of their most controversial aspects: the high rate of false positives when analyzing actual traffic and their difficulties at model updating. Unlike its predecessors, the approach extends the training process and facilitates the generation of rule sets. With this purpose, countering Bloom filters and the generation of K-values for defining decision thresholds are implemented. The latter optimize the system workload by identifying the most representative contents of payload.

On the other hand, an alert correlation framework aimed on the management of incidents reported by anomaly-based malware detection on communication network payloads has been introduced. Its development was motivated by facilitating the understanding of the outputs of the aforementioned NIDS, in this way considering the analytic and modeling methodologies of the sensors to be complemented. The approach poses a multi-layered architecture where both individual layers and sequences of them are studied. It assumes a pair of similarity criteria: the nature of the incidents and their degree of discordance (i.e. likelihood of being true). Henceforth the proposed framework only takes into account features inherent to the discordances discovered by the sensors, and it is allowed to be integrated into general purpose incident management schemes.

With the purpose of contributing in the defense against DDoS flooding threats, a strategy for their detection and mitigation was introduced. It implied the deployment of a sensor network that integrates an Artificial Immune System (AIS) inspired by the biological defense mechanisms of human beings. Unlike similar proposals, conventional bio-inspired methods for pattern recognition were not applied. Instead a combination of strategies for DDoS detection based on the study of variations of the entropy on the network traffic by thresholding, with the adaptation of the biological immune reactions is proposed. This makes it possible to apply real-time countermeasures, building an immune memory and

establishment of quarantine areas, all in accordance with the current state of the protected network. Another relevant contribution is the novel method for detecting DDoS deployed by the immune agents, which was able to forecast anomalies on the entropy of the traffic analyzed, and thereby recognizing flooding attacks. This was achieved by representing the entropy as time series of observations and the definition of prediction intervals. The preliminary experiments showed promising results, which motivates the development of a cooperative deployment strategy. This is the second main contribution on this research field, where a method for management of immune agents that implements the previously described detection method was proposed. Within this, the decisions are made as to when and how they will act and in what level of restriction, all this depending on the status of the network and orchestrated by an artificial immune approach.
With the motivation of facilitating the comprehension of the performed research, every contribution has been evaluated and tested according to functional standardized methodologies and real monitoring environments. The obtained results were extensively discussed and illustrated by figures, tables and other explanatory solutions, so their comparison with the main features of previous proposals in the bibliography is possible. It is important to highlight that regardless the promising results obtained, they suggested different improvable aspects, on which it has been investigated throughout the documentation in order to motivate the undertaking of future related research lines.

## 3.2 Future work

The following described the principal future research lines derived from the performed investigation:

### 3.2.1 Anomaly recognition on mobile devices

The problem of recognizing discordant behaviors in mobile devices focused on discovering malware downloaded from application distribution markets is one of the described use cases that illustrates the problem of adapting conventional detection methods to the emergent scenarios. Although this object of study is reviewed in depth in the first chapters of this document, the development and publication of a sound proposal for its mitigation is an ongoing task; the approach in which we are currently working is based on the methods of sequence alignment and the study of the system calls executed by the monitored applications, thus importing some of the contribution in Chapter 5 for masquerade detection.

### 3.2.2 Anomaly recognition for ransomware detection

Ransomware is defined as any kind of malware that blocks part of the functionalities on the victim system and demands a payment (ransom) for their recovery; only then promises restoring them. Despite its apparent simplicity, its modus operandi has grown alarmingly in the last years, being in the focus of most of the organizations for information security. The anomaly recognition plays an essential role in the fight against this threat,

since the methods reviewed throughout this research facilitate the discovery of the victim enumeration processes, asset cyphering steps and their deletion. Therefor it is a current problem that poses another interesting emergent monitoring scenario.

### 3.2.3 Anomaly recognition for biometrics-based access control

Another topic of interest in the field of anomaly recognition is its adaptation to access control systems based on biometrics. Given the resistance of these technologies to counterfeiting attempts, the use of biometric features is increasingly frequent in certain scenarios: recognition of handwritten signatures and their dynamics, voice, keyboard usage patterns, touchpad movements, etc. However, they are highly dependent on the reference dataset with which they are trained, hence often requiring constant updating. This feature can significantly penalize the quality of user experience, which is continually subjected to new tests to detect changes in its biometric features. The in-depth study of this problem, as well as the elaboration of concrete methods for its palliation based on the identification of discordant characteristics is a topic of interest that should be dealt in future research works.

# Parte II

# Descripción de la investigación

# Capítulo 1

# Introducción

Los sistemas de detección de intrusiones o IDS (del inglés *Intrusion Detection Systems*) son elementos defensivos que monitorizan y analizan los eventos observados en sistemas de la información en busca de indicios de actividades potencialmente maliciosas. En sus inicios, la detección de intrusiones se basaba en el reconocimiento de patrones relacionados con amenazas previamente conocidas, a lo que la comunidad investigadora denominó detección de intrusiones basada en firmas [LLL13]. Sin embargo, la rápida proliferación de nuevas estrategias de ataque, así como la aparición de decenas de miles nuevas amenazas, llevó a la necesidad de desarrollar métodos de detección capaces de reconocer procesos de intrusión desconocidos. De entre ellos cabe destacar las estrategias de detección de intrusiones basadas en el reconocimiento de anomalías. Éstas implementan técnicas de análisis de comportamientos discordantes observados en el entorno de monitorización, y asumen que si alguna de estas observaciones difiere de manera significativa de su modo de uso habitual y legítimo, es indicadora de una posible acción intrusiva. Esta metodología ha demostrado jugar un papel crucial en los esquemas defensivos de la última década [BBK14]. En consecuencia, actualmente existe una gran variedad de propuestas que la implementan. Su modus operandi habitualmente se orienta hacia el modelado de las actividades legítimas observadas en el entorno monitorizado; cuando las muestras analizadas difieren representativamente de estos modelos, se consideran anómalas, notificándose las incidencias.

## 1.1 Problema de investigación

A pesar del progreso en la detección de intrusiones, la eficacia de los ataques dirigidos contra sistemas de la información ha aumentado drásticamente en los últimos años. Las diferentes organizaciones para la seguridad de la información se han hecho eco de este problema, alertando de la necesidad de disponer de herramientas defensivas mucho más eficaces, y adaptadas a las nuevas amenazas. Por ejemplo, la Agencia de Seguridad de las Redes y de la Información de la Unión Europea (ENISA), ha destacado el aumento del daño causado por estos ataques, hecho que principalmente relaciona con el incremento de su capacidad de pasar desapercibidos y la explotación de nuevos servicios [ENI15].

Otras fuentes, como es el caso de los Equipos de Respuesta ante Emergencias Informáticas (CERT), también coinciden con esta observación. Por ejemplo, la división europea CERT-EU periódicamente comparte avisos de nuevos especímenes de malware, en los que se observa una tendencia a cada vez causar un mayor impacto, y en entornos mucho más especializados [CE18]. Desde el sector privado, organizaciones y empresas orientadas a la seguridad de la información, como Symantec [Sym16], ISACA [ISA15] o McAfee Labs [Lab16] también dejan constancia de esta evolución. En sus informes de incidencias coinciden en el alarmante incremento del número de nuevas amenazas, y el crecimiento de las pérdidas que ocasionan.

Esto es debido a diferentes motivos: en primer lugar, cada vez más usuarios hacen uso de tecnologías de la información para llevar a cabo actividades que involucren el intercambio de datos de especial sensibilidad, como comercio electrónico, consultas a historiales clínicos o trámites administrativos. Para ello, a menudo deben de valerse de tecnologías con las que no están del todo familiarizados, lo que les vuelve mucho más susceptibles de llegar a convertirse en víctimas de fraudes, estafas o robos de identidad. Por otro lado, los atacantes cada vez disponen de una mayor cantidad de medios para la ejecución de intentos de intrusión. Esto ha dado lugar a la aparición de un nuevo modelo de negocio, conocido popularmente como cibercrimen como servicio o CaaS (del inglés *Cybercrime-as-a-Service*). El CaaS es un modelo emergente, en el que cada comerciante ofrece una serie de productos, tales como vulnerabilidades sin explotar, protección, ocultación de huellas, acceso a credenciales robados, o facilidades para el blanqueo de capitales. De este modo, cada vendedor se especializa en el desarrollo de un tipo de tecnología maliciosa, debiendo ofrecer un producto cada vez más eficaz, y por lo tanto con más posibilidades de ocupar un espacio de mercado entre tanta competencia. Una de las características más peligrosas del CaaS, es que presenta una relación de retroalimentación con la aparición de nuevas amenazas [Man13]. Por ejemplo, cuando la explotación de un espécimen de malware deja de ser rentable, sus desarrolladores ponen a la venta el código fuente. Esto deriva en la aparición de nuevas cepas que fortalecen los aspectos vulnerables de su esquema original, y que añaden funcionalidades adicionales. Tal y como advirtió la unidad para la seguridad de la información de la Oficina Europea de Policía (Europol) [Eur16], detrás de la mayor parte los ataques registrados en los últimos años se esconden grandes organizaciones criminales capaces de financiar toda esta inversión y generar el mayor beneficio a partir CaaS contratado, cuya mitigación implica la necesidad de organizar y coordinar los mecanismos de defensa de las diferentes agencias para la seguridad de la información.

Otro aspecto fundamental a tener en cuenta es la evolución de los escenarios de monitorización. Este hecho es propiciado por el avance tecnológico, dando lugar a sistemas de cómputo mucho más complejos, con mayor capacidad de procesamiento y que son capaces de manejar la información proporcionada por gran cantidad de fuentes de diferente naturaleza, hallándose claros ejemplo de ello en las redes de quinta generación 5G, virtualización, computación ubicua, computación en la nube e Internet de las cosas o IoT (del inglés *Internet of things*). Por lo tanto, los nuevos sistemas defensivos deben hacer frente al análisis de una mayor cantidad de información, cuyas características son

mucho más variables y heterogéneas. A este problema se le añaden los desafíos que ya planteaban los sistemas de detección de intrusiones convencionales, como la evolución de las técnicas intrusivas previamente mencionada, o los nuevos retos de la sociedad de la información, como su accesibilidad universal o la salvaguarda de la privacidad de los usuarios. La consecuencia directa de estos cambios es que los resultados obtenidos a partir de la evaluación de los IDS convencionales, por medio de metodologías estandarizadas por la comunidad investigadora, no presenten coherencia con los resultados obtenidos al operar sobre entornos de monitorización reales, ya sujetos a las características imbuidas por las nuevas tecnologías y las restricciones legales [Zim14, BSMT14, MVK+15]. Esta tendencia ha ido a más en los últimos años, llevando a la comunidad investigadora a cuestionar la eficacia de sus principales metodologías y principios de diseño [HSB+12]. Con el fin de contribuir al estudio de las causas y soluciones derivadas de la adaptación de los sistemas de reconocimiento de anomalías a estos nuevos entornos de monitorización, esta tesis revisa sus principales aproximaciones y dificultades, planteando nuevos principios de diseño, metodologías y estrategias de detección. A partir de los resultados obtenidos es posible observar cómo la adaptación de los métodos de detección convencionales es viable, incluso a pesar de las dificultades que plantean los nuevos escenarios. Con el fin de facilitar la comprensión del trabajo realizado, las siguientes secciones resumen nuestras principales motivaciones, objetivos fijados, principales contribuciones, y la organización detallada del resto del documento.

## 1.2 Motivación

En los últimos años, la detección de intrusiones basada en anomalías se ha convertido en un elemento indispensable de los esquemas defensivos que tratan de garantizar la seguridad de la información. A diferencia de la detección basada en firmas, estos sistemas requieren de la adquisición de conocimiento específico acerca de los escenarios de monitorización sobre los que van a ser desplegados. Esto permite elaborar una imagen de las características de su modo de uso habitual, y a partir de ella identificar comportamientos discordantes. Sin embargo, los avances en las tecnologías de la información han dado lugar a cambios representativos en estos escenarios, lo que conlleva la necesidad de tratar con datos muy diferentes, y la elaboración de modelos en base a sus nuevas características. Como consecuencia de esta evolución, los atacantes también han sido capaces de aprovechar estos cambios para introducir nuevos tipos de amenazas, las cuales ahora son capaces de pasar desapercibidas ente los esquemas de detección de intrusiones tradicionales.
Otra importante consecuencia de estos cambios es la falta de coherencia entre los resultados que obtienen los sistemas de detección de intrusiones al ser validados por medio de conjuntos de evaluación estandarizados funcionalmente por la comunidad investigadora, y su despliegue en casos de usos reales. Es decir, es frecuente que no sea factible desplegar sobre nuevos entornos de monitorización, sistemas de detección de anomalías que en su momento demostraron un impresionante comportamiento sobre escenarios de monitorización obsoletos (i.e. tasas de aciertos cercanas al 100%, y tasas de falsos positivos próximas al 0%). Existen diferentes motivos que llevan a esta situación, como su elevado

consumo de batería, incapacidad de operar en tiempo real sobre grandes volúmenes de datos, tendencia a la emisión de una mayor cantidad de falsos positivos, o vulnerabilidad frente a técnicas de evasión. Las principales motivaciones de esta investigación son el estudio de las circunstancias que han llevado a esta situación, y el planteamiento de nuevas metodologías capaces de adaptar las ideas esenciales de los esquemas de detección de intrusiones basados en anomalías convencionales, a los nuevos escenarios, demostrando de esta manera que siguen siendo elementos defensivos eficaces e imprescindibles para garantizar la seguridad de la información

## 1.3 Objetivos

Teniendo en cuenta el estado del arte en seguridad de la información y la adaptación de los escenarios de monitorización a las nuevas tecnologías, esta investigación plantea un objetivo principal: el estudio de los sistemas de detección de intrusión basados en anomalías al operar sobre nuevos escenarios de monitorización. Para alcanzarlo se han propuesto tres tareas fundamentales. La primera de ellas es la revisión en profundidad de los esquemas de detección de anomalías más relevantes, analizando sus características y las consecuencias que implican al operar sobre escenarios actuales. La segunda tarea es la identificación de las dificultades y riesgos inherentes a estos casos de uso, indagando en sus causas y el cómo han sido tratados desde la comunidad investigadora. Finalmente, se propondrán nuevos principios de diseño, metodologías y estrategias de detección capaces de reducir el impacto de dichos entornos, demostrándose su eficacia en algunas de sus aplicaciones reales más habituales.
Para alcanzar el objetivo principal de este trabajo y cumplimentar cada una de sus tareas, las siguientes actividades debieron llevarse a cabo en el transcurso de la investigación:

1. Identificación de los diferentes tipos de amenazas, valoración de la precisión de las diferentes propuestas para su mitigación y selección de los casos en que van a ser estudiadas.

2. Extracción y evaluación de las características más comunes de las técnicas aplicadas para la evasión de los sistemas defensivos, así como las dificultades que proponen los entornos de monitorización seleccionados.

3. Selección de un conjunto de sistemas referentes en el reconocimiento de atacantes, en base a su eficacia sobre escenarios experimentales.

4. Desarrollo de marcos para la detección de ataques considerando los aspectos más representativos de dichos entornos monitorizados.

5. Implementación de herramientas para la aplicación de técnicas de evasión en los procesos de intrusión convencionales y/o repetición de las dificultades de los escenarios seleccionados.

6. Evaluación de la precisión de las diferentes propuestas realizadas, en la identificación de ataques en entornos de monitorización actuales, y discusión de los resultados tomando como referencia esquemas de detección convencionales.

7. Valoración del comportamiento de estas propuestas en los diferentes casos de uso reales.

8. Conclusiones e identificación de futuras líneas de investigación.

## 1.4 Contribuciones

Las contribuciones principales de la investigación realizada se organizan tal y como se ilustra en la Figura 1.1, donde un bloque central asienta las bases de las propuestas realizadas, y son cubiertos cinco casos de uso: reconocimiento de atacantes internos enmascarados, análisis de la carga útil del tráfico en redes de comunicaciones, identificación de malware en dispositivos móviles, mitigación de denegación de servicio en redes de quinta generación y correlación de alertas emitidas por sistemas de detección de anomalías.

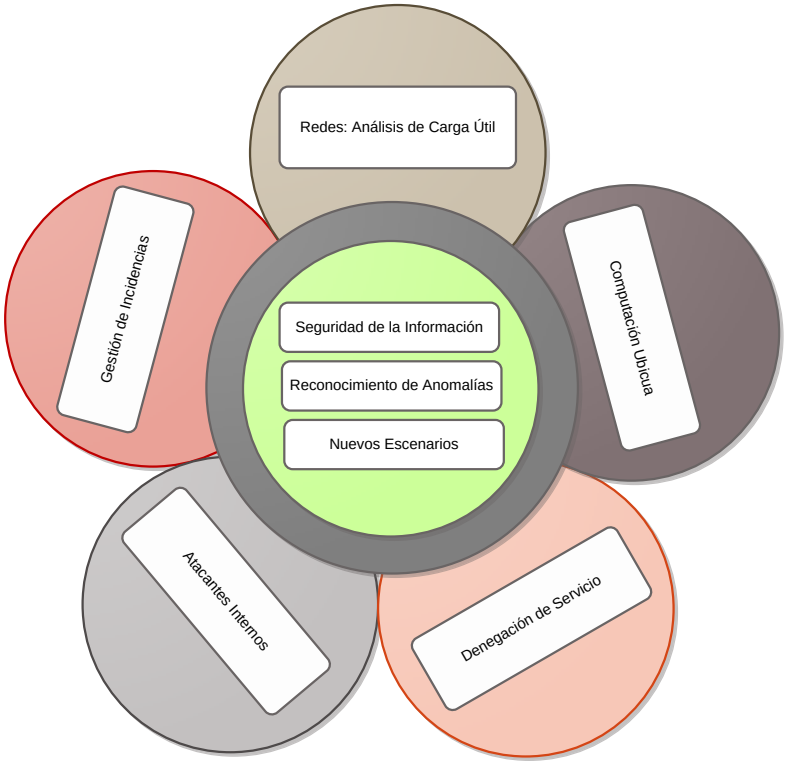



Figura 1.1: Contribuciones según su área de interés.

El contenido del bloque central es principalmente teórico, y en él se revisan los principios de la seguridad de la información, detección de intrusiones, reconocimiento de anomalías y desafíos en nuevos escenarios de monitorización. En base a este estudio, en [MVSOGV15a] se propone una guía para el despliegue de este tipo de tecnologías en centros de procesamiento de datos. Por otro lado, las contribuciones relacionadas con la defensa

contra atacantes internos se inician con [MVSOGV14], donde se propone una estrategia bio-inspirada para la identificación de enmascarados. Esta estrategia es refinada y optimizada en [MVSOGV16], con el fin de ganar robustez frente a métodos de evasión basados en imitación.
Las contribuciones relacionadas con el reconocimiento de malware en redes tienen su origen en [GVSOMV15], donde se propone un método de detección basado en el análisis de su carga útil. Esta estrategia es revisada y mejorada en [GVMVSO17], donde se adapta a las redes actuales adoptando algunas de las estrategias introducidas por la familia de sensores PAYL. Con el fin de mejorar aún más su comportamiento por medio de la reducción de su tasa de falsos positivos, en [MVSOGV15b] se propone una estrategia de correlación basada en criterios cuantitativos. Éste es el primer trabajo incluido en el bloque de correlación de alertas, donde posteriormente se añadiría [MVSOGV15c, MVSOGV17] como marco para la gestión de incidencias de propósito mucho más general.
A día de hoy, las contribuciones en la detección de intrusiones basada en el reconocimiento de anomalías sobre tecnologías ubicuas se concentran en [MVSMGV18], donde se propone un método de identificación de contenido en aplicaciones maliciosas basado en el estudio de sus secuencias de acciones de arranque. Una versión extendida de este trabajo se ha publicado en [MVSMGV18]. Por otro lado, en [MVSOGV18, MVSOGV15d] se propone un esquema de detección y mitigación de ataques de denegación de servicio auto-organizativo inspirado en los mecanismos inmunitarios de la naturaleza.

## 1.5 ORGANIZACIÓN

Esta Tesis se organiza de la siguiente manera:
En el **Capítulo 1** se introduce el problema de la detección de intrusiones basada en anomalías en los nuevos escenarios de monitorización, y se revisan los siguientes aspectos de la investigación realizada: motivación, objetivos, contribución y organización del resto del documento.
En el **Capítulo 2** se profundiza en los sistemas de detección de intrusiones y el impacto de su despliegue en los escenarios de monitorización. Para ellos se estudian las características de la seguridad en las tecnologías de la información, las diferentes estrategias de intrusión, la estructura general de los IDS, sus características y clasificación, y el proceso de gestión de las alertas que generan.
En el **Capítulo 3** revisa todos aquellos aspectos relacionados con el reconocimiento de anomalías que deben ser tenidos en consideración para la comprensión de los esquemas de detección de intrusiones convencionales y la introducción de nuevas aproximaciones. De entre ellos cabe destacar la definición del término anomalía y su ámbito de uso, los diferentes tipos de anomalías, los mecanismos de adquisición de conocimiento para su identificación, las distancias y medidas de similitud más utilizadas con este fin, el impacto de los entornos de monitorización no estacionarios en este proceso, las métricas y metodologías para su evaluación.
En el **Capítulo 4** se revisan los principales desafíos que plantea el reconocimiento de anomalías, haciendo hincapié en aquellos que se relacionan directamente con los nuevos

entornos de monitorización. En este capítulo también se describen los casos de uso en que se centra la investigación realizada, y los avances de la comunidad investigadora para acomodar las estrategias de detección convencionales a sus características más específicas.
En el **Capítulo 5** se centra en la problemática que plantean los ataques de evasión en la detección de intrusiones en escenarios locales. Para su mitigación se propone una nueva estrategia de detección y se introduce un esquema de procesamiento capaz de reconocer dichos intentos de evasión. El método propuesto es respaldado por una amplia experimentación y comparado con aproximaciones anteriores.
En el **Capítulo 6** se profundiza en el problema de la detección de intrusiones basada en el análisis del tráfico en redes de comunicaciones actuales; en particular, en las estrategias basadas en la inspección del contenido de la carga útil de los paquetes capturados. Se propone un nuevo método capaz de operar con precisión en dicho contexto, y presentar consistencia con los resultados obtenidos al ser evaluado por estándares funcionales.
En el **Capítulo 7** se tratan las dificultades de la gestión de incidencias reportadas por sistemas de detección de intrusiones basados en anomalías. Con el fin de facilitar su adaptación a los nuevos desafíos, se propone un marco para la correlación de alertas adaptado a los sensores que monitorizan la carga útil del tráfico de las redes de comunicaciones actuales.
En el **Capítulo 8** introduce una estrategia inspirada en los procesos inmunitarios de los seres vivos para la detección, mitigación e identificación del origen de ataques de denegación de servicio en redes de nueva generación.
Finalmente, el **Capítulo 9** reúne las conclusiones y propuestas de trabajo futuro derivadas de la investigación realizada.
Este documento además contiene un anexo en el que se describen las principales técnicas de reconocimiento de anomalías. Con el fin de facilitar la comprensión del resto del documento, se hace hincapié en las estrategias que son mencionadas en capítulos anteriores, así como es sus diferentes taxonomías. Los siguientes cuatro grandes familias de técnicas son revisadas en profundidad: detección basada en modelado, proximidad, agrupamiento e hipótesis estadísticas.

# Capítulo 2

# SEGURIDAD Y SISTEMAS DE DETECCIÓN DE INTRUSIONES

En este capítulo se introducen los Sistemas de Detección de Intrusiones o IDS, y el impacto de su despliegue en los entornos de monitorización actuales. Con este fin se revisan algunos aspectos imprescindibles para su comprensión, los cuales son organizados en las siguientes secciones: en la Sección 2.1 se definen conceptos básicos relacionados con la seguridad de la información y su gestión; en la Sección 2.2 se describe el objeto de análisis de los sistemas de detección de intrusiones: las amenazas a las que se enfrentan y su impacto; en la Sección 2.3 se revisan los esquemas clásicos de detección de intrusiones; finalmente, en la Sección 2.4 se ofrece una visión general de sus diferentes características.

## 2.1 Seguridad en las tecnologías de la información

La protección de la información y el ciberespacio se ha convertido en un aspecto esencial en el soporte que garantiza el avance hacia los principales desafíos que plantean la sociedad de la información y las nuevas tecnologías. Los sistemas de detección de intrusiones, como parte del conjunto de herramientas necesarias para su salvaguarda, han ido adquiriendo cada vez mayor importancia en los diferentes modelos y estrategias defensivas. Dado que su despliegue y organización dependen de dichos planes, el profundizar en sus principales características va de la mano de conocer las circunstancias, políticas y limitaciones que rigen su comportamiento. Con el fin de introducir al lector en el contexto en el que se enmarcan su diseño, implementación, despliegue y evaluación, esta sección revisa dos aspectos clave para su comprensión: la evolución de la definición y el ámbito de la seguridad en las nuevas tecnologías, y las características de los principales modelos para la gestión de su defensa.

### 2.1.1 Definición y ámbito de la seguridad

En las últimas décadas se han utilizado diferentes términos para representar las medidas de protección de la información y sus tecnologías. Como indica C. Paulsen [Pau16], la existencia de tantas definiciones a menudo genera confusión incluso entre los profesionales

con mayor trayectoria en el sector. Con el fin de esclarecer el ámbito de cada término y su relación con el resto, C. Paulser representó la relación que existe entre estas definiciones tal y como se ilustra en la Figura 2.1.

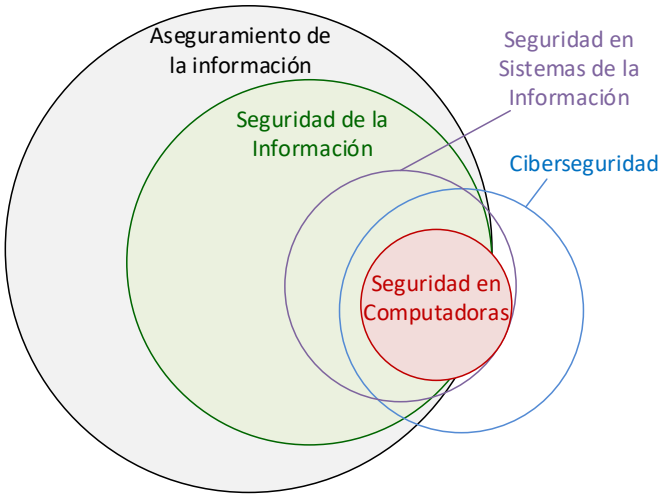



Figura 2.1: Ámbito de las definiciones de seguridad en la sociedad de la información y nuevas tecnologías.

- *Seguridad en computadoras* (del inglés *computer security*). Si bien el término seguridad en computadoras fue formalmente acuñado y descrito en 1970 [War70], ya estaba presente en discusiones e informes registrados a partir de 1950 [Pau16]. Su uso se popularizó en los años 80 al relacionarse directamente con el modelo de gestión de seguridad basado en la tríada Confidencialidad (del inglés *Confidentiality*), Integridad (del inglés *Integrity*) y Disponibilidad (del inglés *Availability*), habitualmente identificado como CIA-triad (ver Sección 2.3 "Estructura general de un IDS"). De hecho, según la definición actual que contempla en NIST (National Institute of Standards and Technology), la seguridad en computadoras (a la que también denominan COMPUSEC) abarca medidas orientadas a garantizar la confidencialidad, integridad y disponibilidad de los activos en sistemas de la información, como hardware, software, firmware o los propios datos [Kis13]. Dado el limitado ámbito que abarca y las restricciones del modelo CIA-triad en que se basa, en la actualidad la expresión seguridad en computadoras cada vez es menos utilizada por la comunidad investigadora.

- *Seguridad de la información* (del inglés *information security*). El término seguridad de la información fue propuesto por J.P. Anderson con el fin de diferenciar la protección de los datos en sí, de la defensa de los elementos físicos encargados de su gestión [And72]. Según la definición aceptada por el NIST, la seguridad de la información es la protección de los datos y los sistemas de información, de intentos de acceso no autorizado, divulgación, alteración o destrucción con el fin de garantizar su confidencialidad, integridad y disponibilidad [And72]. A pesar de

que esta definición también se basa en la CIA-triad, es una de las expresiones más utilizadas en la actualidad. Según C. Paulsen esto se debe principalmente a que también se aplica fuera del ámbito de los sistemas basados en computación que almacenan la información [Pau16]. Esto permite considerar un mayor rango de activos, como documentos impresos, grabaciones de audio o video, fotografías, etc. almacenadas de manera tanto analógica como digital.

- *Seguridad en sistemas de la información* (del inglés *information systems security*). No se conoce con certeza el origen de esta expresión, aunque principalmente fue usada entre los años 70 y 80. Según la definición del NIST (donde también se conoce como INFOSEC), la seguridad en sistemas de la información se centra en proteger los sistemas de la información de acceso y modificaciones de datos no autorizadas, tanto en su etapa de almacenado, procesado o tránsito. También implica la defensa de los usuarios frente a intentos de denegación de servicio, y todas las tareas relacionadas con la detección, documentación y mitigación de dichas amenazas, abarcando incluso componentes no electrónicos [Kis13]. En otros ámbitos se conoce como la unión de la seguridad en computadoras (COMPUSEC), redes en comunicaciones (COMSEC) y el control de fugas de datos (TEMPEST). En la actualidad la comunidad investigadora debate abiertamente acerca del solapamiento de funciones entre la seguridad de la información y la seguridad en los sistemas de la información. Si bien esta última es prácticamente un subconjunto de la primera, existen determinados casos en que no se produce dicha intersección; esto es debido a que la seguridad de la información tiene el punto de mira puesto en la información en sí, mientras que la seguridad en sistemas de la información se centra en garantizar que los elementos que trabajan sobre ella se comporten de manera adecuada, para lo que podrían requerirse acciones que no afecten a lo primero.

- *Aseguramiento de la información* (del inglés *information assurance*). Tal y como resalta C. Paulsen [Pau16], a mediados de los años 90 la palabra seguridad se asociaba directamente con la CIA-triad. En aquella época, en los círculos relacionados con el desarrollo de productos comerciales se popularizó la expresión aseguramiento de la calidad (del inglés *quality assurance*). Ésta abarca el conjunto de estrategias, políticas, criterios, métodos de seguimiento de tareas e implementación de estándares para la prevención de errores de producción que pudieran llegar a reducir la calidad del producto final. El aseguramiento de la información es la adaptación de este concepto a la protección de datos y tecnologías involucradas en su gestión. El NIST definió este término como las medidas para proteger la información y los sistemas de información por medio de garantizar su disponibilidad, integridad, confidencialidad y no repudio [Kis13]. Actualmente hace referencia al conjunto de estrategias para preservar el nivel de seguridad en la información de una organización, lo que incluye el desarrollo de políticas alineadas con sus objetivos, la evaluación e identificación de activos, despliegue de mecanismos de monitorización y detección, evaluación del estado del sistema, toma de decisiones y actuación, o la gestión de los informes relacionados con las incidencias detectadas.

- *Ciberseguridad* (del inglés *cybersecurity*). La palabra cibserseguridad es la unión del término seguridad con la palabra griega *cyber*, que literalmente significaba "experto en dirección y gobernancia". A finales de los años 40 se utilizó como prefijo para acuñar la expresión Cibernética (del inglés *Cybernetics*), directamente relacionada con los sistemas de control y la teoría de sistemas [Pau16]. Ésta hace referencia a los esfuerzos científicos para lograr la organización efectiva en un sistema. En los años 80 y 90 se usaba prácticamente cono sinónimo de algo electrónico. A finales de los años 90 y principios del 2000 empezó a utilizarse la palabra ciberseguridad. El significado que popularmente se le daba coincidió con la definición que posteriormente propuso el NIST [182]: "protección y defensa del cibserspacio y ciberataques", donde "cibserspacio" es el dominio global de propagación de información basado en infraestructuras de redes de comunicaciones independientes y sistemas de computación; y los "ciberataques" son vulneraciones de la seguridad de la información haciendo uso del ciberespacio. Por lo tanto, el término ciberseguridad reduce el alcance de las definiciones previamente descritas (seguridad en computadores, seguridad de la información, etc.) al ámbito del ciberespacio.

Nótese que esta investigación se centra principalmente en el ámbito de la seguridad de la información. El trabajo realizado no se extiende hasta acciones propias del aseguramiento de la información, como por ejemplo la definición de políticas de actuación en acuerdo a los objetivos de las organizaciones, o la identificación de sus activos; tampoco se limita al estudio de las amenazas relacionadas únicamente con el ciberespacio, a pesar de que gran parte de los nuevos escenarios tecnológicos descritos guardan una importante relación con su evolución.

## 2.2 Modelado y gestión de la seguridad

Con el fin de que un despliegue de métodos para la seguridad de la información sea efectivo bajo un determinado contexto, es necesario estudiar en profundidad el espacio a proteger y ser capaces de identificar y evaluar las amenazas a las que potencialmente puede verse sometido. Por lo tanto, y tal y como se demuestra en [BAG15], una mayor especialización en esta área de investigación y sus aplicaciones mejora de manera representativa la eficacia de los despliegues defensivos. Debido a esto, y con el fin de optimizar las tareas relacionadas con la protección de una organización y sus objetivos corporativos, la seguridad de la información habitualmente se basa en modelos, estándares, directivas de evaluación, detección y análisis de los riesgos a los que puede ser sometida. En consecuencia, las etapas de diseño, configuración y despliegue de los sistemas de detección de intrusiones son influenciadas por el alcance del modelo de seguridad para el que actúan. Con el fin de introducir al lector en las características de estos esquemas de seguridad, a continuación se describe brevemente su evolución, haciéndose hincapié en aquellos modelos con mayor influencia en la actualidad, y en el papel que desempeñan los Sistemas de Detección de Intrusiones que operan dentro de ellos.

### 2.2.1 Modelos conceptuales de seguridad

El modelo de seguridad conceptual más conocido, pero también más discutido en las últimas décadas es la CIA-triad. En la Sección 2.1 "Definición y ámbito de la seguridad" pudo deducirse su impacto, siendo sus tres propiedades: Confidencialidad, Integridad y Disponibilidad, aquellas que predominan en la definición de los principales ámbitos de la seguridad de la información y sus tecnologías. Aunque su origen es incierto, una de las primeras investigaciones que mencionan estas tres propiedades y las relacionan con la seguridad en los sistemas de información, fue publicada por J. Saltzer y M. Schroeder en el año 1975 [Smi12]. Sin embargo, el término CIA-triad no fue acuñado hasta 1986, donde formaba parte del Plan de Seguridad de la Información JSC-NASA, al que coloquialmente se denominó "el libro rosa" (del inglés *The Pink Book*) [CH13a]. Según la definición del NIST [Kis13], la propiedad Confidencialidad indica la salvaguarda de las restricciones de acceso y divulgación de la información. Este organismo también incluye en la acepción de confidencialidad todos aquellos tópicos relacionados con la privacidad y la propiedad intelectual, los cuales a menudo son tratados por separado en otros modelos conceptuales. Por otro lado, Integridad se refiere a la protección de la información frente a modificaciones inapropiadas o su eliminación. Para el NIST todo aquello relacionado con defender la autenticidad (i.e. ser genuino, de confianza y poder demostrarlo) y el no repudio (i.e. no poder negar la participación y el papel desempeñado dentro de un proceso de comunicación) se enmarca en esta propiedad. Finalmente, Disponibilidad es garantizar la posibilidad de acceder en cualquier momento y como sea preciso, a la información o cualquiera de los procesos que involucra. En la Figura 2.2 se muestra la representación más habitual de la CIA-triad, con forma de triángulo equilátero en la que cada arista representa una de sus tres propiedades fundamentales.

Las principales críticas a este modelo se centran en el alto nivel de abstracción de cada una de sus propiedades, la ausencia de propiedades de especial importancia en casos de uso concretos, o el hecho de que algunos elementos puedan perder protagonismo al formar para de propiedades mucho más grandes. Probablemente la corriente crítica con la CIA-triad más famosa fue la que encabezó D.B. Parker [Par98], quien propuso un modelo de mayor complejidad añadiendo tres nuevas propiedades a su antecesor: utilidad (i.e. la información debe ser un activo para el usuario), autenticidad y posesión (i.e. control de la información). Esto es conocido popularmente como sexteto Parkesiano (del inglés *Parkerian hexad*). Otra variación clásica de la CIA-triad es el modelo propuesto por J. McCumber [McC91], también conocido como cubo de McCumber (del inglés *McCumber's Cube*). Éste propone un marco para el aseguramiento de la información que asume como principal objetivo, salvaguardar las propiedades enunciadas por la CIA-triad. Pero a diferencia de su predecesor, el cubo de McCumber también tiene en cuenta las estrategias necesarias para lograrlo. Esto hace que no sea un modelo meramente conceptual, incorporando ideas propias de las estrategias de gestión de la seguridad de la información (ver Sección 2.2.2 "Gestión de la seguridad"). Alternativas más recientes a la CIA-triad son, [GGI+15] donde se incluyen nuevas propiedades para su adaptación a la seguridad en sistemas físicos, o [CH13a] donde se avanza hacia contrarrestar el efecto de la diversificación y

*des-perimetralización* inherentes a los nuevos escenarios de monitorización. Cabe destacar que a pesar de la controversia que suscita, la CIA-triad es el modelo conceptual que goza de mayor popularidad tanto en el ambiente académico como en las diferentes organizaciones para la seguridad de la información. Es también la base de casi todos los sistemas de gestión de seguridad actuales, por lo que sus pilares han constituido la base que soporta la mayor parte de la investigación descrita en este documento.


Figura 2.2: CIA-triad.

### 2.2.2 Gestión de la seguridad

La mayor parte de las estrategias de gestión de la seguridad de la información y los sistemas relacionados con ella asumen como principal objetivo, salvaguardar las propiedades definidas por los modelos conceptuales de seguridad (ver Sección 2.2.1 "Modelos conceptuales de seguridad"). Tal y como resaltan R.O Albuquerque et al. [OAGVSO+16], la gestión de la seguridad de los datos y las tecnologías que los procesan se lleva a cabo a partir de normas, directivas o estándares que deben de alinearse con los objetivos de la organización/infraestructura a proteger. Ejemplos clásicos de marcos de seguridad son las series ISO/IEC 27001:2013 [ISO13], NIST-SP800 [NIS18] o MAGERIT [CC12], centradas en señalar aquellos aspectos que conviene tener en cuenta a la hora de realizar la gestión de la seguridad. Otras plataformas, como La Biblioteca de Infraestructura de Tecnologías de Información o ITIL (del inglés *Information Technology Infrastructure Library*) [SSA08], o los Objetivos de Control para Información y Tecnologías Relacionadas o COBIT (del inglés *Control Objectives for Information and related Technology*) [ISA12], además aportan indicaciones para optimizar el beneficio que conlleva el despliegue de medidas defensivas. Finalmente cabe destacar la existencia de aproximaciones centradas en la identificación y evaluación de riesgos, muchos más enfocadas a la defensa de las tecnologías de la información y el ciberespacio. De entre ellas cabe destacar las métricas de evaluación de vulnerabilidades CVSS-SIG (del inglés *Common Vulnerability Scoring System*) [FIR15]. En [HCPD+16] se recopilan otras

directivas similares y se discuten sus elementos en común.
En términos generales, y siguiendo la distribución enunciada en NIST-SP800 [NIS18] (ver Figura 2.3), las acciones orientadas a la gestión de la seguridad establecidas por los distintos marcos pueden alinearse en cuatro grandes bloques de tareas: contextualización, evaluación, monitorización y respuesta. A continuación se describe brevemente cada una de ellas:

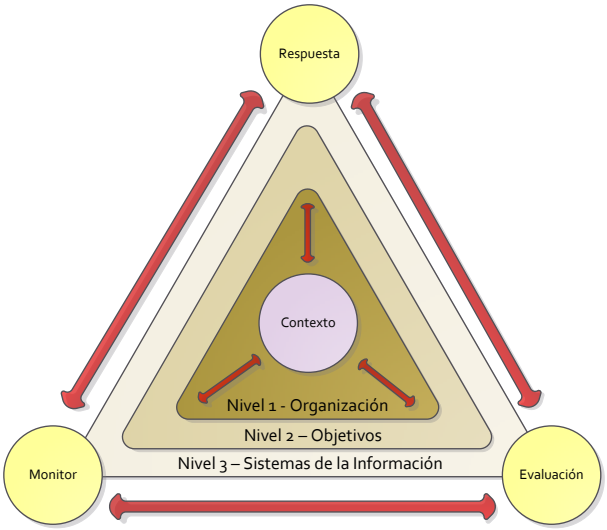



Figura 2.3: NIST/SP800.

- *Contextualización.* En la etapa de contextualización se establece la tolerancia a riesgos del sistema de gestión de incidencias y las prioridades a la hora de tomar decisiones. Éstas pueden estar relacionadas con leyes, políticas, regulaciones, directivas o contratos. Es habitual que las tareas de contextualización se alineen con los objetivos de la organización o el entorno de monitorización, lo que hace que varíen dependiendo del caso de uso. Esta etapa también establece las premisas iniciales del sistema, así como las limitaciones que se asumen antes de su desarrollo.

- *Evaluación de riesgos.* Las tareas enmarcadas en la evaluación de riesgos se centran en identificar las posibles amenazas dirigidas contra el sistema a proteger, el establecimiento de métricas que permitan valorar su impacto en base a los activos que puedan comprometer, y la definición del conjunto de contramedidas [SRM11]. Según el estándar ISO/IEC 27001:2013 [ISO13] la evaluación de riesgos se divide en dos tareas: 1) identificación de activos y riesgos; 2) valoración del impacto de las amenazas. Existe una amplia bibliografía centrada en estos desafíos, pudiéndose encontrar en [SSABC16] un resumen de las aproximaciones más recientes. Según esta publicación, todas ellas comparten cuatro importantes etapas de desarrollo: selección de métricas, establecimiento de mecanismos que puntúen los riesgos en función del contexto en que se detecten, la valoración e identificación de los activos a proteger y la evaluación de su posible propagación.

- *Monitorización.* En la etapa de monitorización se recopila información real del estado del sistema a proteger y se analiza en busca de intentos de intrusión o riesgos capaces de comprometer sus objetivos. En este proceso se aplicarán las métricas definidas en la fase de evaluación de riesgos y se estudiarán las posibles consecuencias de las amenazas.

- *Respuesta.* En la gestión de seguridad, la etapa de respuesta se activa una vez detectado un riesgo o cada cierto intervalo de tiempo. En ella se decide si deben de aplicarse contramedidas, y en ese caso, cuáles de ellas son las más apropiadas. Asimismo, esta tarea se encarga la elaboración de informes periódicos sobre el estado del sistema a proteger; además, si se detectan riesgos los comunica a los administradores de seguridad.

A raíz de esta clasificación es posible deducir que, en el marco de un sistema de gestión de incidencias, los Sistemas de Detección de Intrusiones principalmente se ubican en los procesos de monitorización, actuando como elementos capaces de identificar los riesgos que previamente se han definido. Sin embargo, es importante tener en cuenta su gran dependencia de las etapas de gestión anteriores (contextualización y evaluación de riesgos). Éstas van a tener un gran impacto sobre ellos, el cual se verá reflejado en sus principios de diseño y limitaciones. Además, van a decidir cuál será su objeto de análisis, y bajo qué circunstancias una observación puede relacionarse con un riesgo del sistema. Finalmente se espera que el Sistema de Detección de Intrusiones provea información que facilite la decisión de qué contramedidas deben aplicarse en el caso de identificarse una incidencia. En consecuencia, todas las tareas relacionadas con los Sistemas de Detección de intrusiones, desde su especificación hasta su evaluación, están influenciadas por el sistema de gestión de seguridad sobre el que operan, y por el contexto en el que son implementados.

## 2.3 Estrategias de intrusión

Según la definición de J.P. Anderson [And72], una intrusión es un intento deliberado de acceso no autorizado a información restringida, modificarla o dificultar que usuarios legítimos puedan hacer uso de ella. Teniendo en cuenta que el modelo conceptual de seguridad más secundado en aquella época era la CIA-triad, la definición de J.P. Anderson puede generalizarse a la expresión: "intento de romper la seguridad de un sistema". En consecuencia, el rango del término "intrusión" es el alcance de las propiedades del modelo conceptual de seguridad en el que se enmarca el sistema de gestión de seguridad que las reconozca. Dado que las intrusiones son el objeto de estudio de los Sistemas de Detección de Intrusiones, y que éstos son el eje central de esta investigación, su correcta comprensión depende de conocer las características de estas amenazas y cómo han evolucionado hasta la actualidad.

En los últimos años se han contemplado diferentes clasificaciones de intrusiones. El avance en las nuevas tecnologías y el incremento en el nivel de sofisticación de los ataques han llevado a la necesidad de replantear continuamente los ejes que separan las categorías en que los distribuyen. Un ejemplo de taxonomía clásica, y muy vigente en la actualidad, es la

clasificación AVOIDIT propuesta por Simmons et al. [SES+14] en el año 2009. A diferencia de otras aproximaciones, AVOIDIT combina cinco ejes de clasificación independientes: 1) su modus operandi o vector de ataque (del inglés *Attack Vector*), 2) la habilidad que tiene el ataque de alcanzar sus objetivos o impacto operacional (del inglés *Operational Impact*), 3) los métodos con que puede mitigarse o detectarse (del inglés *Defense*), 4) el daño que es capaz de causar o impacto informativo (del inglés *Informational Impact*), y 5) su objetivo (del inglés *Attack Target*). Se trata de una taxonomía de propósito general que reúne aspectos del proceso de intrusiones independientes del entorno en el que es llevada cabo. Sin embargo, para facilitar la correcta evaluación de riesgos de ciertos casos de uso, a menudo es necesario contar con clasificaciones mucho más concretas, que reúnan aspectos técnicos propios del escenario de monitorización sobre el que serán desplegados los sensores. Ejemplos de ellos se observan en [BG15] para redes, en [MVSOGV16] para atacantes internos o en [FBL15] para dispositivos móviles.
Según el Centro Criptológico Nacional del gobierno de España (CCN) y su organismo de respuesta ante incidencias (CCN-CERT), los factores que pueden considerarse a la hora de establecer criterios de clasificación de intrusiones son entre otros: el tipo de amenaza, origen, víctimas, tipología de los sistemas afectados o los requerimientos legales y regulatorios que desencadenan [CC16]. Éstos deben de combinarse para facilitar la decisión de contramedidas, valorar su impacto y priorizar la ejecución de las acciones de mitigación. Considerando el vector de ataque de las intrusiones como criterio de clasificación, el CCN propone los siguientes grupos de riesgo:

- *Código dañino.* Software específicamente desarrollado para comprometer la seguridad de la información. En esta categoría se agrupan amenazas como virus, gusanos, troyanos, spyware, rootkits, ransomware o Herramientas para el Acceso Remoto o RATs (del inglés *Remote Access Tools*).

- *Disponibilidad.* Riesgos capaces de reducir la accesibilidad y la calidad del servicio del entorno de monitorización. Son miembros de esta clase los Ataques de Denegación de Servicio Distribuidos o DDoS (del inglés *Distributed Denial of Service Attacks*), el sabotaje, y los errores tanto humanos como de software/hardware.

- *Obtención de información.* Esta familia de riesgos reúne el conjunto de ataques dirigidos a recabar información que permita desatar amenazas de mayor complejidad. Ejemplos de integrantes de este grupo son los métodos de escaneo de vulnerabilidades, *sniffing*, ingeniería social y la suplantación de identidad (del inglés *phising*).

- *Irrupciones.* Ataques que tienen por finalidad la explotación de vulnerabilidades que faciliten el acceso no autorizado al sistema víctima. Esta clase incluye los intentos de inyección SQL, desfiguración (del inglés *defacement*), Secuencias de órdenes en Sitios Cruzados o XSS (del inglés *Cross-Site Scripting*), *spear phising*, *pharming*, la inyección remota de ficheros o los ataques de fuerza bruta.

- *Compromiso de la información.* Riesgos relacionados con brechas en la confidencialidad e integridad de la información clasificada. Esto incluye accesos

no autorizados, modificación, borrado, divulgación y exfiltración de manera no autorizada.

- *Fraude.* Clase que agrupa las acciones relacionadas con la suplantación de identidad, como por ejemplo el uso de recursos no autorizado, acceso con credenciales ilegítimos o la violación de derechos de propiedad industrial e intelectual.

- *Contenido Abusivo.* Riesgos que acarrean daño en la imagen de la organización o el sistema protegido, y que se valen de sus recursos para perpetrar acciones malintencionadas, como envío de correo basura (*spam*), acoso, extorsión o emisión de mensajes ofensivos.

- *Política de seguridad.* Conjuntos de incidentes relacionados con la violación de las políticas de seguridad de la empresa previamente aprobadas en la fase de contextualización de su estrategia de gestión de seguridad (ver Sección 2.2.2 "Gestión de la seguridad"). Esta categoría incluye el uso abusivo de privilegios o el acceso a servicios no autorizado.

- *Otros.* Cualquier otro tipo de amenaza que no figure entre las anteriores.

Nótese que en [CC16] se profundiza en las características de los integrantes de cada una de estas categorías, no siendo el objetivo de este documento desarrollar en detalle cada una de ellas. En la taxonomía del CCN, la clase que aquí se llama "irrupción" es originalmente denominada "intrusión". En este documento ha sido renombrada para desambiguar su significado, evitándose confusiones entre su significado en el marco de la taxonomía, y la definición propuesta por J.P. Anderson [And72] y sus posteriores adaptaciones. Dado que esta clasificación sigue en vigor, y que su eficacia ha sido demostrada a lo largo de los años, es la que más se ha tenido en consideración durante el trascurso de la investigación realizada.
Finalmente cabe destacar la existencia de diferentes metodologías para la representación de este tipo de riesgos. Las más utilizadas en la actualidad son el lenguaje implementado por el centro de respuesta de incidencias US-CERT y el lenguaje descriptivo CVE (del inglés *Common Vulnerabilities and Exposures*) [YHK+15]. Ambos esquemas identifican unívocamente cada tipo de incidencia y la representan con lenguaje comprensible por seres humanos, distribuyendo sus características en diferentes campos semánticos. El CVE además es adoptado por el estándar de evaluación de vulnerabilidades OVAL (del inglés *Open Vulnerability and Assessment Language*) y sus repositorios.

## 2.4 Estructura general de un IDS

A finales de los años 90, la agencia de defensa norteamericana DARPA (del inglés *Defense Advanced Research Projects Agency*), constituyó un grupo de investigación enfocado al desarrollo de una plataforma de propósito general para la detección de intrusiones conocida como CIDF (del inglés *Common Intrusion Detection Framework*) [PSSC+90]. La arquitectura de su propuesta considera una división de la funcionalidad del IDS en

diferentes bloques, independientemente de si éstos pueden localizarse en una misma máquina o distribuidos en diferentes equipos. En el año 2000 este grupo se integra al IETF (del inglés *Internet Engineering Task Force*) bajo el acrónimo IDWG (del inglés *Intrusion Detection Working Group*). En el año 2006, la Organización Internacional de Normalización (del inglés *International Organization for Standardization*) conocida como ISO, estandariza el concepto de IDS en la ISO/IEC 18043:2006 [ISO06] y adopta el esquema CIDF como base de su arquitectura (ver Figura 2.4). Posteriormente, la necesidad de normalizar la información emitida entre componentes derivó en el estándar IDMEF (del inglés *Intrusion Detection Message Exchange Format*) [DCF07] para el intercambio de mensajes entre IDS. Nótese que la revisión actual de la estandarización de IDS se recoge en la ISO/IEC 18043:2015 [ISO15]. En ella además se proponen directivas para la elección de las técnicas que mejor se adapten a la contextualización del sistema de gestión de seguridad sobre los que operan, optimizar su despliegue y facilitar la toma de decisiones. La Figura 2.4 ilustra la estructura general de un IDS de acuerdo con el esquema CIDF. A continuación se describe cada uno de sus elementos.

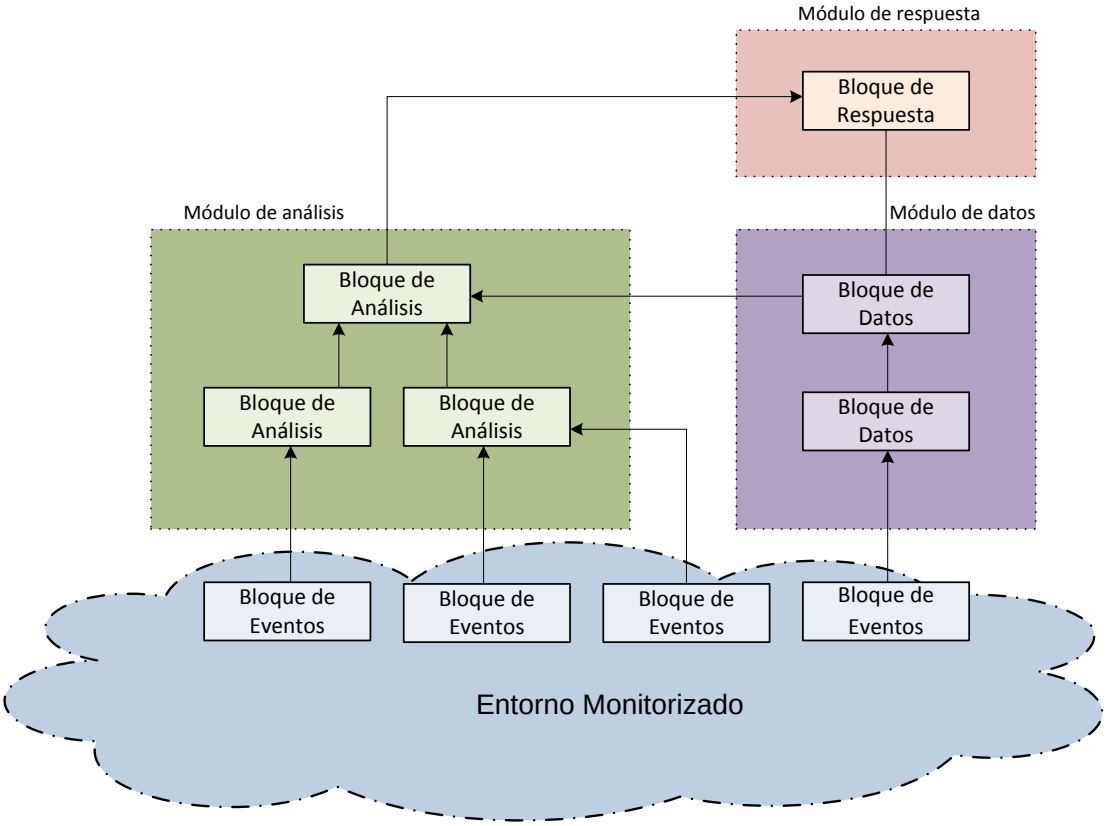



Figura 2.4: Arquitectura del CIDF.

- *Entorno Monitorizado.* El entorno de monitorización es el medio del que se extrae la información a analizar, y su elección repercute en la fase de diseño del resto de módulos del IDS. Por ejemplo, existe una gran diferencia entre monitorizar los

eventos producidos en una red, que involucran parámetros como las direcciones IP, los puertos, los protocolos o la carga útil del tráfico involucrados en el proceso de intrusión, con la monitorización de los eventos producidos a nivel local, que involucran parámetros como procesos, direcciones de memoria o registros.

- *Bloques de evento (E-bloques).* Los bloques de evento proporcionan información sobre los eventos del entorno al resto de módulos del IDS. Son los sensores que extraen información del medio, y expresan los resultados en forma de objetos de comunicación; en el CIDF original, lo hacen mediante objetos GIDO (del inglés General Intrusion Detection Object) especificados en el lenguaje CISL [ISO06].

- *Bloques de análisis (A-bloques).* Los bloques de análisis se encargan de analizar los datos recogidos por los bloques de evento en busca de actividades potencialmente maliciosas. Al igual que los bloques de evento, en el CIDF consideraban objetos de comunicación GIDO para el intercambio de información. Estos módulos son capaces de sintetizar los eventos de entrada con el fin de aligerar la velocidad de procesamiento de información. En el caso de que se detecten eventos maliciosos emiten alertas.

- *Bloques de datos (D-bloques).* Los bloques de datos ayudan a los bloques de evento y respuesta, originalmente almacenando los objetos GIDO correspondientes a eventos pendientes de ser procesados.

- *Bloques de respuesta (R-bloques)* Los bloques de respuesta procesan los objetos correspondientes a eventos etiquetados como maliciosos, y deciden las medidas preventivas a adoptar. La respuesta generada puede ser pasiva o activa. En el caso de que sea pasiva, las tareas de mitigación son delegadas a un operador. Cuando la respuesta es activa, estas labores se deciden automáticamente y son ejecutadas.

Pese a que el CIDF no tuvo éxito como modelo para el desarrollo de nuevos sistemas de detección de intrusos, asentó las bases para la estandarización de la ISO. La mayor parte de los IDS actuales consideran las bases, y la división en bloques que sus autores propusieron. Además, a lo largo de los años se ha considerado como un ejemplo genérico de los componentes de cualquier IDS y las relaciones existentes entre ellos, siendo mandatorio que se adapten a las características particulares del entorno de monitorización y el sistema de gestión de seguridad sobre el que operan.

## 2.5 CARACTERÍSTICAS DE LOS IDS

Desde la estandarización de los esquemas de detección de intrusiones de propósito general hasta la actualidad, la comunidad investigadora ha planteado una gran cantidad de propuestas relacionadas con su mejora y optimización. Esto ha sido propiciado por la rápida evolución y adaptación de los ataques a las nuevas tecnologías, el incremento en su sofisticación, crecimiento en variedad y la facilidad que ofrecen algunas de las nuevas herramientas ofensivas al perpetrar amenazas sin que el atacante presente un perfil con

conocimientos avanzados en seguridad. Con el fin de agilizar las tareas relacionadas con la identificación y despliegue de aquellas estrategias defensivas que mejor pudieran llegar a comportarse en cada caso de uso, la comunidad investigadora ha tratado de organizar todo este conocimiento por medio de taxonomías y ontologías. Algunas de estas clasificaciones tienen más de una década, pero se han convertido en una referencia muy importante para los nuevos investigadores. Por ejemplo, la taxonomía propuesta por H. Debar et al. [DDW99] publicada en el año 1999 es un documento clásico de cómo organizar todas estas contribuciones de manera intuitiva, asumiendo como criterios de clasificación las diferentes propiedades del sensor, como su método de detección, comportamiento, escenario de monitorización o frecuencia del análisis. Una clasificación más reciente puede encontrarse en [LLL13], donde además también se recopilan muchas de las taxonomías previas. En ella se definen nuevos criterios, como el tipo de datos a analizar, tiempo de respuesta, arquitectura del sensor o granularidad en el procesamiento de la información. En esta investigación también se deja constancia de la clara distinción que existe entre Sistemas de Detección de Intrusión (IDS) y Sistemas de Prevención de Intrusiones (del inglés *Intrusion Prevention Systems* o IPS); nótese que los IPS, a diferencia de los IDS, permiten desencadenar acciones reactivas y proactivas ante amenazas potenciales. Tanto en [LLL13] como en la presente sección, la clasificación realizada se centra únicamente en características relacionadas con la identificación de ataques, dejándose la toma de decisiones y activación de contramedidas fuera del alcance del estudio realizado. En base a este criterio, la revisión de la bibliografía nos ha permitido reconocer tres elementos comunes en las diferentes taxonomías: estrategia de detección, entorno de monitorización y arquitectura; los cuales establecen las mayores diferencias entre aproximaciones. Éstas son descritas a lo largo de esta sección.

### 2.5.1 Estrategia de detección

En los trabajos previos se distinguen dos estrategias fundamentales de detección de intrusiones: reconocimiento de firmas y reconocimiento de anomalías. Éstas a menudo han sido combinadas para aprovechar sus ventajas y minimizar su impacto sobre el sistema protegido. Dada su relevancia, es conveniente aclarar sus características más relevantes:

#### 2.5.1.1 Detección de intrusiones basada en firmas

En sus inicios, las estrategias para la identificación de amenazas se basaban en el reconocimiento de patrones conocidos o *firmas* de las características de la intrusión, lo que llevó a que la comunidad investigadora acuñara la expresión *detección de intrusiones basada en firmas*. La eficacia de estas estrategias dependía directamente de la calidad de la base del conocimiento que describía los patrones característicos de las amenazas. Sin embargo, la adquisición de estos rasgos a menudo no es una tarea trivial, habiendo sido éste su aspecto más estudiado en las últimas décadas. Un ejemplo clásico de ello se ilustra en [LSM99], donde las firmas son definidas como reglas de detección. Más recientemente, en [GTDVTSH15] se observa un ejemplo de estrategia de obtención de firmas de amenazas dirigidas contra los servicios HTTP de redes actuales. La principal

ventaja de la detección basada en firmas es su precisión; dado que únicamente emite alertas al reconocerse ataques previamente conocidos, la probabilidad de confundir observaciones legítimas con amenazas es muy baja (i.e. la tasa de falsos positivos del sistema es baja, ver Sección 4.6.1.1 "Precisión"). Además, permiten actualizar de manera sencilla sus bases del conocimiento, evitándose complejos procesos de entrenamiento y adaptación a escenarios no estacionarios (ver Sección 4.5 "Anomalías en entornos de monitorización no-estacionarios"). Las diferentes propuestas para la detección basada en firmas también presentan dos grandes inconvenientes; en primer lugar, su alto consumo de recursos de cómputo. W. Meng et al. [MLK14] identificaron algunas de sus causas, destacando de entre ellas el alto coste de sus algoritmos de encaje de patrones. El otro gran problema es su incapacidad de detectar ataques desconocidos, también conocidos como ataques de día-cero (del inglés *zero-day attacsk* o *0-day attacks*), y que por lo tanto no figuran en sus bases del conocimiento.

#### 2.5.1.2 Detección de intrusiones basada en anomalías

Como alternativa a la detección de firmas, la *detección de intrusiones basada en anomalías* implementa técnicas de análisis de comportamientos discordantes en el escenario de monitorización. Por lo tanto, requieren de la elaboración de algún tipo de representación del estado habitual y legítimo del sistema. Su método parte de la asunción de que, si alguna observación difiere de manera significativa de dicha representación, es considerada anómala, y por lo tanto podría ser un indicador de una acción intrusiva. Dado que esta estrategia es revisada en profundidad en los dos capítulos siguientes (Capítulos 4 y 5) y que se muestran ejemplos de su aplicación en lo que resta del documento, en esta subsección no se procederá a su descripción detallada. Sin embargo, es importante dejar constancia de que en la actualidad es el método más estudiado debido principalmente a su capacidad de identificar ataques desconocidos. .

#### 2.5.1.3 Combinación de firmas y anomalías

Con el fin de compensar las ventajas y desventajas de los métodos de detección basados en anomalías y firmas, también han sido propuestas estrategias híbridas. Tal y como se indica en [GPLL16], la *detección de intrusiones híbrida* habitualmente agrupa sus aproximaciones en las siguientes tres categorías: sistemas de detección de firmas cuyas salidas son procesadas por un sistema de detección de anomalías, sistemas de detección de anomalías cuyas salidas son procesadas por sistemas de detección de firmas, y despliegue en paralelo de métodos de detección de firmas y anomalías. Un ejemplo típico de esquema de detección híbrido se ilustra en el sistema EMERALD [NP97], donde ambas estrategias de detección actúan en paralelo y combinan los resultados de su análisis.
La Figura 2.5 muestra un ejemplo de sensor que combina la información provista por métodos de detección de anomalías, y la información provista por sensores basados en firmas. De manera similar, en [WCCQ07] el módulo de detección de anomalías se combina mediante el sistema de reglas usado por Snort y Bro. Esto permite aprovechar las ventajas de proyectos soportados por comunidades grandes, diseñados específicamente

para la instalación de nuevos módulos de preprocesamiento. A pesar de los buenos resultados arrojados en las diferentes propuestas híbridas, algunos investigadores advierten de que los resultados obtenidos por sistemas híbridos no siempre son mejores que los obtenidos aplicando las estrategias por separado, aunque conllevan las desventajas que ellos conllevan.

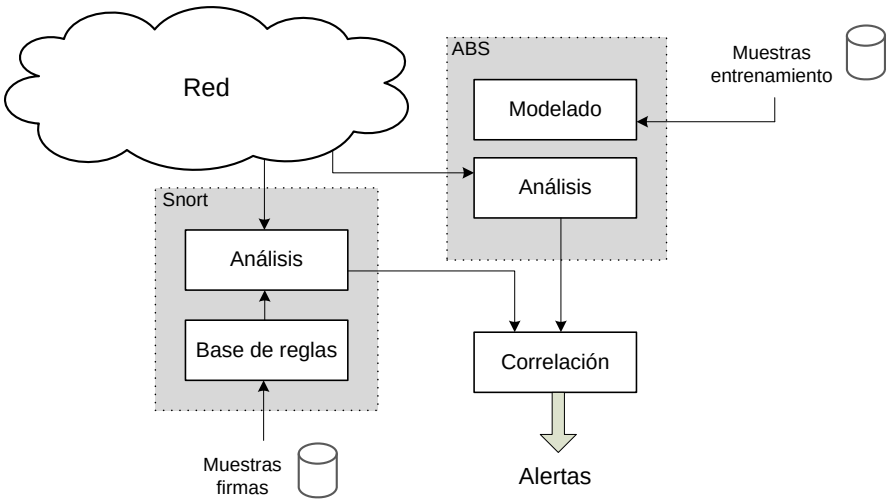



Figura 2.5: Ejemplo de arquitectura híbrida.

### 2.5.2 Entorno de monitorización

El entorno de monitorización sobre el que actúa un sistema de detección de intrusiones repercute directamente en los métodos de monitorización, extracción de características de la información y requisitos en el análisis llevado a cabo por el sensor. De manera tradicional, estos sistemas se clasifican en dos grandes grupos en base a su escenario de despliegue: sistemas de detección de intrusiones en hosts o locales (del inglés *Host-based Intrusion Detection Systems* o HIDS) y sistemas de detección de intrusiones en redes (del inglés *Network-based Intrusión Detection Systems* o NIDS). Nótese que en la bibliografía también es frecuente considerar otras categorías más específicas, como sistemas de detección de intrusos en redes inalámbricas (del inglés *Wireless-based Intrusion Detection Systems* o WIDS), sistemas de análisis de comportamiento en redes (del inglés *Network Behavior Analysis systems* o NBA) o sistemas de detección de intrusiones mixtos (del inglés *Mixed Intrusion Detection Systems* o MIDS). En [LLL13] se revisan en profundidad estas nuevas clasificaciones. No obstante, en el marco de nuestra investigación son consideradas como subconjuntos de las familias HIDS y NIDS, por lo que esta subsección se centra en las dos primeras y sus posibles combinaciones. A continuación se describe brevemente cada una de ellas:

#### 2.5.2.1 Detección de intrusiones local

Los HIDS son elementos defensivos que operan a nivel local por medio del despliegue de agentes encargados del análisis de información monitorizada específicamente en este

tipo de escenarios, como por ejemplo procesos del sistema operativo, llamadas al sistema, consumo de recursos, acceso a directorios y ficheros, cuentas de usuarios, políticas de auditoría o registros de eventos. A diferencia de los NIDS, son los únicos sensores capaces de estudiar todas las acciones perpetradas por los usuarios del sistema, reconocer ataques iniciados desde el mismo host, verificar si una amenaza ha alcanzado su objetivo y analizar el contenido cifrado en el tráfico de una red. Además, no requieren del despliegue de hardware adicional, ya que se aprovechan los propios recursos del host. Sin embargo, esta última característica puede conllevar el aumento de su consumo de recursos, y hacer que sean susceptibles de ser comprometidos en los casos en que no hayan sido capaces de defender con éxito el sistema. En [CH13b] se revisa en profundidad esta problemática y se propone una estrategia de detección basada en la identificación de anomalías en las secuencias de llamadas al sistema. Según esta publicación, existen dos grandes tendencias en la detección de intrusiones local. La primera de ellas se basa en el análisis de logs y acceso a ficheros/directorios. A partir de ellos es posible la identificación de ataques perpetrados por los propios usuarios legítimos de las organizaciones. En [CnHGMT14] se ilustra un claro ejemplo de cómo aplicar esta información para la identificación de atacantes enmascarados. Por otro lado, cabe destacar el estudio de las llamadas al sistema. Ésta permite detectar código malicioso antes de que se ejecute en el entorno protegidos [FBL15] y reconocer comportamientos anómalos en los perfiles de uso del sistema [MVSOGV16].

#### 2.5.2.2 Detección de intrusiones en redes

Los avances en la detección de intrusiones en redes son los que agrupan una mayor cantidad de publicaciones en los medios de diseminación relacionados con la seguridad de la información. Tal y como se indica en [MVSOGV15a], el *modus operandi* de un NIDS es analizar el contenido de cada uno de los paquetes que circula por la red, o estudiar la información que se produce en un intercambio de información entre sistemas. El primer caso está directamente relacionado con las técnicas de inspección profunda de paquetes (del inglés *Deep Packet Inspection* o DPI), y por lo tanto, sujeto a las restricciones legales que esta práctica conlleva [SBPC14]. Las propuestas que aplican este método a menudo se dividen teniendo en cuenta su objeto de estudio: carga útil, encabezados, o mixto. El análisis de la carga útil es especialmente eficaz en la detección de amenazas que pretenden explotar vulnerabilidades a nivel de aplicación [WS04]. Por otro lado, el considerar los datos contenidos en el encabezado del paquete facilita la identificación de amenazas que tengan por objetivo la explotación de vulnerabilidades en la implementación de protocolos de red e intentos de enumeración. Este campo además provee la información necesaria para la correlación de incidencias e identificación del origen del ataque. Finalmente, el análisis híbrido audita el contenido total del paquete; por lo tanto es mucho más preciso, pero conlleva un mayor consumo de recursos de cómputo. Un ejemplo clásico de propuesta híbrida se ilustra en PAYL [WS04], donde las amenazas se detectan por reconocimiento de anomalías aplicado sobre modelos construidos en función del contenido de la carga útil y tres características del encabezado de los paquetes: el puerto, la longitud y la dirección del flujo de tráfico (entrada y salida).

Como alternativa a la inspección de paquetes, en análisis de conexiones considera

métricas relacionados en el proceso completo de comunicación, como el número de bytes transmitidos, duración de las sesiones, protocolos o extremos del envío. El desarrollo de este tipo de sensores se ha motivado por los problemas de rendimiento relacionados con el análisis paquete a paquete, y la dificultad al tratar con información cifrada en tiempo real. En la actualidad a menudo se basa en el estudio de flujos de datos, cuya representación ha sido estandarizada en diversos formatos, como IPFIX [Cla08] o netFlow [Cla04]. En ellos cada flujo de datos se representa por el vector (IP origen, IP destino, puerto origen, puerto destino, protocolo). En [SSS+10] se explica más detalladamente el uso de este tipo de métricas y las tendencias que derivan de ello.

#### 2.5.2.3 Detección de intrusiones híbrida

Los esquemas de detección híbridos combinan sensores encargados del análisis de las actividades monitorizadas tanto a nivel de red como local. Esto permite la realización de un análisis minucioso de uno de los dos escenarios, considerando la información contextual aportada por el otro. Un ejemplo clásico de esta aproximación se observa en [DKPS05], donde se propone el uso de HIDS para mejorar la eficacia del conocido NIDS Bro. Por medio del análisis local se adquiere información que es muy difícil o imposible de observar desde el NIDS, como el contenido cifrado de los paquetes, datos relacionados con el procesamiento interno de peticiones/respuestas a servidores y la resolución de ambigüedades con nombres de dominios. Los IDS híbridos también pueden desplegarse como soluciones de propósito general [MVSOGV15a], en los que se brida protección en ambos planos de información. En la Figura 2.6 se ilustra un ejemplo de esta aplicación, en el que los sensores que actúan a nivel local emiten alertas relacionadas con virtualización, registros o llamadas al sistema; mientras que los sensores de red aplican DPI para detectar malware en la carga útil del tráfico, y analizan flujos de información en busca de ataques de denegación de servicio e intentos de robo de sesión.

### 2.5.3 Arquitectura

La arquitectura de un IDS viene determinada por las características de su entorno de despliegue y la información que debe analizar. Ésta puede agrupar todos sus elementos de análisis en un único componente, o expandirlos a lo largo del entorno de monitorización, lo que conlleva la necesidad de plantear un diseño más complejo y de definir los procesos de comunicación entre ellos. Una publicación clásica relacionada con esta problemática es [SZ00], donde se revisan las principales ventajas de cada aproximación. En ella se destaca la simplicidad del diseño centralizado frente a la complejidad, pero también capacidad de adaptación de los sistemas de detección de intrusiones colaborativos o CIDS (del inglés *Collaborative Intrusion Detection Systems*). Con el fin de introducir al lector en estos paradigmas de diseño, a continuación se describen las tres arquitecturas más frecuentes en la bibliografía: centralizada, distribuid y jerárquica.

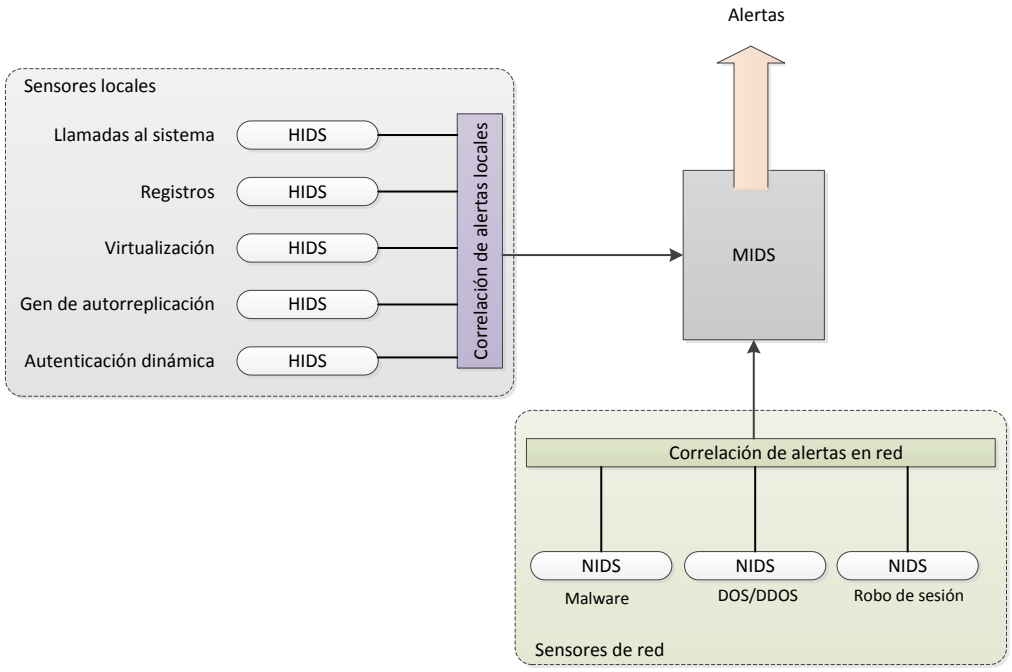



Figura 2.6: Ejemplo de esquema de detección híbrido.

#### 2.5.3.1 Centralizada

Los IDS de arquitectura centralizada agrupan todos sus componentes de análisis en un mismo nodo. Nótese que esto no implica que la información a analizar no pueda ser capturada por herramientas de monitorización repartidas a lo largo del sistema a proteger, tal y como se ilustra en la Figura 2.7. Por lo tanto, y según señalaron E. Spafford et al. [SZ00], se trata de la estrategia de despliegue más simple e intuitiva. Pero esta sencillez a menudo se penaliza con una menor potencia de análisis, lo que tiende a resultar en una peor precisión. Otros inconvenientes son su poca escalabilidad y la presencia de un único punto de fallo, lo que a menudo lleva a su complementación por medio de elementos defensivos adicionales como tarros de miel (del inglés *honeypots*) o trampas y señuelos. En [VKMF15] se propone un estado del arte actualizado acerca de este tipo de sistemas. En él se menciona a la herramienta SURFcert IDS [SUR18] como ejemplo ilustrativo de IDS capaz de aprovechar otras tecnologías (en concreto, tarros de miel) en las tareas de adquisición de información.

#### 2.5.3.2 Distribuida

Los IDS con arquitectura distribuida despliegan componentes con capacidad de análisis a lo largo del entorno protegido. Este tipo de sistemas son habitualmente conocidos como sistemas de detección de intrusiones distribuidos o DIDS (del inglés *Distributed Intrusion Detection Systems*). Estos elementos pueden operar de manera autonómica o cooperativa, tal y como se describe a continuación:

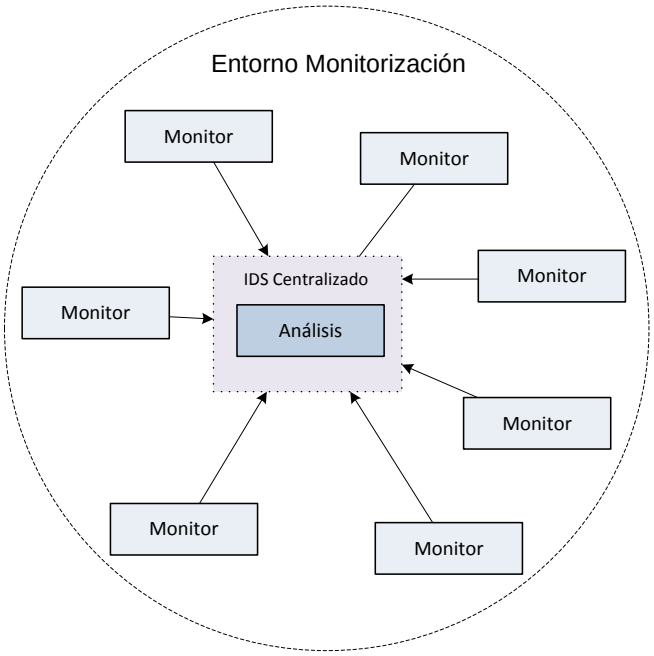



Figura 2.7: Ejemplo de IDS con arquitectura centralizada.

- *Modo autonómico.* Los elementos de análisis del DIDS actúan de manera independiente y sin compartir información (ver Figura 2.8). Cada uno de ellos habitualmente presenta un propósito específico, y por lo tanto se encarga de afrontar una amenaza concreta. En términos generales ofrecen una alta tasa de acierto, y un escaso consumo de recursos de cómputo. Sin embargo, la falta de coordinación entre sensores puede saturar el sistema, siendo propenso a padecer altas tasas de falsos positivos. Además, cuando la respuesta del NIDS es activa, pueden decidirse distintas medidas de prevención para mitigar una misma intrusión, lo que puede agotar los recursos del sistema o crear inconsistencias en su comportamiento. Un ejemplo típico de este modus operandi se ilustra en [JD07a], donde se propone una arquitectura para la detección de intrusiones en una red móvil MANET (del inglés *Mobile Ad Hoc Network*) mediante la monitorización del consumo de batería de distintos dispositivos móviles.

- *Modo cooperativo.* Cuando un DIDS actúa en modo cooperativo, distribuye el conjunto de nodos en una inmensa telaraña de elementos con capacidad de análisis comunicados entre sí (ver Figura 2.9), ya sea por medio de un servidor central o redes de pares (del inglés *Peer-to-peer* o P2P). Al igual que cuando sus sensores se comportan de manera autónoma, su propósito es específico, pero ahora sí son capaces de compartir información. Las diferentes alertas emitidas son puestas en común y el módulo de respuesta considera la información provista por cada elemento. Se trata de una estrategia precisa, que sacrifica parte del rendimiento del sistema en su etapa de detección debido a la latencia de las comunicaciones entre los nodos, en favor de la optimización de las etapas de análisis y prevención. Un claro ejemplo de esta metodología se ilustra en [ACM15], donde se propone un NIDS distribuido

cooperativo para redes MANET adaptado para resistir ataques de evasión basados en la explotación de la movilidad de sus nodos. Para alcanzar este objetivo, es necesario que los diferentes sensores tengan la capacidad de correlacionar información extraída de la carga útil del tráfico monitorizado.

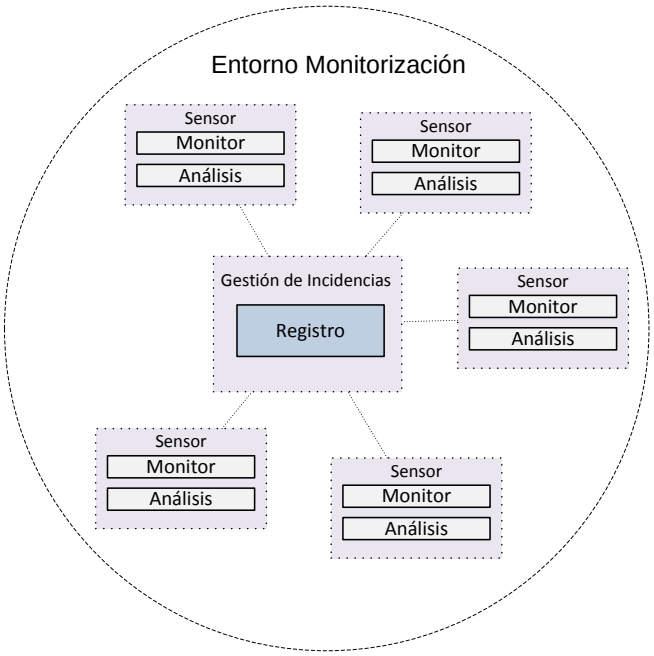



Figura 2.8: Ejemplo de IDS con arquitectura distribuida en modo autonómico.

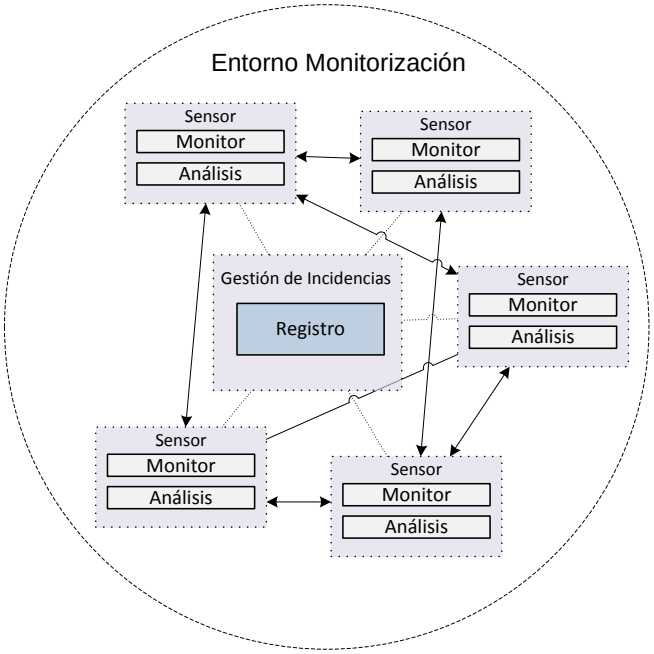



Figura 2.9: Ejemplo de IDS con distirbuida en modo cooperativo.

#### 2.5.3.3 Jerárquica

Los NIDS de arquitectura jerárquica, también conocidos como descentralizados, se inspira en las redes con infraestructuras de múltiples capas, donde las subredes se disponen en grupos o clusters [MVSOGV15a]. Dentro de estas organizaciones, los denominados nodos principales o supervisores (del inglés *clusterhead*) son aquellos que tienen mayor repercusión en el correcto funcionamiento de la red, ya que su tarea es actuar como gestor y puerta de enlace para el resto de nodos de que la integran. En los NIDS jerárquicos, los sensores se distribuyen de tal manera que algunos de ellos se comportan como supervisores de ciertos niveles de procesamiento de datos, lo que facilita un tratamiento de la información multinivel. En cada uno de estos niveles se lleva a cabo el análisis y la correlación de las alertas emitidas por los elementos que lo integran, y se procede a su correspondiente etiquetado, de manera que se facilita la labor de procesamiento de los niveles superiores. Generalmente, los NIDS de arquitectura jerárquica son los más precisos, ya que las distintas capas de procesamiento permiten un análisis más profundo de los eventos. Nótese que de estos despliegues típicamente se espera que, a mayor cantidad de niveles de procesamiento de información, mayor sea la precisión y escalabilidad obtenidas. Sin embargo, peor serán su rendimiento y consumo de recursos de cómputo. Un ejemplo clásico de NIDS con arquitectura jerárquica es EMERALD [NP97], donde se distinguen tres niveles diferentes: análisis de servicios, dominio y empresa (ver Figura 2.10).

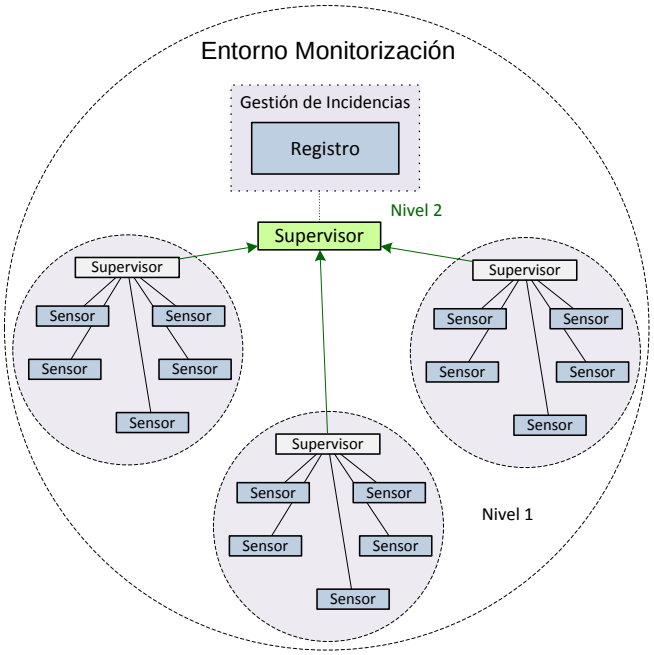



Figura 2.10: Ejemplo de IDS con arquitectura jerárquica.

# Capítulo 3

# Reconocimiento de anomalías

Este capítulo revisa los aspectos relacionados con el reconocimiento de anomalías que son necesarios para la correcta comprensión del resto del documento. Se ha organizado en las siguientes seis secciones: en la Sección 3.1 se define el concepto anomalía y se introducen las principales líneas de investigación relacionadas con este término; en la Sección 3.2 se repasan los diferentes tipos de anomalías; en la Sección 3.3 se ofrece una visión general de las estrategias de adquisición de conocimiento aplicadas a la identificación de anomalías; en la Sección 3.4 se introducen las diferentes medidas de similitud que habitualmente son tenidas en cuenta para su identificación; Finalmente en la Sección 3.5 se expone la problemática de la detección de anomalías en escenarios no estacionarios; en la Sección 3.6 se describen sus criterios de evaluación.

## 3.1 Introducción

El problema de la identificación de anomalías lleva siendo objeto de estudio desde hace décadas, pudiéndose observar en trabajos como [Edg87], donde en lugar del término anomalía se empleaba la expresión "observación discordante". Tal y como señalaron V. Chandola et al. [CBK09], la palabra anomalía ha sido sustituida por conceptos equivalentes a lo largo de los años, siendo también denominada: partes aisladas (del inglés *outliers*), excepciones, aberraciones, sorpresas, peculiaridades o elementos contaminantes. El uso de cada una de estas etiquetas habitualmente ha variado en función del dominio en el que ha sido empleada, del mismo modo que ha sucedido con su definición. Con el fin de familiarizar al lector con este término, a continuación se reúnen algunas de las definiciones de anomalía más populares en la bibliografía y se repasan los temas de investigación directamente relacionados con ellas.

### 3.1.1 Definición de anomalía

De entre las acepciones clásicas del término anomalía con mayor repercusión en la actualidad, cabe destacar la que propuso D. Hawkings [Haw80], refiriéndose a anomalías como "observaciones que se desvían lo suficientemente de las observaciones realizadas habitualmente como como para levantar la sospecha de que hayan sido originadas por

diferentes fuentes a las tenidas en consideración". Otras definiciones altamente recurridas son la de F.E. Grubbs [Gru69], en la que las anomalías indican "observaciones que parecen desviarse de manera representativa del resto de elementos de la muestra en la que se encontraban", la de V. Barnett et al. [BL94], en la que hacen referencia a "observaciones que parecen ser inconsistentes con el resto del conjunto de datos", o la de Aggarwal et al. [AY01], donde las anomalías son "puntos de ruido que quedan fuera de las agrupaciones de información previamente establecidas, o bien puntos que permanecen fuera de dichas agrupaciones y que además se distinguen del ruido". En un ámbito mucho más cercano a la seguridad de la información, V. Chandola et al. [CBK09] definieron anomalías como "patrones en los datos que no forman parte de la correcta definición de comportamiento normal". Esta última aceptación ha sido adaptada a diferentes escenarios de monitorización, como por ejemplo, la modificación de D. Savage et al. [SZY+14] para su aplicación en el análisis de redes sociales, siendo anomalías las "regiones de la red cuya estructura difiere de la esperada en su modelo normal".
La gran variedad de definiciones de anomalías y dominios en el que son estudiadas ha llevado a que los intentos de unificación hayan sido recurrentes a lo largo de la bibliografía. Según la generalización de E.M. Knorr et al. [KN97] las anomalías pueden definirse de la siguiente manera:

**Lemma 3.1.1** *Un objeto $O$ en un conjunto de datos $T$ es una $UO(p, D) - anomalía$ si al menos una fracción $p$ de los elementos de $T$ son menores o iguales que la distancia $D$ respecto de $O$.*

Sus propios autores justifican su elección teniendo en cuenta que ésta es independiente de la distribución que presenta la información observada, y que también es aplicable incluso antes de la ejecución de cualquier tipo de prueba de discordancia. Más recientemente, A. Zimek et al. [ZCS13] analizaron el problema de la generalización de esta definición. Esto los llevó a concluir que independientemente de la naturaleza de las observaciones, ya sea tráfico de red, datos de tarjetas de crédito, información capturada por sensores, etc., éste ofrecerá características que podrán ser predichas siempre y cuando se haya comprendido adecuadamente su comportamiento. Teniendo esto en cuenta, la presencia de observaciones no predecibles demuestra un desconocimiento parcial o total del entorno de monitorización, lo que probablemente conlleve la necesidad de investigar en mayor profundidad los nuevos acontecimientos. Este tipo de situaciones no predecibles son denominadas anómalas.
Al igual que la generalización de Knorr et al., esta nueva acepción es completamente independiente del contexto en el que es utilizada. Sin embargo, tiene en cuenta las ideas de percepción del entorno y proyección del conocimiento adquirido, las cuales se alinean intuitivamente con el modelo de Consciencia Situacional propuesto por Endsley [End88]. Este modelo establece las bases de los sistemas de identificación, evaluación y gestión de incidencias actuales, construyendo un modelo mental del entorno dividido en tres grandes niveles de asimilación: percepción, comprensión y proyección; donde la percepción es la fase de monitorización, recolección de evidencias y adquisición de conocimiento básico; comprensión es la correlación, reconocimiento de patrones, evaluación e interpretación de los datos obtenidos; y proyección es la anticipación y simulación de su evaluación

[BLVCMV+17]. Debido a que esta tesis se centra en el problema de la detección de intrusiones basada en anomalías, y dada la relación de este tópico con el modelo propuesto por Endsley, la generalización del concepto de anomalía propuesta por Zimek et al. es la que mejor se adapta al contexto en el que se ha llevada a cabo. Por lo tanto, es la que mayormente ha sido considerada a lo largo del estudio realizado.

### 3.1.2 Temas de investigación relacionados

Tal y como fue resaltado en [CBK09], es importante tener en cuenta la presencia de áreas de investigación fuertemente relacionadas con el concepto de anomalía y su identificación, las cuales han dado pie a discusiones acerca de si deben ser consideradas sub-tópicos dentro de la detección de anomalías. Esta proximidad ha sido ampliamente discutida por la comunidad investigadora, dando lugar a diversos intentos de unificación, como por ejemplo [TY06, SWZ15]. De entre las líneas de investigación relacionadas con nuestro objeto de estudio, cabe destacar aquellas que centran sus esfuerzos en la eliminación y acomodación de ruido, detección de novedades, identificación de puntos de cambio, y el descubrimiento de tendencias. A continuación son descritas brevemente, así como sus principales puntos de intersección y diferencias:

- *Eliminación y acomodación de ruido.* El ruido es un fenómeno frecuente en las colecciones de muestras e información extraída para el estudio que, a diferencia de las anomalías, no tiene interés desde el punto analítico, pero que es capaz de alterar los resultados obtenidos. Tal y como es discutido en [FV14], las técnicas de detección de anomalías forman parte del conjunto de estrategias aplicables a la identificación de ruido, facilitando de este modo su eliminación y acomodación.

- *Detección de novedades.* Según la definición de V. Chandola et al. [CBK09], la detección de novedades aborda el problema de reconocer patrones que no hayan sido previamente observados en el entorno de monitorización. Por lo tantos las novedades son un tipo particular de anomalías, que a menudo involucran el uso de estrategias específicas para su identificación [DLBM14]. Además, como se verá más adelante (ver Sección 3.5 “Anomalías en entornos de monitorización no-estacionarios”), juegan un papel muy importante en los métodos de detección adaptados a escenarios de monitorización no estacionarios, ya que sirven de referencia a la hora de actualizar su entrenamiento en tiempo de ejecución [OGIR14].

- *Identificación de puntos de cambio.* Según se describe en [TY06], el problema de la identificación de puntos de cambio se centra en determinar en qué instante de tiempo concreto se ha producido un cambio estadístico significativo en el entorno de monitorización, así como cualquier otro tipo de comportamiento inusual. Esta definición deja en evidencia la importante conexión entre el estudio de las anomalías y este campo, los cuales a menudo comparten y/o complementan sus técnicas de análisis y tratamiento de información [CBK09].

- *Descubrimiento de tendencias.* La detección de tendencias y la identificación de nuevos tópicos son dos problemas habituales en la minería de textos. Tal y como es

discutido en [SWZ15], su objetivo es detectar cambios en las distribuciones de flujos de datos que denoten el inicio de nuevos eventos, lo que esencialmente etiquetan como reconocimiento de anomalías en flujos de texto. Sin embargo, los métodos aplicados para este fin presentan claras diferencias con la detección de anomalías tradicional. Las más clara es que el descubrimiento de tendencias no está interesado en instancias de anomalías puntuales, sino en conjuntos de ellas y su relación, de manera que sea posible la definición de nuevos tópicos o variaciones de los ya existentes.

## 3.2 TIPOS DE ANOMALÍAS

Dado el desacuerdo presente en la propia definición del término anomalía, resulta difícil generalizar una clasificación que facilite su distinción. De entre las diferentes ontologías presentes en la bibliografía cabe destacar la propuesta por V. Chandola et al. [CBK09] por ser una de las más referenciadas. En ella se establecen tres grandes conjuntos de anomalías: *puntuales*, *contextuales* y *colectivas*. En [AM16] se muestra un ejemplo claro de su aplicación, en el que los principales ataques dirigidos contra redes pueden mapearse en cada una de ellas. Como alternativa a esta agrupación, en el estado del arte es frecuente su generalización en dos únicos conjuntos: anomalías *globales* y *locales* [SZK14]. El término anomalías globales abarca las anomalías puntuales de la taxonomía de V. Chandola et al.. Por otro lado, las anomalías locales hacen referencia a las anomalías contextuales, heredando dicho término de las aproximaciones para el reconocimiento de discordancias basadas en el análisis de densidades. En esta taxonomía, las anomalías colectivas pasan a ser casos particulares dentro de cualquiera de estos dos grupos, ofreciéndose una visión mucho más general de su naturaleza. Nótese que estos tipos de discordancias se han planteado con propósito general, lo que a menudo lleva a la necesidad de distinguir anomalías de manera mucho más específica dentro de cada caso de uso particular. Esto ha dado pie a nuevas clasificaciones adaptadas a cada escenario de monitorización, como por ejemplo sucede en [EH07, SZY+14], donde se han establecido anomalías concretas para el análisis de redes representadas por estructuras de datos con forma de grafos. A partir de ellas es posible distinguir anomalías relacionadas con el comportamiento de sus nodos o discordancias vinculadas a variaciones en la topología de la red. En [OGIR14] se emplea una jerarquía de anomalías específica para reconocer amenazas contra redes de sensores inalámbricos en entornos de monitorización no estacionales, la cual establece tres categorías: anomalías de primer, segundo y tercer orden. Las anomalías de primer orden son las que se disparan cuando un nodo sensor reconoce discordancias en parte de las métricas que monitoriza; las de segundo orden cuando para dicho sensor, todas las métricas muestran valores no esperados. Finalmente, las anomalías de tercer orden se detectan al poner en común la información capturada por varios sensores.

Dado que la clasificación V. Chandola et al. [CBK09] es una de las más completas y está presente en la mayoría de las publicaciones actuales, será la que se considere durante el resto del documento. A continuación se describe brevemente cada uno de los tipos de discordancias que define.

### 3.2.1 Anomalías puntuales

La categoría de anomalías puntuales reúne a todas aquellas instancias particulares de discordancia con respecto al resto de las observaciones realizadas. Un ejemplo de anomalía puntual puede observarse en el retraso de un autobús al realizar un recorrido en su línea regular: supongamos que cada vez que el vehículo completa su recorrido se toma una muestra del tiempo transcurrido, pudiendo concluir que aproximadamente requiere 1 hora y media para terminar su ruta. Si habiendo transcurrido dos horas el autobús aún no ha llegado a su destino, se producirá una anomalía puntual, llevando a los operadores a investigar por qué motivo se ha producido el retraso. En la Figura 3.1 se ilustra otro ejemplo de anomalía puntual, esta vez tomando como referencia el espacio euclidiano. Tal y como se observa, a partir de las muestras tomadas pueden distinguirse dos claras regiones de similitud ($C_1$, $C_2$) y dos divergencias ($A_1$, $A_2$), las cuales pueden etiquetarse como anomalías puntuales dentro de dicho conjunto de datos.

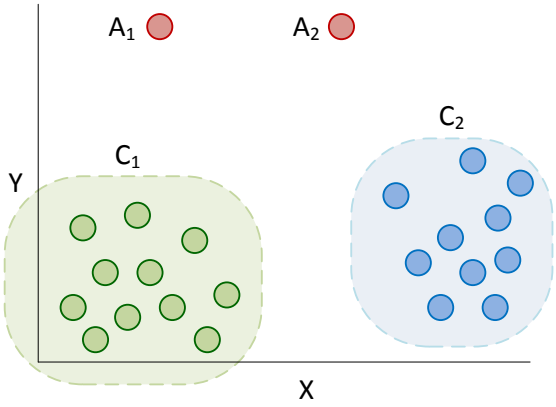



Figura 3.1: Ejemplo de anomalías puntuales.

Pero a pesar de que las anomalías puntuales son el conjunto de discordancias más sencillo e intuitivo, su detección plantea diferentes desafíos, como por ejemplo el problema de identificar la medida óptima de similitud de su desviación [HKP11]. Además, Kriegel et al. [KKZ10] resaltaron otros importantes aspectos para tener en cuenta de cara a su identificación, como el hecho de que el conjunto de muestras de referencia aplicado en el entrenamiento del sensor y/o el modelado del entorno de monitorización pueda contener otras anomalías puntuales, dando lugar a resultados que no correspondan con la realidad. De manera análoga, en detectores no supervisados la observación reiterada de anomalías puntuales podría llevar al sensor a concluir en que corresponden con parte del comportamiento normal y esperado.

### 3.2.2 Anomalías contextuales

Según la definición de Song et al. [SWJR07], cuando una observación es discordante si se enmarca en un determinado contexto, pero puede no serlo si éste varía, se trata de una anomalía condicional, también denominadas anomalías contextuales [CBK09]. Un ejemplo clásico de este tipo de discordancias es la temperatura en diferentes regiones geográficas

y/o momentos estacionales: si bien un registro de 32° puede ser considerado normal al medirse en Madrid en verano, la lectura de este mismo valor en la misma ciudad, pero en los meses invernales se tratará sin duda una anomalía contextual (asumiendo que la temperatura media en esta estación se aproxima a 12°). En la Figura 3.2 se ilustra otro ejemplo, en el que se muestran variaciones estacionales normales en varios periodos de tiempo ($D_1$, $D_2$), y tres registros puntuales con idéntico valor que pueden ser anomalías contextuales ($A_1$, $A_2$) o no ($P_1$) en función del momento en que son observados.

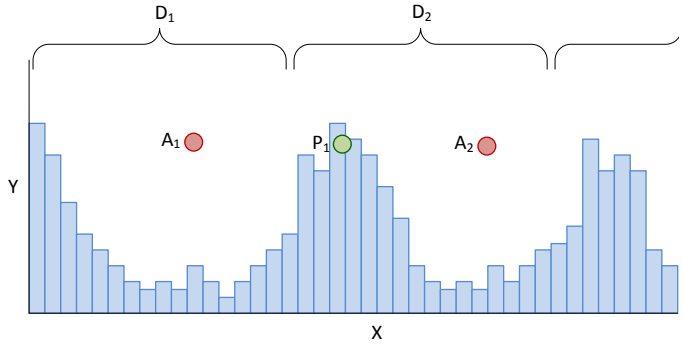



Figura 3.2: Ejemplo de anomalías contextuales.

Con el fin de facilitar el estudio de este tipo de observaciones, es habitual acompañarlas de dos tipos de atributos: *contextuales* y de *comportamiento*. Los atributos contextuales determinan las circunstancias en las que se enmarca la observación; por ejemplo, para el estudio de la temperatura en Madrid, un buen atributo contextual es el mes del año en que se mide. Por otro lado, los atributos de comportamiento son las características en sí de la observación, siendo en el caso anterior, la propia temperatura que ha sido registrada. Dada la dificultad que conlleva definir los atributos contextuales en determinados escenarios de monitorización, el uso de este tipo de anomalías prácticamente se reduce a la exploración de series temporales y objetos en el plano.

### 3.2.3 ANOMALÍAS COLECTIVAS

Cuando un conjunto de observaciones presenta discordancia respecto al comportamiento que se espera monitorizar, es denominado anomalía colectiva [HKP11]. Por lo tanto, las anomalías colectivas implican que se esté teniendo en consideración algún tipo de relación entre las observaciones que integran el conjunto de muestras de referencia. Además, al separar los elementos que componen la anomalía colectiva podrían no ser considerados discordantes de manera individual. Las anomalías colectivas pueden ilustrarse muy claramente en el ejemplo de los ataques de denegación de servicio. Estas amenazas consisten en el agotamiento de los recursos de cómputo, habitualmente por medio de la inundación por inyección de un gran volumen de información que debe de ser procesada por la víctima. Al estudiar los avances en la identificación de los ataques de denegación de servicio en redes se observa una gran variedad de métricas y relaciones entre observaciones [BBK15]. Pero hay una idea que es común en todas ellas: si se analizan únicamente los paquetes maliciosos que el atacante dirige hacia la víctima de manera individual, es muy

difícil desenmascarar estos intentos de agotamiento de recursos. Para determinan si existen intenciones maliciosas, es importante tener en cuenta el conjunto de paquetes enviados, hallándose anomalías colectivas en variaciones de las relaciones entre dichas muestras.

## 3.3 Adquisición de conocimiento

La naturaleza de la información que los sistemas de detección de anomalías consideran en las tareas de definir qué es normal y qué es discordante, limita la elección de las estrategias a implementar y afecta a su comportamiento. Según Chandola et al. [CBK09], en el ámbito de la detección de intrusiones su mayor distinción puede realizarse a partir del etiquetado de las muestras que contiene, siendo las categorías *normal* y *anómala* las más frecuentes en las publicaciones previas. En base a esto, P. Laskov et al. [LDSR05] realizaron dos observaciones esenciales necesarias para comprender las diferentes estrategias de adquisición de conocimiento y decidir la más apropiada para cada caso de uso: en primer lugar debe tenerse en cuenta que las muestras etiquetadas son muy difíciles de obtener, situación que puede agravarse si el sistema de detección opera en tiempo real en escenarios donde probablemente no pueda etiquetarse toda la información extraída. Por otro lado, es imposible disponer de una colección de muestras maliciosas que cubra todos los tipos de amenazas existentes, por lo que el sistema es vulnerable a ataques desconocidos. En base a la naturaleza de las muestras de referencia, es posible dividir los métodos de detección en cinco grandes grupos, los cuales corresponden con los principales paradigmas del aprendizaje automático: aprendizaje supervisado, semi-supervisado, no supervisado, reforzado, transducción y multitarea. En [BG15] se recopila una gran cantidad de trabajos que implementan cada una de estas metodologías. A continuación se describe brevemente cada una de ellas y su implicación en el reconocimiento de anomalías.

### 3.3.1 Aprendizaje supervisado

Los sistemas de detección de anomalías que adoptan técnicas de aprendizaje supervisado tienen en cuenta colecciones de muestras tanto normales como anómalas a la hora de construir los modelos de uso del sistema. Por lo tanto, son aproximaciones generalmente más precisas, y con menor tendencia a la emisión de falsos positivos. Sin embargo, tienen por principales inconvenientes la dificultad de obtener muestras y etiquetados fiables (especialmente de observaciones anómalas), y el hecho de habitualmente se dispone de menos muestras anómalas que normales, dando pie a los inconvenientes relacionados con el desequilibrio entre clases [HKP11]. A pesar de ello es una de las estrategias más utilizadas en la detección de intrusiones, empleándose en muy diversas áreas, como la detección de malware en redes [26], la identificación de amenazas internas [MVSOGV16] o el análisis de aplicaciones para dispositivos móviles [FBL15]. En [GKRB13] se revisa en detalle este paradigma y se recopilan sus principales técnicas.

### 3.3.2 Aprendizaje semi-supervisado

A diferencia que en el aprendizaje supervisado, la detección de anomalías semi-supervisada únicamente considera una clase da datos de referencia, la cual frecuentemente es de naturaleza normal. Tal y como es descrito en [CBK09], la aproximación típica hacia la detección semi-supervisada construye un modelo del modo de uso habitual y legítimo del entorno de monitorización, y analiza las nuevas observaciones en busca de diferencias significativas. Por lo tanto, lleva al desarrollo de herramientas habitualmente más fáciles de configurar y de mayor sensibilidad, generalmente ofreciendo una mejora de la capacidad de identificación de anomalías desconocidas a costa de penalizar su tasa de falsos positivos. En la detección de intrusiones se utiliza en los mismos campos que el aprendizaje supervisado, permitiendo la implementación de algoritmos alternativos, como Máquinas de Vectores de Soporte o SVM (del inglés *Support Vector Machines*), autocodificadores, modelos mixtos gaussianos, etc. [GU16].

### 3.3.3 Aprendizaje no supervisado

La detección de anomalías que adquiere conocimiento de manera no supervisada no considera categorías a priori, es decir, prescinde de colecciones de datos etiquetados como referencia. En su lugar trata las observaciones realizadas como un conjunto de variables aleatorias, a partir de las cuales es posible construir modelos de densidad y definir grupos. Tal y como indican J. Han et al. [HKP11], para minimizar el problema de los errores de etiquetado, los sistemas de detección que adoptan este paradigma deben tener en cuenta la siguiente asunción: las observaciones "normales" han de seguir cierto tipo de patrones o distribuciones mucho más frecuente que en las discordancias. A medida que la información monitorizada se aleja de esta premisa, los sistemas de detección tenderán al incremento de la emisión de errores de etiquetado, lo que hace que sea su principal vulnerabilidad. En la práctica, el aprendizaje no supervisado es muy explotado en entornos de monitorización especialmente complejos, como sistemas SCADA [AYTF14] o en la gestión de redes [CBMP16].

### 3.3.4 Aprendizaje reforzado

El objetivo del aprendizaje por refuerzo es ayudar a los sistemas de detección de anomalías y actuadores a tomar las mejores decisiones. En los sistemas que aplican este paradigma, cada vez que un agente toma una decisión, su repercusión en la calidad del servicio prestado será tenida en cuenta en futuros razonamientos [MK15]. Esto ha sido utilizado en la mejora de diferentes características de los sistemas de detección de anomalías convencionales, tales como la decisión de qué datos deben analizarse cuando el sensor opera en tiempo real [SSSS13], la optimización de la predicción de secuencias de patrones en entorno local [Xu06] o en la reconfiguración dinámica de los sistemas de detección en función de las características de las anomalías [HDND15].

### 3.3.5 Transducción

El razonamiento por transducción permite inferir conocimiento tomando como referencia un conjunto de observaciones etiquetadas, y conociendo previamente las muestras específicas que van a ser analizadas. A diferencia de los esquemas inductivos (aprendizaje supervisado, semi-supervisado, etc.), la transducción no construye un modelo genérico del entorno de monitorización y lo aplica a predicciones concretas; en su lugar plantea una solución específica a un conjunto de objetos a analizar concreto y finito [BDR06]. La principal ventaja de este método es que es mucho más preciso que los algoritmos inductivos en los casos en que el conjunto de muestras de referencia es mucho más pequeño. Sin embargo, al no construir un modelo, el proceso de transducción debe repetirse para cada nuevo conjunto de elementos a etiquetar, lo que involucra un mayor consumo de recursos de cómputo. Aunque en la bibliografía su uso no es tan frecuente como el de los métodos previamente descritos, existen diferentes publicaciones que revelan sus eficacia en la detección de intrusiones basada en anomalías, como por ejemplo el reconocimiento de patrones de ataques en redes [LG07] o la identificación de comportamientos discordantes para la autenticación dinámica en dispositivos móviles [MSBF15].

### 3.3.6 Aprendizaje multitarea

La adquisición de conocimiento por medio de aprendizaje multitarea permite complementar el conocimiento generado a partir de una tarea principal con la información provista por otras tareas secundarias, pero relacionadas con ella. Por lo tanto, parte de dos premisas básicas: la información relevante para la solución del problema a tratar puede ser compartida por todas las tareas, y su unión debe mejorar la capacidad de predicción del sistema. Sin embargo, la implementación del aprendizaje multitarea no es trivial en la mayor parte de los escenarios de monitorización, donde debe decidirse qué elementos tienen en común las distintas tareas y cómo van a compartir información [KD12]. A pesar de ello se ha aplicado en muy diversas áreas, como por ejemplo, clasificación de tráfico de red [LHWP15], autenticación por reconocimiento de rasgos biométricos [ZGTJ16] o el etiquetado de riesgos en sistemas de la información [JDC+16].

## 3.4 Distancias y medidas de similitud

Los sistemas de detección de anomalías a menudo deben comparar la percepción de las observaciones “normales” realizada durante su etapa de adquisición de conocimiento con los elementos que son analizados. Por lo tanto, las características de las distancias y medidas de similitud que son tenidas en consideración afectan de manera directa a su eficacia.

Según D.J. Weller-Fahy et al. [WFBS14], la definición de medida de distancia conlleva el cumplimento de tres grandes requisitos: no negatividad, identidad de los indiscernibles y simetría. Si la medida de distancia satisface el principio de desigualdad triangular, pertenecerá a la categoría de métricas de distancia. Cuando el método aplicado no cumple las tres primeras propiedades se denomina medida de similitud. Para definir cada una de

estas propiedades, D.J. Weller-Fahy et al. hicieron uso de la función dist: $A \times B \rightarrow \mathbb{R}$, la cual considera dos posiciones genéricas de entrada $A$ y $B$, y devuelve el valor de su distancia. En base a esto, expresaron estos requisitos de la siguiente manera:

**Lemma 3.4.1** *No Negatividad. La distancia entre A y B es siempre mayor o igual que cero:*

$$dist(A, B) \geq 0 \tag{3.1}$$

**Lemma 3.4.2** *Identidad de los indiscernibles. La distancia entre A y B es igual a cero si y solo si A equivale a B:*

$$dist(A, B) = 0 \leftrightarrow A = B \tag{3.2}$$

**Lemma 3.4.3** *Simetría. La distancia entre A y B es igual a la distancia entre B y A:*

$$dist(A, B) = dist(B, A) \tag{3.3}$$

**Lemma 3.4.4** *Desigualdad triangular. Al considerar un tercer elemento C, la distancia entre A y B es siempre menor o igual a la suma de la distancia entre A y C con la distancia entre B y C:*

$$dist(A, B) \leq (dist(A, C) + dist(B, C)) \tag{3.4}$$

En la actualidad existen cientos de distancias y medidas de similitud, las cuales se adaptan a la naturaleza de los datos a analizar. En el ámbito de la detección de anomalías predominan tres grupos de aproximaciones de propósito general: las que comparan datos cuantitativos, cualitativos y mixtos. A continuación se describe brevemente cada uno de ellos, así como algunas algunos de los escenarios particulares que han llevado a su especialización.

### 3.4.1 DISTANCIAS DE SIMILITUD EN DATOS CUANTITATIVOS

Las distancias de similitud en datos cuantitativos se centran en el estudio de variables representadas en el dominio de los números reales, es decir, en las que $\{A, B\}\epsilon\mathbb{R}$. Los datos cuantitativos pueden ser discretos o continuos; en variables discretas el dominio se limita a un rango de valores conocido, mientras que en datos continuos puede darse cualquier valor. Un caso particular de datos discretos son las variables binarias, las cuales asumen $\{A, B\}\epsilon\{0, 1\}$. Pueden encontrarse muchos ejemplos de este subconjunto en [CCT10], donde se recopila y se clasifica una lista con 76 distancias de similitud específicas para datos binarios.

Según S.H. Cha [Cha07] et al. el cálculo de distancias de similitud en datos cuantitativos tiene un importante parecido con la comparativa de los datos nominales representados en histogramas. En base a esto, plantean taxonomías de funciones asumiendo tres criterios: similitud sintáctica, advertencias de cara a su implementación y semántica. Análogamente, en [LRB08] estas medidas son agrupadas en medidas de disimilitud y medidas derivadas de productos escalares, planteando una distinción mucho más explícita de su naturaleza.

En [BBK14] se indican las distancias de similitud más frecuentes en el estudio de datos cuantitativos, las cuales son resumidas en la Tabla 3.1 como ejemplo ilustrativo.

Tabla 3.1: Ejemplos de distancias de similitud para datos cuantitativos

| Nombre | Medida $dist(A, B)$ | Nombre | Medida $dist(A, B)$ |
|---|---|---|---|
| Euclideana | $\sqrt{\sum_{i=1}^{d} \lvert A_i - B_i \rvert^2}$ | Euclideana ponderada | $\sqrt{\sum_{I=1}^{D} \alpha_I \lvert A_I - B_i \rvert^2}$ |
| Euclideana cuadrática | $\sum_{i=1}^{d} \lvert A_i - B_i \rvert^2$ | Cuerca cuadrática | $\sum_{i=1}^{d} (\sqrt{A_i} - \sqrt{B_i})^2$ |
| Cuadrática $X^2$ | $\sum_{i=1}^{d} \frac{(A_i - B_i)^2}{A_i + B_i}$ | geometría del taxista | $\sum_{i=1}^{d} \lvert A_i - B_i \rvert$ |
| Minkowski | $\sqrt[p]{\sum_{i=1}^{d} \lvert A_i - B_i \rvert^p}$ | Chebyshev | $\max i \lvert A_i - B_i \rvert$ |
| Canberra | $\frac{\sum_{i=1}^{d} \lvert A_i - B_i \rvert}{A_i + B_i}$ | Coseno | $\frac{\sum_{i=1}^{d} A_i B_i}{\sqrt{\sum_{i=1}^{d} A_i^2} \sqrt{\sum_{i=1}^{d} B_i^2}}$ |
| Jaccard | $\frac{\sum_{i=1}^{d} A_i B_i}{\sum_{i=1}^{d} A_i^2 + \sum_{i=1}^{d} B_i^2 + \sum_{i=1}^{d} A_i B_i}$ | Bhattacharyya | $-\ln \sum_{i=1}^{d} \sqrt{(A_i B_i)}$ |
| Pearson | $\sum_{i=1}^{d} (A_i - B_i)^2$ | Divergenia | $2 \sum_{i=1}^{d} \frac{(A_i - B_i)^2}{(A_i + B_i)^2}$ |
| Mahalanobis | $\sqrt{(A - B)^t \sum^{-1} A - B}$ | | |

### 3.4.2 Distancias de similitud en datos cualitativos

Desde un punto de vista estadístico, las variables categóricas (también conocidas como variables cualitativas o de atributo) son aquellas que pueden tomar como valores cualidades o categorías. Por ejemplo, una instancia de un medio de transporte puede presentar diferentes características, como su color (rojo, azul, verde, etc.), tipo de vehículo (motocicleta, coche, barco, etc.) o lugar de fabricación (Madrid, Barcelona, Valencia, etc.). La mayor dificultad a la hora de procesar este tipo de información se encuentra en el hecho de que sus atributos no presentan ninguna noción explícita de orden. Esto hace que la eficacia de las distancias dependa directamente de la naturaleza de los datos. En [BCK08] se discute más detalladamente este problema y se recopilan las medidas de similitud más utilizadas en el análisis de datos cualitativos, las cuales son resumidas en la Tabla 3.2 como ejemplo ilustrativo. Nótese que las ecuaciones de la tabla respetan la siguiente formalización:
Sean las observaciones $A$ y $B$ pertenecientes a la colección de muestras $D$ de extensión $N$, su similitud es definida como:

$$dist(A, B) = \sum_{k=1}^{d} w_k(A_k, B_k) \tag{3.5}$$

donde $k$ determina el atributo en la posición $k - esima$ de cada muestra y $w_k$ el peso asignado a dicho valor.

Tabla 3.2: Ejemplos de distancias de similitud para datos cualitativos.

| Nombre | Medida $dist(A,B)$ | $w_k = 1, \cdots, d$ |
|---|---|---|
| Superposición | $\begin{cases} 1 & \text{si } A_k = B_k \\ 0 & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Eskin | $\begin{cases} 1 & \text{si } A_k = B_k \\ \frac{n_k^2}{n_k^2+2} & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| IOF | $\begin{cases} 1 & \text{si } A_k = B_k \\ \frac{1}{1+\log f_k(A_K) \times \log F_K(B_k)} & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| OF | $\begin{cases} 1 & \text{si } A_k = B_k \\ \frac{1}{1+\log \frac{N}{f_k(A_k)} \times \log \frac{N}{f_k(B_k)}} & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Lin | $\begin{cases} 2\log p_k(A_k) & \text{si } A_k = B_k \\ 2\log(p_k(A_k) + p_k(B_k)) & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{\sum_{i=1}^{d} \log p_i(A_i) + \log p_i(B_i)}$ |
| Lin1 | $\begin{cases} \sum \log p_k(q) & \text{si } A_k = B_k, q \epsilon Q \\ 2\log \sum_{q \epsilon Q} p_k(q) & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{\sum_{i=1}^{d} \sum_{q?Q} \log p_i(q_i)}$ |
| Goodall1 | $\begin{cases} 1 - \sum_{q \epsilon Q} p_k^2(q) & \text{si } A_k = B_k \\ 0 & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Goodall2 | $\begin{cases} 1 - \sum_{q \epsilon Q} p_k^2(q) & \text{si } A_k = B_k \\ 0 & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Goodall3 | $\begin{cases} 1 - p_k^2(A_k) & \text{si } A_k = B_k \\ 0 & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Goodall4 | $\begin{cases} p_k^2(A_k) & \text{si } A_k = B_k \\ 0 & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Smirnov | $\begin{cases} 2 + \frac{N - f_k(A_k)}{f_k(A_k)} + \sum_{q \epsilon \frac{X_k}{A_k}} \frac{f_k(q)}{N - f_k(q)} & \text{si } A_k = B_k \\ \sum_{q \epsilon \frac{X_k}{A_k, B_k}} \frac{f_k(q)}{N - f_k(q)} & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{\sum_{k=1}^{d} n_k}$ |
| Gambaryan | $\begin{cases} 1 & \text{si } A_k = B_k \\ [20pt] \log \frac{\sum_{q \epsilon X_k} 2\log(1-p_k(q))}{\frac{p_k(A_k) p_k(B_k)}{(1-p_k(A_k))(1-p_k(B_k))} + \sum_{q \epsilon X_k} 2\log(1-p_k(q))} & \text{si } A_k \neq B_k \end{cases}$ | $\frac{1}{d}$ |
| Andelberg | $dist(A,B) = \frac{\sum_{k?1 \leq k \leq d: A_k = B_k} (\frac{1}{p_k(A_k)})^2 \frac{2}{n_k(n_k+1)}}{\sum_{k \epsilon 1 \leq k \leq d: A_k = B_k} (\frac{1}{p_k(A_k)})^2 \frac{2}{n_k(n_k+1)} + \sum_{\epsilon 1 \leq k \leq d: A_k \neq B_k} (\frac{1}{2 p_k(A_k) p_k(B_k)}) \frac{2}{n_{k(n_k+1)}}}$ | |

También debe tenerse en cuenta que $f_k(x)$ indica el número de veces que aparece el valor $x$ en $D$ , siendo $\hat{p}_k(x)$ la probabilidad de aparición de $x$ en $D$ expresada como:

$$p_k(x) = \frac{f_k(x)}{N} \tag{3.6}$$

y $p_k^2(x)$ es otra probabilidad que estima el número de apariciones de $x$ en $D$ teniendo en cuenta $n$, el cual representa el número de valores que puede asumir el atributo $k$, tal que:

$$p_k^2(x) = \frac{f_k(x)(f_k(x)-1)}{N(N-1)} \tag{3.7}$$

Nótese que en la tabla para la medida *Lin1* se cumple $Q \subseteq X_k : \forall_q \epsilon Q, p_k(A_q) \leq p_k(q) \leq p_k(B_q)$ asumiendo $p_k(A_q) \leq p_k(B_q)$. Para la medida *Goodall1* se cumple $Q \subseteq X_k : \forall_q \epsilon Q, p_k(q) \leq p_k(A_q)$ y para *Goodall2* $Q \subseteq X_k : \forall_q \epsilon Q, p_k(q) \geq p_k(A_q)$.

### 3.4.3 Distancias de similitud en datos mixtos

Las variables mixtas contienen valores tanto numéricos como cualitativos, siendo muy frecuentes en sistemas de detección de anomalías distribuidos con diferentes fuentes de información. La aproximación más frecuente a la comparación de este tipo de datos consiste en la aplicación de algoritmos de agrupamiento, lo que requiere de la identificación del número de grupos a definir y la selección de los atributos que serán tenidos en consideración dentro de cada uno de ellos. Tal y como indicaron A. Foss et al. [FMRH16], la literatura relacionada con este problema se centra en la construcción de grupos de un único atributo, siendo habitual la conversión de los valores categóricos a numéricos o viceversa. En la actualidad existe una gran variedad de medidas de proximidad para datos mixtos, hallándose en [GMW07] un estudio en profundidad de cada uno de ellos. Según M.H. Bhuyan et al. [BBK14], los dos más habituales son el coeficiente general de similitud y el coeficiente general de distancia, a los que [GMW07] le añaden la distancia generalizada de Minkowski. En la Tabla 3.3 se resume cada uno de ellos.

### 3.4.4 Distancias de similitud para casos de uso específicos

En determinados contextos es necesario adaptar las funciones para la comparativa de datos convencionales (cuantitativos, cualitativos y mixtos) a las características de la información a tratar. Esto ha derivado en la publicación de nuevas distancias, válidas únicamente en escenarios mucho más restrictivos, pero sobre los que operan de manera más eficaz. A continuación se repasan dos de los escenarios específicos más importantes en la bibliografía de la detección de anomalías relacionada con el objeto de estudio de esta tesis: el análisis de series temporales y los conjuntos de datos agrupados.

#### 3.4.4.1 Similitud en series temporales

Las series temporales son secuencias de datos observados en el tiempo medidos en intervalos regulares, en los que debe estudiarse el orden en que aparece cada registro. Es importante tener en cuenta que algunas de las distancias de propósito general son ampliamente utilizadas en este contexto, como por ejemplo las distancias Minkowski, euclidiana o euclidiana cuadrática [GMW07]. Con el fin de mejorar la calidad de su análisis, las series temporales frecuentemente son pre-procesadas y transformadas. Esta tarea permite estabilizar sus valores eliminando ruido e incluso separando sus diferentes componentes, dando pie a métodos avanzados de análisis [EA12]. Ejemplos de estrategias de comparación de series temporales son la medición de la Subsecuencia Común más Larga o LCS (del

Tabla 3.3: Ejemplos de distancias de similitud para datos mixtos.

| Nombre | Medida $dist(A, B)$ | Condiciones |
|---|---|---|
| Coeficiente General de Similitud | $\frac{1}{\sum_{K=1}^{D} W(A_k,B_k)} \sum_{k=1}^{d} w(A_k, B_k)$ | ▪ Para datos cuantitativos: $dist(A,B) = 1 - \frac{\lvert A_k - B_k \rvert}{R_k}$ donde $R_k$ es el rango del $k^{th}$ atributo; $w(A_k, B_k) = 0$ si $A$ o $B$ no tienen valor en $k$; en caso contrario $w(A_k, B_k) = 1$<br>▪ Para datos cualitativos: $dist(A_k, B_k) = 1$ si $A_k = B_k$; en caso contrario $dist(A_k, B_k) = 0$; $dist(A_k, B_k) = 0$ si $A$ o $B$ no tienen valor en $k$; en caso contrario $w(A_k, B_k) = 1$ |
| Coeficiente General de Distancia | $(\frac{1}{\sum_{K=1}^{D} W(A_k,B_k)} \sum_{k=1}^{d} w(A_k, B_k) d^2(A_k, B_k))^2$<br>▪ $d^2(A_k, B_k)$ es la distancia cuadrática para el $k^{th}$ atributo;<br>▪ $w(A_k, B_k)$ es igual que en el Coeficiente General de Similitud | ▪ Para datos cuantitativos: $dist(A,B) = \frac{\lvert A_k - B_k \rvert}{R_k}$, donde $R_k$ es el rango del $k^{th}$ atributo<br>▪ Para datos cualitativos: $dist(A_k, B_k) = 0$ si $A_k = B_k$; en caso contrario $dist(A_k, B_k) = 1$; |
| Distancia de Minkowski Generalizada | $(\sum_{j=1}^{d} \varphi(A_k, B_k)^p)^{\frac{1}{p}}$ | ▪ Sea $A_k \epsilon B_k$ la unión cartesiana entre$A_k y B_k$ del $k^{th}$ atributo; $A_k \epsilon B_k$ es un intervalo cerrado:$A_k \epsilon B_k = \min(A_{kL}, B_{kL}), \max(A_{kU}, B_{kU})$ donde $A_{kL}$ y $A_{kU}$ son los limites superior e inferior del intervalo $A_k$ |

inglés *Longest Common Subsequence*) [WF74], alineamiento de secuencias [MVSOGV16], alineamiento temporal dinámico o DTW (del inglés *Dynamic Time Warping*) [PG14], búsqueda de coincidencias probabilista [ASW15] y medición basada en puntos de referencia [GMW07]. La detección de anomalías por medio del análisis de series temporales ocupa un papel relevante en la bibliografía, siendo su uso habitual en el reconocimiento de patrones y en la identificación de comportamientos inesperados tomando como referencia modelos predictivos. En su aplicación a la seguridad de la información forman parte de estudios de muy diversa índole, que abarcan desde el análisis del comportamiento de los usuarios dentro de un sistema [MVSOGV16] hasta la identificación de discordancias en el volumen de tráfico habitual y legítimo de una red [BBK15].

#### 3.4.4.2 SIMILITUD EN DATOS AGRUPADOS

El agrupamiento de información es una de las ramas más importantes de la minería de datos. En ella se proponen soluciones al problema de la identificación de grupos de observaciones de características parecidas, que sean diferentes entre sí. Según dos Santos et al. [SZ15], evaluar la calidad de un agrupamiento requiere considerar distancias internas y externas, por lo que estas medidas no sólo tienen una repercusión importante en la detección de anomalías. De entre ellas, las distancias internas tienen en cuenta la

proximidad y distribución de las observaciones asignadas a un mismo grupo, lo que abarca métricas como su entropía [Rez19] o compactación [ZM97]. En [RAAQ11] se recopilan muchas de estas aproximaciones; por otro lado, las distancias externas determinan la relación entre elementos de diferentes grupos o de los grupos en sí. En [GMW07] se repasan las principales medidas entre grupos, siendo las distancias basadas en la media, vecinos más próximos y vecinos más lejanos las más referenciadas por asentar la base de algoritmos clásico como k-medias o K-NN [Jai10]. En términos generales, las distancias y medidas de similitud en datos agrupados desempeñan un papel esencial en la detección de intrusiones basada en el reconocimiento de anomalías. Un claro ejemplo de ello puede observarse en [LKT15], donde para este fin se propone el uso de dos medidas: la distancia de cada observación respecto al valor central del grupo al que pertenece, y su distancia con su vecino más próximo dentro de su mismo grupo.

## 3.5 Anomalías en entornos de monitorización no-estacionarios

La mayor parte de los métodos de aprendizaje automático y minería de datos, y en especial, aquellos que se orientan al reconocimiento de anomalías, asumen la premisa de que las colecciones de datos de referencia presentan distribuciones estacionarias. También asumen que la información que tendrán que analizar proviene de un entorno de características similares, situación que no siempre se satisface en su despliegue en casos de uso reales. En consecuencia, y tal y como es discutido en [Han06], esto puede llevar a comportamientos poco realistas e impredecibles. Con el fin de introducir al lector en las dificultades que plantean estos escenarios de monitorización, en esta sección se describen las causas que llevan a cambios representativos en la información monitorizada, así como sus consecuencias y las diferentes estrategias desarrolladas para su mitigación.

### 3.5.1 Escenarios no-estacionarios y sus consecuencias

Según R.C. Holte [Hol93], la aproximación tradicional al reconocimiento de patrones supone que las muestras de referencia son representativas de las observaciones que serán realizadas en el futuro, resaltando la posibilidad de que, a partir de un evento concreto, la relación entre las variables a estudiar cambie considerablemente. Kelly et al. [KHA99], ayudándose del teorema de Bayes describieron las tres maneras más probables de que se produzcan estas fluctuaciones: dada la observación $x$ y la clase $w$, la probabilidad de cambios de distribución en el entorno de monitorización se define a partir de la expresión:

$$P\left(w|x\right) = \frac{P\left(x|w\right)P(w)}{P(x)} \tag{3.8}$$

donde 1) $P(w)$ es susceptible de cambiar en el tiempo; 2) la distribución de $w$ puede variar; 3) $P\left(w|x\right)$ podría cambiar en el futuro.

Otro problema que puede mermar la calidad de las clasificaciones es la presencia de cambios paulatinos a lo largo del tiempo en las características estadísticas de la clase

a la que pertenece una observación, de tal manera que $P_t(w|x) \neq P_{t+1}(w|x)$. En la bibliografía esta variación es conocida como concepto-deriva (del inglés *concept drift*), [EP11]. Cuando los cambios se producen en un periodo corte de tiempo, son denominados concepto-giro (del inglés *concept shift*). En [DRAP15] se profundiza en las características de estas fluctuaciones en diferentes escenarios de monitorización Nótese que algunos escenarios son especialmente susceptibles a este tipo de incongruencias, como las redes de comunicaciones o los datos económicos. Las consecuencias de analizar muestras pertenecientes a distribuciones no estacionaras en la detección de anomalías, fueron descritas por C. O'Reilly et al. [OGIR14] de la siguiente manera:

- Si se producen variaciones en la distribución de la clase que agrupa las muestras consideradas normales, es probable que cambie su delimitación, y por lo tanto también lo hace $P(normal|x)$.
- Si se producen variaciones en las proporciones de datos anómalos y normales, también lo hace $P(anomalía)$, y esto puede llegar a afectar $P(normal)$.

Por lo tanto, el tratamiento de este tipo de información debe de ser tenido en cuenta en la etapa de diseño de las herramientas de clasificación, y su comportamiento afectará al proceso completo de análisis.

### 3.5.2 Estrategias de detección en entornos no-estacionarios

Con el fin de adaptar las estrategias de detección convencionales a los desafíos que plantean los escenarios de monitorización no-estacionarios, la comunidad investigadora ha desarrollado diferentes aproximaciones. Según C. O'Reilly et al. [OGIR14], la manera más apropiada de agrupar todas estas propuestas aplica como eje de clasificación las características de las etapas involucradas en el proceso de análisis. La primera de estas etapas es la detección de cambios relevantes en la distribución de los datos monitorizados. Esto se lleva a cabo combinando técnicas de identificación de novedades y detección de puntos de cambio (ver Sección 3.1.2 "Temas de investigación relacionados"). El siguiente paso es la actualización de los modelos asociados a las clases a identificar. Nótese que existen aproximaciones que no requieren de la detección, permitiendo su actualización constante a lo largo del tiempo. C. O'Reilly et al. [DRAP15] dividieron todas estas estrategias en dos grandes grupos: métodos de respuesta activa y pasiva. A continuación se describe brevemente cada uno de ellos.

#### 3.5.2.1 Métodos basados en respuestas activas

Las respuestas activas se caracterizan por abarcar métodos de modelado y aprendizaje automático que actúan tras identificarse un cambio en el entorno de monitorización, o al considerarse que el error derivado de la identificación de anomalías depende de manera representativa del proceso de modelado. En [Ali14] a este método se le denomina *detección y reacción*; una vez que se detecta el cambio, el sistema descarta el conocimiento obsoleto y se adapta al entorno. Estos mecanismos habitualmente se agrupan en tres familias: uso

de ventanas, ponderación y muestreo aleatorio. La primera de ellas es la más frecuente en la bibliografía, y se basa la aplicación de una ventana deslizante que permite seleccionar las muestras de referencia más recientes y descartar las antiguas. A partir de ellas se reconstruye el modelo. En [AR08] se describe en profundidad este método, se indican algunas de sus variantes, y se discuten algunos de los principales problemas que acarrea, destacando de entre ellos la dificultad de seleccionar un tamaño de ventana óptimo.
A diferencia de los métodos basados en el uso de ventanas deslizantes, las propuestas basadas en ponderación consideran la colección completa de muestras de referencia. Al detectarse el cambio en el entorno de monitorización, aquellas con mejores pesos tendrán una mayor relevancia en la construcción del nuevo modelo. Nótese que el criterio de ponderación puede variar en función del caso del uso, siendo frecuente la consideración de su antigüedad [Koy00], o la valoración del error de etiquetado y su factor de cambio [Kli04].
Finalmente, los métodos basados en muestreo aleatorio tienen en cuenta un subconjunto particular de la colección de muestras de referencia, cuyos componentes han sido seleccionados aleatoriamente. Existen diferentes variaciones de este paradigma, como el muestreo sin reemplazamiento (i.e. cada muestra solo puede seleccionarse una vez), muestreo con desplazamiento o el reservorio de muestreo (del inglés *reservoir sampling*) [ND08]. En [DRAP15] se profundiza en éste área y se muestran más ejemplos.

#### 3.5.2.2 Métodos basados en respuestas pasivas

Las estrategias de detección de anomalías basadas en respuestas pasivas no requieren de la identificación de un evento para activarse. En lugar de esto, asumen que el entorno de monitorización varía de manera constante a lo largo del tiempo. Para acomodarse a estas fluctuaciones, realizan pequeños cambios de manera continuada sobre los modelos/regresiones de referencia, manteniendo de este modo información actualizada acerca del contexto en que la información es observada. Según [DRAP15], las diferentes contribuciones a este campo pueden agruparse en dos categorías: las que se centran en la actualización de un único sistema de detección, y las que afectan a clasificadores que integran diferentes sensores. Las primeras de ellas son más eficientes, y por lo tanto más recomendables para sistemas que operan en tiempo real. Esto se ilustra con claridad en [LLZ09], donde se aplican árboles de decisión para el análisis de secuencias de información. Otro ejemplo es [CALK08], donde se combinan lógica difusa y métodos basados en el uso de ventanas deslizantes sobre las muestras de referencia. Finalmente, en [YSP13] se aplican estrategias de aprendizaje automático extremo o ELM (del inglés *Extreme Learning Machines*) sobre redes neuronales cuyo ajuste varía a lo largo del tiempo.
Por otro lado, las combinaciones de sensores (del inglés *ensembles*) han demostrado comportarse de manera mucho más estable que los sensores individuales, en entornos de monitorización estacionarios. Esto es debido a que tienden a compensar el error de los clasificadores que peor se comportan en cada caso de uso. Además, facilitan la incorporación de nuevos datos en los modelos que han construido, y proveen estrategias para descartar la información menos relevante [ZCS13]. Por lo tanto, en circunstancias estacionales, la diversidad de sensores en una combinación afecta de manera beneficiosa

a sus resultados. Tal y como demostraron L. L. Minku et al. [MWY10], en entornos no estacionarios sucede exactamente lo mismo. En [MY12] se ilustra un ejemplo de la aplicación de diferentes niveles de diversidad con el fin de mejorar la adaptación al medio. En [BS14] se reúnen algunas de estas propuestas orientadas al análisis de secuencias, y se propone una nueva técnica basada en ponderación y árboles de decisión.

## 3.6 MÉTRICAS Y METODOLOGÍAS DE EVALUACIÓN

En las últimas décadas se han publicado diferentes metodologías para la evaluación de sistemas de reconocimiento de anomalías. En ellas no sólo se tiene en cuenta su capacidad de identificación de discordancias; también se miden otros parámetros que facilitan la decisión de qué estrategias de detección se adaptan mejor a cada caso de uso. Debido a la gran cantidad de metodologías de evaluación presentes en la bibliografía, en la actualidad existe controversia acerca de cuáles de ellas ofrecen información más fiable, y cuáles permiten una mejor comparativa del sistema a evaluar respecto a propuestas anteriores. Tal y como indican M.H. Bhuyan et al. [BBK14], la evaluación de un sistema de reconocimiento de anomalías únicamente considerando su precisión, tan sólo ofrece una instantánea de su efectividad en un instante de tiempo concreto. Por lo tanto, a medida que se producen cambios en el escenario de monitorización, su comparativa con futuras propuestas tiende a perder relevancia. Además, si se usa únicamente este criterio de evaluación, no se adquiere una visión general de las consecuencias de desplegar el sensor en los distintos casos de uso. En consecuencia, la comunidad investigadora también se ha ayudado de otros criterios a la hora de estimar el impacto de un sistema de detección de anomalías sobre un entorno de monitorización concreto. Los más frecuentes en la bibliografía son: precisión, rendimiento, tiempo de respuesta, facilidad de actualización, escalabilidad, resistencia a ataques de evasión y consumo de energía. Con el fin de introducir al lector en esta problemática y de facilitar la comprensión de las metodologías de evaluación implementadas en las siguientes secciones de este documento, el resto de esta sección revisa los principales criterios de evaluación de sistemas de reconocimiento de anomalías.

### 3.6.1 PRECISIÓN

La precisión de un sistema de reconocimiento de anomalías es su capacidad de detectar observaciones discordantes y distinguirlos de datos normales. La naturaleza del etiquetado de las muestras a analizar se resume en cuatro clases: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. En la Figura 3.3 se muestran las distintas evaluaciones del etiquetado emitido por un IDS basado en anomalías con dos clases de datos ("Anomalía" y "Normal"). Los conjuntos de alertas de los cuadros verdes (verdaderos negativos y verdaderos positivos) representan categorías de etiquetado correcto mientras que el resto corresponde a errores de detección. A continuación se describe cada una de ellas:

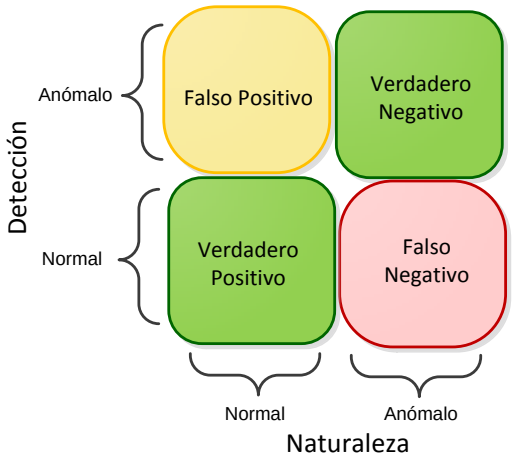



Figura 3.3: Evaluación del etiquetado de un detector de anomalías de dos clases.

- *Verdaderos positivos* o TP (del inglés *True Positives*). Tomando como referencia la clase $C$, los verdaderos positivos se observan cuando el sensor indica que las muestras analizadas pertenecen a $C$ y es correcto. En el ejemplo los verdaderos positivos o son clasificaciones de observaciones discordantes etiquetadas correctamente como anómalas. La tasa de falsos positivos del sistema a menudo es denominada tasa de acierto o sensibilidad.

- *Verdaderos negativos* o TN (del inglés *True Negatives*). Tomando como referencia la clase $C$, los verdaderos positivos se observan cuando el sensor indica que las muestras analizadas no pertenecen a $C$ y es correcto. En el ejemplo, son clasificaciones de observaciones normales etiquetadas correctamente como normales.

- *Falsos positivos* o FP (del inglés *False Positives*) Tomando como referencia la clase $C$, los verdaderos positivos se observan cuando el sensor indica que las muestras analizadas pertenecen a $C$, pero en realidad pertenecen a una clase distinta. En el ejemplo son clasificaciones de observaciones normales, incorrectamente clasificadas como anómalas. Además de suponer un problema para la calidad de servicio ofrecida por el detector, este tipo de errores pueden ser aprovechados por atacantes para forzar la emisión de grandes cantidades de alertas, de este modo causando el agotamiento de recursos de cómputo del sistema [TA05].

- *Falsos negativos* o FN (del inglés *False Negatives*). Tomando como referencia la clase $C$, los verdaderos positivos se observan cuando el sensor indica que las muestras analizadas no pertenecen a $C$, pero en realidad pertenecen a una clase distinta. En el ejemplo son clasificaciones de actividades anómalas que erróneamente han sido identificadas como normales. Se trata del peor caso posible, ya que representan lo opuesto al objetivo del sistema de detección: no identificar anomalías. La tasa de falsos negativos habitualmente es denominada especificidad.

En base a esta clasificación, M.H. Bhuyan et al. [BBK14] concluyeron que el objetivo del sistema de reconocimiento de anomalías es alcanzar las mayores tasas de verdaderos positivos y negativos posibles, y el menor número de falsos positivos y negativos posibles. La precisión de muchos de estos métodos se ajusta mediante el balanceo entre la cantidad de falsos negativos y falsos positivos, por medio de los parámetros de ajuste del sensor. De este modo, cuando el sistema opera sobre entornos de monitorización que requieren mayor protección, lo habitual es disminuir su nivel de permisividad; así se mejora su tasa de falsos negativos, pero se incrementa la de falsos positivos. Por el contrario, la adaptación a entornos de monitorización menos sensibles habitualmente conlleva la reducción de la tasa de falsos positivos por medio de penalizar la tasa de acierto del sistema. En las últimas décadas se han empleado diferentes herramientas para el análisis de la calidad del etiquetado de los detectores de anomalías, destacando por su relevancia en la bibliografía las curvas ROC y las métricas derivadas de la construcción de matrices de confusión. Ambos son descritos a continuación. Para conocer más acerca del resto de criterios de evaluación, se recomienda consultar [FHOM09].

#### 3.6.1.1 CURVA ROC

La curva ROC (del inglés *Receiver Operating Characteristic*) es una representación gráfica de la sensibilidad del sistema frente a su especificidad [Faw06]. A partir de ellas se genera el estadístico Área Bajo la Curva o AUC (del inglés *Area Under the Curve*) para la comparación de detectores en función de su calibrado, expresado mediante:

$$AUC = \int_{-\infty}^{\infty} TPR(X)FPR'(X)dX \tag{3.9}$$

La la Figura 3.4 ilustra un ejemplo de curva ROC, donde en el eje superior izquierdo se agrupan los mejores ajustes, es decir, las más próximas al calibrado óptimo $TP = 1$ Y $FP = 0$. Los peores calibrados reflejan los valores más cercanos a la esquina inferior derecha, es decir, $TP = 0$ y $FP = 1$. El AUC de esta curva ROC es 0.9.

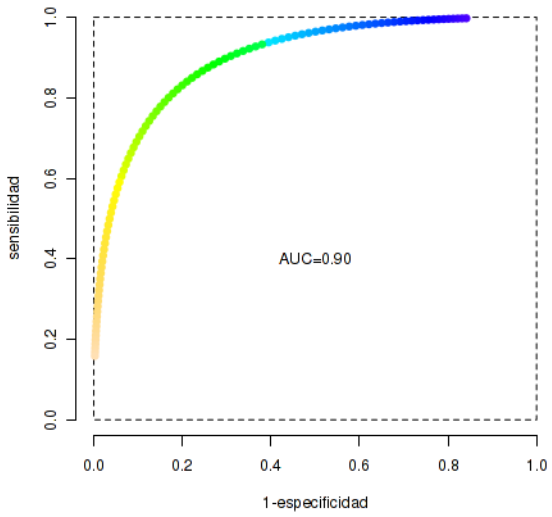



Figura 3.4: Ejemplo de curva ROC.

En [SWZK12] se lleva a cabo un estudio exhaustivo acerca del uso de la curva ROC y el AUC en el reconocimiento de anomalías. En esta publicación además se revisa la problemática que suscita su implementación, resaltando como principal causa el hecho de que el análisis ROC no tiene en cuenta el grado de discordancia de cada clasificación realizada. Para evitar este problema proponen el uso de métricas basadas en el grado de discordancia detectado. En [CZS+16] se recopilan muchas de ellas, destacando Precision-at-n (prec@n) e IREOS [MCZS15] por ser algunas de las más recientes. Nótese que a pesar de este inconveniente, las curvas ROC son la herramienta más utilizada en la bibliografía.

#### 3.6.1.2 Matriz de confusión

Como alternativa al análisis ROC, también es frecuente el uso de matrices de confusión, también conocidas como matrices de error o de contingencia. Estas estructuras son de dimensiones $nxn$, siendo $n$ el número de clases a tener en cuenta, las columnas son los datos de referencia, y las filas la cantidad de observaciones etiquetadas como pertenecientes a cada clase. Por lo tanto, la diagonal indica la clasificación correcta. En la Tabla 3.4 se muestra un ejemplo de matriz de confusión, donde las clases *Legi1* y *Legi2* agrupan las observaciones normales, siendo *Anomalía* la que contiene las discordancias.

Tabla 3.4: Ejemplo de matriz de confusión para tres clases.

| | Predicciones | | |
|---|---|---|---|
| Clase | *Legi1* | *Legi2* | *Anomalía* |
| *Legi1* | 900 | 800 | 123 |
| *Legi2* | 1070 | 893 | 126 |
| *Anomalía* | 875 | 753 | 175 |

Nótese que la detección de anomalías raramente considera $n > 2$, siendo las clases definidas: datos anómalos y normales. Una vez construida la matriz de confusión, es posible deducir diferentes índices relacionados con el comportamiento del sensor, destacando de entre ellos su precisión o ACC (del inglés *Accuracy*), precisión o Pre (del inglés *precision*), exhaustividad o Re (del inglés *Recall*) y proposición de fallo o Fa (del inglés *Fall-Out*) [BBK14]. Éstos están definidos por las siguientes expresiones:

$$ACC = \frac{\sum P + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN} \tag{3.10}$$

$$Pre = \frac{\sum TP}{\sum TP + \sum FP} \tag{3.11}$$

$$Re = \frac{\sum TP}{\sum TP + \sum FN} \tag{3.12}$$

$$Fa = \frac{\sum FP}{\sum TP + \sum FP} \tag{3.13}$$

En el ejemplo de matriz de confusión de la Tabla 3.4, la precisión global del sensor es $ACC = 0.34$; la precisión para cada clase por separado es $Pre(Legi1) = 047$, $Pre(Legi2) = 0.42$ y $Pre(Anomal\acute{i}a) = 0.09$; la exhaustividad de cada clase por separado es $Re(Legi1) =$

0.31, $Re(Legi2) = 0.35$ y $Re(Anomalía = 0.41)$. Finalmente, las proposiciones de fallo son $Fa(Legi1) = 0.53$, $Fa(Legi2) = 0.58$ y $Fa(Anomalía) = 0.91$. A la vista de estos resultados, es posible concluir que este sensor es muy poco eficaz.
Para unificar en un solo valor el contraste entre la precisión *Pre* de cada clase de datos y su exhaustividad *Re*, habitualmente se consideran las medidas-F (del inglés *F-measure*). Cada valor $F_\beta$ es una medida armónica generalizada en la siguiente expresión:

$$F_\beta = (1 + \beta^2)\frac{Pre \times Re}{(\beta^2 Pre) + Re} \tag{3.14}$$

siendo $\beta$ un valor positivo real. La medida-F tradicional, conocida como $F1$, es una de las más utilizadas en la bibliografía. A partir de ello se obtiene la expresión anterior en la siguiente ecuación:

$$F_1 = 2\frac{Pre \times Re}{Pre + Re} \tag{3.15}$$

Nótese que los valores $F_1$ para el ejemplo de la Tabla 3.4 son: $F_1 Legi1 = 0.37$, $F1_L egi2 = 0.38$ y $F_1 Anomalía = 0.147$, lo que reafirma la conclusión de que el sensor del ejemplo es poco eficaz. En [PFR17] se revisa en profundidad estos criterios de evaluación y se describe la evolución de los sistemas de clasificación de información hacia la optimización de sus resultados.

### 3.6.2 RENDIMIENTO

El rendimiento de un sistema de detección de anomalías determina su capacidad de procesamiento de información en función del tiempo. Este parámetro habitualmente depende de la estrategia de detección, y de la capacidad de procesamiento del entorno de monitorización. Nótese que, si el rendimiento de un sensor es inferior a la capacidad de procesamiento de información del escenario sobre el que actúa, garantizar su operatividad en tiempo real implica la limitación de la tasa de procesamiento de datos del entorno de monitorización. Esta situación habitualmente acarrea una importante penalización en su calidad de servicio. En la actualidad existen diferentes métricas para evaluar el rendimiento de un sistema de reconocimiento de anomalías. Algunas de ellas son de ámbito general, como por ejemplo el consumo de memoria del sistema o tiempo de procesamiento por unidad de información; otras se adaptan a las características del caso de uso, como sucede al analizar tráfico de redes y considerar su tasa de pérdida de paquetes [BBK14]. En [LMS+02] se explica detalladamente cada uno de los costes computacionales involucrados en el proceso de detección, así como los mecanismos para alcanzar el estado de equilibrio entre la precisión y el rendimiento del sensor.

### 3.6.3 TIEMPO DE RESPUESTA

Los criterios basados en tiempo de respuesta evalúan la rapidez con la que un sistema es capaz de informar de la presencia de una anomalía desde que esta sucede en el entorno de monitorización. A diferencia de las métricas para la estimación del rendimiento, el tiempo de respuesta tiene en cuenta que determinados procesos necesitan una ventana

de observación previa al análisis de los datos. Por lo tanto, al tiempo invertido en el procesamiento de información se le añade el tiempo de su obtención. Un ejemplo que ilustra con claridad esta idea se observa en [OB15], donde se lleva a cabo el estudio del impacto de la variación del tamaño de las unidades básicas de información a analizar (observaciones), con el fin de reconocer ataques de denegación de servicio en redes. A partir de los datos monitorizados, en cada periodo de observación se extrae la métrica fundamental para el proceso de análisis (entropía del número de paquetes recibidos). Según [OB15], las observaciones pueden definirse como conjuntos de datos (paquetes) de dimensión predefinida d extraídos de manera secuencial del entorno de monitorización, o bien como el conjunto de datos (paquetes) monitorizados en un intervalo de tiempo concreto $T$. El impacto sobre la eficacia del sensor, de tomar una u otra decisión, depende de las características del entorno de monitorización: si se analiza una red con un nivel de tráfico alto, probablemente se cumpla que el tiempo de captura de d paquetes sea menor que el intervalo $T$. En este caso, el tiempo de respuesta será mejor. En el caso contrario es preferible la segunda opción. En [BBK14] se profundiza en los parámetros a tener en cuenta para la evaluación del tiempo de respuesta en sistemas de detección de anomalías.

### 3.6.4 Facilidad de actualización

Dada la naturaleza no-estacionaria de la mayor parte de los escenarios de monitorización (ver Sección 3.5.2 “Anomalías en entornos de monitorización no-estacionarios”), a la hora de evaluar un sistema de reconocimiento de anomalías es conveniente tener en cuenta el coste que conlleva la modificación de las reglas de detección, modelos y demás elementos necesarios para la elaboración de los perfiles de uso normal y anómalo del sistema. Por lo tanto, la facilidad de actualización tiene un impacto directo en el tiempo de respuesta del sensor [OB15] y en la calidad de experiencia QoE (del inglés *Quality of Experience*) que perciben sus usuarios. Esto último se observa claramente en ciertos casos de uso, como por ejemplo en los sistemas de acceso a sistemas basados en biometría. Tal y como se demuestra en [DP09], algunos rasgos biométricos tienden a variar con el tiempo. Para que el detector de anomalías actúe de manera eficaz, los usuarios registrados en el sistema deben actualizar periódicamente su información de referencia. Si este proceso requiere de la inserción de muchas muestras, o si su elaboración resulta especialmente tediosa, la QoE es penalizada. En términos generales, la facilidad de actualización es evaluada a partir de métricas objetivas y subjetivas de QoE. En [BH10] se profundiza en ellas, y en las metodologías relacionadas con su adquisición. La facilidad de actualización también puede medirse teniendo en cuenta la observación de su impacto en el tiempo de respuesta (ver Sección 3.6.3 “Tiempo de respuesta”).

### 3.6.5 Escalabilidad

La escalabilidad de un sistema de reconocimiento de anomalías es su capacidad de adaptarse al crecimiento del escenario de monitorización sobre el que actúa. Esta situación a menudo tiene un impacto directo sobre su capacidad de procesamiento de datos, siendo un sistema escalable aquel que es capaz de adaptarse a estos cambios sin consecuencias

representativas en su eficacia. La escalabilidad puede medirse en diferentes dimensiones como, por ejemplo, su capacidad de incorporar nuevas funcionalidades o la facilidad de adaptación a nuevos niveles de carga de trabajo. En consecuencia, los criterios a tener en cuenta a la hora de evaluar la escalabilidad de un detector de anomalías dependen directamente de su implementación y el caso de uso para el que es desplegado. Por ejemplo, la escalabilidad a nivel lógico del sensor puede considerar en qué medida pueden admitirse nuevas clases de información correspondientes a nuevos perfiles de comportamiento normal. A nivel físico, podría ser importante estudiar si es posible su despliegue distribuido en diferentes sistemas de cómputo, y hasta qué punto este despliegue no afecta a la eficacia del sensor. Muchas de las métricas tradicionalmente empleadas en la evaluación de la escalabilidad son recopiladas en [JW00]. Pero a pesar de su gran variedad, tal y como indican Xiong et al. [XZZ+14] en la actualidad no existen criterios unificados para su aplicación. Cada propuesta analiza la escalabilidad desde un punto de vista concreto, orientada a un caso de uso específico.

### 3.6.6 ROBUSTEZ ANTE MÉTODOS DE EVASIÓN

Las estrategias para el reconocimiento de anomalías a menudo forman parte de sistemas encargados de analizar y permitir el acceso a información sensible. Debido a esto, muchos atacantes han desarrollado métodos para tratar de disminuir su eficacia, los cuales abarcan desde intentos de denegación de servicio [TA05], hasta la alteración de las secuencias de acciones de la intrusión con el fin de simular patrones de uso legítimo [MVSOGV16]. Muchas de estas técnicas son recopiladas en [CGR13], donde además se discuten sus contramedidas más relevantes y los nuevos desafíos hacia su mitigación. Para que un sistema de reconocimiento de anomalías sea eficaz, debe de ser robusto ante este tipo de acciones. Pero a pesar de la importancia de esta premisa, en la actualidad no existen criterios unificados para su valoración, siendo habitualmente evaluada por medio de la comparación de su precisión (ver Sección 3.6.1 “Precisión”) con la de propuestas similares.

### 3.6.7 CONSUMO DE ENERGÍA

Los avances en el reconocimiento de anomalías han conllevado la publicación de algoritmos mucho más eficaces, pero que a su vez demandan el consumo de una mayor cantidad de recursos de cómputo. Esta característica tiene un impacto especialmente importante en las tecnologías móviles, donde el elevado consumo energético es capaz de reducir de manera muy representativa la autonomía del dispositivo; esto conlleva una penalización en la calidad de experiencia QoE que percibe el usuario. Por lo tanto, es recomendable que el impacto de los sistemas de reconocimiento de anomalías sobre la batería del dispositivo sea mínimo. En [PDP+15] se lleva a cabo un estudio sobre las consecuencias de ejecutar aplicaciones para la detección de intrusiones, donde se demuestra la relevancia de este rasgo. También se propone una metodología para alcanzar un punto de equilibrio entre la eficacia de los algoritmos y sus requisitos energéticos. En [CGL+16] se muestra cómo a partir de la detección de anomalías en las características de consumo energético del dispositivo, se facilita la identificación de acciones malintencionadas.

# Capítulo 4

# Desafíos y nuevos escenarios de monitorización

Este capítulo se centra en la revisión y posterior discusión de los desafíos que plantea el reconocimiento de anomalías aplicado a la detección de intrusiones. Una vez identificados sus principales retos, se profundiza en varios casos de uso de especial interés, algunos de los cuales han asentado las bases de las contribuciones que se describen en capítulos posteriores. El contenido del capítulo está organizado de la siguiente manera: en la Sección 4.1 se describen las principales dificultades y desafíos del reconocimiento de anomalías, haciéndose hincapié en los retos que plantean los nuevos escenarios de monitorización. Las siguientes cinco secciones se centran en diferentes casos de uso; en concreto, la Sección 4.2 revisa el problema de la detección de atacantes enmascarados; la Sección 4.3 la detección de malware por medio del análisis de la carga útil en redes de comunicaciones; la Sección 4.4 profundiza en la correlación de alertas derivadas de la identificación de discordancias; la Sección 4.5 analiza las estrategias de mitigación de ataques de denegación de servicio; y finalmente la Sección 4.6 discute los avances en la detección de malware en dispositivos móviles.

## 4.1 Dificultades y desafíos en los nuevos escenarios de monitorización

La evolución en los escenarios de monitorización ha sido propiciada por el avance tecnológico. Esto ha llevado a la aparición de sistemas de cómputo mucho más complejos, con mayor capacidad de procesamiento y que son capaces de manejar la información proporcionada por gran cantidad de fuentes de diferente naturaleza. En consecuencia, las nuevas propuestas relacionadas con la identificación de anomalías deben de hacer frente a una mayor cantidad de información de características mucho más heterogéneas. A este problema se le añaden los desafíos que ya planteaban los esquemas de detección de anomalías convencionales, como la pericia de algunos atacantes para lograr su evasión, o los nuevos retos de la sociedad de la información, como su accesibilidad universal o la salvaguarda de la privacidad de los usuarios. Con el fin de facilitar el desarrollo de

nuevas herramientas de detección, esta sección reúne y discute las principales dificultades y desafíos que plantea el reconocimiento de anomalías en escenarios de monitorización actuales. Los siguientes problemas son discutidos a continuación: altas tasas de falsos positivos, ausencia de una estrategia de detección universal, métodos de evasión, entornos de monitorización de características variables, disponibilidad de conjuntos de muestras de entrenamiento y validación, dificultad en la elección de las distancias y medidas de similitud, y consumo de recursos.

### 4.1.1 Altas tasas de falsos positivos

La dificultad de establecer las métricas y delimitaciones que distinguen el conjunto de observaciones normales de las que no lo son no es un problema nuevo. V. Chandola et al. [CBK09] ya avisaron de que es frecuente que en sistemas de reconocimiento de anomalías se confundan observaciones normales con anómalas y viceversa; esto sucede incluso en aproximaciones capaces de adaptarse a cambios en el entorno de monitorización (ver Sección 3.5 "Anomalías en entornos de monitorización no-estacionarios"). Dichos errores de etiquetado son mucho más frecuentes al tratar conjuntos de datos cuyas características son más propensas a experimentar variaciones, como por ejemplo sucede en la información que circula a través de redes [BBK14]. La aparición de los nuevos escenarios de monitorización lleva a la necesidad de analizar una mayor cantidad de información, y de rasgos mucho más heterogéneos, lo que en términos generales no sólo implica la perseverancia de este problema; también lo amplifica. En consecuencia, muchos investigadores han iniciado líneas de estudio centradas en la reducción de las tasas de falsos positivos y la mitigación de su impacto en el sistema, recopilándose en [HS14] algunas de las aproximaciones más relevantes.

### 4.1.2 Ausencia de una estrategia universal

Tal y como se resalta en [AM16], en la actualidad no existe una estrategia de reconocimiento de anomalías unificada, capaz de operar con efectividad en cualquier escenario de monitorización. Esto es debido entre otras cosas, a que las características de cada caso de uso plantean requisitos y limitaciones demasiado específicas, lo que a menudo dificulta la interoperabilidad de las propuestas. En consecuencia, los esfuerzos realizados en este campo han evolucionado hacia la especificidad en lugar de la unificación, situación que ha conllevado que en muchas aproximaciones se haya pasado por alto avances en la identificación de discordancias, que sí que han sido considerados en otros casos de uso.

### 4.1.3 Métodos de evasión

El reciente incremento de la popularidad de las nuevas tecnologías, así como el gran crecimiento de la sociedad de la información, han llevado a que los ataques dirigidos contra estos sistemas sean cada vez más rentables, situación que ha motivado el desarrollo de técnicas de evasión capaces de inutilizar incluso las estrategias de reconocimiento de anomalías más efectivas. Este hecho ha puesto en alerta a diferentes organismos de carácter público y privado [Eur16]. Tal y como fue descrito en la Sección 3.6.6 "Robustez ante

métodos de evasión", la consecuencia directa de estas amenazas es la necesidad de elaborar esquemas de detección robustos, capaces de resistir intentos de denegación de servicio y de detectar actividades maliciosas ofuscadas por medio de la imitación de las observaciones de referencia consideradas normales, hallándose en [TA05, CGR13] claros ejemplos de ello.

### 4.1.4 Entornos de monitorización de características variables

A medida que crece el nivel de heterogeneidad de los escenarios de monitorización, se vuelven más susceptibles a cambios que lleven a la presencia de inconsistencias entre la distribución en los datos de referencia respecto a la información a analizar [EP11]. Para Ahmed et al. [AM16] esto conlleva que los comportamientos considerados normales cambien, y no vuelvan a ser considerados normales más adelante. Del mismo modo, las observaciones inicialmente discordantes pueden llegar a constituir patrones de actividades normales. Por lo tanto, la adaptación de los métodos de detección de anomalías a estos escenarios no estacionarios (ver Sección 3.5 "Anomalías en entornos de monitorización no-estacionarios") es necesaria en la mayor parte de los casos de uso reales, siendo un tema candente en aquellas líneas de investigación relacionadas con los entornos de monitorización más susceptibles a este problema. En [DRAP15] se revisan muchas de estas aproximaciones.

### 4.1.5 Disponibilidad de conjuntos de muestras de entrenamiento y validación

La dificultad en la adquisición de colecciones de muestras para el entrenamiento y la validación de estrategias de detección de anomalías es un problema clásico en muchas de sus aplicaciones [CBK09]. Tal y como indicaron M.H. Bhuyan et al. [BBK14], el hecho de capturar y recopilar información lo suficientemente representativa como para entrenar un sensor, ya de por sí supone una tarea complicada. Además, las colecciones de muestras de dominio público a menudo contienen muchas más muestras normales que discordantes, situación que para algunos investigadores puede poner en entredicho las tasas de falsos negativos que presumen de alcanzan algunas propuestas. A. Zimmermann también estudió este problema en profundidad, resaltando de entre otros aspectos a tener en cuenta, la antigüedad de las colecciones públicas de muestras y la existencia de errores de etiquetado dentro de ellas [Zim14]. Esto último lleva a la observación de importantes diferencias entre la precisión obtenida por el sistema de detección en los estándares funcionales de evaluación, respecto a su eficacia en casos de uso reales. Una buena recopilación de conjuntos de muestras se ilustra en [MVK+15], donde además se discute en detalle las características de cada una de ellas. Como alternativa al uso de muestras recogidas directamente del entorno de monitorización, en [BSMT14] se analiza el problema de la generación de muestras sintéticas, y se propone un marco para la generación de tráfico artificial que emule el comportamiento real de una red.

### 4.1.6 Dificultad en la elección de las distancias y medidas de similitud

Tal y como se describió en la Sección 3.4 "Distancias y medidas de similitud", la elección de la distancia y las medidas de similitud apropiadas para reconocer anomalías sobre un escenario de monitorización concreto puede no ser una tarea trivial. Si bien ciertos parámetros de ajuste son fácilmente configurados a partir de un proceso de aprendizaje, la elección de las distancias apropiadas va a condicionar el comportamiento del sensor incluso desde su etapa de adquisición de conocimiento. En [WFBS14] se identifican cuatro errores típicos relacionados con el uso de distancias y medidas de similitud en las propuestas actuales: 1) selección de distancias y medidas inapropiadas, 2) parámetros de configuración de distancias inapropiados, 3) errores al justificar la elección de las distancias o medidas de similitud, y 4) errores al tratar las distancias y medidas de similitud como otros factores a tener en cuenta en la experimentación.

### 4.1.7 Consumo de recursos

La necesidad de analizar una mayor cantidad de información a menudo conlleva la necesidad de desplegar algoritmos más complejos, y por lo tanto más costosos computacionalmente. Tal y como se discutió en las secciones 3.6.2 "Rendimiento" y 3.6.3 "Tiempo de respuesta", los escenarios de monitorización actuales tienen un elevado impacto en la capacidad de procesamiento de datos de los sensores, lo que repercute directamente en su eficiencia y consumo de memoria. Al mismo tiempo, estos nuevos casos de uso cada vez requieren con mayor frecuencia de la emisión de respuestas en tiempo real [BBK14], siendo su elevado consumo de recursos una importante limitación. Asimismo, y tal y como se discutió en la Sección 3.6.7 "Consumo de energía", la reciente tendencia a implementar métodos de detección de anomalías en dispositivos móviles también demanda el desarrollo de estrategias que causen un menor impacto en la batería del dispositivo [PDP+15]. Por último, y tal y como se ilustra en [HNH13], es importante tener en cuenta que las métricas basadas en el consumo de recursos pueden ser de especial utilidad a la hora de identificar ciertas amenazas.

## 4.2 Detección de atacantes enmascarados

Tradicionalmente, la mayor parte de la investigación en el área de la seguridad de la información se ha centrado en el desarrollo de estrategias para prevenir, detectar y mitigar amenazas con origen externo al sistema protegido. En consecuencia, estas estrategias se han desarrollado con una orientación clara hacia proteger los activos del sistema frente a accesos no autorizados, generalmente ayudándose de la elaboración de perímetros defensivos. No obstante, la mayor parte de los accesos no autorizados se producen desde el interior, lo que comúnmente se conoce como *ataques internos*. Tal y como indicaron M.B. Salem et al. [SHS08], este tipo de ataques no necesitan explotar vulnerabilidades para atravesar los diferentes controles de acceso ya que, de alguna manera, satisfacen los requisitos para acceder haciéndose pasar por usuarios legítimos. Esto hace que sea muy

difíciles de detectar por los despliegues defensivos perimetrales. Según el último informe anual sobre amenazas internas publicado por el US-CERT [Cen15], la consecuencia directa de esta característica es un importante incremento en la cantidad e impacto de los ataques dirigidos desde el interior de las organizaciones; en particular, un 28% de las incidencias registradas corresponden con este tipo de amenazas, superando en un 32% de los casos las pérdidas de cualquier otro tipo de intrusión.

Salem et al. [SHS08] clasifican los ataques internos tomando como referencia las características de los atacantes. Para ello definen como *traidores* a los usuarios legítimos del sistema que tratan de ganar privilegios para acceder a información restringida. Por otro lado, definen como *enmascarados* a los usuarios no autorizados que de alguna manera han conseguido las credenciales de acceso de usuarios legítimos. Aunque la diferencia entre ambos grupos parezca insignificante, a efectos de detección es muy grande. Mientras que el perfil del traidor coincide con el de un usuario autorizado, que frecuenta el sistema protegido y que, por lo tanto, conoce a la perfección su organización, a menudo se asume que los enmascarados carecen de conocimientos previos sobre ello. Debido a esta última característica, los enmascarados pueden ser descubiertos por medio del reconocimiento de comportamientos discordantes respecto a las actividades habituales perpetradas por los usuarios legítimos. Para la detección de traidores es frecuente el despliegue de trampas y señuelos, como por ejemplo tarros de miel o marcaje de ficheros [SS11a, VJBS15]. Otra manera de hacerlos es la monitorización de cambios en los permisos gestionados por el sistema [PSW16]. Dada la importancia del reconocimiento de anomalías en el área de la detección de enmascarados, se ha considerado como objeto de estudio en el marco de esta investigación realizada. En consecuencia, a lo largo de esta sección se revisa el estado del arte relacionado con la detección de atacantes enmascarados, y se discuten aquellos aspectos de interés al plantear su mitigación por medio del reconocimiento de anomalías.

### 4.2.1 Trabajos relacionados

Las diferentes propuestas para la detección de atacantes enmascarados pueden agruparse en función de su paradigma de diseño, distinguiéndose dos grandes familias: aquellas propuestas que se basan en el estudio de 1) cómo son ejecutadas las acciones en el sistema, y 2) qué acciones son ejecutadas. Las primeras de ellas aplican los fundamentos de la autenticación dinámica basada en rasgos biométricos, como por ejemplo los movimientos del ratón del computador [AT07], la manera en que el usuario interactúa con las pantallas táctiles [LL16], giros y orientación de la pantalla de dispositivos móviles [SSYP16], o el uso del teclado [AT14]. Estas aproximaciones heredan las ventajas y desventajas inherentes al uso de biometría, destacando su gran resistencia a las falsificaciones e imitaciones, a costa de la necesidad de disponer de conjuntos de muestras actualizados periódicamente, gran sensibilidad, y tendencia a la emisión de falsos positivos. Cabe destacar que únicamente son eficaces cuando el atacante accede físicamente al sistema, y difícilmente permiten la detección de enmascarados que operen de manera remota. Por este motivo, nuestra investigación se ha centrado en el segundo grupo de estrategias.

Típicamente, la detección de enmascarados basada en el estudio de las acciones en sí, parte del análisis de los comandos o llamadas al sistema ejecutadas por el usuario. Algunas

métricas alternativas son recopiladas por V. Chandola et al. [CBK12], destacando de entre ellas las actividades de búsqueda del usuario o actividades de red. Las primeras propuestas en esta área típicamente elaboraban modelos de uso legítimo a partir de estos comandos [DH88]. Éstas eran analizadas por métodos Bayesianos [Dum99] o cadenas de Markov [ST00], y habitualmente se basaban en la identificación de comandos inusuales [JV01] o en el análisis de su frecuencia de aparición en las secuencias [YLC+02]. La evolución de estas aproximaciones comienza con la incorporación de métodos de entrenamiento semi-supervisados de una clase (ver Sección 3.3.2 "Aprendizaje semi-supervisado"). Tal y como demostraron K. Wang et. al [WS03], el uso de únicamente muestras legítimas de un usuario en la detección de enmascarados basada en anomalías es prácticamente igual de eficaz que considerar las de varios de ellos. Esto simplifica las tareas de entrenamiento y captura de datos. Sin embargo, plantea esquemas mucho más difíciles de adaptar para hacer frente a intentos de evasión basados en imitación.

En [CBSB03, CS08] S. Coull et al. introducen el uso de técnicas de alineamiento de secuencias bioinformáticas para el tratamiento de las secuencias de acciones llevadas a cabo por los usuarios del sistema. Entre las principales aportaciones de sus publicaciones cabe destacar la discusión sobre la implementación de las diferentes técnicas de alineamiento de aminoácidos, la propuesta de diferentes sistemas de puntuación (del inglés *scoring systems*), la introducción de métodos para adaptar el sistema a variaciones del comportamiento del usuario legítimo (reentrenamiento), y la propuesta de heurísticas para reducir el consumo del sistema, a costa de penalizar levemente su precisión. Posteriormente S. Sen et al. revisaron el uso de esta metodología aplicada a la detección de enmascarados [Sen15] y demostraron su capacidad para reconocer patrones complejos en este tipo de secuencias, así como su gran variedad de parámetros de ajuste, concluyendo que estas características los hacen especialmente aptos para este tipo de problemas. No obstante, también destacaron su elevado consumo de recursos, así como su vulnerabilidad ante métodos de evasión. En [KBH15] se discute el problema de la adaptación de los algoritmos de alineamiento semi-globales al reconocimiento de anomalías, explorándose diferentes estrategias de optimización de rendimiento.

Como alternativa, Oka et al. [OOAK04], propone el análisis de las acciones de los usuarios, sin considerar únicamente los eventos producidos de manera adyacente. Para ello proponen su correlación mediante matrices ECM (del inglés *Eigen Co-ocurrence Matrix*), estableciendo su relación en base a intervalos de secuencias de información. Jian et al. [JST+07] introducen la combinación de n-grams para recorrer las secuencias de llamadas al sistema con estrategias de decisión mediante podas (del inglés *decision stump*), derivada de los árboles de decisión de un nivel. De este modo se logra que el proceso de detección sea mucho más visible para los operadores. Geng et al. [GOKO10] también implementan n-gram con este fin, pero complementándolo con modelado de secuencias basado en gramáticas STF-IDF. En [HS11] se introduce el uso de modelos ocultos de Markov basados en perfiles o PHMMs (del inglés *Profile Hidden Markov Models*) al problema de la detección de enmascarados. Se trata de una estrategia muy utilizada en bioinformática que en su aplicación a este caso de uso, ha demostrado superar en precisión a los sensores basados en el modelo oculto de Markov cuando se dispone únicamente de conjuntos pequeños

de muestras referencia. Sin embargo, esta mejora ha conllevado un incremento de su complejidad computacional.
Finalmente cabe destacar la aproximación de M.B. Salem et al. [SS11b] como una de las más influyentes en la actualidad. Al igual que muchos trabajos previos, su objeto de análisis son los comandos introducidos por los usuarios. Sin embargo, en esta ocasión proponen su tratamiento tras un agrupamiento y etiquetado en base a la finalidad de cada acción, como la recopilación de recursos, búsquedas o procesos de comunicaciones. Gracias a esto es posible modelar las intenciones del usuario, las cuales constituyen uno de los mayores indicadores de su verdadera identidad. Esta estrategia demostró ser capaz de ofrecer excelentes resultados en términos de precisión, pero pasando completamente por alto su fortalecimiento frente a evasión.

### 4.2.2 Observaciones finales

La revisión en profundidad de la bibliografía permite deducir que, a excepción de las aproximaciones para la detección de enmascarados basadas en biometría, el grueso de publicaciones estudiadas se centra en el estudio del comportamiento de los usuarios y su modelado teniendo en cuenta las acciones ejecutadas principalmente, por los usuarios legítimos del sistema. En términos generales, las mayores preocupaciones de la comunidad investigadora recaen en la mejora de su tasa de acierto, reducción del número de falsos positivos, y más recientemente, en su fortalecimiento frente a métodos de evasión basados en imitación. Esta última adquiere especial importancia tras la publicación [TC11] de J.E. Tapiador et al. en el año 2011, donde se demuestra que la mayor parte de las propuestas actuales son susceptibles a este tipo de amenazas. Dado que estos ataques son cada vez más frecuentes en los sistemas de información actuales [Cen15], cada vez es más necesario el planteamiento de estrategias defensivas capaces de hacerles frente. En vistas a esta problemática, en el capítulo 6 de esta investigación se describe en detalle una de las principales contribuciones de esta tesis, centrada en la elaboración de una estrategia detección de enmascarados robusta frente a métodos de evasión basada en el estudio de las discordancias en la actividad de los usuarios del sistema protegido [MVSOGV14, MVSOGV16].

## 4.3 Análisis de la carga útil en redes de comunicaciones

En la última década, la detección de malware en redes mediante el análisis estadístico de la carga útil del tráfico en buscan de anomalías, se ha convertido en una medida esencial para la identificación de nuevos especímenes de malware. En consecuencia, actualmente existe una gran variedad de propuestas para la detección de malware en redes basadas en este paradigma, tal y como es recopilado en [BBK14]. Según esta publicación, su modus operandi habitualmente se centra en el modelado de las actividades legítimas observadas en el entorno monitorizado; cuando la carga útil del tráfico analizado difiere representativamente de ellas se considera anómalo, reportándose la posible amenaza. Sin embargo, su despliegue ha sido objeto de controversia por parte de la comunidad

investigadora, abriéndose el debate sobre sus posibles consecuencias al operar sobre redes actuales. De entre las diferentes publicaciones que han establecido las bases para el desarrollo de este tipo de aproximaciones, cabe destacar la propuesta PAYL [WS04, WCS05], la cual inspiró una de las familias de detectores de malware en redes más relevantes de la última década, incluyendo propuestas como POSEIDON [BEHZ06], AnPDPP [TKBK09], ANAGRAM [WPS06], McPAD [PAF+09] o RePIDS [JTH+13]. Pero a pesar de su eficacia, este tipo de herramientas heredaron los inconvenientes propios de la detección basada en anomalías, como la emisión de tasas altas de falsos positivos, un elevado consumo de recursos de cómputo, o dificultad de modelado en entornos heterogéneos. Esto hace que con frecuencia, los resultados obtenidos a partir de conjuntos de evaluación estandarizados por la comunidad investigadora, no presenten coherencia con los obtenidos al operar sobre entornos de monitorización reales, tal y como es discutido en [HSB+12, VTN13]. Asimismo, sus estrategias de modelado a menudo son susceptibles a ciertos tipos de métodos de evasión, como por ejemplo ataques basados en la imitación de la actividad legítima de la red [POTPL14]. Dado el interés que suscita esta área de estudio en la comunidad que investiga la detección de intrusiones basada en el reconocimiento de anomalías, así como la importancia que ha adquirido la aplicación de este paradigma en la mitigación de los problemas previamente descritos, esta sección revisa la bibliografía relacionada con la identificación de malware en redes basada en el estudio de su carga útil. Con este fin se hace hincapié en los sensores de la familia PAYL, identificándose aquellos aspectos clave a tener en cuenta de cara a su despliegue en las redes de comunicaciones actuales.

### 4.3.1 Trabajos Relacionados

En la actualidad existe una gran cantidad de propuestas para la identificación de intrusos en redes por medio del reconocimiento de anomalías. Esto ha impulsado que muchos autores hayan tratado de recopilarlas y organizarlas en taxonomías, ilustrándose en el trabajo de M.H. Bhuyan et al. [BBK14] un claro ejemplo de ello. De entre todas estas contribuciones, cabe destacar aquellas basadas en la detección de código malicioso a partir del análisis estadístico del contenido binario de la carga útil del tráfico de la red a proteger. Éstas típicamente operan por medio de una etapa de entrenamiento y otra de detección. En la primera de ellas se construye una representación estadística del modelo de uso habitual y legítimo de la red por medio de la observación de carga útil con contenido legítimo. En la etapa de detección se analizar el tráfico en busca de discordancias significativas con dicho modelo, y los paquetes con carga útil anómala son etiquetados como potencialmente intrusivos. En general, las publicaciones más representativas a esta área de investigación son variantes del sistema originalmente propuesto por K.Wang et al. conocido como PAYL [WS04, WCS05], una de las primeras aportaciones que implementó esta técnica con éxito. Desde entonces habitualmente se conoce como familia PAYL al conjunto de propuestas derivadas de este trabajo. A continuación se describen sus integrantes más relevantes.

#### 4.3.1.1 PAYL

El método PAYL original considera 256 características de la carga útil del tráfico [WS04]. Cada una de ellas representa la frecuencia de aparición en la carga útil de cada uno de los 256 posibles bytes de su contenido binario. En la construcción del modelo de uso legítimo se tiene en cuenta la media y desviación típica de cada uno de ellos. El contenido de un paquete es considerado anómalo si la distancia Mahalanobis (ver Sección 3.4.1 "Distancias de similitud en datos cuantitativos") entre sus valores en el modelo legítimo respecto de los de la observación a analizar supera ciertos umbrales predefinidos. Esta aproximación demostró comportarse con bastante precisión al reconocer amenazas reales, pero con tendencia a la emisión de una gran cantidad de falsos positivos. Para solucionar este problema, sus autores propusieron una nueva versión basada en el uso de una ventana deslizante 1-gram [GOO14] para la extracción de la información binaria [WCS05]. Si bien, alcanzaron el objetivo planteado, posteriormente se demostró su vulnerabilidad ante ataques de imitación [WPS06]. Tal y como indicaron I. Corona et al. [CGR13], estos ataques se basaban en la inserción de contenido de relleno en la carga útil con el fin de lograr un mayor parecido con el modelo de uso legítimo. Dado que el atacante es conocedor de la utilización de una ventana 1-gram para la extracción de la información, pueden distribuir el contenido malicioso entre información de relleno, y así lograr que pase desapercibido.

#### 4.3.1.2 ANAGRAM

Con el fin de mitigar esta vulnerabilidad, K. Wang et al. [WPS06] presentaron la propuesta ANAGRAM. Al igual que PAYL, ANAGRAM extrae la información de la carga útil por medio de n-grams. Sin embargo, ésta implementa filtros Bloom [RK15] para su almacenaje y consulta, lo que permite trabajar con n-gram de mayor extensión sin penalizar el consumo de memoria o su rendimiento. Nótese que los filtros Bloom son estructuras que registran de manera binaria si un n-gram ha aparecido o no en la carga útil. Durante la etapa de entrenamiento de ANAGRAM se rellenan dos filtros: uno que registra los n-gram observados en las muestras legítimas, y otro con los n-gram de las muestras maliciosas. Al analizarse la carga útil del tráfico monitorizado se establece una puntuación en base al número de coincidencias con cada filtro. Si esta puntuación supera un cierto umbral, la observación es considerada discordante y se emite una alerta. Gracias al uso de n-gram de mayor dimensión, ANAGRAM demostró ser mucho más preciso que sus predecesores, mejorando considerablemente su tasa de falsos positivos. Además, incorpora esquemas aleatorizados de distribución de n-grams, lo que dificulta la construcción de ataques de imitación y facilita su identificación. No obstante, S. Pastrana et al. [POTPL14] demostraron que sigue siendo vulnerable a muchas de sus variantes.

#### 4.3.1.3 Resto de familia PAYL

En POSEIDON [BEHZ06] se ilustra otra mejora clásica de PAYL, centrada en mejorar su eficiencia y corregir algunos de sus errores relacionados con su etapa de elaboración de modelos y falta de consistencia al analizar paquetes de diferente tamaño. A diferencia que

su predecesor, PAYL introduce el uso de redes neuronales artificiales, en particular mapas autoorganizados SOM (del inglés *Self-Organizing Maps*), en sus tareas de modelado y clasificación. Se trata de una alternativa que demostró ser mucho más rápida, pero sujeta a problemas de granularidad: si el ataque ocupa una porción muy pequeña de la carga útil, puede ser mucho más difícil de detectar. Otra propuesta orientada a mejorar la eficiencia de PAYL es AnPDPP [TKBK09]. A diferencia que sus predecesores, AnPDPP no analiza toda la carga de útil; la particiona y divide en clases, lo que permite su despliegue en redes de alta velocidad, pero penaliza su precisión. Finalmente, de entre las aproximaciones orientadas a la mejorar de su eficiencia cabe destacar la propuesta RePIDS [JTH+13]. Ésta introduce el uso de técnicas avanzadas de preprocesamiento y extracción de características de mayor relevancia en la carga útil, como análisis de componente principal o PCA (del inglés *Principal Component Analysis*). De este modo se facilita el análisis en entornos de monitorización heterogéneos y grandes datos.
Como alternativa al uso de filtros Bloom de ANAGRAM, McPAD [PAF+09] introduce una versión modificada de su método de análisis, basada en la información extraída a partir de una ventana 2-gram y el uso de Máquinas de Vector Soporte con entrenamiento semi-supervisado de una clase. En su etapa de modelado, McPAD construye representaciones de la distribución de la frecuencia de aparición de los 2-grams en diferentes espacios. A partir de la comparación de los paquetes a analizar con estos modelos es posible determinar si pertenecen al modo de uso habitual y legítimo, o son discordantes. A pesar de su probada capacidad de identificar amenazas, la pequeña distinción entre ambos modelos hace que este método sea propenso a emitir tasas altas de falsos positivos. Nótese que al igual que McPAD, otras propuestas han tratado de introducir herramientas de clasificación alternativas a la comparativa de filtros Bloom. Por ejemplo, en HMMPayl se implementan Modelos Ocultos de Markov [ATG11]. Esto demostró una gran eficacia en la detección de ciertos grupos de amenazas, como por ejemplo directivas de inyección SQL o secuencias de órdenes en sitios cruzados XSS (del inglés *Cross-site scripting*). Pero al aplicarse como detector de malware de propósito general también estaba sujeto a la emisión de tasas altas de falsos positivos.

### 4.3.2 Observaciones finales

En la actualidad existe una gran controversia acerca del uso del análisis estadístico de la carga útil del tráfico para el reconocimiento de anomalías que indiquen la presencia de malware. En algunos estudios, como por ejemplo [HSB+12], muchos de estos sensores clásicos son desplegados y evaluados en entornos de monitorización actuales. Los resultados obtenidos demostraron que la precisión observada al analizar colecciones estandarizadas a nivel funcional entre la comunidad investigadora, no eran escalables a casos de uso reales. Cabe destacar que tan sólo ANAGRAM conservó gran parte de su fiabilidad, pero experimentando un importante incremento de su tasa de falsos positivos. Por otro lado, algunos autores defienden abiertamente esta metodología, resaltando su gran utilidad al ser adaptada a casos de uso mucho más concretos. Éste es el caso de A. Oza et al. [ORLS14], donde el análisis estadístico del contenido binario de la carga útil permite identificar ciertas amenazas en el tráfico HTTP. Su experimentación demostró

una gran precisión en ambos escenarios, probando que su despliegue es factible incluso en escenarios de monitorización actuales. Tras el estudio en profundidad de la bibliografía es fácil identificar la necesidad de plantear soluciones fáciles de adaptar a la heterogeneidad y el volumen de información inherentes a estos nuevos contextos. Asimismo, se requiere de estrategias robustas frente a métodos de evasión, capaces de abordar el problema de la ofuscación del malware. Por lo tanto, la adaptación de técnicas de reconocimiento de anomalías al análisis de la carga útil del tráfico de redes de comunicaciones plantea un importante desafío a la comunidad investigadora. Con el fin de contribuir a su desarrollo, en el capítulo 7 de esta investigación se describe en detalle una de las principales contribuciones de esta tesis, centrada en la elaboración de una estrategia de detección de malware de fácil configuración e interoperabilidad [GVSOMV15, GVMVSO17].

## 4.4 Gestión de alertas

En situaciones normales, el IDS tiende a reportar una enorme cantidad de incidencias en intervalos cortos de tiempo. A la vista de un operador humano, el análisis de estas alertas resulta poco viable si no se dispone de mecanismos que permitan su gestión y clasificación. Por otro lado, cuando el proceso de respuesta es automatizado, el exceso de alertas puede afectar drásticamente a la efectividad del análisis realizado, así como la calidad del servicio del sistema protegido. La gestión de estas incidencias también desempeña un papel primordial en IDS con arquitecturas distribuidas cooperativas y jerárquicas, donde la información extraída por los diferentes sensores del sistema debe ser compartida y tenida en cuenta en los procesos de análisis. Debido a esto, los IDS habitualmente deben integrar estrategias de gestión de alertas. A continuación se profundiza en estos métodos, haciendo hincapié en el proceso de tratamiento de alertas y la evolución de sus diferentes aproximaciones.

### 4.4.1 El proceso de tratamiento de alertas

Los sistemas de correlación de alertas tienen como objetivo facilitar la gestión de las alertas emitidas por los distintos sensores, permitiendo su agrupamiento y aportando información adicional acerca de los eventos que las generan. Además, la tendencia al despliegue de sistemas distribuidos y su jerarquización en diferentes módulos de preprocesamiento especializados en la detección de determinadas amenazas, potencian que su uso sea necesario para complementar la labor del IDS. Por estos motivos, es frecuente la integración de este tipo de soluciones en sus propias etapas de diseño. Los sistemas de correlación de alertas se componen de diferentes módulos, destacando la presencia de etapas de normalización, verificación, correlación y agregación [ZYZ08]. Además de estas tareas, H. Hubballi et al. [HS14] destacaron la necesidad de incorporar mecanismos de reconstrucción de escenarios de ataques, fusión de alertas, priorización y generación de informes. En la Figura 4.1 se ilustra un ejemplo de sistema de correlación de alertas. En ella las alertas emitidas por diversos IDS son normalizadas en un formato único. Una vez concluido este proceso, son tratadas en las diferentes etapas del gestor, y

los resultados son transformados en informes para el operador. Nótese que el orden en que se efectúan dichas etapas depende de las necesidades de diseño del sistema y de en qué medida se pretende complementar al sensor. Cada una de estas etapas de procesamiento de información es descrita a continuación:

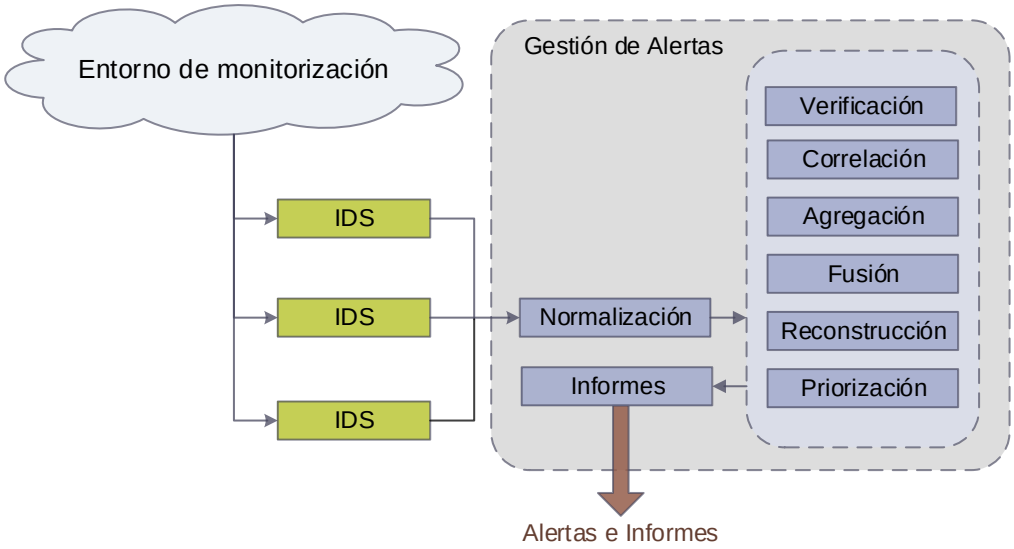



Figura 4.1: Ejemplo de sistema de gestión de incidencias.

- *Normalización.* Esta etapa tiene como objetivo la conversión del formato de las alertas emitidas por los distintos sensores del IDS a un formato único, que es el que será considerado en los niveles superiores de tratamiento de información. Existen diferentes formatos para representar informes de IDS normalizados, siendo el estándar IDMEF [DCF07] el más utilizado, ya que, además de unificar su representación, soluciona el problema de la sincronización de los tiempos de emisión de alertas, gracias a su compatibilidad con el protocolo NTP (del inglés *Network Time Protocol*).

- *Verificación.* La etapa de verificación de alertas determina si un ataque puede afectar al sistema protegido, y en tal caso, evalúa el daño que puede llegar a causar. Esta tarea está directamente vinculada con la estrategia de gestión de riesgos del sistema (ver Sección 2.2.2 “Gestión de la seguridad”), así como a su definición de amenazas, activos y su valoración [SSABC16].

- *Correlación.* En la correlación de alertas se filtran las incidencias redundantes y se identifican las posibles relaciones que puedan presentar entre sí. Este proceso es el núcleo del sistema de gestión de alertas, lo que hace que estas herramientas sean habitualmente conocidas como sistemas de correlación de alertas. Dada su relevancia, en la siguiente sección se revisa en detalle la evolución de las principales líneas de investigación en este campo.

- *Agregación y fusión.* Los procesos de agregación y fusión aprovechan la relación entre incidencias identificadas en su correlación, para fusionarlas y compactarlas en base a sus elementos comunes.

- *Reconstrucción del escenario de ataque.* Esta etapa asocia los conjuntos de alertas agregadas con los riesgos y amenazas determinados por la estrategia de gestión de riesgos del sistema (ver Sección 2.2.2 "Gestión de la seguridad")

- *Priorización.* La priorización de alertas optimiza las tareas de gestión, estableciendo en qué momento debe procesarse cada incidencia, y facilitando la decisión de las medidas de mitigación.

- *Generación de informes.* Una vez identificado un riesgo y reconstruido el escenario en el que ha sido detectado, el componente de generación de informes comunica al operador los sucesos descubiertos. Es frecuente que este tipo de documentación se ajuste a algún tipo de estandarización, como por ejemplo IDMEF [DCF07].

### 4.4.2 Correlación de alertas

Recientemente se han publicado diferentes estados del arte relacionados con la gestión de las alertas emitidas por sistemas de detección de intrusiones. Algunos de ellos abarcan un enfoque general, como es el caso de [SMFDV13, MAJ13]. Sin embargo, otros hacen hincapié en aspectos más específicos, como en la colaboración entre sensores [EO11], o la reducción de su tasa de falsos positivos. En [HS14] se repasan las principales técnicas aplicadas para su preprocesamiento en redes. A lo largo de la bibliografía, los sistemas de correlación de alertas han sido agrupados de manera muy similar a la propuesta planteada por S. Salah et al. [SMFDV13], donde su eje de clasificación está determinado por los métodos de correlación. En base a este criterio se distinguen tres grandes grupos: estrategias basadas en similitud, métodos secuenciales o casos. Éstos son descritos a continuación..

#### 4.4.2.1 Correlación basada en similitud

Este conjunto de aproximaciones habitualmente se centra en la agrupación y reducción de las alertas emitidas. Para ello se tienen en cuenta diferentes atributos o características, que abarcan desde las direcciones IP y puertos relacionados con la intrusión, hasta la frecuencia de aparición de ciertos patrones en su contenido. En [TFPC10] se observa un claro ejemplo de ello, donde además de estos atributos, son considerados otros rasgos, como su protocolo o prioridad de tratamiento. M. Cam et al. [CMP13] aplicaron características parecidas al tratamiento de eventos de redes móviles, adoptando atributos específicos, como banderas o la duración de las comunicaciones. La correlación basada en similitud a menudo también tiene en consideración las relaciones espacio-temporales de las incidencias. Un ejemplo de ello se encuentra [AZM+12], donde a partir de estas características es posible inferir el escenario del ataque. Otras propuestas toman como referencia atributos propios del sistema de detección. Éste es el caso de la aproximación de T. Chen et al. [CZJK14], donde

la correlación se realiza a partir de una versión comprimida del conjunto de muestras que ha participado en el entrenamiento del sensor.

#### 4.4.2.2 Correlación basada en similitud

El análisis secuencial de alertas tiene en cuenta las relaciones de causalidad de los incidentes, y se basa en el estudio de sus precondiciones y postcondiciones. En términos generales se define como precondiciones al conjunto de requisitos necesarios para desencadenar una amenaza; por el contrario, las postcondiciones son las consecuencias de su ejecución. En [ZSY10] se muestra un claro ejemplo de esta metodología, donde mediante el enlace de prerrequisitos y postcondiciones de diferentes alertas, es posible la construcción de escenarios de ataques. En [RAA15] A.A. Ramaki et al. proponen llevarlo a cabo mediante la elaboración de matrices de correlación causal. Otra representación común son los grafos. En [AJPS11] se profundiza en esta tendencia, y se propone la aplicación de grafos de incidencias generalizados basados en las relaciones de dependencia entre alertas. La construcción del escenario del ataque a menudo involucra la aplicación de diferentes estrategias, como sucede en [AMZ08] mediante el uso de gramáticas, o en [FAK11, AAB12] con algoritmos de minería de datos; el primero de ellos se apoya en modelos de Markov, y el segundo en la lógica difusa.

#### 4.4.2.3 Correlación basada en casos

Los métodos basados en casos se fundamentan en bases del conocimiento, denominadas en este contexto bases de casos. Por lo tanto, dependen de algoritmos encargados de reconocer los patrones específicos de comportamiento representados en ellas. Un ejemplo de su aplicación se observa en [EO13], donde los escenarios de ataques que integran la base de casos son construidos a partir de agrupaciones de incidencias. La propuesta infiere futuras amenazas, lo que permite actualizar la base del conocimiento con nuevas intrusiones. En [AJA11] se introduce una aproximación similar, centrada en la actualización automática de grafos de dependencias. Con el fin de priorizar su tratamiento, R. Shittu et al. [SHGH$^+$15] aplicaron algoritmos de agrupamiento capaces de clasificar las incidencias en función de la base de casos.

#### 4.4.2.4 Casos de uso específicos

Aunque la mayor parte de estas propuestas presentan soluciones de ámbito general, diversos trabajos han optado por plantear soluciones específicas, fácilmente adaptables a casos de uso recientes. Un ejemplo de ello es la línea de investigación relacionada con la verificación de alertas. H.W. Njogu et al. discutieron la necesidad de validar incidencias en redes de gran extensión [NJKH13]. De este modo es posible separar los avisos aislados, de aquellos que realmente forman parte de escenarios de ataques. Otro ejemplo es [PTC$^+$14], donde se propone un sistema de reputaciones para la verificación y priorización de reportes en sistemas colaborativos. Adicionalmente, la necesidad de desplegar los sistemas de correlación convencionales en entornos de monitorización específicos ha derivado en publicaciones que advierten de la necesidad de su adaptación. Por ejemplo, en [RED12]

se muestra la diferencia entre aplicar diferentes esquemas de correlación, en [MDJ12] se demuestran las limitaciones de los sistemas de correlación de ámbito general al operar sobre tarros de miel y en [SSB13] se propone su adaptación a la computación en malla (del inglés *grid computing*).

### 4.4.3 Observaciones finales

La gestión de las alertas emitidas por los IDS plantea un interesante escenario, donde la información a analizar no procede directamente del entorno de monitorización, sino de los informes recibidos de perímetros externos de seguridad. Se trata de un área de investigación centrada en la complementación de los sensores y facilitar la tarea de toma de decisiones relacionada con el despliegue de contramedidas. Tras finalizar el desarrollo del sistema de detección de intrusiones APAP (ver Capítulo 6), la correlación de sus alertar parecía una tarea necesaria, y capaz de mejorar considerablemente su adaptación al medio. Sin embargo, y a pesar de la extensa bibliografía revisada, no se hallaron propuestas centradas en el manejo de este tipo de informes que compartieran el objeto de análisis de nuestro sensor, es decir, la carga útil del tráfico. Esto planteaba la necesidad de desarrollar propuestas capaces de llevar a cabo dicha acción, de este modo complementando en mayor profundidad las ventajas ofrecidas por los esquemas de correlación actuales. Por lo tanto, la correlación de alertas en función de las características de las anomalías identificadas resulta ser una prometedora línea de investigación. Nótese que la complementación de APAP ha conllevado el análisis de las anomalías detectadas en la red protegida desde un nivel de procesamiento de información superior, facilitando la reducción de su tasa de falsos positivos [MVSOGV15b] y el manejo de los reportes enviados al operador [MVSOGV15c]. Con este propósito en el Capítulo 6 se introduce una estrategia de correlación de incidencias adaptada a este tipo de sensores. En ella son considerados dos niveles de clasificación: el riesgo de que las amenazas sean reales y su naturaleza. Esto es logrado a partir de una arquitectura de dos niveles, donde el primero analiza la información recibida a nivel de paquete y el segundo a nivel de traza. El estudio de paquetes es más eficiente y permite detectar ciertos métodos de evasión, y el segundo permite la reconstrucción del escenario del ataque y ofrece una visión general de las amenazas detectadas.

## 4.5 Mitigación de ataques de denegación de servicio

Los ataques de denegación de servicio tienen por objetivo el agotamiento de cómputo de los sistemas de información y redes de comunicaciones. Cuando se originan en múltiples fuentes, habitualmente se denominan ataques de denegación de servicio distribuidos o DDoS. En los últimos años el número de incidentes relacionados con estas amenazas ha crecido de manera alarmante, tal y como advierten las principales organizaciones para la seguridad de la información. Según la Agencia Europea de Seguridad de las Redes y de la Información ENISA (del inglés *European Union Agency for Network and Information Security*), tan solo entre los años 2013 y 2014 se experimentó un aumento del 70%, el cual ha continuado hasta el día de hoy [ENI15]. Además, estos ataques

suponen una amenaza también aprovechada con el fin de que otros tipos de intrusiones alcancen sus objetivos, entre las que se incluyen estrategias de evasión relacionadas con la propagación del malware, la ocultación de transferencias fraudulentas de dinero [Sym16], o el comprometer sistemas de anonimato como Tor o Freenet [JTJS14]. A este crecimiento se le atribuyen diferentes causas: en primer lugar, los ataques DDoS habitualmente son ejecutados desde sistemas previamente infectados, los cuales pueden llegar a formar parte de redes de terminales comprometidos a las que popularmente se conoce como *botnets*. En los últimos años éstas han sido adaptadas para resistir los métodos de mitigación convencionales y gestionar mayores cantidades de nodos, lo que hace que los atacantes de DDoS dispongan de una mayor cantidad de puntos de origen [Sop15]. Otro motivo del crecimiento de estas amenazas es que los atacantes son cada vez más capaces de aprovechar elementos intermedios de red para amplificar el impacto de la intrusión, mejorando considerablemente su capacidad de dañar el sistema víctima. Para lograrlo es frecuente la explotación de vulnerabilidades en la implementación de protocolos de red, como DNS, NTP o SNMP. Además, y tal y como advierte Europol [Eur16], el aumento de los ataques DDoS se relaciona directamente con el incremento de la participación de las organizaciones criminales en ciberdelitos, y con el hecho de que el alquiler de botnets para la ejecución de estos ataques sea muy rentable como servicio CaaS. Finalmente, los atacantes con poca formación disponen de una importante variedad de herramientas de fácil configuración que agilizan la ejecución de estas amenazas. El mercado negro además ofrece soporte técnico, guiando a sus nuevos usuarios hasta alcanzar sus objetivos.
Los métodos de DDoS más utilizados se basan en inundación [ENI15], por lo que son el principal objeto de estudio de esta sección. El *modus operandi* de los ataques de inundación consiste en la generación de un enorme volumen de tráfico con el fin de saturar los sistemas víctima que deben procesarla [ZJT13]. Por lo tanto, es un método barato y fácil de implementar, haciéndolo especialmente rentable y popular. Dada su gran impacto en las redes de comunicación actuales, se han publicado diferentes propuestas orientadas a su detección y mitigación, las cuales a menudo no son capaces de satisfacer todos los requisitos necesarios para que sean efectivas en casos de uso reales. La detección de anomalías está presente en la mayor parte de ellas, por lo que su desarrollo conlleva su adaptación a estos nuevos escenarios. A continuación se describen las principales aproximaciones a este problema.

### 4.5.1 DDoS basada en inundación: ataques y contramedidas

Según W.W. Wei et al. [WCXJ13] existen dos tipos de inyección de tráfico capaces de comprometer un sistema o red de comunicaciones por inundación. El primero se basa en la continua y constante generación de grandes volúmenes de tráfico, a lo que se conoce como inundación de tasa alta (del inglés *high rate flooding*). Este método es habitualmente muy visible y capaz de desbordar con facilidad la capacidad de cómputo de la víctima. Por otro lado, la víctima puede llegar a ser comprometida por estrategias menos ruidosas, como la explotación de vulnerabilidades en los protocolos de comunicaciones. Los ataques de inundación que las aprovechas reciben el nombre de inundación de tasa baja (del inglés, *low rate flooding*), hallándose un claro ejemplo de ellos en los ataques que siguen patrones

on/off dirigidos contra el protocolo TCP [TLHC14]. Tanto en tasa alta como en tasa baja, los ataques de inundación pueden llegar a la víctima de manera directa o reflejada [BBK15, AKK+13]. Existen diferentes maneras de aprovechar la capacidad de agotamiento de recursos que facilitan estos métodos, como por ejemplo por medio de la inundación de conexiones (del inglés *link-flooding*) [WLJW16] o la conexión dirigida a conexiones concretas (del inglés *target link-flooding*) [GKLD16]. Éstas se basan en agotar el ancho de banda de la víctima concentrando la inyección de tráfico malicioso en conexiones o regiones de mayor sensibilidad. Dado que son capaces de reutilizar tráfico legítimo con dicho fin, y que los ataques dirigidos son menos ruidosos, son amenazas mucho más difíciles de detectar que los ataques DDoS convencionales. La mayor parte de los esfuerzos de la comunidad investigadora para paliar estas amenazas asumen estos comportamientos, y a partir de ellos plantean estrategias de detección, mitigación e identificación de su origen, tal y como se describe a continuación.

La detección de ataques de DDoS normalmente es necesaria para poder desencadenar las otras dos acciones defensivas. Se basa en el estudio del tráfico que fluye a lo largo del entorno protegido en busca de patrones conocidos o comportamientos discordantes. Con este fin se han implementado diferentes métodos de análisis, como modelos probabilistas basados en Markov [SLKK13], algoritmos genéticos [LKLP12], teoría del caos [CMW13], análisis estadístico CUSUM con transformadas wavelet [CGPP12], métodos de análisis forense basados en visualización [CFGH10], lógica difusa [KS13], o el estudio de las variaciones en la entropía del tráfico monitorizado [TLHC14, BBK15]. En aproximaciones como [ZJW+14] se discute el problema de la similitud que existe entre la denegación de servicio y ciertos eventos de naturaleza legítima, como el acceso de muchos usuarios verdaderos a un recurso en periodos de tiempo concretos (del inglés *flash crowds*). Las características no estacionarias de las redes actuales, su falta de homogeneidad que a menudo lleva a la emisión de falsos positivos tras la observación de discordancias de naturaleza legítima, o la frecuente confusión de eventos como *flash crowds* al analizar el tráfico, son algunas de las principales preocupaciones en las nuevas contribuciones es este campo. Por otro lado, las tareas de mitigación se centran en reducir de manera parcial o total en impacto de los ataques. La bibliografía reúne diferentes propuestas para lograrlo, como el despliegue de tarros de miel [HWS+13], puzles para distinguir usuarios no humanos [ZYB+14], redirección e incremento del ancho de banda en las regiones afectadas [KVF+12], filtrado o adopción de protocolos de seguridad como IPsec [Eur16]. Nótese que el conjunto de técnicas de mitigación contiene aproximaciones también relacionadas con su prevención. Sin embargo, cuando presentan esta última función, no dependen de la identificación previa de la amenaza, lo que hace que sean mucho más eficaces, a costa de causar un mayor impacto sobre el entorno monitorizado [LLZZ13].

Finalmente, la identificación del origen del ataque persigue el descubrimiento de los nodos comprometidos que han participado en la intrusión. De manera ideal, su objetivo es descubrir al ciberdelincuente. Pero dadas las dificultades administrativas que esto conlleva (consulta a servidores proxy de diferentes proveedores ISP, legislación en favor de la privacidad de los usuarios, etc.) y los avances en la ocultación del rastro, se ha convertido en una meta muy difícil de lograr. En consecuencia, la mayor parte de las publicaciones de

la bibliografía se han centrado en llegar lo más cerca posible del atacante, lo que permite ampliar el despliegue de contramedidas a una mayor cantidad de las regiones afectadas. De entre estas aproximaciones cabe destacar aquellas basadas en el marcado de paquetes. N.M. Alenezi et al. [AR14] discutieron este enfoque en profundidad y revisaron una gran cantidad de publicaciones relacionadas, proponiendo además un nuevo esquema de rastreo. En [YBV15] se propone un método de rastreo pasivo de direcciones IP o PIT (del inglés *Passive IP Traceback*) basado en el análisis de los avisos de error del protocolo ICMP. Finalmente, cabe destacar la relevancia de las características de la topología de la red en la eficacia de las estrategias de seguimiento, tal y como se discute en [JL14].

### 4.5.2 Observaciones finales

Tras la revisión del estado del arte relacionado con la defensa frente a los ataques DDoS, cabe destacar la relevancia del reconocimiento de anomalías en las tareas relacionadas con su detección, y la insistencia de la comunidad investigadora en mejorar su tasa de aciertos, tasa de falsos positivos, consumo de recursos y rendimiento en tiempo real. Esto ha dado lugar a una gran cantidad de publicaciones, repartidas entre acciones de detección, mitigación, prevención e identificación del origen de las amenazas. Sin embargo, es importante resaltar que en la bibliografía apenas se recogen aproximaciones que consideren los avances y las nuevas tendencias en el desarrollo de redes de comunicación. Es de esperar que tomando el ejemplo las redes de quinta generación o 5G, éstas avancen hacia una autogestión inteligente [PSS16]. La implementación de tecnologías como redes definidas por software SDN (del inglés *Software-Defined Networking*), funciones de red virtualizadas NFV (del inglés *Network-Function Virtualization*), inteligencia artificial o computación en la nube facilitan el diseño de redes auto-organizativas SON (del inglés *Self-Organizing Networks*). Éstas deberían incentivar la aparición de propuestas capaces de brindar respuestas activas, pero también proactivas, y analizar el estado de la red de una manera mucho más cognitiva.
Con el fin de contribuir a su desarrollo, en [MVSOGV18, MVSOGV15d] se describe una de las principales contribuciones de este documento (ver Capítulo 8). Ésta introduce el despliegue de una red de sensores distribuidos a lo largo del espacio protegido integrando un sistema inmunitario artificial o AIS (del inglés *Artificial Immune System*). A diferencia de las propuestas tradicionales, implementa métodos bioinspirados de reconocimiento de anomalías, y permiten la emulación de los mecanismos defensivos del sistema inmunitario de los seres humanos. De esta manera es posible la variación del nivel de restricción en que actúan los detectores de anomalías, aprovechar el conocimiento adquirido para mejorar su eficacia contra amenazas previamente identificadas o establecer regiones de cuarentena. Para ello se hace uso de tecnologías propias de escenarios 5G, como SDN, NFV, etc.

## 4.6 Identificación de malware en dispositivos móviles

Debido a la gran capacidad de conectividad, accesibilidad, y versatilidad de los dispositivos móviles, en los últimos años se ha experimentado un importante crecimiento de su

popularidad. En consecuencia, cada vez más usuarios se apoyan en estas tecnologías para el desempeño de actividades de especial sensibilidad, tales como el comercio electrónico, intercambio de activos o accesos a información confidencial. Esto hace de esta tecnología un objetivo muy deseado por los cibercriminales, tal y como ha advertido ENISA en su último informe anual [ENI15]. En este informe no sólo se predice un importante crecimiento de las amenazas dirigidas contra dispositivos móviles; también se alerta de su peligrosa sofisticación, lo que las hace difíciles de detectar por los esquemas de defensa actuales. Dentro de este problema, cabe destacar la importancia de la migración de los ataques convencionales a la infraestructura móvil, siendo la adaptación del software una de las prácticas más habituales. Según la Oficina Europea de Policía (Europol), detrás de esta laboriosa tarea a menudo se escoden complejos entramados de crimen organizado [Eur16]. De entre sus estrategias de propagación más frecuentes, predomina el uso de los mercados de distribución de aplicaciones oficiales. En este caso, los delincuentes ofrecen variaciones de productos originalmente legítimos, que han sido manipulados para albergar el vector de infección que permite la instalación del software malicioso. Dada la poca efectividad de los métodos defensivos ofrecidos por estos mercados, y el exceso de confianza de gran parte de los usuarios (a menudo incentivado por falta de conocimiento), los delincuentes consiguen propagar los especímenes con gran rapidez, y de manera indiscriminada. Por otro lado, estos estudios apuntan a la plataforma Android, como principal objetivo de los estos atacantes [Cis17]. Su principal motivación radica en que Android es el sistema operativo más extendido del mercado. Debido a la ineficacia de los métodos de seguridad que practican sus tiendas de aplicaciones, y el exceso de confianza de usuarios poco concienciados a la hora de dar permiso de acceso al software descargado, los atacantes son capaces de propagar software cada vez con mayor precisión y velocidad.
Con el fin de combatir esta amenaza, la comunidad investigadora ha desarrollado diferentes propuestas, el las cuales el reconocimiento de anomalías adquiere un papel de especial relevancia. G. Suarez-Gil et al. [STTPLR14] recopilaron muchos de estos trabajos y analizaron en detalle la evolución de dicho software. A partir de su estudio es posible observar las principales causas que han llevado al fracaso de gran parte de las aproximaciones actuales, siendo la limitación de recursos de cómputo una de las más problemáticas. Dado que los dispositivos móviles, y en especial, la plataforma Android suponen un emergente caso de uso de la detección de discordancias, a lo largo de la presente sección se revisan las aproximaciones más comunes para la identificación de este tipo de software, y aquellos aspectos de relevancia de cara a plantear nuevas propuestas.

### 4.6.1 Malware contra Android

El estado del arte actual acerca del software malicioso específico para la plataforma Android está directamente ligado a los avances en el software para dispositivos móviles. Sus primeros especímenes, como *Cabir* (2004) o *CommWarrior* (2005) se basaban en la explotación de vulnerabilidades en estas tecnologías con el fin de lograr su propagación por protocolos específicos de comunicaciones, como por ejemplo el servicio de mensajería multimedia o MSS (del inglés *Multimedia Messaging Service*) o Bluetooth [Apv14]. Es importante tener en cuenta que a pesar de que al compararse con especímenes actuales

resultasen poco peligrosos, ya eran capaces de causar pérdidas monetarias derivadas del envío de mensajes de pago. El éxito que alcanzaron motivó a que los atacantes dieran un paso más, llevándolos a centrar sus esfuerzos en obtener beneficio económico de la intrusión, dando lugar a especímenes como *RedBrowser* (2005) o *Yxes* (2009), con los cuales suscribían al dispositivo a servicios de pago vía SMS [DMC15]. El último de estos dos ejemplares está considerado el precursor de las *botnets* para dispositivos móviles. La adaptación de software de propósito general a la computación ubicua alcanzó un importante hito al descubrirse el espécimen *Zitmo* (2010), la versión para dispositivos móvil de la conocida *botnet* para bancos *Zeus* [Gaf13]. Para entonces Android ya ocupaba el mayor nicho del mercado, lo que resultó en la aparición de los primeros ejemplares específicos para este sistema operativo, entre ellos *Gemini* (2010), *DroidKungFu* (2011) o *Plankton* (2011). Se distribuían tanto desde la tienda de aplicaciones oficial de Android, como desde servicios de distribución de terceras partes [CADZ15]. El éxito alcanzado ha motivado a que los atacantes desarrollen nuevas maneras de sacar beneficio de su trabajo, dando pie a una gran variación de software malicioso que reúne términos como *adware*, *riskware* o *spyware*. Un buen ejemplo de ello se observa al estudiar el fenómeno del *ransomware*. Esta amenaza se caracteriza por extorsionar a sus víctimas, y a menudo incorpora la capaz de bloquear parte de la funcionalidad del dispositivo hasta que la víctima pague un rescate económico por recuperarla. El primer espécimen conocido de *ransomware* para dispositivos móviles es *FakeDetect* (2013), una familia de software malicioso que imita los avisos del software antivirus comercial y exige al usuario un rescate a cambio de recuperar el acceso a sus activos [KYZ15]. Otros ejemplos de *ransomware* son *FakeAV* (2013), *Cryptolocker* (2014), *Koler* (2014) o *Locker* (2015) [Eur14].

El software específico para Android, al igual que el resto de aplicaciones desarrolladas para este sistema operativo, es distribuido comprimido en formato APK (del inglés *Application PacKage file format*). Los archivos APK maliciosos son distribuidos a través de su tienda oficial de aplicaciones o por servicios de distribución de terceras partes; no obstante, existe otros mecanismos algo menos frecuentes, como la ingeniería social (*spear fishing*, *baiting*, etc.) o la explotación de vulnerabilidades. Una vez infectado el dispositivo, adquieren la capacidad de propagarse vía protocolos de comunicación (Bluetooth, Wi-Fi, NTC, etc.) y por diferentes canales (correo electrónico, mensajería instantánea, redes sociales, etc.). Con el fin de dificultar su distribución, las tiendas oficiales obligan a que sus productos estén firmados digitalmente y presenten su certificación antes de permitir su descarga. También incorporan algunas herramientas de detección, principalmente por medio del análisis estático y dinámico de su contenido. Pero a pesar de que la iniciativa es buena, en la práctica han demostrado ser incapaces de prevenir la distribución de este tipi de software, el cual a menudo trata de pasar desapercibido por técnicas de evasión basadas en ofuscación. En [MAC+15] se discute este problema en profundidad, y se demuestra que por medio de la mutación del motor de infección del espécimen es posible engañar a estos mecanismos.

### 4.6.2 Trabajos relacionados

Recientemente se han publicado diferentes estados del arte relacionados con la seguridad en dispositivos móviles. Algunos de ellos presentan un enfoque generalizado, como es el caso de [STTPLR14, LPMS13], otros tratan temas más concretos, siendo el sistema operativo Android uno de los más frecuentes [FBL15]. Finalmente existen recopilaciones centradas en las propias amenazas, como por ejemplo [PYY14], donde se profundiza en el problema del malware. A lo largo de la bibliografía, la limitación de los recursos de estos dispositivos ha determinado la manera en que la información monitorizada es procesada en busca de discordancias. Una primera manera de hacerlo consiste en la ejecución del análisis en los propios dispositivos. Esto tiene la ventaja de no depender de sistemas externos, o de tener acceso a la Red. Además, son menos propensos a vulnerabilidades relacionadas con la privacidad [DMC15]. Pero tal y como indicaron D. Maiorca et al. [MAC+15], también tienden a la emisión de más falsos positivos, y son especialmente sensibles contra técnicas de evasión. Esto es debido a que los métodos más precisos requieren de mayor cantidad de memoria, capacidad de procesamiento o consumo de batería. Como alternativa muchos autores han optado por delegar las tareas más complejas del análisis a servicios externos, aun asumiendo los riesgos que esto conlleva. Otro aspecto de especial relevancia en las propuestas estudiadas ha sido la decisión de las características analizadas por los sistemas de detección. Este problema no es especialmente representativo en aproximaciones basadas en el reconocimiento de firmas, como por ejemplo [ZSL13]. Sin embargo, cuando la detección se basa en anomalías, resulta crucial para garantizar el éxito de las propuestas, tal y como señalaron P García-Teodoro et al. [GTDVMFV09]. Debido a esto, es comprensible que muchas clasificaciones hayan considerado estos rasgos como principal distinción entre los diferentes métodos de análisis, hallándose en la taxonomía de M. Zhang et al. [FASW15] un claro ejemplo de ello. En base a esta publicación, las propuestas pueden dividirse en cuatro grandes grupos: análisis de características estáticas, dinámicas, mixtas o metadatos.

#### 4.6.2.1 Rasgos estáticos

El análisis basado en rasgos estáticos inspecciona las aplicaciones antes de su ejecución, en busca de contenido malicioso. Para ello se extraen distintos datos, tales como su código binario, privilegios solicitados, recursos hardware o conectividad. Por ejemplo, en [TY06] se consideran estas características para explorar distintos mercados de distribución en busca de instancias de un mismo espécimen. En [XPW+14] se estudia el código fuente de las aplicaciones del servicio de gestión de paquetes o PMS (del inglés *Package Management Service*) de Android en busca de malware capaz de escalar privilegios. En [GFKS15] se propone la detección de intrusiones en basada en la identificación de anomalías en los privilegios que solicitan. De manera alternativa, en [ASH+14] esta distinción es realizada por medio de la consulta del *AndroidManifest*, donde se observa el hardware que la aplicación solicitará. En [STTPLA14] se analiza la estructura del código en busca de similitudes entre diferentes muestras de malware. A partir de ellas es posible establecer relaciones entre especímenes de una misma cepa, y por lo tanto estudiar su evolución. En

general, el análisis estático presenta la ventaja de su sencillez en el proceso de extracción de datos, y eficiencia. Sin embargo, es un método susceptible a ser engañado por técnicas de ofuscación. Además, debido a su incapacidad de definir el comportamiento de las aplicaciones en tiempo de ejecución, a menudo no alcanza la precisión esperada.

#### 4.6.2.2 Rasgos dinámicos

Como alternativa, el análisis basado en rasgos dinámicos monitoriza el comportamiento del sistema y extrae la actividad de las aplicaciones instaladas a través de diversos identificadores. Tal y como se muestra en [FASW15], la estrategia dinámica más frecuente es el análisis de las llamadas al sistema involucradas en la ejecución de cada aplicación y el estudio de sus discordancias. Un ejemplo típico de ello se ilustra en el sistema *Crowdroid* [BZNT11], donde se considera su frecuencia de aparición. En otros casos, como [LLCT13], se analiza su relación con el sistema de planificación de hilos en ejecución. Este tipo de propuestas a menudo efectúan la extracción de la información en entornos aislados y controlados conocidos como *sandboxes*, hallándose en [SMP13] un completo estado acerca de esta metodología. El análisis dinámico también involucra el estudio de otras características. Por ejemplo, en [ZYYG14] se construyen modelos de comportamiento de aplicaciones en base a los permisos que solicitan. Asimismo, en [HNH13] se discute la eficacia de diferentes métricas basadas en el consumo de energía de los dispositivos. En general, la identificación de anomalías por medio del análisis de rasgos dinámicos es muy precisa. Sin embargo, requiere del uso de una cantidad importante de recursos, situación que puede hacer inviable su despliegue, y en su defecto, requerir de la disponibilidad de infraestructura adicional.

#### 4.6.2.3 Rasgos mixtos

Los entornos más complejos o con mayor susceptibilidad a ser perpetrados, a menudo afrontan la identificación de intrusiones mediante la combinación de ambas técnicas. A esto se le conoce como análisis basado en rasgos mixtos o híbrido. Un ejemplo de ello es [WGNF12], donde se lleva a cabo el análisis estático del *AndroidManifest* y el código de las aplicaciones, y se estudian diferentes rasgos dinámicos, como los registros de llamadas o el tráfico de red. Otro trabajo de interés es [RCEC13], donde se trata el problema de la sobrecarga causada por sistemas de análisis dinámicos puros. Para ello se realiza un estudio del código de las aplicaciones, etiquetando todas aquellas llamadas que desaten sospechas; de este modo, el análisis dinámico solo necesita monitorizar las actividades derivadas del código etiquetado, reduciendo considerablemente su consumo de recursos. En general, las propuestas híbridas compensan los beneficios y contramedidas de los métodos combinados. Esto implica ceder parte de sus ventajas, en pro de fortalecer sus puntos débiles.

#### 4.6.2.4 Metadatos

El último grupo de aproximaciones basa su análisis en el estudio de metadatos. Los metadatos fueron definidos por A. Feizollah et al. [FASW15] como la información de las aplicaciones conocida antes de su descarga, lo que involucra una gran variedad de

fuentes, tales como los requisitos establecidos por sus autores, opinión de otros usuarios, reputación o distribución geográfica. Un ejemplo de ellos se ilustra en [PXY+13], donde se analiza información procedente del mercado de distribución relacionada con los permisos solicitados por cada aplicación. Con este fin son considerados métodos de procesamiento de lenguaje natural, los cuales analizan en detalle los motivos con que los autores justifican la habilitación de cada uno de ellos. Otro ejemplo es [TFF+16], donde se tiene en cuenta una mayor variedad de información, entre la que se encuentra la valoración, precio o última modificación de la aplicación. La principal ventaja del análisis basado en metadatos es que permite identificar el malware antes de que sea descargado. Sin embargo, su éxito depende de información fácilmente manipulable por los atacantes, situación que facilita su evasión y a menudo conlleva la emisión de errores de clasificación.

#### 4.6.2.5 Observaciones finales

La revisión de la bibliografía lleva a resaltar diferentes aspectos a tener en cuenta. El más relevante es el impacto que supone la limitación de recursos en las tecnologías ubicuas. La mayor parte de las estrategias de modelado y reconocimiento de anomalías son costosas a nivel de cómputo, ya sea en términos de rendimiento o consumo de memoria. Esto impide que los algoritmos de análisis exploten todo su potencial, restringiendo la selección de sus parámetros de ajuste, y por lo tanto su precisión. En muchas ocasiones esto suele solucionarse derivando parte de su procesamiento a servicios externos. Otro aspecto interesante es la tendencia al uso del *sandbox*; los métodos de detección basados en rasgos estáticos y metadatos permiten descartar una parte importante del malware, pero los especímenes más evasivos son detectados en tiempo de ejecución. El uso de esta estrategia supone un incremento adicional en los recursos requeridos por el sistema de detección de intrusiones, lo que puede tener impacto en otras capacidades de la estrategia de detección. Con el fin de contribuir a su optimización, en [MVSMGV18] se introduce una de nuestras líneas de investigación en curso. En ella se propone un sistema de reconocimiento de malware para dispositivos móviles. Su principal objetivo es evitar que software malicioso procedente de terceras partes sea instalado en los sistemas protegidos. Para ello, las aplicaciones descargadas de los diferentes medios de distribución oficiales son evaluadas en un entorno de ejecución seguro y aislado, previo a su despliegue sobre el sistema real. El proceso de análisis involucra la construcción de secuencias a partir de las llamadas al sistema ejecutadas en los procesos de arranque de las aplicaciones, y su posterior análisis por técnicas de alineamiento. Considerar únicamente las actividades iniciales de la muestra reduce considerablemente el espacio de búsqueda del sensor, y por lo tanto su consumo de recursos. De este modo es posible la identificación del malware antes de que actúe contra el sistema protegido. Además, la aplicación de algoritmos de alineamiento de secuencias tiene en cuenta la proximidad y relación temporal de las acciones ejecutadas por la aplicación, permitiendo la mejor comprensión de su comportamiento. Ambas características han sido demostradas en los experimentos preliminares realizados, los cuales han involucrado muestras de las colecciones de dominio público Genome y Debrin [ASH+14]. No obstante, aún quedan diferentes aspectos por evaluar, siendo ésta una línea de investigación en curso.

# Capítulo 5

# Detección de enmascarados robusta a ataques de imitación

La propuesta para la detección de enmascarados introducida en este capítulo se centra en mejorar el comportamiento de los esquemas de detección convencionales basados en el análisis de las secuencias de acciones ejecutadas por los usuarios del sistema. La clasificación de la actividad monitorizada es modelada y clasificada en base a algoritmos de alineamiento de secuencias locales. Para la validación del etiquetado se incorpora la prueba estadística no paramétrica de Mann-Whitney. Esto permite el análisis de secuencias en tiempo real, y evita las restricciones habituales relacionadas con su longitud. Asimismo, se propone una estrategia para identificar técnicas de evasión basadas en mimetismo, mediante el análisis de sub-secuencias en paralelo derivadas de la secuencia de ejecución original. Sus objetivos principales son el incremento de su tasa de acierto, reducción de tasa de falsos positivos, y fortalecimiento frente a los nuevos métodos de evasión basados en imitación (ver Sección 4.2 Detección de Atacantes enmascarados"). El contenido del capítulo está organizado de la siguiente manera: en la Sección 5.1 se describen las principales características de los algoritmos de alineamiento de secuencias que han sido tenidos en consideración; en la Sección 5.2 se introduce un método para la detección de atacantes enmascarados; en la Sección 5.3 se describe cómo se ha logrado su fortalecimiento frente a técnicas de evasión basadas en imitación; en la Sección 5.4 se presenta la experimentación realizada; y en la Sección 5.4 se discuten los resultados obtenidos.

## 5.1 Alineamiento de secuencias

En bioinformática, el alineamiento de secuencias constituye una técnica para establecer el grado de similitud entre diferentes cadenas de ADN, ARN o proteínas. Las secuencias alineadas generalmente corresponden a nucleótidos o aminoácidos, y se identifican mediante símbolos de un alfabeto *sum*. Cuando el ancestro de un linaje de individuos es común, sus diferencias son consideradas mutaciones puntuales (sustituciones). A los huecos observados en ellas se les denomina *indels* (inserciones o eliminaciones). La

similitud entre secuencias habitualmente es considerada como una medida de conservación entre linajes, que habitualmente conlleva una importancia funcional y estructural de las muestras. El alineamiento de secuencias ha sido adaptado para resolver problemas muy alejados de sus propósitos originales, como es el caso de análisis de acciones y activos financieros, o la comparación entre secuencias de eventos en sistemas computacionales. Su aplicación para esta finalidad fue propuesta por primera vez por Wespi et al. [WDD99] en el año 1999, dando lugar al sistema que denominaron TEIRESIAS, el cual se centraba en el estudio de las secuencias de llamadas al sistema ejecutadas por los distintos usuarios.
Las diferentes técnicas de alineamiento de secuencias habitualmente son consideradas como generalizaciones del problema de la detección de la longitud de la sub-secuencia más larga común entre dos cadenas, conocida como LCS (del inglés *Longest Common Subsequence*), y de la solución propuesta por Wagner y Fischer [WF74]. En ella se propone que, para la obtención de la LCS de dos cadenas de símbolos, basta con eliminar símbolos y añadir huecos (*gaps*) en cada una de ellas hasta que aparezcan sub-secuencias similares, determinándose las de mayor dimensión. En la Figura 5.1 se muestra el ejemplo de evolución de un genotipo que parte de los cromosomas del ancestro original del linaje. Las variaciones genéticas son constituidas por elementos vacíos (*indels*). La aplicación de técnicas de alineamiento permite la identificación de las estructuras con mayor importancia funcional o estructural, como es el caso de la cadena "ACT". Tal y como puede observarse, dicha estructura se conserva en las distintas generaciones que descienden del mismo ancestro.

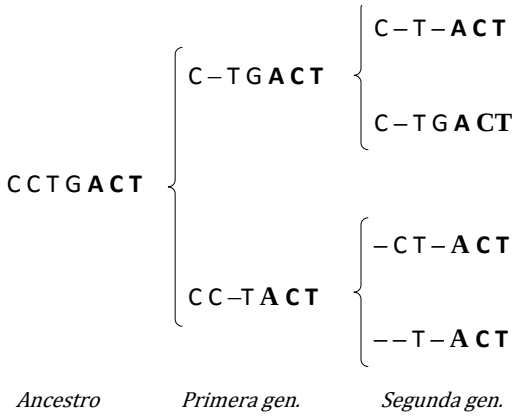



Figura 5.1: Alineamiento de secuencias sobre un linaje.

En la Figura 5.2 se ilustra un ejemplo de inserción de huecos (*gaps*) para maximizar el número de coincidencias entre las secuencias comparadas. Las secuencias originales únicamente presentan 3 coincidencias, pero al aplicar técnicas de alineamiento se producen 25 coincidencias. En las secuencias originales, la sub-secuencia común de mayor longitud es "TC", con una dimensión de 2. Pero tras su alineamiento, la sub-secuencia común de mayor longitud es "CGATGCTAGCGT", de dimensión 12; a partir de ella será posible determinar el grado de similitud de las secuencias, de una manera mucho más precisa. Existen tres esquemas fundamentales de alineamiento de secuencias de aminoácidos: alineamiento global, local y semi-global. A continuación se describe cada uno de ellos.

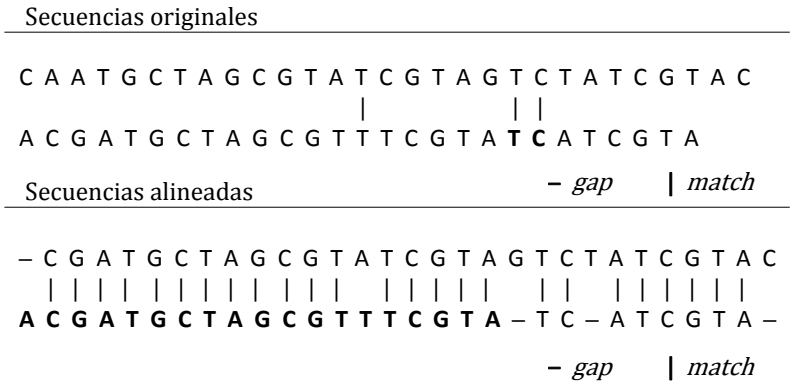



Figura 5.2: Ejemplo de alineamiento de secuencias.

- *Alineamiento global.* Esta aproximación se basa en maximizar la longitud de las sub-secuencias comunes entre las cadenas considerando su extensión total. El método más utilizado es el de Needleman-Wunsch [NW70], el cual aprovecha las características de la programación dinámica. Entre sus características cabe destacar la inserción de *gaps* para igualar la longitud de las secuencias, lo que hace que no sea la mejor opción al tratar con cadenas de dimensión significativamente dispar. Esta aproximación se utiliza con frecuencia en la comparación de varias secuencias.

- *Alineamiento local.* El alineamiento local se basa en maximizar la longitud de las sub-secuencias comunes entre las cadenas, considerando las diferentes sub-cadenas de cada una de ellas. Esta característica lo hace especialmente apto para la comparación de secuencias de longitudes muy dispares, ya que tiene en consideración las estructuras más representativas de las muestras. Generalmente se aplica mediante el algoritmo de Smith-Waterman [SW81], en cual se combinan las puntuaciones parciales de los diferentes alineamientos globales

- *Alineamiento semi-global.* Los métodos de alineamiento semi-globales combinan las dos técnicas anteriores. Habitualmente se implementan como variaciones del algoritmo de Smith-Waterman, que, a diferencia de su versión original, no aplican penalizaciones al principio y al final de las secuencias. Típicamente se utilizan cuando se requiere considerar la similitud de una secuencia completa, respecto a sub-secuencias de otra secuencia diferente. Dado que este método se comporta especialmente bien cuando una cadena es mucho más larga que la otra, Coull et al. [CBSB03, CS08] introdujeron el uso de técnicas bioinformáticas de alineamiento de secuencias en la detección de enmascarados por medio de la adaptación de esta estrategia.

## 5.2 Detección de atacantes enmascarados

El sistema de detección propuesto analiza las actividades llevadas a cabo por los usuarios legítimos del sistema protegido en busca de indicios de intrusiones. Con el fin de facilitar su comprensión y desarrollo, se han establecido las siguientes consideraciones iniciales:

- Los usuarios interactúan con el sistema protegido por medio de la ejecución de secuencias de acciones. Se asume que estas actividades pueden ser analizadas en diferentes niveles de abstracción: desde instrucciones en código máquina hasta comandos y llamadas al sistema. Esta aproximación debe de ser capaz de tratar con acciones en cualquier nivel.
- Tanto los usuarios legítimos como los atacantes enmascarados pueden iniciar sesión en el sistema. Se asume que ambos podrían incluso operar al mismo tiempo.
- El sistema de detección debe de ser capaz de identificar enmascarados. Esta tarea se llevará a cabo mediante la observación del menor número posible de acciones, pero sin penalizar su rendimiento, de tal manera que se facilite su operatividad en tiempo real.
- Cada vez que un usuario ejecute una nueva acción, el sistema debe de ser capaz de revisar su comportamiento y determinar si es "legítimo", "anómalo" o "desconocido".
- La etiqueta "desconocido" se aplica a los usuarios acerca de los cuáles no se dispone de suficiente información como para llegar a una conclusión fiable. Ante esta falta de datos, el sistema de detección espera a monitorizar nuevas acciones hasta que se capaz de determinar su naturaleza.

Con el fin de satisfacer los requisitos derivados de estas consideraciones, el sistema propuesto gestiona la información monitorizada en dos niveles de procesamiento de datos: análisis de secuencias y refinamiento del etiquetado (ver Figura 5.3). En la etapa de análisis, el sistema compara las secuencias de acciones ejecutadas por el usuario con los modelos de uso legítimo y malicioso del sistema. Esto resulta en un primer etiquetado de sus actividades del tipo "legítimo" o "malicioso", despendiendo del modelo con el que la secuencia a analizar presente mayor similitud. En la etapa de refinamiento se identifica el nivel de confianza de estos etiquetados. Si presenta valores altos, la clasificación es emitida a los operadores. Cuando esta etiqueta es del tipo "malicioso", además se emite una alerta. Nótese que cuando el etiquetado no ofrece la suficiente confianza, la secuencia permanece en estado "desconocido" hasta que el usuario ejecute nuevas acciones. A continuación se profundiza en cada uno de estos componentes.

### 5.2.1 Análisis de secuencias

En esta subsección se revisan las principales características de la etapa de análisis de nuestra propuesta, haciendo hincapié en los modelos de uso que construye, la representación de las puntuaciones resultantes del alineamiento y el etiquetado provisional de secuencies, y el sistema de asignación de puntuaciones.

#### 5.2.1.1 Modelos de uso

El sistema de detección propuesto considera dos modelos de uso del entorno protegido: legítimo y malicioso. El primero es construido a partir de actividades habituales y legítimas

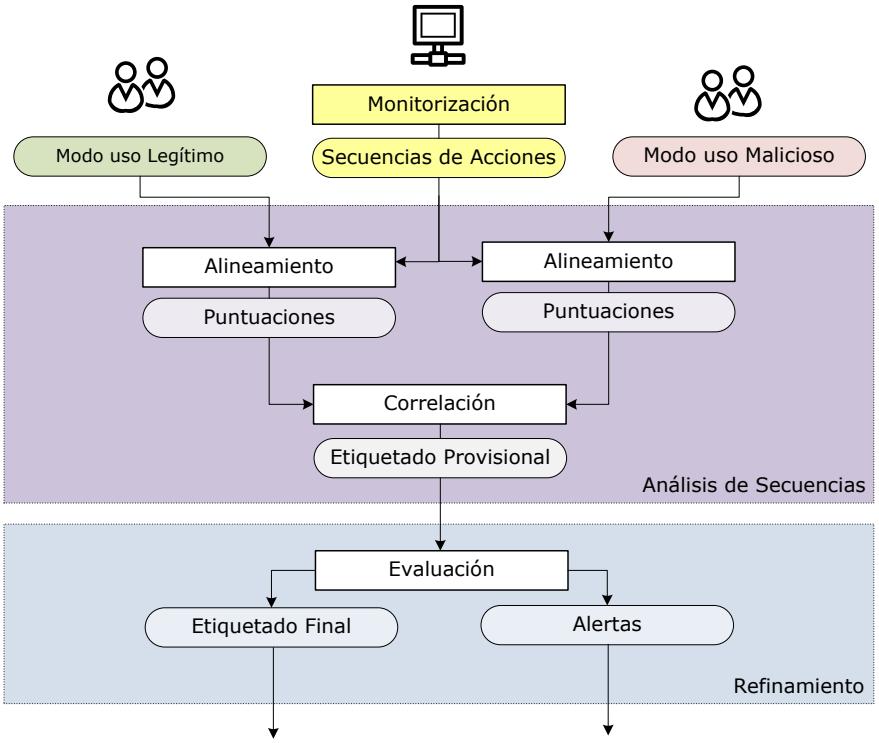



Figura 5.3: Esquema de detección y validación de resultados.

ejecutadas por usuarios reales. Por lo tanto, y con el fin de ofrecer completitud, incorpora acciones capturadas regularmente en diferentes intervalos de tiempo. Esto además reduce el problema del concepto deriva (del inglés *concept drift*) en la detección de enmascarados, el cual es estudiado en profundidad en [Sen14] (ver Sección 3.5 "Anomalías en entornos de monitorización no-estacionarios"). Además, si diferentes usuarios realizan actividades similares, es posible utilizar esquemas supervisados parecidos a los que se proponen en [WS03] para reducir el impacto del entrenamiento.

Por otro lado, el modelo de uso malicioso se construye a partir de secuencias de acciones ejecutadas por atacantes. Puede contener actividades directamente intrusivas practicadas por los atacantes, o intentos de imitación del comportamiento malicioso. La eficacia de los modelos construidos considerando estas últimas, parte de la habitual falta de conocimiento que tiene el atacante del sistema comprometido. Tal y como señalaron M.B. Salem et al. [SS11b], éstas tienen tendencia a presentar actividades de exploración, como búsquedas o movimientos entre directorios. En su propuesta, trataron de aproximar modelos de uso malicioso construidos a partir de secuencias de acciones derivadas de juegos de "capturar la bandera", donde voluntarios intentaron localizar un activo oculto en un sistema que desconocían. La experimentación realizada demostró que el conjunto de muestras construido con este método era prácticamente igual de efectivo que aquellos construidos a partir de amenazas reales, lo que demostró la validez de sus premisas.

#### 5.2.1.2 Representación de puntuaciones y etiquetado provisional

Dada la secuencia de acciones a analizar $S = S_1 \dots S_n$, y las colecciones $L = L_1 \dots L_r$ y $M = M_1 \dots M_S$, tales que $0 < n, r, s$, $L$ es el modelo de uso legítimo del sistema y $M$ el modelo de uso malicioso. En la etapa de análisis de secuencias, el sistema propuesto emite dos conjuntos de puntuaciones:

- $PL = PL_1 \dots PL_r$, donde cada puntuación parcial $PL_i$ tal que $1 \leq i \leq r$, es el resultado de alinear la secuencia legítima $L_i$ con la secuencia a analizar $S_i$.
- $PM = PM \dots PM_S$, donde cada puntuación parcial $PM_i$ tal que $1 \leq i \leq s$, es el resultado de alinear la secuencia maliciosa $M_i$ con la secuencia a analizar $S_i$.

El etiquetado provisional se obtiene tras decidir cuál de los dos conjuntos de puntuaciones ($PL$ o $PM$) es mejor. Esto se consigue mediante el cálculo de $\bar{x}$ e $\bar{y}$, donde $\bar{x}$ es la media aritmética de las puntuaciones en $PL$, e $\bar{y}$ es la media aritmética de las puntuaciones en $PM$. Cuando $\bar{x} > \bar{y}$, S es etiquetada provisionalmente como "legítima". En caso contrario es considerada como provisionalmente "maliciosa". Nótese que el uso de medias aritméticas hace que las características del etiquetado sean especialmente susceptibles a cambios en los conjuntos de puntuaciones. Por lo tanto, es recomendable el uso de otra métrica en el caso de que estos presenten distribuciones altamente asimétricas, o cuando las muestras de acciones sean especialmente heterogéneas entre usuarios. Dado que éste no es el caso habitual en el estado del arte, nuestra propuesta asume su homogeneidad. La implementación de métricas alternativas o la decisión de cuál de ellas optimiza la eficacia del sistema se pospone a trabajos futuros.

#### 5.2.1.3 Sistema de puntuaciones

Sean las secuencias de acciones $S = S_1 \dots S_n$, $D = D_1 \dots D_k$ tales que $0 < n, k$ y $\forall S_i S_j$ se cumple $0 < i, j \leq T$. Además $S_i, D_j \in A$, donde $A$ es el conjunto de acciones de dimensión $0 < T$ observables en el entorno monitorizado. La puntuación obtenida al correlacionar $S$ con $D$ procede de la búsqueda de su mejor segmento alineado, lo que corresponde con el paradigma del alineamiento local.
Para lograr el mejor alineamiento local de $S$ y $D$, el sistema propuesto aplica las bases del algoritmo de Smith-Waterman [SW81] y su implementación por medio de esquemas de programación dinámica. Por lo tanto, a cada par de elementos $(S_i, D_j)$ que ocupan la misma posición, tal que $i = j$, se le asigna una puntuación. Su valor viene dado por funciones de similitud del tipo $s(S_i, D_j : A \times A \rightarrow \mathbb{R}$. En el caso en que $S_i = D_i$ la puntuación resultate es positiva. Pero si $S_i \neq D_i$ o alguno de los símbolos es un *gap*, es negativa. Nótese que el algoritmo de Smith-Waterman penaliza la aparición consecutiva de *gaps*. Dada una secuencia del tipo $g = g_1 \dots g_v$ en la que cada elemento $g_i$ es un *gap*, el símbolo $g_1$ es denominado *gap* inicial (del ingés *opening gap*). Al tratar con cadenas de aminoácidos, la posición del *gap* inicial determina la penalización incremental de la puntuación del alineamiento asociada a esta subsecuencia. Sin embargo, es habitual que fuera de este campo se considere un valor constante $\delta$, tal y como sucede en nuestra propuesta. En la Figura 5.4 se muestra un ejemplo de alineamiento local.

```
X: T C C A T C T A C T C G G G
   |   | | |       |   | | | |
Y: T * C A T G G G C * C G G G
```

Figura 5.4: Ejemplo de alineamiento de secuencias local.

Si se asume la función de puntuación $s(S_i, D_j)$ mostrada a continuación, y se penaliza $\delta = -1$ por *gap*, el valor del alineamiento de las secuencias $X$ e $Y$ es 16.

$$s(S_i, D_j) = 2sis(S_i = D_j - 1sis(S_i \neq D_j) \tag{5.1}$$

Nótese que la puntuación óptima entre dos secuencias $S$ y $D$ es la que se obtiene cuando $\forall S_i D_j, S_i, = D_j$. Por otro lado, la peor puntuación se obtiene si $\forall S_i D_j, S_i, \neq D_j$. En situaciones parecidas a esta última es posible que el alineamiento devuelva puntuaciones negativas. Por convenio, estos casos son ajustados a 0, considerándose así que su similitud es nula.

Los segmentos de $S$ y $D$ habitualmente son representados mediante $S_{i:p}$ y $D_{j:q}$, tales que $1 \leq i \leq p \leq n$ y $1 \leq j \leq q \leq k$. La función $f(S_{i:p}, D_{j:q}) : i : p \times j : q \rightarrow R$ devuelve la puntuación resultante de su alineamiento. De este modo, y considerando las secuencias del ejemplo de la Figura 5.4, se cumple que $f(X, Y) = 16$ , siendo $X = S_{i:p}$ e $Y = D_{j:q}$.

El objetivo de los algoritmos de alineamiento locales entre dos secuencias $S$ y $D$ es hallar el resultado del alineamiento óptimo entre los posibles pares de segmentos de cada una de ellas. Esto es representado mediante $Local_{Alignm}(S, D)$, y corresponde con la expresión

$$Local_{Alignm}(S, D) = \max_{0 \leq i \leq p \leq n, 1 \leq j \leq q \leq k} \{f(S_{i:p}, D_{j:q}), 0\} \tag{5.2}$$

Para calcular $Local_{Alignm}\,(S, D)$, el algoritmo de Smith-Waterman implementado propone la construcción de una matriz $H$ bidimensional cuyos ejes representan los símbolos de cada elemento de $S$ y $D$. Cada una de sus celdas $H_{p,q}$ contiene la puntuación de mayor valor obtenida para los segmentos que concluyen en $S_p$ y $S_q$. Esto corresponde con la expresión recursiva:

$$H_{p,q} = \max \begin{cases} 0 & \text{(5.3a)} \\ H_{p-1,q-1} + s(S_p, S_q) & \text{(5.3b)} \\ \max w \geq 1\{H_{p-w,q} + \delta\} & \text{(5.3c)} \\ \max b \geq 1\{H_{p,q-b} + \delta\} & \text{(5.3d)} \end{cases}$$

donde $1 \leq p \leq n, 1 \leq q \leq k$. Sus casos base son $H_{p,0} = 0$ y $H_{0,q} = 0$, dado que al faltar uno de los elementos a comparar, su similitud es nula. En la recursión, cada una de las expresiones parciales a comparar tiene un significado distinto: por un lado, $H_{p-1,q-1} + s(S_p, S_q)$ representa la situación de acierto, $\max_{w \geq 1} H_{p-w,q} + \delta$ la de sustitución de elementos por gaps, y $\max_{b \geq 1} H_{p,q-b} + \delta$ la de inserción de gaps entre elementos. El valor 0 evita la generación de puntuaciones negativas. Por lo tanto, para calcular $Local_{Alignm}(S, D)$ el procedimiento es el siguiente:

1. Se construye la matriz $H_{p,q}$ y se inicializan las celdas correspondientes con los casos base ($H_{p,0} = 0$ y $H_{0,q} = 0$).

2. Se rellena la matriz aplicando la expresión recursiva. El camino a seguir recorre desde la esquina superior izquierda a la esquina inferior derecha

3. Una vez completado el recorrido se identifica su valor más alto.

4. A partir de él se retrocede hasta alcanzar una celda de valor 0, acumulando las puntuaciones recorridas. Cada paso retrocede hacia $(i-1, j)$ o $(i, j-1)$, dependiendo del movimiento seguido durante la construcción de $H_{p,q}$.

En esta aproximación la presencia de *gaps* se penaliza de manera uniforme, asignando un valor $\delta = -1$; el alfabeto $\Sigma$ es el conjunto de acciones realizables por el usuario, y la función de puntuación aplicada es la siguiente:

$$s(S_i, D_j) = \begin{cases} 1 & if\ S_i = D_j \\ 0 & if\ S_i \neq D_j \end{cases} \tag{5.4}$$

#### 5.2.1.4 Refinamiento de etiquetados

La validación del etiquetado decide cuando una alerta debe de ser emitida. Sus elementos de partida son los vectores de puntuaciones $P_L = P_{L1},\ \ldots,\ P_{Lr},$, $P_M = P_{M1},\ \ldots,\ P_{Mr},$ y el etiquetado provisional generados durante la etapa de análisis. Para determinar si la diferencia entre $P_L$ y $P_M$ es lo suficientemente representativa como para emitir un etiquetado final, se aplica la prueba no paramétrica de Mann-Whitney, conocida como U-test [MW46]. Ésta es una alternativa al T-test de Student para comparar las medias entre dos distribuciones en base al estudio de sus medias. A partir del estadístico $U$ se obtiene el nivel de significancia de la prueba. Si su valor supera cierto intervalo de confianza $p$ (0.05 en la experimentación), se concluye que la hipótesis nula $h_o$ es correcta. En este caso el parecido entre $P_L$ y $P_M$ es muy grande, y por lo tanto el etiquetado no es de confianza. En el caso contrario $h_o$ es rechazada. Esto significa que $P_L$ y $P_M$ difieren lo suficiente como para pasar del etiquetado provisional de la etapa anterior, a un etiquetado definitivo. Adicionalmente, cuando la secuencia a analizar se etiqueta definitivamente como perteneciente a un usuario enmascarado, también es emitida una alerta.
El intercambio de etiquetados que lleva a la emisión de una alerta, en una situación ideal puede representarse por medio de un autómata finito $M = (Q, \Sigma, q_0, F, \delta)$ en el que:

- $Q = \{q_0, q_1, q_2\}$ donde $q_0$ representa el etiquetado de "usuario desconocido", $q_1$ el de "usuario legítimo" y $q_2$ el de "usuario enmascarado".

- $\Sigma = \{l, m\}$ donde $l$ representa la monitorización de una o varias acciones realizadas por usuarios legítimos, y $m$ las de usuarios enmascarados.

- El estado inicial es $q_0$, ya que sin acciones monitorizadas es imposible determinar la naturaleza del usuario.

- $F = \{q_2\}$, debido a que solo se emiten alertas cuando el usuario es considerado enmascarado

- Dados $a, b \in Q$, $\delta$ es la función de transición de estados tal que $\delta(a, b) : Q \times \Sigma \to Q$. Cada transición es representada por Tabla 5.1 y la Figura 5.5.

Tabla 5.1: Transición de estados real para la emisión de alertas.

| $\delta$ | $l$ | $M$ |
|---|---|---|
| $q_0$ | $\{q_0, q_1\}$ | $\{q_0, q_2\}$ |
| $q_1$ | $\{q_0\}$ | $\{q_0, q_2\}$ |
| $q_2$ | $\{q_2\}$ | $\{q_0, q_1\}$ |

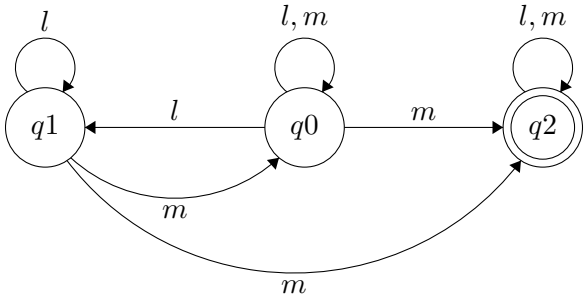



Figura 5.5: Transición de estados ideal para la emisión de alertas.

Pero en un caso de uso real deben considerarse los posibles errores de etiquetado. En $\acute{M} = (Q, \Sigma, q_0, F, \acute{\delta})$ se representa la emisión de alertas en dicha situación. Los parámetros $Q, \Sigma, q_0, F$ coinciden con los de $M$, pero esta vez la función de transición $\acute{\delta}$ se comporta de la siguiente manera (ver Tabla 5.2 y la Figura 5.6):

Tabla 5.2: Transición de estados real para la emisión de alertas.

| $\acute{\delta}$ | $l$ | $M$ |
|---|---|---|
| $q_0$ | $\{q_0, q_1, q_2\}$ | $\{q_0, q_1, q_2\}$ |
| $q_1$ | $\{q_0, q_1, q_2\}$ | $\{q_0, q_1, q_2\}$ |
| $q_2$ | $\{q_2\}$ | $\{q_2\}$ |

Como puede observarse, las transiciones $\acute{\delta}(q_0, l) = q_2$ y $\acute{\delta}(q_1, l) = q_2$ conllevan falsos positivos y las transiciones $\acute{\delta}(q_0, m) = q_1$ y $\acute{\delta}(q_1, m) = q_1$ falsos negativos. Por su parte, tanto en el comportamiento ideal como en el real, las transición que llevan de vuelta al estado inicial $q_0$ (a saber $\acute{\delta}(q_0, l) = q_0$, $\acute{\delta}(q_0, m) = q_0$, $\acute{\delta}(q_1, l) = q_0$ y $\acute{\delta}(q_1, m) = q_0$) proceden de los casos en que la prueba de verificación no ha sido superada.

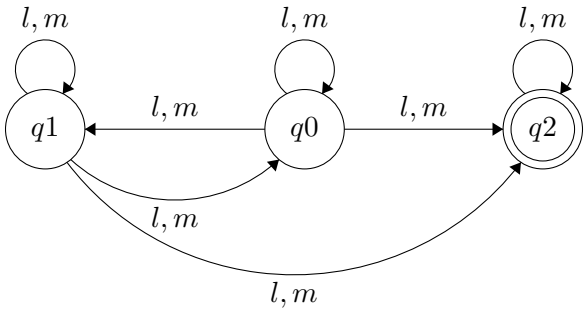



Figura 5.6: Transición de estados real para la emisión de alertas.

## 5.3 Fortalecimiento frente a técnicas de evasión

La elección del uso de técnicas de alineamiento de secuencias local presenta ventajas en determinados casos de uso, pero también desventajas en otros. Por ejemplo, considérese las siguientes tres situaciones:

1. Los usuarios legítimos del sistema inician la sesión que es monitorizada. Tras realizar actividades legítimas, son suplantados por enmascarados.

2. Los usuarios legítimos del sistema inician la sesión que es monitorizada. Los atacantes enmascarados actúan al mismo tiempo, pero en segundo plano.

3. Los atacantes enmascarados inician la sesión que es monitorizada. El usuario legítimo es suplantado a lo largo de toda la sesión.

Dado que el alineamiento local estudia segmentos en las secuencias de acciones, tanto 1) como 2) son casos de uso que se adaptan muy bien a sus características. En ellos no es recomendable el uso, ni de alineamiento global, ni híbrido, ya que el ataque es una porción de la actividad auditada. Sin embargo, en 3) cualquiera de las tres metodologías es eficaz. El problema del uso de alineamiento local se observa en las siguientes situaciones:

4. La secuencia a analizar es muy larga. En comparación con ella, los segmentos correspondientes a ataques enmascarados son muy pequeños [SYR15].

5. El ataque se parece al modelo de uso legítimo del sistema.

En ambos casos es más probable que el segmento de mayor longitud pertenezca al modo de uso legítimo, antes que al malicioso. En 4) esto sucede porque la presencia de actividades legítimas es mucho mayor. En 5) porque parte del ataque es considerado erróneamente como legítimo, llevando a la situación 4). Los atacantes enmascarados a menudo tienen la capacidad de explotar estas vulnerabilidades, lo que les permite evadir los sistemas de detección por medio de imitación. J. Tapidador y Clark [TC11] describieron las tres principales premisas que hay que tener en cuenta a la hora de afrontar estas amenazas:

- El adversario conoce las características de los algoritmos de detección, así como la información necesaria para lograr su evasión.

- El sistema de detección aplica modelos ideales de uso legítimo y malicioso. Por lo tanto, no es posible ni su envenenamiento por inserción de muestras maliciosas, ni su mejora.

- El atacante puede insertar acciones de relleno en cualquier lugar de los segmentos de los ataques a analizar, sin restricciones en torno a su tipo o su cantidad.

Con el fin de reforzar la eficacia del sistema en situaciones como 4) y 5), y de este modo, mejorar su robustez frente a ataques de imitación, el sistema desarrollado realiza un análisis multi-secuencial de las series de acciones monitorizadas. Esto permite el estudio de la partición de secuencias largas 4), y el análisis de nuevas secuencias inicializadas en situaciones poco discordantes 5). Las claves de esta propuesta residen en la estrategia de decisión del inicio de nuevas secuencias a analizar, y el cómo son gestionadas. A continuación se explica detalladamente cada una de estas características.

### 5.3.1 Decisión del inicio de nuevas secuencias a analizar

Cada vez que se emite un nuevo etiquetado definitivo, o que se detectan determinados eventos, existe la probabilidad de que el algoritmo de secuenciación inicie un nuevo proceso de análisis. Ambas situaciones son detalladas a continuación.

#### 5.3.1.1 Secuenciación base

Sea la secuencia de acciones $S = S_1, \ldots, S_n, 1 \leq n$ cuyo inicio coincide con el comienzo de la sesión de audición. Mientras que el sistema de análisis no es capaz de establecer su etiquetado definitivo, le son incorporadas las nuevas acciones monitorizadas $S_{n+1}, \ldots, S_m, 1 \leq n \leq m$, siendo $S_m$ la que finalmente elimina la incertidumbre. Por lo tanto, el sistema es capaz de reconocer con confianza la naturaleza de la secuencia resultante $S = S_1, \ldots, S_m$. Al establecerse el etiquetado validado, y por lo tanto definitivo de $S$, el sistema comienza la construcción de una nueva secuencia $\acute{S} = S_{m+1}, \ldots, S_r$, $1 \leq n \leq r$, que también es analizada y etiquetada con éxito (esta vez se consigue en la acción $S_r$). Este proceso se repite durante toda la etapa de detección. De esta manera, el total de actividades auditadas desde el inicio de sesión es fraccionado. Esto reduce la complejidad operacional de los procesos de análisis, y evita la presencia de segmentos muy largos de acciones legítimas, lo que facilita el tratamiento de problemas como los planteados en 4) y 5).

El total de acciones monitorizadas es denominado hilo principal a analizar. A la decisión de la inicialización de secuencias de acciones consecutivas se la denomina secuenciación base. En la Figura 5.7 se muestra un ejemplo de secuenciación base. En ella la primera secuencia analizada comienza en el inicio de sesión de audición ($S_1$) y acaba con la emisión del primer etiquetado definitivo ($S_m$). La segunda sesión analizada comienza en la acción siguiente a la que delimita la primera secuencia ($S_{m+1}$). Las secuencias a analizar se delimitarán de manera similar hasta el final de la sesión de monitorización.

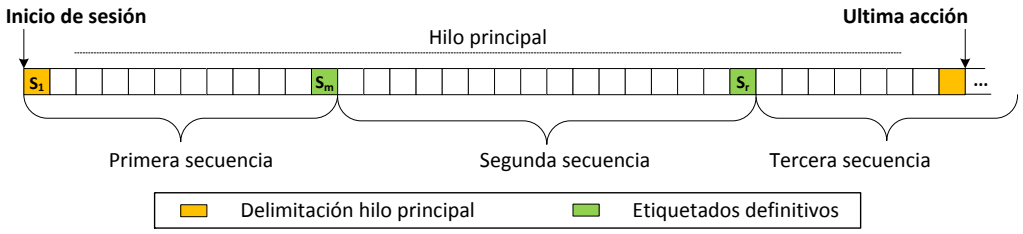



Figura 5.7: Ejemplo de secuenciación base.

#### 5.3.1.2 Secuenciación en paralelo

Al igual que sucede con la secuenciación base, la secuenciación en paralelo tiene como objetivo principal la reducción de la presencia de actividades legítimas en las secuencias a analizar. Esto se hace mediante el análisis en paralelo de sub-secuencias pertenecientes al hilo principal. Sus delimitadores son acciones poco comunes (desconocidas o anómalas), potencialmente peligrosas o arbitrarias. Con las dos primeras se busca la proximidad de las acciones maliciosas. Con la tercera, añadir un cierto grado de aleatoriedad al comportamiento del sistema. A continuación se describe cada tipo de delimitador:

- *Acciones Desconocidas.* Son acciones sin precedentes en el perfil de los usuarios legítimos. Su presencia indica el riesgo de no ser conocidas por ellos, o de pertenecer a la ejecución de tareas no habituales.

- *Acciones anómalas.* Son acciones con escasa frecuencia de aparición en el perfil de los usuarios legítimos. Por lo tanto, sí son conocidas, pero no se relacionan con la actividad habitual en el sistema.

- *Acciones potencialmente peligrosas.* Es la actividad potencialmente nociva para el sistema. Por ejemplo, en entornos GNU-Linux, el comando "*chmod -r 777 /*" autoriza la modificación de archivos en todo el sistema. De la misma manera, el comando "*rm -rf/boot/*" elimina todo el contenido del kernel, o el comando "*dd if command =/dev/urandom of =/dev/sda*" rellena de contenido aleatorio el disco duro. Este tipo de acciones son potencialmente peligrosas, y frecuentes en determinados atacantes.

- *Aleatorio.* Son actividades elegidas arbitrariamente, con el fin de dificultar el proceso de evasión del sistema de detección.

En la Figura 5.8 se muestra un ejemplo de secuenciación en paralelo partiendo de la secuenciación base de la Figura 5.7. Por lo tanto son analizadas diversas secuencias en concurrencia; por ejemplo, antes de etiquetarse la primera secuencia, son generadas tres sub-secuencias para analizar en paralelo. Su inicialización se debe al reconocimiento de acciones desconocidas y anómalas, o simplemente a factores aleatorios. Se llegan a ejecutar hasta cuatro tareas de análisis en paralelo: las de las nuevas secuencias, y la secuenciación base. Durante la generación de la segunda secuencia desde la secuenciación base, también se generan tres nuevas secuencias a analizar en paralelo. Dos de ellas se

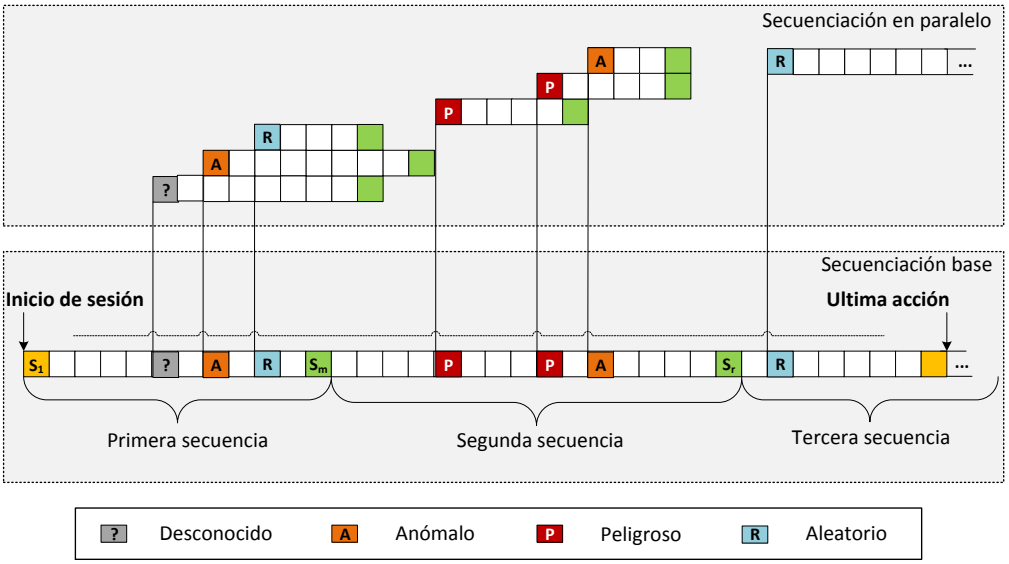



Figura 5.8: Ejemplo de secuenciación en paralelo.

inicializan al reconocerse actividades peligrosas, y la última por el reconocimiento de comandos anómalos. En análisis de las secuencias concurrentes es idéntico al de las secuencias base: mientras que no se determine su etiquetado definitivo se añaden las últimas acciones al conjunto de observaciones a estudiar. El resultado se resume en el Pseudocódigo 1.

### 5.3.2 Gestión de procesos de análisis concurrentes

La gestión de análisis concurrentes establece condiciones adicionales para la generación de nuevos procesos en paralelo. Para ello se limitan los recursos dedicados, y se aplica un cierto grado de no determinismo, tal y como se describe a continuación.

#### 5.3.2.1 Limitación de recursos

Se define como secuenciación ideal, a aquella en la que se construye y se ejecuta una nueva secuencia a analizar cada vez que es reconocido un delimitador. En ella no existen limitaciones computacionales, pudiendo ejecutarse infinitos procesos en paralelo. Pero en los casos de uso reales, las características del entorno de monitorización hacen que esto no sea factible; de sus carencias surgen vulnerabilidades que pueden ser aprovechadas por el atacante para tratar de denegar el servicio del sistema. Por lo tanto, en el despliegue del sistema propuesto sobre entornos reales deben de considerarse las características del equipo anfitrión y el escenario de monitorización. A partir de esto es posible establecer el parámetro $N$ que indica la cantidad de secuencias que pueden ser analizadas en paralelo sin comprometer sus recursos computacionales. La existencia de N acarrea entre otras, las siguientes consecuencias:

1. Cuando el número de secuencias en ejecución $n$ equivale a $N$, aunque se dé alguna de las situaciones que implique generar nuevas secuencias, éstas no pueden ser creadas

**Algoritmo 1:** Fortalecimiento frente a evasión basada en imitación.

**Entrada:** Cada acción $S_n$ realizada por el usuario protegido, analizadas como una secuencia $S = S_1 \dots S_n$. La longitud máxima de secuencia a analizar $N$. Las secuecuas de referencia $D$. El intervalo de confianza $I$.

**Salida :** Etiquetado de la secuencia $S$ como perteneciente a un "usuario desconocido", "usuario legítimo" o "usuario enmascarado".

```
Para cada acción X monitorizada hacer
    Si n > N entonces
        Eliminar_Primera(S);
        Insertar(S,X);
    Fin
    Si X es delimitador de nueva subsecuecnia entonces
        Análizar_Subsecuencia(X,S);
    Fin
    Mientras usuario desconocido hacer
        Alineamiento_Local(S, D);
        Refinamiento(S, etiquetado, I);
        Si distancia < I entonces
            Esperar_Nuevas_Acciones(S,D);
        si no
            Notificar_etiquetado(S,X,delimitador);
            Reiniciar_Secuencia(S,X);
        Fin
    Fin
    Mostrar etiquetado definitivo e incertidumbre;
Fin
```

2. El atacante es capaz de forzar esa situación mediante la inyección de determinadas acciones delimitadoras. De esta manera puede lograr que durante un periodo de tiempo no se creen nuevas secuencias, denegando el servicio del detector.

En consecuencia, es necesario tener en cuenta $N$ para proteger al sistema de ataques de evasión. Por otro lado, el sistema puede presentar problemas al decidir el etiquetado definitivo de determinadas secuencias. En este caso, la secuencia a analizar tarda demasiado en superar el U-test, o no es capaz de hacerlo. A raíz de ello se produce el bloqueo del proceso de análisis, y se penaliza el rendimiento del sistema, dando lugar a secuencias extremadamente largas, de naturaleza desconocida. En el peor de los casos, el sistema puede llegar a ejecutar $N$ procesos infinitos, denegando completamente su servicio. La construcción de secuencias tan grandes puede producirse cuando alguno de los dos modelos de uso no es definido correctamente, o cuando el atacante ejecuta determinadas acciones en el orden adecuado. Para combatir este problema se establece un número máximo de acciones por secuencia. Su valor se genera aleatoriamente en el momento de su inicialización, y su rango varía en función de los recursos computacionales del sistema.

#### 5.3.2.2 Indeterminismo

Tal y como se muestra en [TC11], la actividad realizada por los usuarios legítimos puede ser capturada con facilidad. A partir de este hecho, puede deducirse que el atacante también es capaz de modelar su comportamiento. Esto revela una parte importante del modo de uso legítimo, facilitando la distinción entre acciones delimitadoras y legítimas. Con este conocimiento, el atacante puede tratar de perpetrar alguna de las amenazas previamente mencionadas. Del mismo modo, el atacante también es capaz de inyectar secuencias de acciones de prueba que arrojen todavía más información acerca de las características de la implementación del detector. Para dificultar estos intentos de enumeración, el sistema propuesto incorpora un pequeño nivel de aleatoriedad en la toma de decisiones. Cada vez que en la monitorización de llamadas al sistema se reconozca una que sea delimitadora, se aplica una probabilidad de "no comenzar a generar la nueva secuencia". Así se incrementa el no determinismo del sistema, y se dificultan las acciones de espionaje.

## 5.4 Experimentación

Esta sección describe los detalles acerca de las colecciones de muestras, la generación de ataques de imitación y las características del escenario de pruebas considerados.

### 5.4.1 Colección de muestras

Para la evaluación del sistema propuesto se ha considerado la colección de muestras publicada por Schonlau et al. conocida como SEA [SEA00]. Esta decisión se fundamenta en que a pesar de la controversia que pueda involucrar su uso, la mayor parte de los trabajos previos también los han considerado. De esta manera es posible comparar las estadísticas obtenidas con las de aproximaciones similares. SEA está compuesta por capturas de las actividades realizadas por 50 usuarios distintos, que operaban sobre entornos Unix. Están organizados de manera que a cada usuario le corresponde un único fichero. Dicho fichero contiene una serie de 15,000 de las acciones que ha llevado a cabo durante el periodo de monitorización. Los primeros 5,000 comandos de cada usuario corresponden a actividades legítimas, por lo que han sido extraídos para la elaboración del modelo de uso legítimo del sistema. Los siguientes 10,000 comandos pueden tratarse tanto de ataques de enmascaramiento, como de actividades legítimas. Sus autores han propuesto la división de las acciones en bloques de 100 comandos (100 bloques por cada usuario), y han publicado una matriz con valores 0 y 1, que distinguen los bloques de actividades que representan acciones legítimas de los bloques relacionados con intrusiones. La distribución de los bloques correspondientes a ataques es la siguiente: a partir de los primeros 5000 comandos legítimos, existe una probabilidad del 1% de que el siguiente bloque sea un ataque enmascarado. Tras identificarse un ataque, existe una probabilidad del 80% de que el siguiente bloque represente la continuación del mismo ataque. De esta manera, aproximadamente el 5% de las muestras contienen actividades maliciosas. En la experimentación realizada, el modelo de uso malicioso es generado a partir de un subconjunto de los bloques con intrusiones.

### 5.4.2 Ofuscación de ataques enmascarados por imitación

A pesar de que SEA ofrecen una completa colección de muestras, no especifica si los ataques identificados han aplicado técnicas de ofuscación similares a los ataques de imitación. En consecuencia, la evaluación de la robustez del sistema requiere del desarrollo de una herramienta de ofuscación capaz de emular sus características. En la experimentación realizada, el sistema propuesto tiene como referencia un conjunto de ataques enmascarados. El proceso de ofuscación se basa en la inserción de acciones de relleno en sus secuencias originales. De esta manera, la distribución de las diferentes acciones que componen el ataque presenta un mayor grado de similitud con las secuencias que componen el modelo de uso legítimo, imitando en la mayor medida la apariencia de las actividades legítimas.
El aspecto más sensible del proceso de ofuscación es la decisión de qué acciones de relleno se deben ejecutar, y en qué lugares deben insertarse. El sistema elegido para para tomar dicha decisión es una adaptación de la estrategia de muestreo estocástico conocida como probabilidad proporcional al tamaño o PPS (del inglés *Probability-Proportional-to-Size*) [CK13]. En ella se parte de un conjunto de pesos $P = p_1, \ldots, p_r$ asociados a las acciones que se repiten con mayor frecuencia en la distribución del perfil del usuario legítimo $C = c_1, \ldots, c_r$, tales que $0 < r$. Por lo tanto, la acción $c_i$ presenta el peso $p_1, 1 \leq i \leq r$, siendo ésta su frecuencia de aparición. Para cada acción $c_i$, su aptitud $A_i$ viene dada por la fórmula:

$$A_i = \frac{p_i}{\sum_{i=1}^{r} p_i} \tag{5.5}$$

Esto garantiza que $\sum_1^r p_i = 1$, como corresponde a una distribución de probabilidades. Entre cada acción del ataque original, se inserta una cantidad aleatoria de acciones de relleno. Su elección parte de la generación de un valor aleatorio $R$ comprendido entre $0 \leq R \leq 1$. Por lo tanto, cada acción de relleno $c_i$, debe cumplir que $A_{i-1} < R < A_i$.

### 5.4.3 Configuración

La propuesta realizada ha sido implementada en un entorno GNU-Linux. Dado que los hilos son procesos que ocupan su propio espacio en la memoria virtual, su máximo número N depende del total de la memoria virtual asignada y del espacio de pila, tal y como se muestra en la expresión:

$$N = \frac{Total\ Memoria\ Virtual\ Asignada}{Dimensiones\ de\ la\ pila} \tag{5.6}$$

En el marco de la experimentación realizada, los equipos utilizados permiten hasta $N = 62572$, y el tamaño máximo de secuencia en proceso de análisis es de 1000 acciones. Si se supera este límite o concluye la sesión sin establecerse un etiquetado definitivo, es emitido el etiquetado provisional. En caso de contingencias, el etiquetado por defecto emitido corresponde con el de "usuario legítimo". Por otro lado, el alfabeto $\Sigma$ está integrado por los diferentes comandos de las muestras SEA que componen los modelos de uso del sistema. Adicionalmente se añade la acción NBSCs (del inglés *Never-Before-Seen Commands*). En

ella se agrupa cualquier comando que no aparezca en dichos modelos. En función de $\Sigma$, los delimitadores establecidos son:

- *Acciones desconocidas.* Son los comandos NBSCs.
- *Acciones anómalas.* Son los comandos cuya frecuencia de aparición en el modelo de uso legítimo es inferior al 5%.
- *Acciones peligrosas.* Son los comandos pertenecientes a $\Sigma$ relacionados con el montado de sistemas de archivos, administración de usuarios y grupos, consulta de Información del sistema o búsquedas.
- *Aleatorio.* Existe una probabilidad de que cualquier comando perteneciente a $\Sigma$ inicialice un nuevo proceso de análisis.

Otro aspecto a tener en consideración es la manera en que se aplica el no determinismo. Las probabilidades de generación de secuencias las da el vector $Prob = [P_1, P_2, P_3, P_4]$ donde $P_1$ es la probabilidad de generarse una nueva secuencia cuando se detecta una acción desconocida, $P_2$ cuando la acción es anómala, $P_3$ cuando es peligrosa, y $P_4$ es la probabilidad de iniciarse arbitrariamente. La configuración por defecto del sistema es $Prob = [0\%, 0\%, 0\%, 0\%]$. Esto quiere decir que inicialmente no se aplica determinismo, y únicamente es considerada la secuenciación base. Posteriormente se estudia qué sucede al variar alguno de estos parámetros y cómo influye en la precisión del sistema.
En la primera fase de experimentación se aplica la configuración por defecto. El comportamiento del sistema es valorado por medio de un proceso de evaluación cruzada que involucra secuencias de SEA que no han sido utilizadas en el modelado de los usuarios. La tasa de acierto o TPR se obtiene al analizar secuencias de acciones maliciosas. Por otro lado, la tasa de falsos positivos o FPR se calcula a partir del resultado de analizar secuencias de acciones legítimas (ver Sección 3.6.1 “Precisión”). En la etapa de modelado de los usuarios se determina la cantidad de secuencias de referencia que representa cada uno de ellos. El modelado de los enmascarados implementa secuencias de acciones maliciosas de 48 bloques de 100 comandos cada una. Inicialmente, el modelo de uso legítimo se genera con los primeros 48,000 comandos ejecutados por cada usuario, aunque varía a lo largo de la experimentación. También lo harán la longitud de las secuencias de referencia y la probabilidad de inicialización de nuevos procesos en concurrencia. Para evaluar la resistencia de la propuesta frente a evasión basada en imitación, se utilizan las mismas muestras de ataques, pero modificadas por el método de ofuscación descrito en la subsección anterior.

### 5.4.4 Resultados

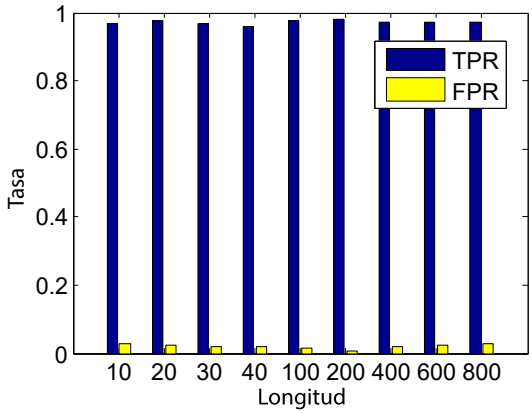
A continuación se describen los resultados obtenidos por el sistema propuesto al aplicar la colección SEA y ataques ofuscados por imitación. Esta sección describe su eficacia al operar únicamente considerando la secuenciación base y al integrar la secuenciación en paralelo. Los resultados obtenidos son comparados con los del resto de la bibliografía.

### 5.4.5 Evaluación de la secuenciación base

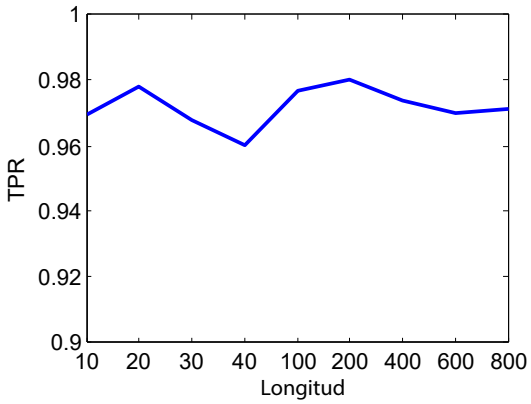
En esta primera fase de la experimentación es considerada la configuración por defecto del detector y se evalúa el impacto de la longitud de las secuencias en la precisión del sistema, las variaciones del comportamiento de los usuarios y su resistencia a intentos de imitación.

#### 5.4.5.1 Impacto de la longitud de las secuencias

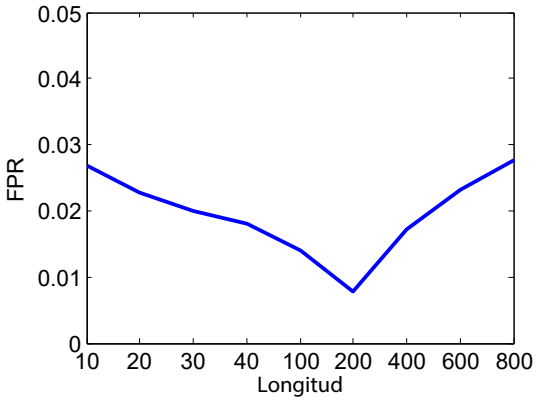
En la Figura 5.9(a) se muestra la tasa de acierto TPR y falsos positivos FPR obtenidos al determinar diferentes longitudes de secuencia en la construcción de ambos modelos (en concreto, longitudes de 10, 20, 30, 40, 100, 200, 400, 600 y 800 comandos). Sobre el eje Y se observan las tasas TPR/FPR, mientras que en el eje $X$, las diferentes longitudes aplicadas.




(a) Comparación de TPR/FPR.




(b) Variación de TPR.




(c) Variación de FPR.

Figura 5.9: Precisión en base a la longitud de las secuencias.

La desviación típica del TPR obtenida es 0.00571686. La del FPR es 0.00596657. El mejor resultado se obtiene con secuencias de longitud 200 (compuestas por dos bloques), siendo $TPR = 0.9804$ y $FPR = 0.0077$. Sin embargo, el peor resultado se obtiene con secuencias de longitud 10, con $TPR = 0.969$ y $FPR = 0.268$. En la Figura 5.9(b) y la Figura 5.9(c) se muestra la evolución de cada tasa por separado. Sus ejes indican los mismos parámetros que en la figura anterior. A raíz de los resultados obtenidos puede concluirse

que independientemente de la longitud utilizada, la precisión alcanzada ha sido alta. Las variaciones observadas sobre la tasa TPR son poco representativas, con oscilaciones del orden $0.9750 \pm 0.0054$ y una diferencia de 1.11% entre valores máximos y mínimos (ver Figura 5.9(b)). Sin embargo, la tasa FPR presenta una evolución más constante, del orden $0.001765 \pm 0.00995$ con una diferencia del 27.89% (ver Figura 5.9(c)). En consecuencia, durante el resto de esta primera evaluación son considerados modelos con secuencias de longitud 200.

#### 5.4.5.2 Estudio del comportamiento de los usuarios

Otro aspecto interesante que evaluar es el comportamiento del sistema para cada usuario por separado. Al igual que como comprobaron R.A. Maxion et al. [MT04], la experimentación realizada corrobora que determinados usuarios son más propensos a ser suplantados. En la Figura 5.10 se muestran las tasas TPR y FPR obtenidas para cada uno de los 50 usuarios. El eje X indica el identificador de cada usuario, y el eje Y el valor de cada una de las tasas. Para su valoración debe tenerse en cuenta la ubicación de cada uno de ellos en el espacio ROC (ver Sección 3.6.1.1 "Curva ROC"). En la Figura 5.11 se muestra la representación en el plano ROC de las tasas obtenidas para cada usuario. En ella puede observarse como su proximidad con el eje superior es muy pequeña, lo que indica la proximidad de los resultados obtenidos con el comportamiento óptimo de un sensor.

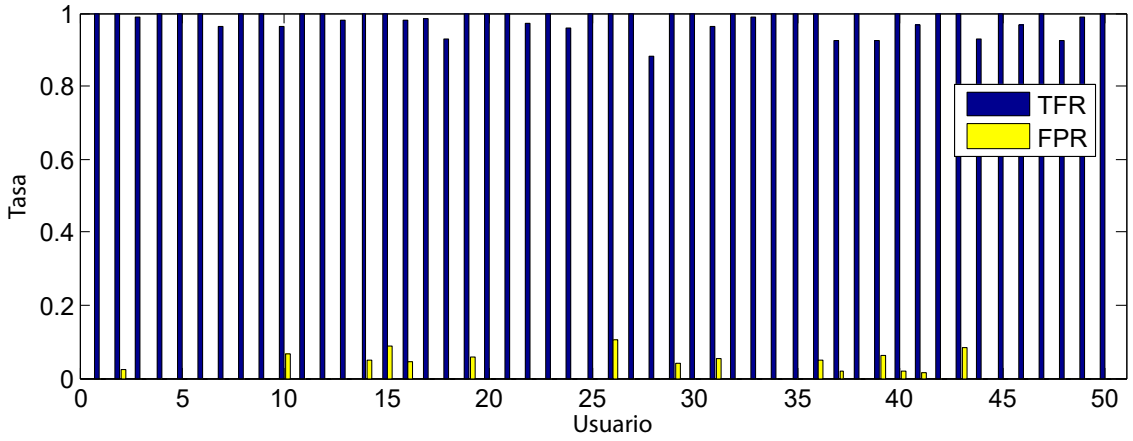



Figura 5.10: FPR/TPR por usuario con longitud 200.

En este experimento el 88% de los usuarios ha arrojado una precisión del 100% a la hora de detectar atacantes enmascarados, mientras que el 12% ha presentado errores. La precisión a la hora de detectar verdaderos positivos ha oscilado en el intervalo $0.98 \pm 0.02$. Sin embargo, los resultados en la prueba de falsos positivos han sido algo peores: el 70% de los usuarios han obtenido un 0% de error, mientras que el 30% restante en algún momento del experimento han generado falsos positivos. El FPR ha variado en el intervalo $0.05 \pm 0.05$, lo que indica una desviación mucho más representativa.

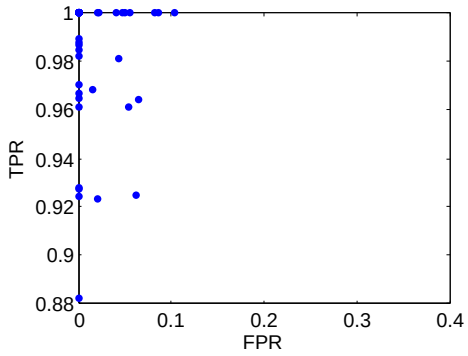


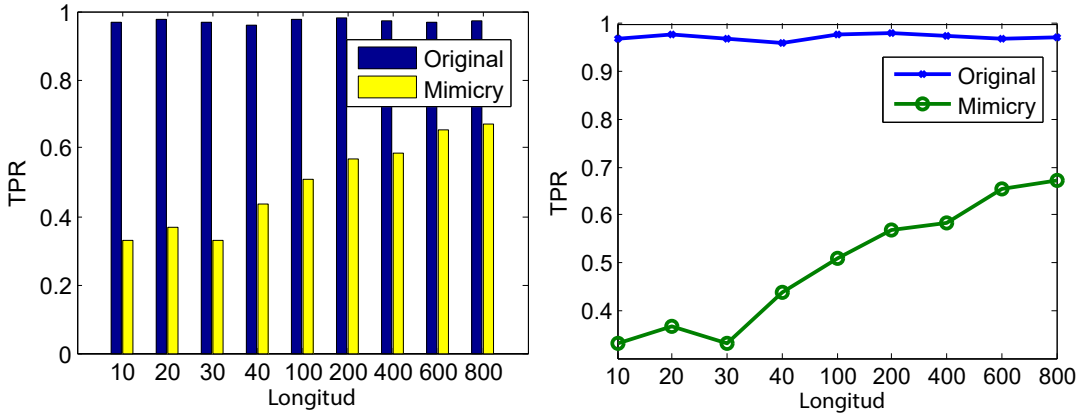

Figura 5.11: Precisión para cada usuario en el espacio ROC.

#### 5.4.5.3 Resistencia a ataques de imitación

Para probar la robustez del sistema frente a ataques de mimetismo se han ofuscado los ataques enmascarados utilizados en la evaluación anterior. Inicialmente, los modelos aplicados utilizan la configuración del detector por defecto. También se han considerado diferentes longitudes de secuencia, lo que lleva a resaltar que una vez completada la ofuscación de una muestra, su longitud aumenta con el contenido de relleno. En consecuencia, se ha convenido que los ataques ofuscados sean agrupados en función de la dimensión de su vector de ataque original, previo a su modificación.

En Figura 5.12(a) se muestran la precisión del sistema en cada uno de estos casos. En ellas también puede observarse cómo el detector ha sido evaluado con muestras de diferente longitud de vector de ataque, y que únicamente se mide la tasa TPR. Esto es debido a que el valor de FPR ya fue calculado en las pruebas anteriores para las mismas longitudes. El eje Y de la Figura 5.12(a) muestra el valor de las tasas, y el eje X la longitud del de los vectores. En ella es posible observar cómo en la configuración por defecto, la capacidad del sistema de detectar ataques de imitación es muy baja. Al compararse los resultados de ambas configuraciones, se observan importantes diferencias: cuando el vector es ofuscado, y su longitud $|10|$, el valor TPR desciende del 0.9696 original al 0.331. Por otro lado, cuando su longitud es $|800|$ el valor TPR cambia del 0.9712 al 0.671. Ésta es la variación menos significativa registrada.

A raíz de los resultados obtenidos también puede observarse que al aumentar la longitud del vector de ataque original, aumenta la tasa TPR. El caso peor se da cuando la longitud de su vector es $|10|$ con $TPR = 0.331$. El caso mejor se da cuando es $|800|$ con $TPR = 0.671$. La diferencia entre la elección de uno y otro tamaño implica variaciones del orden $TPR = 0.501 \pm 0.170$. Por lo tanto, la precisión obtenida prácticamente se reduce a la mitad. Esto es debido a que cuando las secuencias tienen una mayor representación de acciones maliciosas, la probabilidad de encontrar segmentos alineados que encajen con el modelo de uso malicioso aumenta. En la Figura 5.12(b) se muestra más detalladamente dicha progresión.




(a) Precisión ganada al ofuscar los ataques

(b) Evolución de la tasa de acierto

Figura 5.12: Precisión al analizar ataquies de imitación.

### 5.4.6 Evaluación de la secuenciación en paralelo

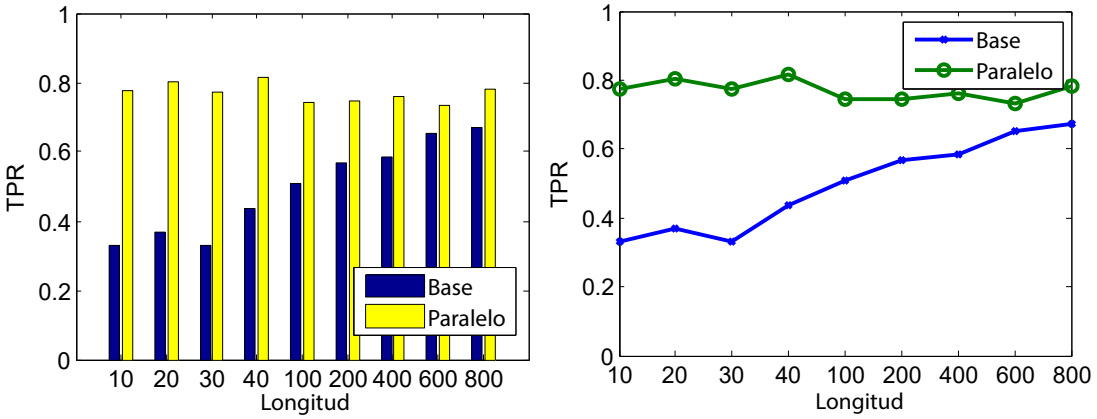
Esta etapa de la experimentación determina la efectividad de la secuenciación en paralelo al tratar con ataques ofuscados por imitación. Para ello se estudian diferentes aspectos, como su precisión y la repercusión en ella del consumo de recursos, cambiar la probabilidad de inicialización de secuencias, o su impacto en la tasa de falsos positivos.

#### 5.4.6.1 Robustez frente a imitación

En esta primera prueba, la configuración implementada coincide con el vector de probabilidades $[20\%, 20\%, 20\%, 5\%]$. En la Figura 5.13(a) se comparan los resultados obtenidos previamente, con los de esta nueva configuración. El eje Y de la Figura 5.13(b) muestra el valor de las tasas TPR, y el eje X la longitud del de los vectores. A la vista de los resultados, la mejora al aplicar la secuenciación en paralelo es evidente: se alcanzan tasas TPR mucho más altas y el detector es menos sensible a la longitud del vector de los ataques. La variación de la precisión oscila en el rango $0.7749 \pm 0.0410$, y con una desviación máxima del 11.1%. Esto sucede porque el peor resultado es TPR=0.7339 con longitud $|600|$ y el mejor resultado TPR=0.8154 con longitud $|40|$. Sin embargo, a diferencia de lo que sucede con la configuración por defecto, en este caso el incremento de la longitud de los ataques no implica una tendencia a la mejora en la precisión, la evolución de la precisión alcanza un estado de saturación.

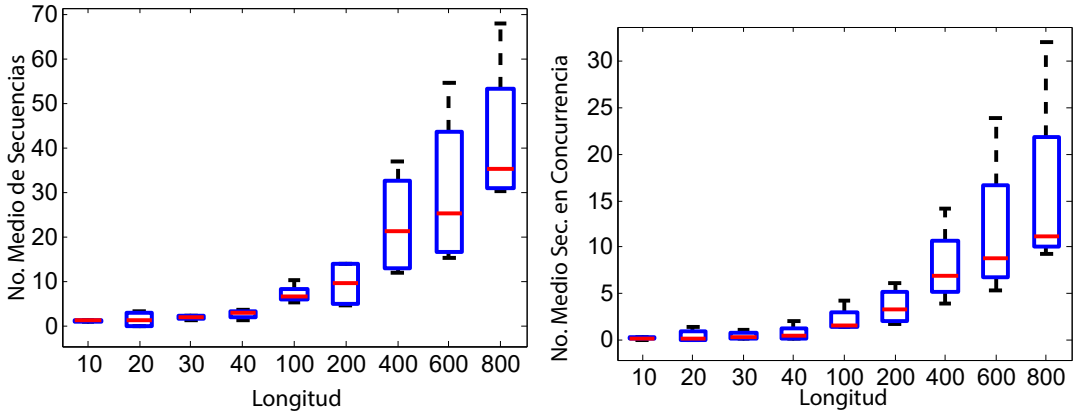
#### 5.4.6.2 Influencia del consumo de recursos

Otro aspecto que evaluar es el impacto de la generación de nuevas secuencias en los recursos de cómputo del sistema anfitrión. En la Figura 5.14(a) se muestran estadísticas acerca de la cantidad de secuencias generadas y analizadas en concurrencia, en función de la longitud del vector de ataque bajo esta misma configuración. En concreto, para cada grupo de vectores de ataque se muestra la cantidad media de secuencias generadas a partir de cada tipo de delimitador, y la media de las que han permanecido en proceso de análisis al mismo




(a) Comparativa. (b) Evolución de la tasa de acierto.

Figura 5.13: Precisión en $[20\%, 20\%, 20\%, 5\%]$ y ajuste por defecto.



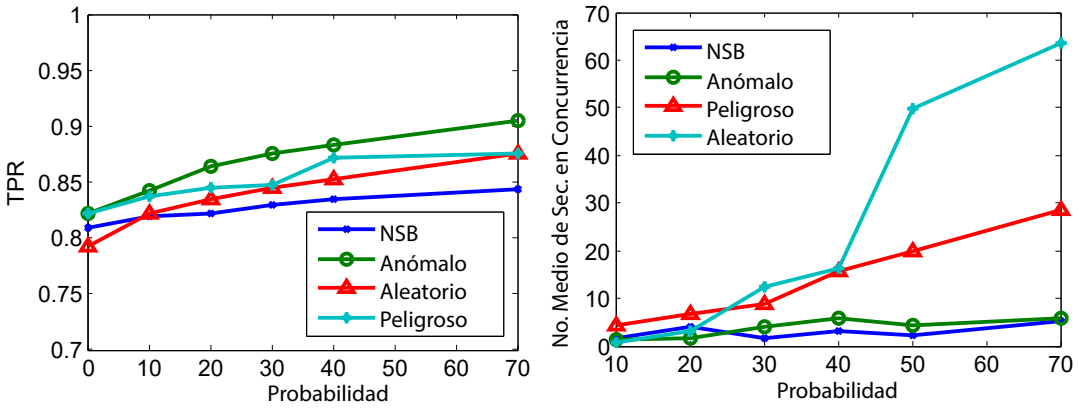

(a) Número medio de secuencias ejecutadas. (b) Promedio de secuencias analizadas en paralelo.

Figura 5.14: Consumo de recursos en $[20\%, 20\%, 20\%, 5\%]$.

tiempo. También se muestra la cantidad total de secuencias analizadas, la cantidad media del total de secuencias analizadas en concurrencia y el número máximo de secuencias que han permanecido en proceso de análisis al mismo tiempo (caso peor computacionalmente). Dado que el equipo anfitrión permite el análisis de hasta $N = 62572$ secuencias en paralelo, el caso peor computacional (104 secuencias simultáneas de longitud $|800|$) se encuentra muy lejos de los límites establecidos. Por lo tanto, la configuración aplicada apenas sobrecarga los recursos computacionales disponibles. Por otro lado, en la Figura 5.14(a) se muestra la evolución del total de secuencias analizadas en función de la longitud de los vectores de ataque, y en la Figura 5.14(b), los valores promedios de secuencias que permanecen en ejecución. En ambos casos, el consumo demuestra ser proporcional a la longitud del vector de ataque. En acuerdo con la información mostrada, el aumento del vector de los ataques ofuscados apenas hace fluctuar la capacidad de detección del sistema. No obstante, su aumento sí que incrementa el consumo de recursos computacionales; en consecuencia, no es recomendable el análisis de secuencias excesivamente grandes.

#### 5.4.6.3 Efecto de los cambios en las probabilidades de inicialización

Llegados a este punto, la experimentación únicamente ha considerado la configuración del detector por defecto sin secuenciación en paralelo, y la configuración con probabilidades $[20\%, 20\%, 20\%, 5\%]$. Pero también es importante determinar de qué modo afectan las variaciones de dichos parámetros. En la Figura 5.15(a) se muestra el TPR y la cantidad total de secuencias generadas al repetir el análisis de la prueba anterior, pero aplicando diferentes probabilidades de inicialización. Tambien se observa la evolución del TPR en función de la probabilidad asignada a cada uno de ellos. A partir de ella es posible deducir que la variación de dichos parámetros produce cambios poco significativos en la capacidad del sistema de detectar ataques ofuscados. La configuración menos precisa es $[20\%, 20\%, 10\%, 5\%]$, con $TPR = 0.792$. Sin embargo, la más precisa es $[20\%, 40\%, 20\%, 5\%]$, con $TPR = 0.905$. La precisión oscila en el orden $848.5 \pm 56.5$, con una variación máxima del 14.26%. Por lo tanto, y al igual que sucede en la prueba anterior, la precisión no presenta tendencias a mejorar o a empeorar en función del consumo de recursos. Esto puede contrastarse con la evolución del consumo de recursos mostrada en detalle en la Figura 5.15(b). Como puede observarse, el número de secuencias analizadas aumenta al asignarles mayores probabilidades. Pero este aumento no se produce de igual manera para todos ellos. Concretamente, el incremento de la probabilidad de inicialización aleatoria genera muchas más secuencias nuevas que cualquier otro. La segunda mayor variación se observa en el aumento de la probabilidad de inicialización por identificación de eventos peligrosos. Sin embargo, las variaciones en la inicialización por comandos únicos o eventos anómalos apenas producen fluctuaciones. Esto es debido a que son los delimitadores menos frecuentes en las secuencias analizadas.




(a) Precisión según probabilidad.

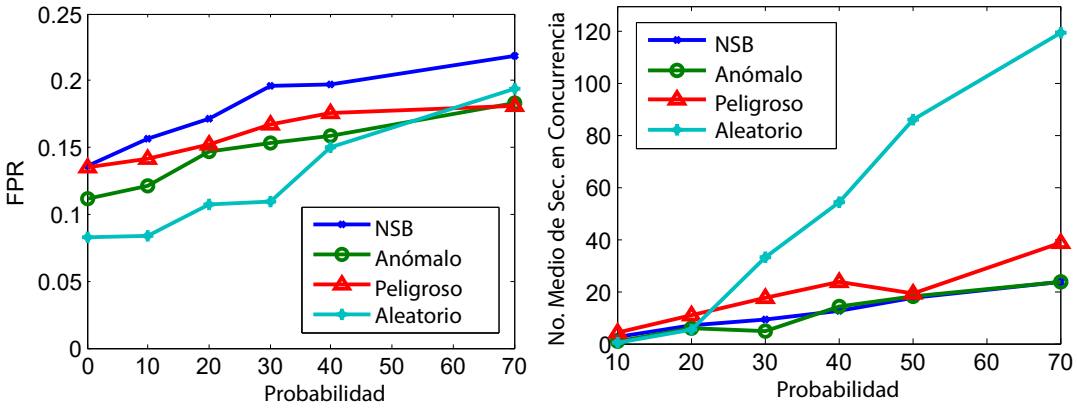
(b) Número de secuencias según probabilidad.

Figura 5.15: Relación entre precisión y número de secuencias analizadas.

#### 5.4.6.4 Impacto del no-determinismo

El último aspecto evaluado es el impacto de la aplicación de no determinismo en la tasa de falsos positivos FPR. Para ello se han considerado las mismas condiciones que en el primer experimento, pero se han aplicado diferentes vectores de probabilidades.

En la Figura 5.16(a) se muestra la precisión obtenida, y la cantidad de secuencias generadas para cada configuración. Los mejores resultados corresponden con la configuración $[20\%, 20\%, 20\%, 70\%]$ con $FPR = 0.083$. Los peores resultados se dan en $[10\%, 20\%, 20\%, 5\%]$ con $FPR = 0.218$. La precisión ha oscilado en el orden $158 \pm 60$, con una variación máxima del 122.4%. A diferencia de la prueba anterior, se observa una tendencia a disminuir la tasa FPR con el incremento del número de secuencias generadas. Por otro lado, en la Figura 5.16(b). se muestra la evolución del total de secuencias analizadas en función de los parámetros del vector de probabilidades. Al igual que sucede en el análisis de ataques de imitación, el aumento de la probabilidad de iniciación de nuevas secuencias aleatoriamente es el que experimenta un mayor crecimiento. El resto de delimitadores mantienen un incremento parecido, destacando la inicialización por actividades potencialmente peligrosas, sobre las de comandos únicos o eventos anómalos. En consecuencia, los puntos con FPR más bajo de la Figura 5.16(a) coinciden con los de más secuencias generadas en la Figura 5.16(b).




(a) FPR según probabilidad.

(b) Secuencias iniciadas según probabilidad.

Figura 5.16: Evolución de FPR número de secuencias iniciadas.

### 5.4.7 Discusión

En la Tabla 5.3 se resumen los resultados obtenidos y se comparan con los de los trabajos previos que también aplicaron SEA. En esta comparativa se tiene en cuenta la eficacia de la secuenciación base 1) y la eficacia de la secuenciación en paralelo 2). Esta tabla resalta los siguientes aspectos de nuestra propuesta:

1. Los resultados alcanzados en términos de precisión son de los mejores de la bibliografía. De este modo queda probado que la aproximación realizada es potencialmente una buena alternativa a cada uno de ellos.

2. La secuenciación base (1) (TPR=0.98, FPR=0.007) obtiene mejores resultados que la secuenciación en paralelo (2) (TPR=0.98, FPR=8.3) al analizar las muestras de SEA. Éstos son además los mejores de la bibliografía en términos de falsos positivos, y los terceros mejores en tasa de acierto. Sin embargo, al no activarse la secuenciación

Tabla 5.3: Precisión de diferentes propuestas al ser evaluadas con SEA.

| Propuesta | TPR (%) | FPR (%) |
|---|---|---|
| **IPAM ([DH88])** | 41.1 | 2.7 |
| **Bayes one-step Markov ([Dum99])** | 69.3 | 6.7 |
| **Uniqueness ([ST00])** | 39.4 | 1.4 |
| **Compression ([ST00])** | 34.2 | 5 |
| **Hybrid multi-step Markov ([JV01])** | 49.3 | 3.2 |
| **Sequence Match ([YLC+02])** | 26.8 | 3.7 |
| **Naïve Bayes (update) ([YLC+02])** | 61.5 | 1.3 |
| **Naïve Bayes ([WS03])** | 66.2 | 4.6 |
| **Semi-Global Alignment ([CBSB03])** | 75.8 | 7.7 |
| **ECM ([OOAK04])** | 72.3 | 2.5 |
| **Two-class NB ([MT04])** | 66.2 | 4.6 |
| **Two-class NB updated ([MT04])** | 61.5 | 1.3 |
| **Decision Stumps ([JST+07])** | 89.2 | 10.1 |
| **Sequence Alignment (actualizado) ([CS08])** | 68.6 | 1.9 |
| **N-Gram STF-IDF ([GOKO10])** | 91.9 | 5.08 |
| **SVM Simple cmds ([SS11b])** | 98.7 | 66.4 |
| **SVM Taxonomy ([SS11b])** | 94.8 | 60.6 |
| **SVM Search Behavior ([SS11b])** | 100 | 1.1 |
| **SVM App-freq ([SS11b])** | 90.2 | 42.1 |
| **PHMM ([HS11])** | 70 | 5 |
| **IWNB ([Sen15])** | 70.2 | 4.5 |
| **DDSGA (Restricted) ([KBH15])** | 83.3 | 3.4 |
| **DDSGA (Free) ([KBH15])** | 80.5 | 3.8 |
| **DDSGA (Restricted Update) ([KBH15])** | 88.4 | 1.7 |
| **(1) Esta propuesta (secuenciación base)** | 98 | 0.07 |
| **(2) Esta propuesta (secuenciación paral.)** | 98.3 | 8.3 |

en paralelo presenta un riesgo alto de ser evadido por imitación, con TPR=0.568 en ataques ofuscados. Esto es similar al resto de trabajos [TC11].

3. Los mejores resultados al implementar la secuenciación en paralelo (2) conservan la tasa alta de acierto de la secuenciación base (1) (TPR=0.98), pero penalizan la tasa de falsos positivos, siendo en este caso FPR= 8.3. Esta desventaja es compensada con una alta capacidad de identificar ataques de imitación, con TPR=0.872. Por lo tanto, esta modificación es una alternativa más robusta.

En términos generales se distinguen dos configuraciones capaces de adaptar nuestra propuesta a los diferentes casos de uso. La primera de ella solo tiene en consideración la secuenciación base, y es recomendada para proteger sistemas con activos de sensibilidad media, en los que la calidad de servicio deba preservarse a toda costa. En este caso el sistema es capaz de brindar una precisión muy alta, pero presenta un importante riesgo de ser evadido por ataques de imitación. Por otro lado, la propuesta puede activar la configuración en paralelo para ganar robustez frente a evasión, mejorando

considerablemente su capacidad de detección de este tipo de amenazas. Pero esto conlleva un incremento del consumo de recursos, y un aumento de su tasa de falsos positivos, lo que reduce la calidad de servicio del sistema sobre el que opera. Esta configuración es apta para escenario con activos mucho más sensibles, y que por lo tanto requieran de mayor protección.

# Capítulo 6

# Detección de malware mediante análisis de carga útil

En la última década, la detección de malware en redes mediante el análisis estadístico de la carga útil del tráfico se ha convertido en una medida esencial para la identificación de nuevos especímenes de malware. Sin embargo, su despliegue ha sido objeto de controversia por parte de la comunidad investigadora, abriéndose el debate sobre sus posibles consecuencias al operar en redes actuales. Con el fin de contribuir a su adaptación a las redes de nueva generación, este capítulo presenta un sistema de detección de malware en redes capaz de comportarse de manera precisa tanto en los estándares de evaluación, como en las redes de comunicaciones reales. Nuestra aproximación consigue aunar todas las ventajas de la detección basada en firmas, gracias a su integración como módulo de preprocesamiento del conocido NIDS *Snort* [Sno18], y los de la detección basada en anomalías, por medio de la creación de patrones que representen el modo de uso habitual y legítimo de la red. APAP es un sistema de detección de intrusiones orientado al reconocimiento de anomalías en la carga útil del tráfico de la red. Está basado en una modificación del sistema PAYL denominada Anagram, y al igual que sus predecesores, requiere de una fase de entrenamiento y una fase de detección: en la fase de entrenamiento crea un modelo estadístico del tráfico legítimo mediante filtros Bloom (del inglés *Bloom filter*) [RK15] y la técnica N-gram [GOO14]. Comparando dicho modelo con el de varios ataques conocidos es capaz de establecer un conjunto de reglas que permitan detectar las anomalías presentes en el tráfico de la red a proteger. La fase de detección compara el tráfico a analizar con el modelo de tráfico legítimo generado en el entrenamiento, y aplica las reglas generadas a partir de muestras de ataques para determinar si contiene algún tipo de amenaza para el sistema protegido. Este capítulo se organiza de la siguiente manera: en la Sección 6.1 se propone un nuevo método para la detección de malware en redes basado en el análisis de la carga útil del tráfico; en la Sección 6.1 se describe la experimentación realizada; y en la Sección 6.2 se discuten los resultados obtenidos.

## 6.1 Detección de malware en la carga útil

Esta sección describe las características fundamentales del sistema de detección de intrusiones APAP. Con este fin, se ha organizada en dos subsecciones: En la primera de ellas se introducen sus principios de diseño, destacando el cómo se han implementado las tecnologías N-gram y filtros Bloom, y la arquitectura general de la propuesta. En la segunda subsección se describen una a una sus etapas de tratamiento de información, abarcándose desde su etapa de entrenamiento y modelado, hasta su despliegue en el entorno de monitorización al operar en modo de detección.

### 6.1.1 Principios de diseño

APAP (del inglés *Advanced Payload Analyzer Preprocessor*) es un sistema de detección de malware en redes basado en el análisis de la carga útil del tráfico en busca de anomalías. Ha sido implementado como módulo preprocesador del popular NIDS Snort [Sno18]. Dado que este último se basa en la aplicación de reglas para la identificación de firmas de ataques, su combinación ofrece un esquema de detección de naturaleza híbrida. APAP incorpora dos tecnologías habituales en la bibliografía: la metodología N-gram [GOO14] para la extracción de información introducida en PAYL, y su almacenamiento a partir de filtros Bloom [RK15] propuesta en ANAGRAM. La primera permite el estudio en profundidad del contenido de la carga útil, mientras que la segunda reduce el tamaño de su representación en memoria y facilita su direccionamiento. A continuación se describe en detalle cómo han sido implementadas.

#### 6.1.1.1 Adaptación de N-gram

La técnica N-gram [GOO14, SVS+14] es una técnica habitualmente utilizada en el área de investigación que abarca el procesamiento del lenguaje natural. Su principal objetivo es la predicción, dentro de una sucesión de elementos, del siguiente elemento conociendo los anteriores y sus distintas probabilidades de aparición. Una de sus múltiples aplicaciones consiste en su implementación para resumir conjuntos de datos que de otra manera serían imposibles de analizar o almacenar, principalmente debido a las limitaciones tecnológicas relacionadas con su almacenamiento, velocidad de análisis mínima requerida o capacidad máxima de procesamiento.

La incorporación de N-gram a APAP conlleva la elección del valor n que determine el número de bytes de la dimensión que abarca cada una de sus estructuras. Por lo tanto, el sistema actúa como una ventana deslizante de tamaño $n$ que recorre todos los bytes de la carga útil del paquete. Cada n-gram extraído de un paquete, es decir, cada secuencia de $n$ bytes consecutivos que contenga el paquete, se procesa como si fuese una entidad propia. De este modo, el resultado de procesar un paquete es determinado por la unión de los resultados obtenidos de analizar cada uno de los n-grams en los que se pueda dividir. Esta técnica permite resumir el contenido de los paquetes y buscar dependencias y similitudes entre ellos. Las principales ventajas de su uso son su bajo consumo de recursos y facilidad de implementación. Sin embargo, el análisis de paquetes por esta técnica

a menudo requiere del tratamiento de una enorme cantidad de n-grams, por lo que es frecuente su complementación con técnicas auxiliares que permitan resumir la información a manejar, y de este modo faciliten su almacenamiento y procesamiento en tiempo real.

#### 6.1.1.2 Adaptación de filtros Bloom

Los filtros Bloom [RK15] son estructuras de datos probabilistas utilizadas para determinar si un dato pertenece a una colección de información o no. Sus principales características son su gran eficiencia, ahorro de memoria y no generación de falsos negativos (ver Sección 3.6.1 "Precisión"), logradas a costa de almacenar únicamente la información estrictamente necesaria para resolver ciertos problemas, en los que tan solo debe guardarse el registro de si un elemento ha sido observado con anterioridad o no. Su coste computacional es $\theta(k)$, y es completamente independiente del número de elementos de la colección a analizar. ANAGRAM adoptó por primera vez el uso de filtros Bloom como una mejora frente a PAYL para permitir trabajar con ventanas de tamaño n-gran más elevado. APAP también adapta esta técnica, desplegándola de la siguiente manera: cuando el detector procesa un n-gram, consulta las funciones de dispersión del filtro obteniendo un conjunto de posiciones que lo representen en su interior. En particular, la experimentación realizada considera funciones de dispersión HASH-MD5 [IET92], unas de las más habituales en la bibliografía. Cuando en las etapas de entrenamiento y detección de APAP es necesario determinar la presencia de un elemento en los modelos de uso, basta con que utilice las mismas funciones de dispersión. De esta forma, si al menos una de las posiciones a las que que se le asocie muestra el valor 0, puede deducirse que ese n-gram no ha sido detectado previamente. En el primer prototipo de APAP se consideró el despliegue de filtros Bloom convencionales. Sin embargo, esta aproximación fue descartada debido a que, al rellenar la estructura con información provista por grandes colecciones de muestras, con frecuencia se llegaba a un estado de sobreentrenamiento en el que la mayor parte de las posiciones del filtro almacenaban el valor 1. Como solución a este problema, en lugar de indicarse de manera binaria si cada n-gram ha sido observado o no, se almacena su frecuencia de aparición en el conjunto de muestras de entrenamiento. A esta modificación del filtro Bloom se la conoce habitualmente como filtros Bloom con conteo o CBF (del inglés *counting Bloom filter*) [SV14], y su adaptación no tiene precedentes en la familia PAYL.

Nótese que tanto el uso de un número de funciones hash inapropiadas como el operar sobre un espacio muestral demasiado grande pueden llevar a que el filtro Bloom genere errores de direccionamiento, que en el contexto de los filtros Bloom son denominados falsos positivos (no confundir con la terminología descrita en la Sección 3.6.1 "Precisión"). Esto se da cuando al consultar un registro del filtro se observa una colisión (del inglés *collision*), es decir, la función de dispersión redirige la búsqueda a dos o más posiciones diferentes, que presentan valores distintos. La colisión es debida a que el elemento buscado no ha sido visto con anterioridad, pero al registrarse previamente otra observación, se ha modificado alguna de las posiciones que la representan. Tal y como señalaron S. Geravand et al. [GA13], la probabilidad de emisión de falsos positivos se aproxima a partir de la siguiente expresión:

$$TFP = ((1 - 1 - \frac{1}{m})^{kn})^{k} \approx (1 - e^{\frac{-kn}{m}})^{k} \tag{6.1}$$

donde $n$ es el número de elementos a clasificar, $m$ es el número de bits que identifican las posiciones del filtro y $k$ es el número de funciones de dispersión. A partir de esta expresión es posible deducir que el número de funciones de dispersión óptimo adaptado a las características del sensor se obtiene de la ecuación:

$$K = \frac{m}{n} \times ln2 \tag{6.2}$$

Como alternativa a este método, también es posible considerar funciones hash universales, pero su uso a menudo conlleva el crecimiento de m, penalizando así el tamaño de la representación en memoria de la estructura. Por este motivo APAP considera la primera opción. Otro aspecto a tener en cuenta acerca del uso de filtros Bloom en el sistema propuesto, es el impacto de la dimensión de los n-gram que extraen la información de la carga útil. K. Rieck et al. revisaron en profundidad este problema [RL06] tomando como referencia propuestas de análisis de tráfico de red que también implementaron esta metodología. Advirtieron de que, a nivel puramente técnico, el uso de n-gram de gran tamaño incrementa el espacio de memoria que ocupa el filtro. Por ejemplo, en la experimentación realizada, trabajar con 1-gram requería $256 \times 2^8 (\approx 1KBytes)$ registros, 2-gram $65536 \times 2^{16} (\approx 262KBytes)$ registros, 3-gram $16777216 \times 2^{24} (\approx 64MBytes)$ registros, 4-gram $429496796 \times 2^{32} (\approx 16GBytes)$ registros, etc., por lo que es necesario evitar valores $n$ demasiado grandes. En la detección de intrusiones, la granularidad de las observaciones también repercute en la precisión a la hora de reconocer componentes maliciosos. Al considerar secuencias binarias grandes, es más probable que las intrusiones con escasa representación en la carga útil pasen desapercibidas. Esto reduce la tasa de acierto del sensor, pero también hace menos probable que muestras legítimas sean etiquetadas erróneamente como maliciosas, reduciéndose la sensibilidad de la detección. En el caso contrario, el sistema opera de manera mucho más restrictiva, asumiendo el riesgo de reportar una mayor cantidad de incidencias. El nivel de restricción con que debe operar el sensor depende directamente del escenario de monitorización y del valor de los activos a proteger. Según [RL06], para el análisis de tráfico con naturaleza principalmente HTTP, SMTP y FTP, que es a través del cual se suele realizar la trasmisión de malware, es recomendable la definición de n-gram de longitud 3. De este modo se consigue un equilibrio entre la capacidad del sensor con la optimización de su impacto en el sistema.

### 6.1.2 Etapas de procesamiento de la información

La arquitectura de APAP conlleva diferentes etapas de tratamiento de información, las cuales se agrupan en dos grandes niveles: entrenamiento y detección (ver Figura 6.1). En la fase de entrenamiento a su vez se distinguen cuatro tareas: inicialización, entrenamiento base, entrenamiento de referencia y definición de valores $K$. Durante su inicialización, APAP procede a la eliminación de la información de entrenamientos anteriores, el vaciado de filtros Bloom y el establecimiento de la función de direccionamiento (del inglés *hashing*).

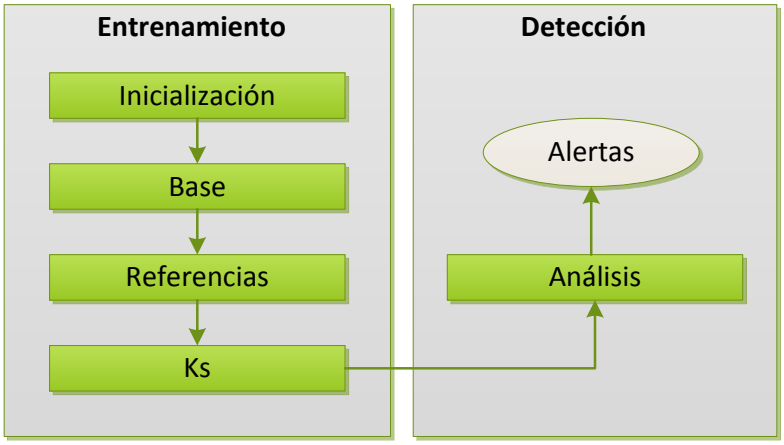



Figura 6.1: Etapas de procesamiento de información en APAP.

En la fase de entrenamiento base se rellena un CBF con información extraída de la carga útil del tráfico habitual y legítimo de la red a proteger. En concreto, se almacena la frecuencia de aparición de cada posible n-gram observable hasta que su contenido sea suficientemente representativo. Las siguientes dos etapas de entrenamiento establecen las mayores diferencias con sus predecesores. En primer lugar, se lleva a cabo el cálculo de las $K_s$ del sistema. Los valores $K$ son métricas que resumen el contenido de los filtros Bloom y facilitan la generación de reglas de detección. El siguiente paso es el cálculo de las puntuaciones de referencia que permiten decidir la naturaleza de los paquetes analizados. Una vez entrenado el sistema, APAP está capacitado para reconocer contenido malicioso en las próximas muestras a analizar. Cada vez que inspeccione un nuevo paquete, generará un conjunto de puntuaciones de referencia por medio de la comparación de las puntuaciones generadas en el modelo de uso legítimo de la red con las nuevas $K_s$ calculadas a partir del paquete a analizar. Cuando la prueba estadística de Wilconxon de los rangos con signo determine que la diferencia entre ellas no es representativa, el paquete es etiquetado como potencialmente malicioso. A continuación se describe en detalle cada uno de estos procesos

#### 6.1.2.1 Inicialización

En esta fase se inicializa las estructuras que son empleadas para representar el tráfico limpio y las anomalías de la red a proteger. Tanto los filtros Bloom como los diferentes valores $K$ reciben un valor inicial de 0 en cada posición.

##### 6.1.2.1.1 Entrenamiento base

En el entrenamiento base se construye el modelo que representa las características de la carga útil legítima que circula por la red a defender. Para su elaboración es necesaria una colección de muestras representativas de tráfico legítimo, es decir, trazas que correspondan al modo de uso habitual de esa red. Nótese que tal y como se demostró en [WPS06], el uso de filtros Bloom en esta fase del modelado reduce la problemática relacionada con pequeños errores de etiquetado en las muestras de referencia, los cuales son habituales en

las colecciones de dominio público típicamente consideradas en la bibliografía. Durante el entrenamiento base, el filtro Bloom que contiene la frecuencia de aparición de cada n-gram en la carga útil es rellenado por la información extraída de las muestras de referencia, hasta concluirse que su contenido es lo suficientemente representativo. Se asume que esto sucede en el momento en el que, al añadir nueva información no se producen cambios representativos en su distribución. A partir de ese momento el sistema es susceptible a ser sobreentrenado. El resultado de añadir nuevas observaciones es estimado tomando como referencia la media de los porcentajes de error entre los valores del filtro antes y después de incorporar nuevas observaciones. Dado el paso de entrenamiento $P$, el porcentaje de error medio $E$ entre el filtro Bloom antiguo y el filtro resultante del incorporar un nuevo conjunto de muestras, es definido mediante la expresión:

$$E = \sum_{i=0}^{T} \frac{|CBF(i) - CBF_{nuevo}(i)|}{CBF(i) + CBF_{nuevo}(i)} \times \frac{1}{T} \tag{6.3}$$

donde $T$ es el número total de registros del filtro, $CBF(i)$ es el valor en su posición $i$ tras los $P-1$ primeros pasos de entrenamiento, y $CBF_{nuevo}$ es el valor de su posición $i$ tras el paso $P$ de entrenamiento. En cada paso de entrenamiento el filtro es actualizado al considerar el valor de los n-gram de un subconjunto del total de trazas que integran la colección de muestras de referencia. En la Figura 6.2, se muestra un ejemplo de la evolución del parámetro $E$ en uno de los experimentos realizados. Para llevar a cabo dicha prueba se ha considerado una colección de muestras de tráfico legítimo cedida por el centro de cálculo de la Universidad Complutense de Madrid (UCM). Sus trazas fueron seccionadas en distintos bloques de 300.000 paquetes con un tamaño aproximado de 250 Mbytes. En la gráfica el eje Y representa el porcentaje de error medio, y el eje X el número de pasos del entrenamiento base llevados a cabo. El comportamiento del valor $E$ observado demuestra cómo el porcentaje de error medio tiende a disminuir, y cómo al llegar a un determinado valor, en concreto el 2%, se produce su saturación; esto es indicador del final del entrenamiento base, ya que la realización de nuevos pasos de entrenamiento no aporta información representativa al contenido del filtro.

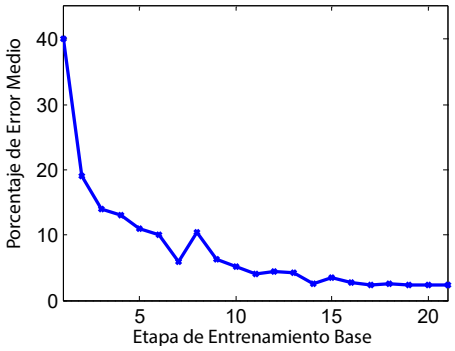



Figura 6.2: Ejemplo de evolución del error $E$ en el entrenamiento base.

Nótese que independientemente del tamaño de la ventana n-gram, el número de pasos máximos que pueden ejecutarse hasta alcanzar el punto de saturación está limitado por

los recursos del sistema. La Figura 6.3 muestra la gráfica que representa la evolución del valor máximo encontrado en el filtro Bloom que representa al tráfico legítimo de la figura anterior tras cada paso de entrenamiento. El eje Y indica el valor máximo encontrado en el filtro, mientras que el eje X representa los distintos pasos del entrenamiento. Es posible observar cómo crece desde un valor inicial de 3.166.438 hasta un valor final de 14.762.427 observaciones. El incremento es muy rápido, pero no parece suficiente para desbordar la capacidad de almacenamiento de las estructuras de datos enteros de la mayor parte de lenguajes de programación de alto nivel. Considérese que, por ejemplo, en el caso del lenguaje C, que es sobre el que ha sido implementado el sistema propuesto, el valor máximo representable en un tipo entero es 2.147.483.647, muy lejano de los obtenidos en los puntos de saturación. No obstante, es conveniente considerar que podrían existir modelos de tráfico con muy poca homogeneidad que requieran una gran cantidad de pasos para completarse. En este caso se recomienda operar directamente sobre datos estocásticos, ocupando cada posición del filtro la probabilidad de hallar el n-gram que representa en la carga útil de cada paquete que ha formado parte del conjunto de entrenamiento.

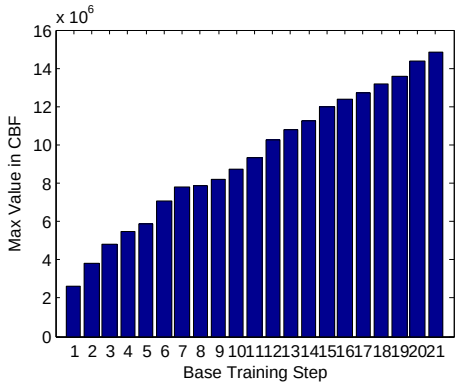



Figura 6.3: Ejemplo de evolución del máximo valor encontrado en filtro Bloom.

#### 6.1.2.1.2 Entrenamiento de referencias

El conjunto de $K$ de APAP es una representación simplificada del contenido del filtro Bloom. Los principales motivos que ha llevado a decidir su implementación son reducir la penalización en el rendimiento del sistema y su consumo de batería, derivados de la gran cantidad de memoria necesaria para su almacenamiento. Otra ventaja es la facilidad que ofrecen en la generación de reglas de detección y en la clasificación de las alertas emitidas. Con el uso de valores $K$, la investigación realizada sume las siguientes premisas:

- La información más útil para la detección de malware del contenido del filtro Bloom se obtienen a partir del análisis de la frecuencia de aparición de cada valor n-gram.
- El estudio de los valores máximos y mínimos de las frecuencias de aparición almacenadas en el filtro Bloom indican con mayor precisión las características de la carga útil monitorizada.
- El entrenamiento base podría introducir ruido en los modelos de uso. Por lo tanto,

debe asumirse cierta tolerancia al ruido para evitarse altas concentraciones de falsos positivos en la fase de detección.

A continuación se describe cómo estos valores son calculados, y extraídos a partir del entrenamiento de referencias.

### Cálculo de valores $K$

En el cálculo de los valores $K$, el filtro Bloom es considerado como una lista de los valores asociados a cada posible n-gram observados en las muestras de referencia durante la fase de entrenamiento de APAP. Suponiendo T posibles n-grams diferentes dentro del filtro, éste puede ser representarse como:

$$CBF = \ frec\,(0)\,,\ frec\,(1)\,,\ \ldots\ ,\ frec(T-1) \tag{6.4}$$

donde la frecuencia de aparición $frec\,(i)$ de cada uno de los posibles n-gram cumple $0 \leq i < T$. Las listas $ORD_{max}$ y $ORD_{min}$ contienen la misma información que el filtro Bloom, pero ordenada; en particular, cada posición de $ORD_{max}$ contiene:

$$ORD_{max}\,(i) = \ Pos_i,\ frec(i) \tag{6.5}$$

tal que $ORD_{max}\,(i),\ 0 \leq i < T$ es la posición de la lista $ORD_{max}$ que contiene el i-esimo valor del filtro con mayor frecuencia de aparición. Es decir, $ORD_{max}\,(0)$ contiene el valor que más veces ha aparecido, $ORD_{max}\,(1)$ contiene el segundo valor que más veces ha aparecido y $ORD_{max}\,(T-1)$ contiene el valor que menos veces ha aparecido. Análogamente, cada posición de $ORD_{min}$ contiene

$$ORD_{min}\,(i) = \ Pos_i,\ frec(i) \tag{6.6}$$

En analogía con el caso anterior, $ORD_{min}\,(i)$, $0 \leq i < T$ es la posición de la lista $ORD_{min}$ que contiene el i-esimo valor del filtro Bloom con menor frecuencia de aparición. Nótese que el análisis de $ORD_{min}$ es algo más complejo, ya que en muchas de las primeras posiciones pueden contener frecuencias de aparición nulas. Para evitar información irrelevante en el cálculo de los valores $K$, éstas no son insertarlas en la lista, de tal manera que $\forall\ ORD_{min}\,(i)$, $0 \leq i < T$ cumple $ORD_{min}\,(i) > 0$. Como alternativa a esta medida, es posible la definición de cotas mínimas que restrinjan los valores a insertar en sus primeras posiciones. No obstante, la implementación de este método ha sido pospuesta a futuras implementaciones de APAP. Con el fin de facilitar la comprensión de la aproximación realizada, de ahora en adelante se va a considerar una lista ordenada genérica $ORD$ que indistintamente puede ser $ORD_{min}$ u $ORD_{max}$. Es importante resaltar que, a la hora de llevar a cabo el diseño, se ha de elegir entre trabajar con los valores máximos o los valores mínimos; el uso de ambos genera una gran cantidad de inconsistencias, y por lo tanto conlleva la emisión de tasas mayores de falsos positivos. Téngase en cuenta que, en los experimentos realizados, considerar valores máximos ha resultado ser más preciso que el uso de valores mínimos.

Una vez ordenadas las frecuencias de aparición, en el peor de los casos la dimensión de

su lista es $T$. Esto quiere decir que, por ejemplo, al operar con una ventana n-gram de dimensión 3, serán organizados los valores de $T = 2^{8\times3}$ posiciones, asumiéndose un byte por cada ranura en los n-gram. Es evidente que considerar una representación tan grande no es recomendable en términos de rendimiento, y por lo tanto se requiere de su simplificación. Una primera posible representación reducida de estas estructuras es considerar únicamente las primeras posiciones de cada lista. De esta manera los valores $K$ se definirían como:

$$K_i = ORD(i) \tag{6.7}$$

siendo $0 \leq i < Cota$. El valor $Cota$ establece cuántas de las primeras posiciones son tenidas en cuenta. Con esto no solamente se reduce el problema de cómputo; también se mitiga el efecto del ruido, dado que las últimas posiciones de $ORD$ son apenas representativas y potencialmente agrupan una mayor cantidad de elementos. Sin embargo, las primeras posiciones tienden a ser similares en muestras del tráfico de la misma red, independientemente de si son legítimas o maliciosas. Para considerar un espectro mayor de frecuencias, APAP considera la relación entre valores de la lista separados de manera exponencial respecto al resto de la sucesión, de tal manera que se tenga en cuenta el valor $i$ de la siguiente sucesión en intervalos exponenciales:

$$i = \ 2^p \ , \ \exists p \ \in [0, \ \log_2 T) \tag{6.8}$$

donde para cada valor $K_i$ se tienen en consideración todos los valores anteriores y se suaviza el efecto del ruido gracias al cálculo promedio. Los valores $K$ calculados tanto en el entrenamiento de referencias, entrenamiento de $K_s$ definitivas o en la fase de detección corresponden con las siguientes expresiones:

$$K_1 \ = Ord_0 \tag{6.9}$$

$$K_2 \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 1 \tag{6.10}$$

$$K_4 \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 3 \tag{6.11}$$

$$K_8 \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 7 \tag{6.12}$$

$$K_{16} \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 15 \tag{6.13}$$

$$K_{32} \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 31 \tag{6.14}$$

$$K_{64} \ = \sum_{i=0}^{n} Ord_i \times \ \frac{1}{n}, \ n = 63 \tag{6.15}$$

$$K_{128} = \sum_{i=0}^{n} Ord_i \times \frac{1}{n},\ n = 127 \tag{6.16}$$

$$\dots$$

$$K_i = \sum_{i=0}^{n} Ord_i \times \frac{1}{n},\ n = i - 1 \tag{6.17}$$

Lo que puede resumirse por la siguiente definición:

$$k_i = \sum_{i=1}^{N} \frac{Ord_i \times ngram}{i} \tag{6.18}$$

Nótese que en la experimentación se han implementado los valores:

$$N : \{|1 \dots 10|, 12,\ 14,\ 16,\ 32,\ 64,\ 96,\ 128\} \tag{6.19}$$

y por lo tanto se consideraron de $K_1$ a $K_{17}$

VALORES $K$ EN EL ENTRENAMIENTO DE REFERENCIAS

El entrenamiento de referencias se centra en el estudio de los valores $K$ calculados a partir del filtro Bloom que contiene las muestras de tráfico legítimo. En esta fase se genera un contenedor que almacena una lista $ORD_{min}$ u $ORD_{max}$ (dependiendo del ordenamiento considerado) que represente la acumulación de cada valor $K_i$ en los distintos paquetes del tráfico legítimo. Por lo tanto, se llevan a cabo las siguientes dos acciones:

1. Para cada paquete se generan sus $K_s$ de la forma previamente explicada.

2. Se construye una lista $ORD_{min}$ u $ORD_{max}$ que resuma todas las $K_s$ generadas para los distintos paquetes. En consecuencia, se genera una lista ordenada para los distintos valores $K_0$, otra para los distintos valores $K_1$, etc... y la sucesión sigue hasta una lista que almacene los valores $K_{128}$.

Una vez concluidos estos pasos se dispone de la información necesaria para generar reglas a partir de las trazas de ataques del entrenamiento de valores $K$ definitivos. En la Figura 6.4 se ilustra un ejemplo del cálculo de valores $K$ en el entrenamiento de referencias. En particular, se extraen $K_1$ y $K_2$ en dos situaciones distintas, en ambos casos partiendo del mismo filtro Bloom rellenado en el entrenamiento base. Tras el análisis del $Paquete1$ y el $Paquete2$, se observa la frecuencia de aparición de los n-grams no nulos. Para cada uno de los paquetes, se muestran los valores $K_1$ y $K_2$, fácilmente deducibles aplicando la definición de Ks anteriormente descrita.

En la Figura 6.5 se muestran los resultados obtenidos en el entrenamiento de referencia para $K_1$ a partir del conjunto de muestras de tráfico legítimo. Es fácilmente observable que la frecuencia de aparición del conjunto de valores 26, 31, 36, 41 y 51 es especialmente representativa. En esta gráfica los valores aún no han sido ordenados en función de su frecuencia.

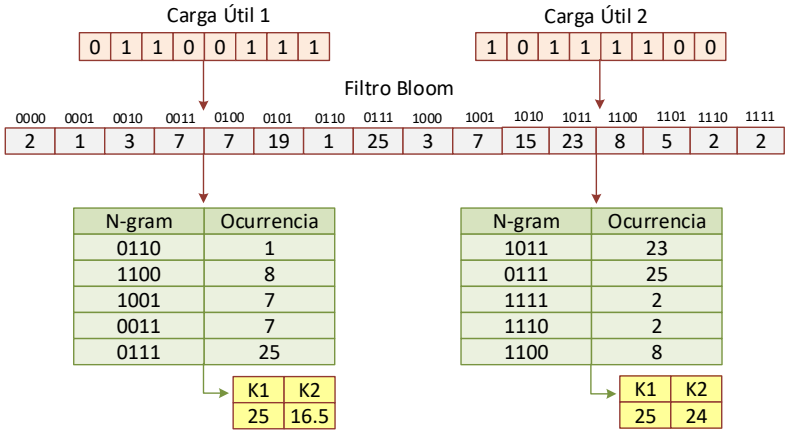



Figura 6.4: Ejemplo de generación de $K_1$ y $K_2$.

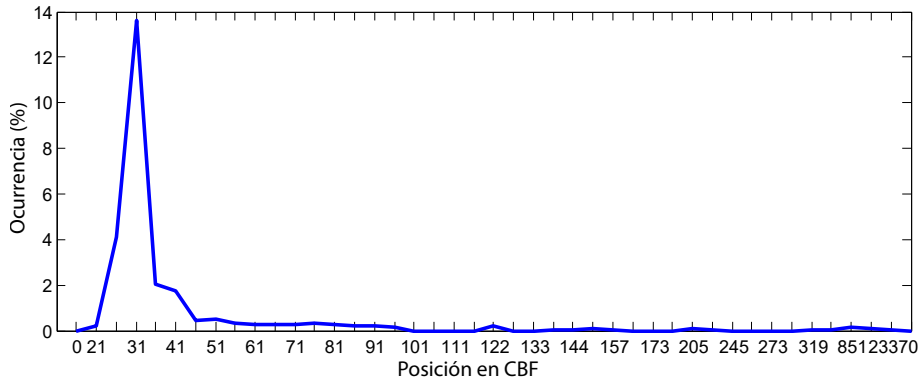



Figura 6.5: Ejemplo de espectro de aparición de $K_1$ en el entrenamiento de referencias.

#### 6.1.2.1.3 Definición de valores $K$

Una vez establecidos los valores $K$ de referencia que resumen el contenido del filtro Bloom rellenado en el entrenamiento de referencia, se construyen las reglas que permiten la identificación de anomalías potencialmente maliciosas. Con el fin de refinar los valores $K$ iniciales, en el entrenamiento de $K_s$ definitivas se considera una colección de muestras de trazas de ataques. El primer paso que realizar, es la generación de las propias $K$ del ataque en función del filtro rellenado en la etapa anterior. Nótese que la manera de generar los valores $K$ para intrusiones es exactamente la misma que con la que se generaron las $K_s$ de referencia explicadas en el apartado anterior. A continuación, se identifican los valores de las $K_s$ de referencia asociados a las posiciones del filtro Bloom cuyos valores $K$ han sido más representativos del ataque. La idea es generar hasta una nueva regla por cada traza de ataque. Las cuáles son constituidas por los valores de dichas $K_s$ de referencia y su frecuencia de aparición. Para evitar confusiones, es importante resaltar que los valores de las $K_s$ definitivas son valores de las $K_s$ de referencia, y que la extracción de las $K_s$ del ataque se realiza para establecer las posiciones de las $K_s$ de referencia que constituirán dichas reglas, y por lo tanto de mayor relevancia. De esta manera son acotados los umbrales de decisión que delimitan el conjunto de tráfico legítimo y el de anomalías. Nótese que el objetivo de incorporar la frecuencia de aparición en las reglas es optimizar el tiempo

de ejecución dedicado a la detección de anomalías en tiempo real. Asimismo, esta técnica permite actualizar rápidamente instancias de APAP ya implementadas por medio de la carga de conjuntos de reglas nuevas.

### Puntuación de la similitud entre paquetes

El proceso de generación de nuevas reglas asociadas a una traza de tráfico malintencionada y al entrenamiento de referencias del sistema, asume las siguientes premisas:

- Cada paquete a analizar debe presentar una puntuación de similitud que permita su comparación con el resto de paquetes.
- Las partes más representativas del malware se concentran en determinadas partes de la carga útil. Los paquetes con puntuaciones más representativas son los que más conviene tener en cuenta en la elaboración de nuevas reglas.
- La puntuación del paquete debe considerar la información contenida en los valores K definitivos que ha generado.
- Las nuevas reglas se definen a partir de los valores k de referencia y establecen los umbrales que distinguen el tráfico anómalo del legítimo. El tráfico del ataque con el que se generan es el encargado de determinas las Ks de referencia que lo constituyen.

En base a esto, el primer problema a resolver es la definición del nivel de significancia de cada paquete en las trazas del ataque con el que es entrenado el sistema. Dada una traza de $m > 0$ paquetes con contenido malicioso, para cada uno de ellos se precisa calcular su puntuación de similitud respecto al modelo de uso legítimo del sistema. Sea la traza maliciosa $A = A_0, A_1, \cdots, A_{m-1}$ donde $A_i$ es el paquete en su posición $i$, y sus puntuaciones de similitud $P = P_0, P_1, \cdots, P_{m-1}$, $0 \leq i < m$. El valor $P_i$ se calcula a partir de la lista ordenada de menor a mayor $ORD^j_{min}$, de las frecuencias de aparición de $K_j$, j$\in 0, 1, 2, 4, 8, 16, 32, 64, 128$ generada a partir de los distintos valores $K$ de referencia generados para los paquetes del tráfico de entrenamiento legítimo. Nótese que si la implementación de APAP opera con valores máximos se utiliza $ORD^j_{max}$ en sustitución de $ORD^j_{min}$. Es prácticamente la misma lista, pero ordenada de mayor a menor, cubriendo así el paradigma opuesto de diseño. Para no discriminar ninguna de las dos opciones, a partir de ahora estas listas son generalizadas como $ORD^j$.

Tras generar los valores $K$ correspondientes al paquete $A_i$, es importante conocer lo representativos que son cada uno de sus parámetros $K$, y de esta manera poder compararlos con el resto de paquetes de $A$. Para ello se definen variables nuevas, cada una de ellas asociada a cada uno de los $K_s$ calculados para el paquete $A_i$. Dichas variables son denominadas $posicion^j$, y se calculan de manera diferente en función del paradigma de diseño:

- $posicion^j$ (con mínimos): Posición de la lista $ORD^j_{min}$ a partir de la cual todos los valores restantes son mayores que el valor de la $K_j$ generada por el paquete $A_i$.

- $posicion^j$ (con máximos): Posición de la lista $ORD^j_{max}$ a partir de la cual todos los valores restantes son menores que el valor de la $K_j$ generada por el paquete $A_i$.

Para establecer la puntuación de $A_i$ se calcula la suma de todas las puntuaciones parciales generadas para cada posible pareja de $K_s$ $p$,$q$, denominándose puntuación parcial de similitud entre $p$ y $q$ $PP_{(p,q)}$ a la puntuación obtenida mediante la expresión:

$$PP_{(p,q)} = \left( \left| posicion^i - posicion^j \right| + 1 \right) \times ((posicion^i \times posicion^j) + 1) \tag{6.20}$$

Esta métrica tiene como finalidad calcular parte de la puntuación de similitud penalizando las parejas de $K_s$ con mayor/menor contenido en su lado derecho de las listas. La puntuación del paquete corresponde con la suma de todas sus sumas parciales, es decir la suma de todas las posibles parejas $p, q$ que se puedan generar. Por lo tanto, cuanto más alto/bajo sea su valor, menos representativo es el paquete, ya que más elementos quedan a la derecha de sus ks en las listas del entrenamiento de referencias.

Generación de nuevas reglas

La generación de nuevas reglas sigue los siguientes pasos:

1. Se calcula la puntuación de similitud de cada uno de los paquetes correspondientes al conjunto de entrenamiento del ataque a partir del que se vaya a generar la nueva regla.

2. Una vez localizado el mejor valor se genera una nueva regla a partir de él. La regla está formada por parejas de valores correspondientes a cada una de las $K_s$ definitivas, donde cada $K_j$ definitiva viene dada por la frecuencia de acumulación localizada en $posicion^j$. De esta manera quedan establecidas las distinciones entre el tráfico legítimo y las anomalías.

En la Figura 6.6 se muestra parte de las reglas generadas por APAP tras finalizar el entrenamiento de $K_s$ definitivas para el conjunto de muestras de los ejemplos anteriores. Cada línea corresponde a una regla, y las columnas corresponden a los valores comprendidos entre $K_1, \cdots, K_7$ y su correspondiente frecuencia de aparición.

#### 6.1.2.2 Detección

En la fase de detección APAP analiza el tráfico entrante paquete por paquete. Para cada uno de ellos genera un filtro Bloom auxiliar copia del generado en el entrenamiento base, insertando las nuevas observaciones. Esto permite determinar sus propias $K_s$. Una vez generadas, son comparadas con las $K_s$ guardadas en las reglas de APAP generadas en la fase de entrenamiento de definición de valores $K$. Si la puntuación asociada a la frecuencia acumulada por alguna de ellas es superada por el de alguna el de las reglas de detección, se desencadenará una alerta. Por lo tanto, su carga útil es considerada anómala y potencialmente maliciosa. En el Pseudocódigo 2 se resume el comportamiento de APAP.

```
58.00 0.0000002186 63.00 0.0000002187 65.25 0.0000002187 67.88 0.0000002189 71.31 0.0000002242 75.09 0.0000009728
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 60.62 0.0000002184 70.31 0.0000002194 74.81 0.0000008748
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 62.38 0.0000002185 70.44 0.0000002196 75.31 0.0000010654
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 62.50 0.0000002185 70.81 0.0000002212 75.84 0.0000018923
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 61.00 0.0000002185 70.38 0.0000002194 75.47 0.0000011805
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 62.25 0.0000002185 70.56 0.0000002198 75.38 0.0000011053
68.00 0.0000002189 70.50 0.0000002219 71.00 0.0000002236 71.25 0.0000002301 73.12 0.0000006644 76.62 0.0000034075
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 63.38 0.0000002186 71.56 0.0000002290 77.56 0.0000054082
54.00 0.0000002184 61.00 0.0000002186 66.25 0.0000002188 70.12 0.0000002223 74.38 0.0000009592 78.50 0.0000097856
59.00 0.0000002186 58.50 0.0000002185 63.75 0.0000002186 67.12 0.0000002187 69.94 0.0000002191 74.44 0.0000006977
69.00 0.0000002191 68.00 0.0000002189 68.25 0.0000002194 68.88 0.0000002198 70.38 0.0000002195 73.91 0.0000004873
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 63.38 0.0000002186 70.88 0.0000002216 75.56 0.0000013151
37.00 0.0000001093 46.50 0.0000001640 53.75 0.0000002184 64.12 0.0000002186 71.31 0.0000002242 76.81 0.0000039090
37.00 0.0000001093 46.50 0.0000001640 53.75 0.0000002184 64.00 0.0000002186 70.75 0.0000002209 75.91 0.0000020566
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 56.25 0.0000002184 65.38 0.0000002186 74.25 0.0000006127
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 61.38 0.0000002185 70.94 0.0000002221 76.97 0.0000041979
37.00 0.0000001093 46.50 0.0000001640 52.50 0.0000002184 63.38 0.0000002186 70.56 0.0000002200 74.66 0.0000007977
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 60.38 0.0000002184 69.94 0.0000002191 74.62 0.0000007798
37.00 0.0000001093 46.50 0.0000001640 53.75 0.0000002184 64.12 0.0000002186 71.62 0.0000002296 76.84 0.0000039880
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 62.38 0.0000002185 70.50 0.0000002197 75.00 0.0000009472
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 61.12 0.0000002185 70.00 0.0000002191 74.41 0.0000006870
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 61.25 0.0000002185 71.06 0.0000002224 77.69 0.0000059034
58.00 0.0000002186 63.00 0.0000002187 65.25 0.0000002187 68.00 0.0000002189 72.12 0.0000003677 76.25 0.0000028178
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 62.50 0.0000002185 70.88 0.0000002216 76.47 0.0000032776
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 59.00 0.0000002184 66.69 0.0000002187 74.41 0.0000006803
37.00 0.0000001093 46.50 0.0000001640 53.75 0.0000002184 64.25 0.0000002186 70.94 0.0000002220 75.66 0.0000014913
37.00 0.0000001093 46.50 0.0000001640 52.50 0.0000002184 62.75 0.0000002186 70.38 0.0000002194 75.09 0.0000009728
37.00 0.0000001093 46.50 0.0000001640 51.50 0.0000001921 60.12 0.0000002184 69.88 0.0000002191 75.06 0.0000009640
```

Figura 6.6: Ejemplo de reglas de detección generadas por APAP.

**Algoritmo 2:** Modos de ejecución de APAP.

**Entrada:** Cada paquete $P_i$ con carga útil binaria $B_1, \cdots, B_n$ monitorizado en la red protegida. El dataset $D$ formado por $L : L_1, \cdots, L_m$ de trazas legítimas, y $M : M_1, \cdots, M_r$ trazas maliciosas. La configuración $C$ en que actúa el sensor.

**Salida :** Etiquetado de $P_i$ como "legítimo" o "sospechoso". CBF resultante de entrenamiento. Conjunto de valores $K_s$ definitivos.

**Si** $C = 0$ **entonces**
| Inicialización de estructuras de datos y borrado de CBF;
**Fin**
**Si** $C = 1$ **entonces**
| **Mientras** $i < m$ *Y Error en entrenamiento base no se satura* **hacer**
| | Entrenamiento_Base($L_i$, CBF);
| **Fin**
**Fin**
**Si** $C = 2$ **entonces**
| **Mientras** $i < r$ **hacer**
| | Entrenamiento_Referencias($M_i$,CBF);
| | Generar_$K_s$(CBF);
| **Fin**
**Fin**
**Si** $C = 4$ **entonces**
| Detección($P_i$, CBF, $K_s$);
| devolver etiquetado;
**Fin**

## 6.2 Experimentación

La necesidad de evaluar NIDS ha llevado a que en las últimas décadas se hayan propuesto diferentes metodologías y colecciones de muestras. La mayor parte de ellas se basan en medir la precisión del sistema por medio del análisis de diferentes colecciones de trazas de tráfico de dominio público, con contenido legítimo y malicioso previamente etiquetado. Las muestras legítimas típicamente son capturas de tráfico habitual de redes. Por otro lado, las muestras de ataques pueden contener registros de intrusiones reales (KDDcup'99 [KDD99], DARPA'99 [Lab99], CAIDA [CAI18], etc.) o de tráfico generado por herramientas que tratan de imitar las actividades perpetradas por ataques (envío de malware convencional, ofuscación, inundación [BSMT14], etc.). Tradicionalmente, el esquema de evaluación predominante en la bibliografía es la colección KDDcup'99, considerada estándar funcional por la comunidad investigadora. KDDcup'99 provee muestras de diferentes tipos de intrusiones (DDoS, escáner de puertos, escalada de privilegios, etc.) y tráfico legítimo, caracterizadas por 41 métricas de diferente naturaleza (protocolo, puerto, servicio, duración de la sesión, número de accesos, etc.). Pero no es compatible con NIDS basados en el análisis de carga útil, ya que no provee el contenido binario de su carga útil. Como alternativa a KDDcup'99, las diferentes propuestas que integran la familia PAYL han sido evaluadas a partir de DARPA'99, considerada un estándar funcional en este contexto. DARPA'99 provee las capturas de tráfico reales a partir de las que fue construida KDDcup'99 en formato *tcpdump*, el cual sí que incluye toda la información necesaria para su análisis. Con el fin de facilitar la comparativa de la eficacia de APAP con la de publicaciones similares, nuestro estudio aplica en primera instancia esta aproximación.

Nótese que el uso de métodos de evaluación clásicos, como KDDcup'99 o DARPA'99 a menudo suscita controversia. Las muestras que contienen generalmente presentan diferentes características que haces que los resultados que arrojen no sean del todo escalables a la actualidad, como por ejemplo la falta de heterogeneidad de sus muestras, presencia de técnicas demasiado antiguas de intrusión o inconsistencias en las capturas de tráfico [HSB+12]. En consecuencia, para probar la eficacia de las nuevas aproximaciones también es habitual el uso de colecciones de capturas propias de tráfico de redes actuales, a cuya carga útil es posible acceder legalmente (las diferentes políticas internacionales de protección de datos son muy estrictas en lo que concierne a la divulgación de este tipo de información [SBPC14]). En sincronía con el resto de publicaciones recientes, APAP también ha sido evaluado por una colección de capturas de tráfico real capturadas en la red de la Universidad Complutense de Madrid (UCM). A continuación se describe cada uno estos experimentos.

## 6.3 Evaluación con DARPA'99

La colección de muestras DARPA'99 [Lab99] fue publicada por el grupo de investigación *Cyber systems and Technology Group* (originalmente *Information Systems Technology Group*) del Laboratorio Lincoln del Instituto de Tecnología de Massachusetts (MIT),

bajo el patrocinio de DARPA y el laboratorio AFRL/SNHS de las fuerzas aéreas de Estados unidos en el año 1999. La colección de muestras DARPA'99 basa su evaluación de sensores en dos colecciones de muestras: *online* y *offline*. Para la evaluación *offline* se entrena el sensor en base a capturas de tráfico llevadas a cabo en 7 días, separadas en sesiones etiquetadas cono normales o ataques. Nótese que al evaluar APAP, las primeras se utilizaron en el entrenamiento base, y las segundas en el entrenamiento de referencias y la definición de valores $K$. DARPA'99 provee además un conjunto de muestras adicional para evaluar el sensor, las cuales fueron capturadas durante dos semanas.
En términos generales, las muestras de DARPA'99 contienen registros de red (originalmente en formato *tcpdump*, en la evaluación de APAP convertidos a *pcap*) llevados a cabo en entornos UNIX, que incluyen un tráfico base (normal) relacionado con actividades marcadas por los protocolos/servicios HTTP, X Windows, SQL, SMTP, DNS, FTP, POP3, Finger, Telnet, IRC, SNMP o Time. El tráfico fue generado por cientos de equipos de red simulados, los cuales trataban de emular patrones estadísticos de tráfico legítimos en función de diferentes criterios, a los que se unieron más de 10,000 envíos de correos electrónicos adecuadamente anonimizados, con origen/destinatario en direcciones pertenecientes a dominios públicos. Cabe destacar que algunos extremos de comunicaciones preservaron direccione de red fijas, mientras que otros cambiaron a lo largo del tiempo. Al mismo tiempo, usuarios legítimos realizaba tareas más complejas que involucraron desde tareas ofimáticas, hasta la instalación/desinstalación de software. Las trazas de tráfico con contenidos maliciosos fueron adquiridas mediante la inyección de ataques desde 120 focos, distinguiéndose un total de 38 categorías de intrusiones [AAL+03], las cuales son resumidas en Tabla 6.1. Los resultados de la eficacia del sensor sobre estas muestras determinan la precisión del sensor, arrojando su tasa de acierto y de falsos positivos (ver Sección 3.6.1 "Precisión"). En [HSB+12] se ilustra su modo de empleo, en el que, de manera similar, las propuestas PAYL, ANAGRAM, POSEIDON y McPAD son evaluadas, estableciéndose así un referente con el que comparar el sistema propuesto.

## 6.4 Evaluación con tráfico HTTP real

Para el entrenamiento y la comprobación de los resultados de APAP, ha sido elaborado un conjunto de muestras de tráfico real a partir de capturas proporcionadas y etiquetadas por el Centro de Cálculo y Procesamiento de Datos de la Universidad Complutense de Madrid (UCM). Esta colección consta de un conjunto de trazas de tráfico limpio y otras de tráfico portador de código malicioso en su carga útil. Para la creación de las muestras de tráfico limpio se monitorizó el tráfico correspondiente a la facultad de informática de la UCM durante varios días en diferentes periodos de tiempo a lo largo del año 2011, en total habiéndose capturado 1.9Gbs de tráfico. La versión final de dataset quedó completada el día 24 de abril del año 2012, dividiéndose su contenido en muestras de 5 minutos (el cual fue su intervalo de muestreo). En total se analizó la carga útil de 2187388 paquetes, habiendo sido almacenada en formato pcap sin anonimizar. No se llevaron a cabo tareas de limpiado ni preprocesamiento, por lo que sus contenidos son exactamente los mismos que se observaron en la red. Las trazas de tráfico capturadas contienen actividades habituales

Tabla 6.1: Contenidos maliciosos en DARPA'99.

| Intrusión | Tipos | Instancias | Solaris | SumOS | Linux |
|---|---|---|---|---|---|
| DoS | 11 | 43 | Back,Neptune, Ping of death, Smurf, syslog, Land, apache2, Mailbomb, Process table, UDP storm | Back,Neptune, Ping of death, Smurf, Land, apache2, Mailbomb, Process table, UDP storm | Back,Neptune, Ping of death, Smurf, teardrop, Land, apache2, Mailbomb, Process table, UDP storm |
| Remoto a local | 14 | 11 | Dictionary, ftp-write, guest, phf, http tunnel, xlock,xsnoop | Dictionary, ftp-write, guest, phf, http tunnel, xlock,xsnoop | Dictionary, ftp-write, guest, imap, phf, named, http tunnel, sendmail, xlock,xsnoop |
| Usuario a root | 7 | 38 | Eject, ffbconfig, Fdformat,ps | Loadmodule, ps | Perl,xterm |
| Monitorización/Probe | 6 | 22 | Eject, nmap, Port sweep, Satan, mscan, saint | Eject, nmap, Port sweep, Satan, mscan, saint | Eject, nmap, Port sweep, Satan, mscan, saint |

perpetradas por sus usuarios de la Facultad de Informática, entre ellas:

- Intercambio de ficheros P2P.
- Envío de archivos adjuntos de diferentes formatos (.doc, .pdf, .mp3, .jpg, etc.) por correo electrónico.
- Navegación web
- Descarga de programas y aplicaciones vía HTTP y FTP
- Acceso a contenidos multimedia como video o música.

La captura de muestras maliciosas se ha llevado a cabo por medio del envío de especímenes de malware a una subred aislada del espacio público y de acceso restringido (la lista completa se muestra en la Tabla 6.2). Los paquetes que contienen una misma amenaza han sido identificados y combinados en una misma traza. Nótese que en la sección 7.3 "experimentación" se provee una descripción detallada de sus contenidos, siendo clasificados en base a diferentes criterios.
Éstas son clasificadas en las siguientes categorías:

- *Virus.* Malware programado con el fin de irrumpir en el funcionamiento normal del sistema víctima sin el permiso o el conocimiento de los usuarios.
- *Gusanos* (del inglés *worms*). Malware con capacidad de auto-replicación.
- *Nukers.* Malware orientado a la ejecución de ataques de denegación de servicio en redes TCP/IP.

Tabla 6.2: Lista de malware en la experimentación.

| | | |
|---|---|---|
| Backdoor.Buttman | I-Worm.Sobig.b | Parity.a |
| Backdoor.DonaldDick.15 | I-Worm.Sobig.f | Parity.b |
| Backdoor.DonaldDick.152 | I-Worm.Swen | PingPong.a |
| Backdoor.SdBot.aa | I-Worm.Tanatos.b | asser.b |
| Backdoor.Zenmaster.102 | I-Worm.Tanatos.dam | sobig |
| BitchSlap | I-Worm.Tettona | Stoned.a |
| blaster | Joke.Win32.Errore | Trojan.JS.Seeker.o |
| Bloodlust | Joke.Win32.Zappa | Trojan.PSW.Hooker.24.h |
| Bomberman | JS.Fortnight.b | Trojan.W32.PWS.Prostor.A |
| Bye | JS.Trojan.Seeker.b | Trojan.Win32.DesktopPuzzle |
| Click 2 | JS.Trojan.Seeker-based | Trojan.Win32.VirtualRoot |
| Die 3 | Junkie.1027 | Trojan.ZipDoubleExt-1 |
| DnDdos | loveletter | Win32.FunLove.4070 |
| Form.a | Macro.Office.Triplicate.c | Win32.HLLP.Hantaner |
| Gimp | Macro.Word.Cap | Win32.Xorala |
| happy99 | Macro.Word97.Ethan | Win95.Dupator.1503 |
| HDKP4 | Macro.Word97.Marker.r | Worm.Bagle.AG |
| IIS-Worm.CodeRed.a | Macro.Word97.Thus.aa | Worm.Mydoom.AS |
| IIS-Worm.CodeRed.c | melissa | Worm.Mytob.IV |
| I-Worm.Fizzer | MMR | Worm.Mytob.V. |
| I-Worm.Klez.e | MXZ II | Worm.P2P.SdDrop.c |
| I-Worm.LovGate.i | MXZ | Worm.SomeFool.Q |
| I-Worm.Mimail.a | NetBus 2.0 Pro | Worm.Win32.Fasong.a |
| I-Worm.Moodown.b | NetDevil 1.5 | Worm.Win32.Lovesan.a |
| I-Worm.Mydoom.a | netsky.z | Worm.Win32.Muma.c |
| I-Worm.NetSky.d | nimda | Worm.Win32.Opasoft.a.pac |
| I-Worm.Rays | NYB | WYX.b |
| I-Worm.Sober.c | orm.Mytob.BM-2 | Zip Monsta |
| I-Worm.Sober.c.dat | OwNeD | |
| HLLC.Crawen.8306 | Macro.Word97.Marker-based | Worm.Bagle.Z |

- *Troyanos* (del inglés *trojans*). Malware que facilita el acceso no autorizado al sistema infectado. Habitualmente es gestionado de manera remota y permite el control a distancia del sistema víctima.
- *Macros*. Malware desarrollado en lenguajes de macro, y que por lo tanto permiten programar y automatizar pequeñas tareas en formatos de archivo típicamente relacionados con la ofimática.

Para realizar el entrenamiento, y posteriormente probar la eficacia de APAP, se aplica el método estadístico de validación cruzada [Koh18]. La idea general de esta estrategia consiste en dividir las muestras de entrenamiento en un número determinado de particiones y probar todas con todas. De este modo se garantiza la independencia de los resultados respecto a la repartición realizada de muestras para el entrenamiento y análisis. El método consiste en repetir y calcular la media aritmética obtenida de las métricas obtenidas a

partir de diferentes particiones. Para cada una de las $N$ iteraciones se realiza la evaluación de la precisión del sistema por medio del cálculo del error de etiquetado. El resultado final se obtiene realizando la media aritmética de los $N$ valores de errores obtenidos, según:

$$E = \frac{1}{N} \sum_{i=1} E_i \tag{6.21}$$

En la experimentación con APAP se llevó a cabo dividiendo las trazas UCM que contienen ataques en 4 partes iguales (A, B, C y D) y realizando 4 experimentos, considerando en cada prueba tres grupos para el entrenamiento y uno para la detección, tal que:

- Entrenamiento con los conjuntos B, C, D y detección con conjunto A.
- Entrenamiento con los conjuntos A, C, D y detección con conjunto B.
- Entrenamiento con los conjuntos A, B, D y detección con conjunto C.
- Entrenamiento con los conjuntos A, B, C y detección con conjunto D.

## 6.5 Resultados

Esta sección describe los resultados obtenidos en la experimentación realizada. En primer lugar, y con el fin de facilitar la comprensión del método propuesto, se ilustra una serie de gráficas con información acerca del rellenado de los filtros Bloom y la generación de valores $K$. A continuación se describe la eficacia de APAP al actuar sobre la colección de muestras DARPA'99, lo que permite su comparativa con trabajos previos. Finalmente se discute su eficiencia al operar sobre tráfico real cedido por el Centro de Cálculo y Procesamiento de Datos de la Universidad Complutense de Madrid (UCM).

## 6.6 Ejemplo de distribución de valores $K_s$

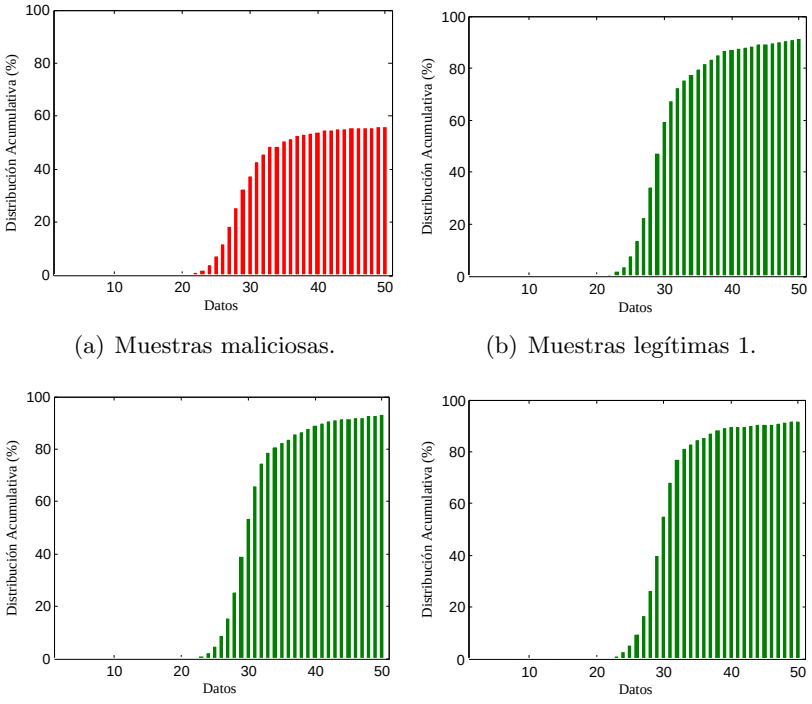
A continuación se muestra una serie de gráficas en las que se ilustran los fundamentos básicos de APAP. De esta manera puede apreciarse cómo los paquetes que presentan contenido malicioso tienen mayor cantidad de valores con $K_s$ significativamente bajos. Esta diferencia particularmente se acentúa para las $K_s$ en posiciones intermedias, puesto que para las $K_s$ en los extremos de la lista ordenada (ya sean tanto sus primeros como sus últimos elementos) las diferencias se reducen, aunque siguen siendo apreciables. También es posible observar (este hecho se da prácticamente en igual medida para todas las $K_s$) que al analizarse carga útil legítima, más del 90% de las frecuencias de aparición se ubican en las primeras 48 posiciones de la lista. Por el contrario, el estudio de tráfico malicioso demuestra que esas posiciones agrupan menos del 60% de las observaciones, hecho que habitualmente es debido a que el malware típicamente forma parte de aplicaciones legítimas que han sido corrompidas, dando pie a una mayor acumulación de observaciones residuales. En la Tabla 6.3 se muestra la regla de detección que condiciona el resto de esta sección. Las pruebas realizadas consideran un conjunto de muestras maliciosas de un mismo espécimen,

y tres colecciones de tráfico legítimo de referencia de la red protegida. A continuación se describen los resultados obtenidos:

Tabla 6.3: Regla de detección aplicada en ejemplos de valores $K_s$.

| $K_1$ | $K_2$ | $K_3$ | $K_4$ | $K_5$ | $K_6$ | $K_7$ | $K_8$ |
|---|---|---|---|---|---|---|---|
| 28.05 | 28.55 | 29.05 | 29.30 | 29.65 | 29.89 | 30.06 | 30.43 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $K_9$ | $K_{10}$ | $K_{11}$ | $K_{12}$ | $K_{13}$ | $K_{14}$ | $K_{15}$ | $K_{16}$ |
| 31.16 | 31.64 | 31.99 | 32.31 | 33.97 | 35.86 | 37.25 | 38.42 |
| ... | ... | ... | ... | ... | ... | ... | ... |

- *Análisis en* $K_1$. APAP tiende a emitir alertas por valores típicamente por debajo de 28. En la Figura 6.7 se observa cómo la colección de muestras legítima es mayor que la maliciosa. Teniendo esto en cuenta, es posible asumir que si el sensor únicamente considera los valores pequeños de ocurrencia de los diferentes n-gram en la carga útil del tráfico, no tendrá en cuenta un conjunto de datos lo suficientemente significativos.

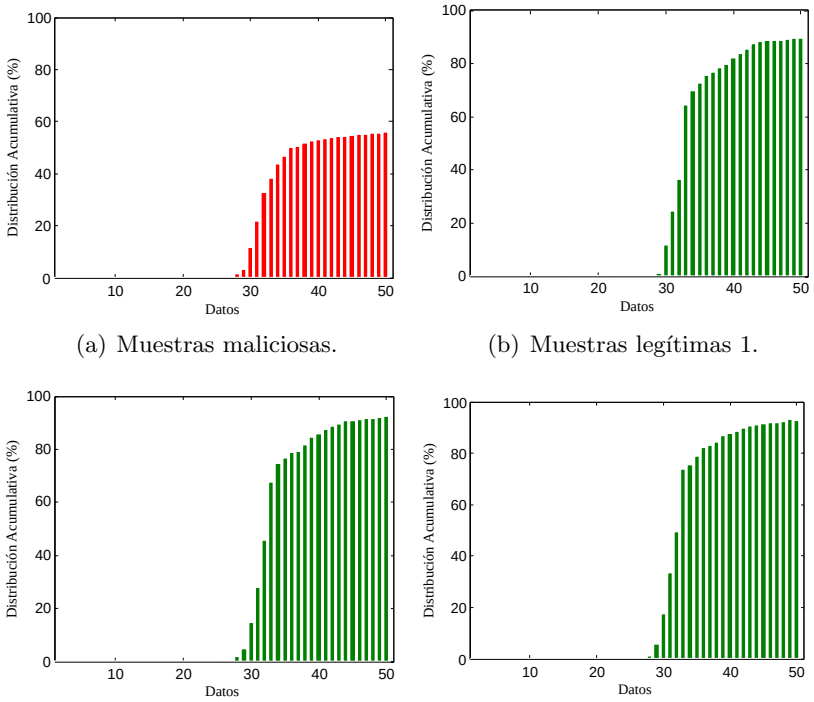



(a) Muestras maliciosas. (b) Muestras legítimas 1.

(c) Muestras legítimas 2. (d) Muestras legítimas 3.

Figura 6.7: Frecuencia acumulada en $K_1$.

- *Análisis en* $K_4$. Para $K_4$, tal y como se ilustra en la Figura 6.8, el sistema lanza alertas para valores típicamente por debajo de 29. Para esta $K$ es posible observar cómo el número de muestras por debajo de 29 se iguala entre los conjuntos de muestras legítimas y maliciosas.



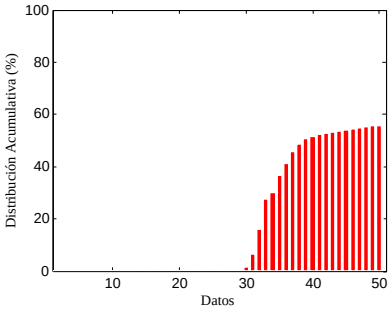

(a) Muestras maliciosas. (b) Muestras legítimas 1.

(c) Muestras legítimas 2. (d) Muestras legítimas 3.

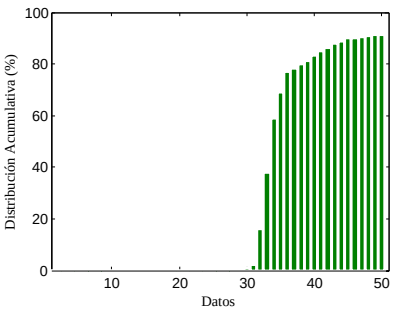
Figura 6.8: Frecuencia acumulada en $K_4$.

- *Análisis en* $K_8$. Para $K_8$, tal y como se ilustra en la Figura 6.9, el sistema lanza alertas para valores típicamente por debajo de 31. Puede observarse cómo para $K_i$ más altos comienza a haber un mayor número de valores en las muestras de ataques.

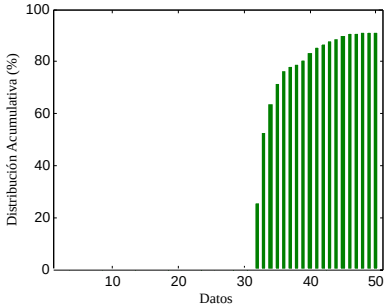
- *Análisis en* $K_{16}$. Para $K_{16}$, tal y como se ilustra en la Figura 6.10, el sistema emite alertas para valores típicamente por debajo de 39. Para los valores $K_i$ más altos se aprecia cómo prácticamente no existen muestras legítimas para valores por debajo de las reglas, pero sin embargo el número de muestras de ataques es considerable.



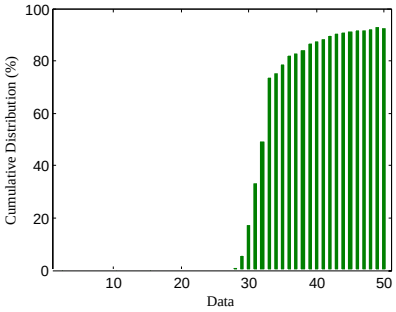

(a) Muestras maliciosas.




(b) Muestras legítimas 1.

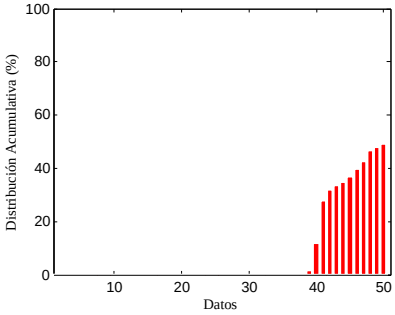



(c) Muestras legítimas 2.

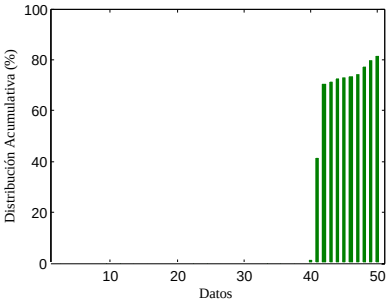



(d) Muestras legítimas 3.

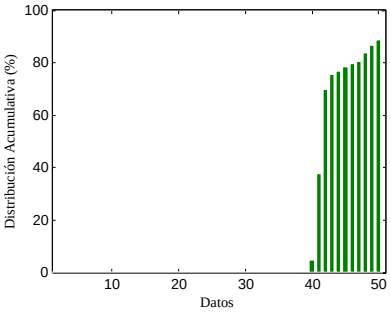
Figura 6.9: Frecuencia acumulada en $K_8$.



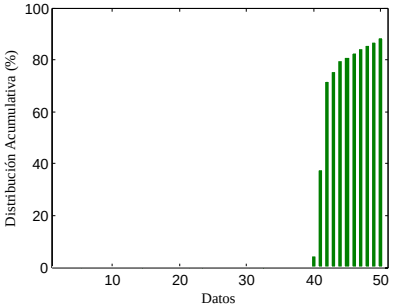

(a) Muestras maliciosas.




(b) Muestras legítimas 1.




(c) Muestras legítimas 2.




(d) Muestras legítimas 3.

Figura 6.10: Frecuencia acumulada en $K_{16}$.

## 6.7 DARPA'99

Con el fin de facilitar la comparativa de APAP con trabajos similares, su evaluación se ha llevado a cabo siguiendo el esquema propuesto para la colección DARPA'99 [Lab99]. Al aplicarse tal y como se describe en [HSB+12] permite la comparativa de APAP con algunos de los miembros de la familia PAYL más relevantes, como por ejemplo PAYL, ANAGRAM, POSEIDON y McPAD. En esta prueba APAP alcanzó una tasa de falsos positivos del 0.15% y detectó todas las trazas maliciosas a las que fue sometido. En la Tabla 6.4 se comparan los resultados obtenidos con los de sus antecesores, y se observa cómo en los trabajos más significativos de la bibliografía es relativamente frecuente llegar al 100% de tasa de acierto al analizar carga útil con malware (como sucede en ANAGRAM y AnPDPP). La tasa de falsos positivos lograda únicamente es superada por ANAGRAM, el cual obtuvo 0%. Sin embargo, y a diferencia de ANAGRAM, APAP aporta un sistema de generación de reglas de detección mucho más completo, ofreciendo una mayor escalabilidad y capacidad de adaptación a cambios en el entorno de monitorización. Finalmente, es importante tener en cuenta que tal y como se demostró en [HSB+12], los resultados obtenidos con DARPA'99 no tienden a ser escalables a redes actuales, debido principalmente al importante incremento de la heterogeneidad en las redes de comunicación actuales y al aumento en la sofisticación del malware. Propuestas como ANAGRAM han demostrado un importante incremento de su tasa de falsos positivos al operar sobre estas redes, la cuales pueden llegar empeorar hasta un 8% su calidad de etiquetado de tráfico legítimo, desencadenando la emisión de una gran cantidad de falsos positivos difícilmente manejable. Con esta prueba se demuestra que APAP está a la altura del resto de miembros de la familia PAYL al ser sometido a su estándar de evaluación funcional, pero aún falta por comprobar su eficacia en las redes de nueva generación.

Tabla 6.4: Comparativa de resultados con DARPA'99.

| Propuesta | FPR (%) | TPR (%) |
|---|---|---|
| PAYL [WCS05] | 0.0 | 90.76 |
| POSEIDON [BEHZ06] | 0.0 | 92.00 |
| AnPDPP [TKBK09] | 0.06 | 100.0 |
| Anagram [WPS06] | 0.0 | 100.0 |
| McPAD [PAF+09] | 0.33 | 87.8 |
| RePIDS [JTH+13] | 0.67 | 99.33 |
| APAP [GVSOMV15] | 0.15 | 100 |

## 6.8 Tráfico real de la Universidad Complutense de Madrid

En la evaluación con trazas de tráfico real, las trazas de ataques cedidas por el Centro de Cálculo y Procesamiento de Datos de la Universidad Complutense de Madrid (UCM) fueron divididas en cuatro conjuntos de ensayo, y el sensor fue sometido a un proceso

de validación cruzada (ver Sección 5.4.2 "Ofuscación de ataques enmascarados por imitación"). El entrenamiento base consideró las muestras de tráfico legítimo y fue ejecutado hasta llegar al estado de saturación, por lo que se asume la significancia del modelo de uso legítimo durante la experimentación. La Figura 6.11 muestra la variación de los resultados obtenidos para cada grupo al considerar como parámetro de ajuste el intervalo de confianza estimado en la decisión de la naturaleza de los paquetes (es decir, 0.005). Las mejores configuraciones observadas arrojan los siguientes resultados (Ver Figura 6.12):

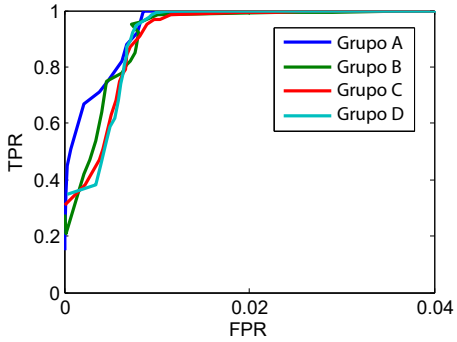



Figura 6.11: Curva ROC en de los resultados de grupos en evaluación cruzada.

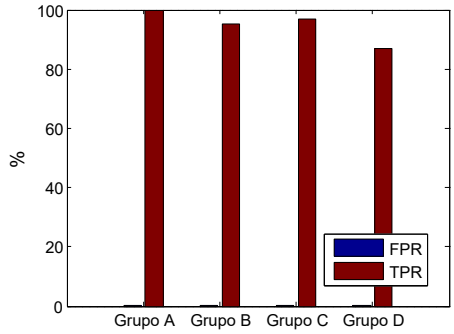



Figura 6.12: Resultados al analizar tráfico real de la UCM.

- Experimento 1 (grupo A) FPR=0.85%, TPR=100% (12 reglas generadas).
- Experimento 2 (grupo B) FPR=0.73%, TPR=95.16% (93 reglas generadas).
- Experimento 3 (grupo C): FPR=0.97%, TPR=96.77% (27 reglas generadas).
- Experimento 4 (grupo D) FPR=0.68%, TPR=87.09% (14 reglas generadas).

En estas pruebas se han considerado 32 valores $K$ $K_1 \ldots K_{32}$. El promedio obtenido demuestra una tasa de acierto del 94.75% y una tasa de falsos positivos del 0.8075%. Como era de esperar, estos resultados demuestran un ligero decremento en la calidad de la precisión del sensor debido principalmente al aumento de la heterogeneidad de la red. No obstante, su despliegue sigue siendo factible en los nuevos entornos de red, observándose una mayor coherencia entre los resultados obtenidos al analizar DARPA'99 que en gran

parte propuestas previas de la familia PAYL [HSB+12]. Puede afirmarse por lo tanto que APAP es un sólido sucesor de esta colección de sensores, cuyas estrategias de suavizado del contenido almacenado en los filtros Bloom y la facilidad en la elaboración de nuevas reglas de detección facilitan su adaptación a los escenarios de monitorización, y por lo tanto mejoran su comportamiento al operar sobre redes actuales.

# Capítulo 7

# Correlación de alertas en NIDS basados en anomalías

En este capítulo se presenta un marco para la correlación de alertas emitidas por sistemas de detección de anomalías en redes basados en el análisis estadístico de la carga útil del tráfico. La propuesta realizada se centra en la correlación de las alertas reportadas por sistemas de detección de intrusiones que operan sobre redes de comunicaciones, y en particular, que basan su estrategia de detección en el análisis estadístico de su carga útil con el fin de reconocer comportamientos anómalos. Por lo tanto se pretende dar solución a algunas de las dificultades planteadas en la Sección 4.3 "Análisis de la carga útil en redes de comunicaciones" relacionadas directamente con el despliegue de este tipo de sensores en redes actuales, como por ejemplo su tendencia a la emisión de tasas altas de falsos positivos, la necesidad de manejar grandes volúmenes de información en tiempo real o la escasa información inherente a los informes de alertas emitidas por sistemas basados en detección de anomalías. La aproximación realizada ha sido motivada por la necesidad de complementar el sistema de detección de intrusiones APAP [GVSOMV15] previamente descrito en el Capítulo 6 "Detección de malware en la carga útil". Nótese que en un intento de mejorar la gestión de la información que ofrecía, se valoró la posibilidad de correlacionar su contenido por técnicas similares a las presentes en la bibliografía. Pero éstas se basaban principalmente en el análisis de datos proporcionados por el encabezado de los paquetes y flujos de datos, como las direcciones IP de origen y destino, puertos, protocolos, duración de la comunicación, etc. los cuales pasaban completamente por alto la información directamente relacionada con las capacidades del sensor: carga útil de los paquetes, modelado o reglas de decisión que llevaron a desencadenar la emisión de alertas. Sobre esta base se propone un marco para la correlación de alertas centrado en aprovechar este tipo de información, y por lo tanto específico para sistemas de detección basados en el análisis estadístico de la carga útil del tráfico en busca de observaciones discordantes. Éste puede a su vez ser complementado por esquemas de gestión de incidencia de propósito general, abarcando de este modo los datos encapsulados en los dos grandes bloques de información presenten en los paquetes que circulan por las redes: encabezado y carga útil. Con el fin de adaptarse a las necesidades del operador, el marco propuesto adopta una

arquitectura multinivel, con etapas que consideran las alertas tanto de manera individual como secuencial. Una primera etapa de procesamiento ofrece clasificaciones en tiempo real a nivel de paquete. El segundo de tratamiento de datos considera grupos de alertas, ofreciendo una visión mucha más general del estado de la red y permitiendo la identificación de ataques divididos en diferentes entramados. La eficacia de la propuesta ha sido demostrada en un caso de uso real, siendo desplegada como complemento de APAP. Por lo tanto, se ha evaluado con tráfico real monitorizado en la red de la facultad de informática de la Universidad Complutense de Madrid (UCM), habiendo arrojando resultados muy prometedores. El contenido del capítulo está organizado de la siguiente manera: en la Sección 7.1 se introduce un marco para la correlación de este tipo de incidencias, describiéndose en detalle su arquitectura y las diferentes etapas de procesamiento de información; en la Sección 7.2 se describe una instanciación del marco, donde cada uno de sus niveles integra diferentes herramientas de adquisición de conocimiento y clasificación; en la Sección 7.3 se describe la experimentación realizada; finalmente, en la Sección 7.4 se discuten los resultados obtenidos.

## 7.1 Marco para la correlación de alertas

El marco propuesto se adapta a las particularidades de los sistemas de reconocimiento de anomalías en la carga útil del tráfico para la identificación de malware. En base a esto se tienen en cuenta los siguientes principios de diseño y limitaciones.

- Se asume que los sensores a complementar tienen arquitectura centralizada. Esto es debido a que muy pocas propuestas de la bibliografía plantean esquemas alternativos para el estudio de la carga útil.
- Los sensores a complementar analizan el tráfico paquete a paquete, estudiando su carga útil de manera independiente. Por lo tanto, basan la construcción de sus modelos del uso en características de la carga útil, que es la portadora del software malicioso.
- Las características de los sistemas de detección deben regir los rasgos que tenga en cuenta el sistema propuesto a la hora de establecer sus propios modelos.
- Se asume que a partir de ciertas características de la carga útil es posible la inferencia de propiedades del malware, y por lo tanto es de esperar que la selección de sus características más significativas condiciones la eficacia del sensor. La identificación de los rasgos más relevantes debe llevarse a cabo teniendo en cuenta la base de conocimiento sobre la que opera el sistema de detección, la naturaleza de las amenazas más probables contra el entorno protegido, y los algoritmos que formen parte de la instanciación del marco. En consecuencia, esta tarea tiene una gran dependencia de las características de caso de uso, cuyo estudio queda fuera del alcance de esta primera aproximación.
- Cuando la carga útil de un paquete satisface ciertas propiedades previamente definidas a lo largo de la etapa entrenamiento del sensor a complementar, procede

a la emisión de alertas. Algunos de estos informes también pueden resultar de la observación de discordancias significativas con el modelo de uso legítimo de la red.

- El malware puede llegar al sistema víctima distribuido en diversos paquetes. En este caso es habitual que cada uno de ellos contenga código encargado de una tarea diferente enmarcada en el proceso de intrusión.
- Los sistemas de detección de intrusiones robustos frente a métodos de evasión a menudo operan de manera no determinista, lo que dificulta que el atacante identifique su modus operandi, y pueda elaborar estrategias capaces de inutilizarlo [POTPL14]. Adicionalmente, los sistemas de detección de intrusiones con estas capacidades facilitan la identificación de malware camuflado por técnicas de ofuscación basadas en polimorfismo. Esto habitualmente se lleva a cabo mediante el estudio de las partes invariantes del vector de infección.

En base a estas premisas, el marco propuesto analiza las alertas emitidas por el sensor aplicando criterios similares a los que rigen su proceso de detección de intrusiones. Por lo tanto, una vez desplegados, comparten métricas y métodos de modelado, situación que lleva a un alto nivel de complementación, ofreciendo características similares. De este modo, si por ejemplo el sistema de detección de intrusiones es capaz de analizar con precisión e indistintamente tráfico derivado de diferentes protocolos de red (IPv4, IPv6, ICMP, etc.), es de esperar que la implementación del marco para la correlación de alertas también ofrezca estas características, permitiendo además mejorar sus desventajas. Como principal inconveniente cabe destacar la alta especificidad en su implementación, lo que reduce la escalabilidad e interoperabilidad de sus instancias. Finalmente es importante tener en cuenta que el marco propuesto actúa en un perímetro de seguridad diferente al sensor, por lo que no debe afrontar las dificultades propias de la monitorización de tráfico en redes de comunicaciones; únicamente los informes emitidos por los sistemas de detección de intrusiones. De este modo, circunstancias tales como que el tráfico circule cifrado, son invisibles para los métodos de correlación desplegados.

### 7.1.1 Arquitectura

El esquema propuesto combinas diferentes técnicas de clasificación con el fin de paliar los inconvenientes derivados del despliegue del sistema de detección de intrusiones a complementar. La idea detrás de este despliegue de clasificadores es que las conclusiones alcanzadas tengan en cuenta el consenso de diferentes "expertos", que en este caso son cada uno de los componentes de la arquitectura planteada. Esto permite, entre otras cosas, considerar diferentes colecciones de muestras de referencia durante el entrenamiento, llevar a cabo diferentes calibrados y desplegar diversos algoritmos de reconocimiento de patrones. Además, facilita el estudio de las diferentes características de la intrusión y las relaciones que existen entre ellas. Zimek et al. [ZV15] discutieron en profundidad los beneficios e inconvenientes de esta metodología, y la describieron valiéndose de la paradoja ilustrada en el cuento tradicional hindú de "los hombres ciegos y el elefante". Éste describe como diferentes "expertos", es este caso los hombres ciegos, alcanzan diferentes conclusiones

acerca de la naturaleza del animal guiándose únicamente de la información que les ofrece el palpar una parte diferente de su cuerpo. Sólo al poner en común el conocimiento adquirido fueron capaces de hallar la solución correcta del acertijo. Los "expertos" del marco propuesto también analizan diferentes características de las alertas, centrándose principalmente en dos criterios de decisión: las diferencia entre las discordancias detectadas respecto a los modelos de uso construidos durante el entrenamiento, y la naturaleza potencial de las amenazas que representan. Teniendo esto en cuenta es posible establecer dos grandes componentes de procesamiento de información: Diagnóstico de Anomalías o AD (del inglés *Anomaly Diagnosis*) y Diagnóstico de naturaleza o ND (del inglés *Nature Diagnosis*). El despliegue del AD lleva a cabo la correlación por medio del análisis del nivel de discordancia en la observación que desencadenó la emisión de las alertas. Por lo tanto, cuanto mayor es la distancia de los rasgos más significativos de la carga útil respecto al modelo de uso legítimo, mayor es su relevancia. Por otro lado, los componentes ND correlacionan incidencias centrándose en su naturaleza, la cual es deducible de las reglas y métricas que han llevado a su identificación. Ambos criterios son importantes, y deben ser tenidos en cuenta conjuntamente de manera previa al despliegue de contramedidas. Subestimar alguno de ellos puede acarrear la ejecución de acciones insuficientes o desproporcionadas. Esto sucede por ejemplo cuando los componentes AD anuncian una incidencia de gran relevancia, pero su correlación basada en naturaleza señala a una colección de amenazas apenas nocivas para el entorno protegido. En el caso contrario, la correlación basada en AD puede reflejar niveles de discordancia poco significativos; sin embargo el estudio de su naturaleza puede apuntar a riesgos de gran impacto, lo que claramente condiciona la toma de decisiones.

La arquitectura de la propuesta se ilustra en la Figura 7.1, y está estructurada en dos etapas de procesamiento de información. La primera de ellas se encarga de la gestión de eventos independientes, y lleva a cabo la correlación en base a la naturaleza y el nivel de discordancia de cada muestra individualmente, a nivel de paquete. Su principal objetivo es la priorización y reducción de falsos positivos en tiempo real. Esta etapa también facilita la toma de decisiones rápidas, y provee la información necesaria para el análisis de las secuencias de incidencias, las cuales únicamente podrán deducir conclusiones tras observar los informes asociados a una cierta cantidad de paquetes. El segundo nivel procesa las alertas de manera secuencial. Cada una de estas secuencias es una serie de alertas identificadas en in intervalo de tiempo concreto, o tras concatenar una cantidad específica de eventos previamente agregados en la etapa previa. El componente ND facilita la asociación de incidencias por medio de técnicas de deducción dependientes de una base del conocimiento relacionada con ataques distribuidos en paquetes.

### 7.1.2 Componente de Diagnóstico de Anomalías

Los diagnósticos basados en el estudio del grado de discordancia se llevan a cabo a nivel de paquete, y representan los criterios de correlación basados en datos cuantitativos. Con este fin las alertas reportadas por el sensor complementado son analizadas, determinándose la distancia que presentan las observaciones discordantes respecto a los modelos de uso del sistema. Antes de decidirse su pertenencia, el sistema de correlación construye diferentes

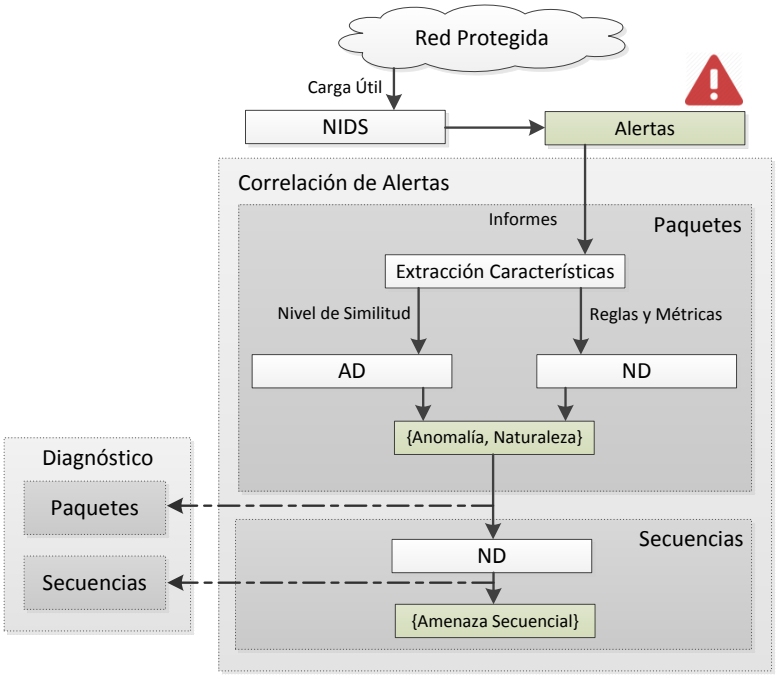



Figura 7.1: Arquitectura para la correlación de alertas.

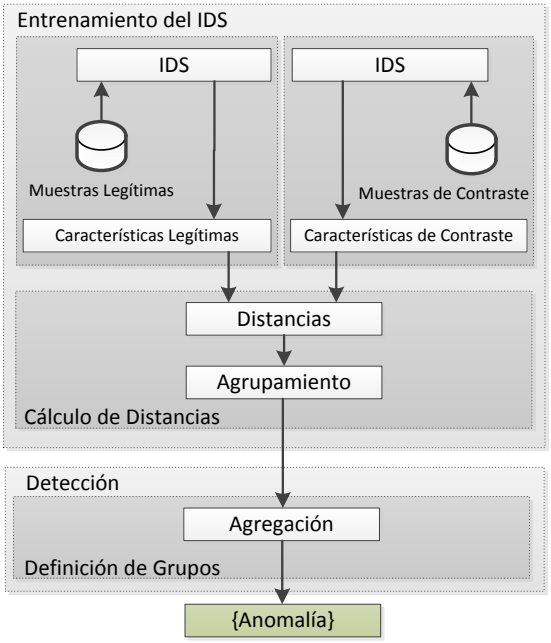



Figura 7.2: Componente de Diagnóstico de Anomalías.

grupos de riesgo por medio de una etapa de agrupamiento (del inglés *clustering*) previa a la etapa de detección del sensor que aprovecha las características con las que es entrenado, tal y como se ilustra en la Figura 7.2.

El conjunto de muestras considerado para el entrenamiento se define como $D = d_1, \ldots, d_n$ tal que $0 < n$. Dentro de este conjunto, el subconjunto de muestras maliciosas es $A =$

$A_1, \ldots, A_{m1}$, $A \subset D$ y el subconjunto de muestras legítimas es $L = l_1, \ldots, l_{m2}$, $L \subset D$, cumpliéndose que $0 < m_1, m_2$ y $A \cup L = D$. El diagnóstico de anomalías puede generalizarse como un problema de agrupamiento que conlleva dos etapas de procesamiento de información:

1. *Definición de grupos.* Las diferentes clasificaciones de los niveles de discordancia se establecen en la etapa de definición de grupos y durante el entrenamiento del sensor, donde $L$ se tiene en cuenta en la construcción del modelo de uso legítimo de la red. Sus rasgos más relevantes se definen como el conjunto $ML = \{ML_1, \cdots, ML_p\}$, $0 < p$. En la definición de grupos de alertas se considera $ML$, y $A$ si es necesario, dependiendo esto último del número de clases establecidas para el entrenamiento. Por lo tanto, si se considera un esquema de una clase, sólo se analizan rasgos inherentes a muestras legítimas; esto también sucede si la colección $A$ es poco significativa, llevando a la definición de grupos únicamente en base a las medidas de similitud que separan las observaciones sospechosas de los elementos de $L$. Por otro lado, si $A$ es significativo se consideran múltiples clases, asociándose una de ellas al modelo legítimo, y el resto a los distintos grupos de inconsistencias definidos a partir de los elementos de $A$. Alternativamente es posible la consideración de únicamente medidas de similitud relacionadas con la observación analizada y los modelos construidos a partir de contraejemplos, lo que resulta intuitivamente más preciso, ya que el valor central de los grupos definidos frecuentemente diverge de los límites del modelo de uso legítimo, tal y como sucede con la carga útil responsable de la emisión de alertas y los modelos construidos por el propio sensor.

2. *Agregación.* Las alertas son agregadas cuando el sensor complementado opera en modo de detección. Concretamente, este paso se lleva a cabo cada vez que el sensor emite una alerta, y conlleva su asociación con el grupo de discordancias definido en la etapa previa con el que presente mayor similitud. Nótese que, al concluir su clasificación, la carga útil asociada a grupos con valores centrales más próximos al grupo de las observaciones legítimas resulta menos divergente, y por lo tanto es más probable que pueda deberse a errores de etiquetado, desencadenando la emisión de falsos positivos. En [WFBS14] se discute una gran cantidad de estrategias para estimar, construir e implementar este tipo de distancias y umbrales. El marco propuesto en este capítulo es de propósito general, y por lo tanto se adapta a cada una de ellas. Sin embargo, la decisión de aquellas que mejor se comporten en cada caso de uso queda fuera del alcance de esta aproximación.

### 7.1.3 Componente de Diagnóstico de Naturalezas

En el ámbito de este marco, la naturaleza de una anomalía es el tipo de amenaza que la desencadena. Por lo tanto, el componente que estudia las alertas en base a su naturaleza se centra en el análisis de las amenazas que desencadenan su emisión; esto requiere de la elaboración previa de una base de conocimiento con elementos factuales acerca de los diferentes ataques potencialmente dirigidos contra el entorno protegido, y conocimiento

procedimental capaz de inferir estos tipos de intrusiones a partir de las características de las incidencias indicadas en las alertas. Esta base de conocimiento se genera durante la etapa de entrenamiento del sensor aprovechando la información que guía la construcción de modelos de uso. Las naturalezas de las anomalías son estudiadas a nivel de paquete y secuencias, lo que lleva a la distinción de dos etapas diferentes de procesamiento de información, tal y como se describe a continuación.

#### 7.1.3.1 ND a nivel de paquete

El estudio de la naturaleza de las alertas a nivel de paquete tiene como objetivo clasificar y agrupar las incidencias reportadas por el sensor de manera individual, y sin tener en cuenta la relación que pudiera existir entre unas y otras. Se trata de un módulo de respuesta rápida orientado al análisis en tiempo real, y que por lo tanto, bajo ninguna circunstancia debe penalizar de manera significativa las tareas de monitorización desempeñadas por el sensor; precisamente este requisito ha llevado a su instanciación por medio de una red neuronal artificial o ANN (del inglés *Artificial Neural Network*), tal y como se detalla en la sección posterior. Sin embargo, e independientemente de los algoritmos de clasificación implementados, este nivel de procesamiento requiere de una base de conocimiento capaz de asociar las discordancias observadas con amenazas conocidas identificables mediante el estudio de la carga útil del tráfico. Afortunadamente hoy en día existen diferentes bases de datos con información detallada acerca de estas características (véase [CE18, UC16]), y fácilmente integrables en el marco propuesto, aunque esta información no es suficiente; para que el sensor sea complementado adecuadamente, el conocimiento debe ser adquirido a partir de características y métricas también manejadas por el sistema de detección de intrusiones a complementar. Esta situación ha llevado a la construcción de una base de reglas de producción capaz de considerar las propiedades del NIDS y el conocimiento disponible acerca de ataques. Con el fin de promover el diseño de una estrategia de inferencia apropiada, se han considerado las siguientes premisas:

- Los paquetes con contenido anómalo podrían desencadenar la activación de más de una regla de detección.
- En la base de reglas, una *característica* es un conjunto de métricas extraídas de la carga útil y los valores que se les asignan. Por ejemplo, podría considerarse que la *característica* "nunca visto con anterioridad" se observa cuando cierto segmento del contenido binario de la carga útil no ha sido registrado en el tráfico monitorizado previamente. Otra *característica* podría ser "poco frecuente", observable cuando se identifica un segmento con poca frecuencia de aparición. Nótese que la presencia de estas *características* en una muestra por sí solas no tienen por qué suponer una amenaza. Pero ciertas combinaciones de *características* a priori inofensivas, pueden delatar la presencia de incidencias relacionadas con la seguridad del entorno protegido, hallándose en [WPS06, GVSOMV15, GVMVSO17] algunos ejemplos de ello.
- Cada regla de clasificación es activada por la presencia de ciertos conjuntos de

*características*, y lleva a la deducción de *categorías*.

- Una *categoría* puede ser inferida por diferentes reglas de clasificación.
- El correcto funcionamiento del componente de correlación basado en el estudio de la naturaleza de las incidencias requiere tolerancia a errores. Por lo tanto, debe considerarse que parte de las reglas de clasificación activadas al analizar una muestra pueden proceden del análisis de ruido en su carga útil.

En base a estos principios de diseño, las *características* que dan forma a la base de reglas deben ser similares a los rasgos considerados por el sensor a la hora de decidir la emisión de una alerta. El conjunto de *características* de la base del conocimiento se define como $C = \{C_1, \cdots, C_n\}$, mientras que el conjunto de reglas se expresa como $R = \{R_1, \cdots, R_m\}$. Cuando una regla es activada, se deduce al menos una *categoría* del conjunto $E = \{E_1, \cdots, E_l\}$. De este modo, si por ejemplo la regla $R_i$, $0 < i \leq m$ es activada debido a la identificación de un subconjunto de $C$ delimitado por $p$ y $q$, $0 < p < q \leq n$ entonces $C_p \wedge C_{p+1} \wedge \cdots \wedge C_{q-1} \wedge C_q \Rightarrow E_k$, donde $E_k$ es la *categoría* inferida por $R_i$. Las *categorías* son las diferentes clasificaciones que la base de reglas es capaz de deducir, lo que lleva a la necesidad de definir una *categoría* por defecto que resulte de interpretar colecciones de *características* incapaces de generar nuevo conocimiento, a la que se ha llamado "anomalía desconocida".
La estrategia de construcción de la base de reglas depende directamente de las métricas y estrategia de modelado que intervienen en las tareas de análisis del sensor a complementar. Por lo tanto, el marco propuesto está abierto a diferentes métodos de aprendizaje (en la Sección 3.3 "Adquisición de conocimiento" se revisan muchas otras opciones). En la instanciación implementada se considera *categorías* a las diferentes familias de ataques presentes en la colección de muestras de referencia. Las *características* son los rasgos que componen las métricas considerada en el proceso de detección del sensor. De este modo, si por ejemplo una métrica es calculada por 12 rasgos diferentes de la carga útil, dichos rasgos son considerados como *características* en las tareas de correlación. En la implementación realizada la frecuencia de aparición de estas *características* al analizar la carga útil de cada muestra es ordenada y tenida en consideración. Por lo tanto, es posible tener en cuenta la distribución de las *características* observadas en las muestras. Nótese que además de esta estrategia, existen muchas otras aproximaciones al problema de heredar y tener en cuenta las métricas del sensor, compilándose en [JBA14] algunas de ellas. La implementación descrita a continuación ha sido elegida con fines didácticos, a fin de mejorar la comprensión del marco propuesto. En base a esta, cada vez que el detector emite una alerta se ejecutan las siguientes acciones:

1. Se extraen los rasgos de la carga útil del paquete sospechoso.
2. Se calcula y enumera la frecuencia de aparición de cada *característica* asociada a los rasgos observados.
3. Se calcula el grado de representatividad de cada *característica* en la carga útil. A estos valores se los denomina "puntuaciones de *características*". En la instanciación

realizada, dado $C_i$, $0 < i \leq n$, la puntuación calculada es su frecuencia de aparición en la carga útil $frec(i)$.

4. Las *características* de la carga útil se ordenan de mayor a menor en función de su puntuación. De este modo, las más significativas pueden ser tomadas rápidamente en mayor consideración. La lista de puntuaciones ordenada se define como $ORD$, donde $ORD(i)$, $o < i \leq n$ es la frecuencia de aparición $frec(i)$ de la *característica* en la posición $i$. Con el fin de evitar incoherencias, $\forall p, q \in C : \{ORD(p) \neq ORD(q)\}$. En la implementación esto se ha logrado por medio de la elaboración de una tabla de desambiguación en la que si se produce $frec(p) = frec(q)$, se indica cual de las dos *categorías* ocupará la posición anterior de la lista. Dentro de esta tabla, las *características* más frecuentes en las *categorías* asociadas a amenazas potencialmente más dañinas ganan un mayor protagonismo.

5. Se determinan las *categorías* inferidas por la base de reglas al considerar $ORD$. En circunstancias ideales, la base de reglas debe elaborarse de tal manera que la respuesta a la activación de cada regla sea única. En este caso puede decirse que la propia *categoría* es la naturaleza de la discordancia. Cuando para una anomalía se infieren diferentes *categorías*, lo más apropiado es que el operador tenga constancia de todas ellas. Esto es importante dada la existencia de técnicas de evasión basadas en camuflar amenazas de gran riesgo en entramados derivados de la emisión de una gran cantidad de alertas relacionadas con riesgos menores, tal y como se describieron T. Cheng et al. [CLLL11]. En términos generales, las *categorías* deducidas constituyen la naturaleza de los grupos a los que puede pertenecer una alerta.

En la Tabla 7.1 se muestra un ejemplo de base de reglas capaz de establecer diferentes *categorías* a partir de *características* derivadas de los rasgos observados en la carga útil del tráfico de una red. En ella se considera: $E = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8\}$, $C = \{C_1, C_2, C_3, C_4\}$ y $R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}\}$. Para definir la naturaleza de las alertas basta con determinar qué reglas son capaces de deducir *categorías*, y qué *categorías* pueden derivar su producción. Algunas de las reglas de la tabla requieren de la identificación de al menos dos *características*, y no tienen en cuenta la presencia de otras $\{R_1, R_2, R_7, R_8, R_{10}, R_{11}\}$. También se observan reglas que indican implícitamente que ciertas *características* no deben hallarse $\{R_3, R_4, R_5, R_6, R_9\}$. La activación de la regla $R_{12}$ lleva a la deducción de la *categoría* por defecto $E_8$, también conocida como "anomalía desconocida". La base de reglas depende de la base del conocimiento y estrategia de modelado del uso de la red llevado a cabo por el sensor. El proceso de generación de conocimiento puede ilustrarse más claramente al asignar valores a algunas de sus variables. Por ejemplo, si $C_1$ indica "ofuscación de malware" y $C_2$ indica "desbordamiento de buffer", entonces $R_1$ puede interpretarse en lenguaje natural como "si la mayor parte de los rasgos hallados en la carga útil denotan la presencia de malware ofuscado y técnicas de desbordamiento de buffer, entonces la alerta se agrupa en la *categoría* $E_1$", donde $E_1$ podría ser "malware polimórfico con escalada de privilegios". Adicionalmente, si por ejemplo $C_4$ es "técnicas de control remoto", entonces $R_7$ y $R_8$ se

describen en lenguaje natural como "si la mayor parte de los rasgos hallados en la carga útil indican la presencia de malware ofuscado y contienen técnicas de control remoto, entonces la alerta se agrupa en la categoría $E_5$", la cual podría ser "presencia de troyano".

Tabla 7.1: Example of rule base in ND.

| $R$ | Activación | $E$ |
|---|---|---|
| $R_1$ | $Ord(1) = C_1 \wedge Ord(2) = C_2$ | $E_1$ |
| $R_2$ | $Ord(1) = C_2 \wedge Ord(2) = C_1$ | $E_1$ |
| $R_3$ | $Ord(1) = C_1 \wedge Ord(2) = C_3 \wedge \{Ord(3, \cdots, n\} = \emptyset$ | $E_2$ |
| $R_4$ | $Ord(1) = C_2 \wedge Ord(2) = C_1 \wedge \{Ord(3, \cdots, n\} = \emptyset$ | $E_2$ |
| $R_5$ | $Ord(1) = C_3 \wedge \{Ord(2, \cdots, n\} = \emptyset$ | $E_3$ |
| $R_6$ | $Ord(1) = C_2 \wedge \{Ord(2, \cdots, n\} = \emptyset$ | $E_4$ |
| $R_7$ | $Ord(1) = C_1 \wedge Ord(2) = C_4$ | $E_5$ |
| $R_8$ | $Ord(1) = C_4 \wedge Ord(2) = C_1$ | $E_5$ |
| $R_9$ | $Ord(1) = C_4 \wedge Ord(2, \cdots, n) = \emptyset$ | $E_6$ |
| $R_{10}$ | $Ord(1) = C_2 \wedge Ord(2) = C_4$ | $E_7$ |
| $R_{12}$ | $Ord(1) = C_4 \wedge Ord(2) = C_2$ | $E_7$ |
| $R_{13}$ | $Others$ | $E_8$ |

#### 7.1.3.2 ND a nivel de secuencia

En una primera aproximación, el marco propuesto tan solo consideraba los dos componentes que han sido previamente descritos, lo que ofrecía un esquema de análisis híbrido capaz de correlacionar en tiempo real las alertas emitidas por el sensor. Pero tras la revisión en profundidad de la propuesta, se hizo constancia de diferentes aspectos relevantes para la correcta complementación del sensor, algunos de los cuales son descritos a continuación:

- El IDS complementado analiza la carga útil del tráfico paquete a paquete y de manera independiente, lo que a priori no permite la detección de malware distribuido en varios de ellos.

- El contexto en el que son emitidas las alertas a menudo provee información cognitiva útil para el análisis de las incidencias y capaz de mejorar la toma de decisiones de los operadores. Cuando el marco correlaciona las alertas de manera no determinista y se informa al operador del conjunto de posibles clasificaciones para cada una de ellas e incertidumbre, es posible llegar a comprender en mayor profundidad la naturaleza de los eventos registrados en la red.

- El análisis forense de trazas de tráfico debe considerar la información adquirida tanto a nivel de paquetes como de secuencias. De este modo los operadores disponen de una visión general de la incidencia, y pueden revisar paquete a paquete su composición.

- La inclusión de un nivel mayor de abstracción en el marco propuesto mejora la comprensión de las incidencias, y de este modo, la eficacia de su despliegue. Además, en ciertas circunstancias es posible que no sea capaz de procesar la información recibida paquete a paquete en tiempo real, como por ejemplo sucede cuando es requerido un continuo despliegue de contramedidas a las que debe hacerse seguimiento, o cuando el sensor opera de manera demasiado restringida generando un exceso de alertas (esperándose en este caso altas tasas de falsos positivos). La ejecución de contramedidas únicamente a partir de eventos registrados a nivel secuencial es una solución eficiente a este problema, aunque puede limitar la capacidad de respuesta del despliegue defensivo.

La correlación basada en el diagnóstico de la naturaleza de la incidencia a nivel secuencial es similar a la que tiene lugar a nivel de paquete. Pero en este caso se parte de la información inicial relacionada con el etiquetado de cada alerta generada en el nivel de procesamiento anterior. Al igual que en dicha etapa, la base del conocimiento se representa por reglas de producción de manera similar a como se ilustró en Tabla 7.1. Pero a diferencia que en la etapa anterior, ahora el conjunto de hechos básico $C = \{C_1, C_2, C_3, C_4\}$ son diagnósticos reportados por la etapa anterior (correlación a nivel de paquetes). Por ejemplo, $C_1$ podría expresarse en lenguaje natural como "malware polimórfico con escalada de privilegios" coincidiendo con $E_1$ a nivel de paquete, o $C_5$ podría indicar "ofuscación de malware y métodos de control remoto" coincidiendo con $E_5$. Por lo tanto, los hechos en el análisis de secuencias son los diagnósticos deducidos en la etapa previa del sistema de correlación. Ejemplo de diagnósticos a nivel de secuencia pueden ser "spyware", "adware" o "ransonmware", los cuales se contextualizan en un nivel de abstracción mayor. Nótese que, dada la complejidad a la hora de formular este tipo de reglas de producción, se requiere de una base de conocimiento relacionada con amenazas distribuidas en paquetes. Pero esta información no es fácil de encontrar, por lo que pueden generarse sintéticamente mediante técnicas de aprendizaje automático. Otro aspecto interesante a considerar es la delimitación de las secuencias a procesar; separarlas en el lugar apropiado puede llegar a establecer la diferencia entre acertar o fallar en la elección de contramedidas. Esto plantea sin duda un elemento de ajuste interesante, pero profundizar en su elección óptima queda fuera del alcance de esta aproximación, habiéndose adoptada a lo largo de la experimentación criterios basados en el conteo del número de alertas recibidas.

## 7.2 Implementación

A modo de ejemplo de instanciación del marco propuesto se ha llevado a cabo su implementación para complementar APAP [GVSOMV15, GVMVSO17]. Este sensor analiza en tiempo real la carga útil del tráfico que circula por una red de comunicaciones en busca de anomalías que resulten en indicios de presencia de malware (ver Capítulo 6). Para ello se aplican reglas autogeneradas en un proceso de aprendizaje semi-supervisado en el que se modela el comportamiento habitual y legítimo de la red; los modelos son refinados al considerar las características de una colección de muestras de referencia de

tráfico malicioso. Al igual que gran parte de los miembros de la familia de sensores PAYL [WS04], APAP adopta el uso de la metodología N-gram para la extracción de las características de la carga útil y la optimización de su almacenado mediante estructuras de datos probabilistas. Cabe resaltar que el sensor de la familia con el que guarda mayor similitud es ANAGRAM [WPS06], aunque APAP también se aprovecha de las características del popular sistema de detección de intrusiones Snort [Sno18], sobre el cual está implementado. A continuación se describe la instanciación de cada uno de los elementos del marco.

### 7.2.1 Instanciación del AD

El componente de correlación basado en diagnósticos de anomalías opera con las mismas métricas que APAP. Tal y como se ilustra en la Figura 7.3, en su etapa de definición de grupos se construye una lista de las distancias Manhattan resultantes de comparar las características del modelo legítimo de APAP con las del modelo construido a partir de la colección de muestras maliciosas de referencia. Los valores obtenidos permiten la generación de grupos de similitud por medio de algoritmos de agrupamiento. De entre las diferentes aproximaciones a este problema se ha optado por el método k-medias con autoajuste de su valor $K$ a partir del método *Elbow*. Este calibrado consiste en la ejecución iterativa de k-medias con diferentes valores $K$ hasta que se observen variaciones representativas en la suma de errores cuadráticos entre cada miembro de los grupos y su valor central. La decisión de este algoritmo se debe a que es una estrategia muy conocida y especialmente tolerante a errores en las distancias debidos a la presencia de discordancias y muestras incorrectamente clasificadas. El método *Elbow* compensa su mayor inconveniente: la necesidad de ajustar el valor $K$. En [XW05] es descrito en detalle y se recopilan diferentes alternativas, cuyo estudio en profundidad y adaptación a este caso de uso podría resultar una interesante línea de trabajo futuro. En la etapa de agregación de este componente, las métricas extraídas de la carga útil de los paquetes que han desencadenado la emisión de alertas son asociadas a los grupos respecto a los que presenten mayor similitud. Los grupos con mayor número de elementos derivados de muestras legítimas son considerados menos discordantes, y por lo tanto las alertas que se les asocian son más propensas a resultar en falsos positivos.

### 7.2.2 Instanciación del ND a nivel de paquete

El primer aspecto a tener en cuenta al instanciar el componente de correlación basado en diagnósticos de naturaleza es el cómo se va a definir la base de reglas de producción. Al complementar APAP, el conocimiento factual y procedimental proviene de los valores $K_s$ que definen los modelos de uso, rasgos de las muestras analizadas, y reglas de detección (ver Figura 7.4). El conjunto de características $C = \{C_1, \cdots, C_n\}$ y de reglas de producción $R = \{E_1, \cdots, R_m\}$ permiten la definición de un clasificador basado en redes neuronales artificiales (ANN). La sustitución de los motores de inferencia convencionales por esta tecnología es una práctica común en la bibliografía, que entre otras ventajas destaca por su gran eficiencia al procesar información en tiempo real y tolerancia a errores de

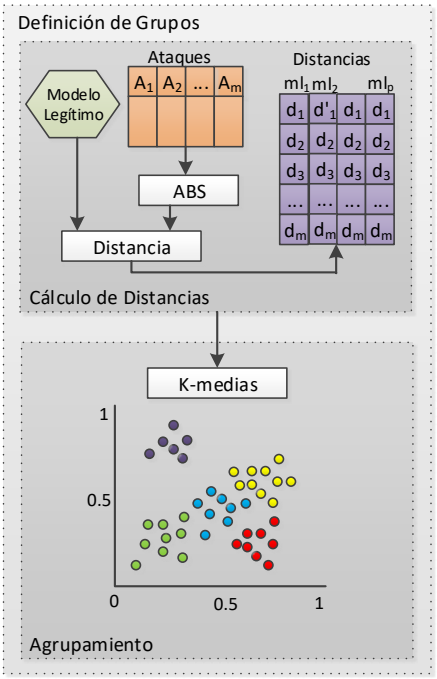



Figura 7.3: Instanciación del componente AD.

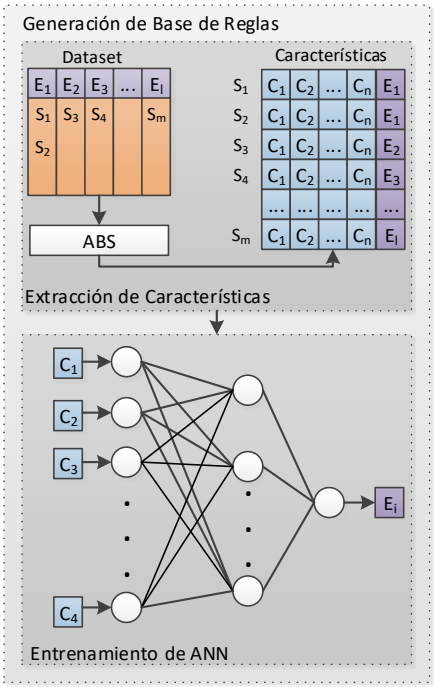



Figura 7.4: Instanciación de ND a nivel de paquete.

etiquetado [ADT95]. La red neuronal artificial considerada presenta $n$ entradas, tantas como características son proporcionadas por las alertas, alineándose cada entrada con cada una de ellas. Cada entrada $v$ acepta un rango de valores $0 < v \leq 1$ que corresponde con la probabilidad de aparición de cada característica. La red neuronal presenta una

única salida, la cual provee la categoría en que la alerta puede ser agrupada. Durante su entrenamiento, cada muestra introducida tiene la forma de un vector $C = [C_1, \cdots, C_n]$ que contiene las características extraídas de las muestras de referencia maliciosas. Se espera que la salida identifique la familia de malware de la intrusión, por lo que las muestras deben ir acompañadas de su grado de pertenencia a cada una de ellas. Una vez concluido el entrenamiento, basta con introducir las características reportadas por las alertas emitidas por el sensor al analizar el tráfico, para establecer la naturaleza de cada paquete. Otros aspectos a resaltar acerca del despliegue de la red neuronal son que está estructurada en tres capas (la intermedia es oculta), el error asumido en el entrenamiento es 0.001 y que se adoptó la función de activación ELLIOT [YAF10]. La última de ellas es una popular aproximación de gran velocidad a la función de activación de la tangente hiperbólica que abarca $0 < y < 1$ y se expresa de la siguiente manera:

$$y = \frac{\frac{x \times s}{2}}{(1 + |x \times s|) + 0.5} \tag{7.1}$$

$$d = s \times \frac{1}{2 \times (1 + |x \times s|) \times (1 + |x \times s|)} \tag{7.2}$$

donde $x$ es la entrada de la función de activación, $y$ es la salida, $s$ es la pendiente y $d$ su derivada. En la Tabla 7.2 se resume la configuración establecida, la cual es sencilla a modo de ejemplo introductorio, pero también capaz de ajustarse a los requisitos del sistema.

Tabla 7.2: Ejemplo de base de reglas en ND a nivel de paquete.

| Parámetro | *Valor* |
|---|---|
| Entrada | $n$ |
| Salida | 1 |
| Capas | 3 |
| Error de entrenamiento | 0.001 |
| Max. Pasos de entrenamiento | 50000000 |
| Capas ocultas | 1(medio) |
| Función de activación | *ELLIOT* |

### 7.2.3 Instanciación del ND a nivel de traza

La instanciación del componente de correlación basado en el diagnóstico de la naturaleza de secuencias de alertas podría haberse realizado de manera análoga al nivel anterior. De este modo, gracias a la red neuronal artificial se obtendría gran eficiencia, pero no se mostraría resultados de manera no determinista. Aunque es posible establecer un proceso de refinamiento a la salida de la ANN capaz de brindar soluciones no deterministas, se ha optado por mostrar una implementación completamente distinta, lo que se espera que sea de mayor utilidad en futuros despliegues. En particular se ha considerado el uso de un algoritmo genético o GA (del inglés *Genetic Algorithm*), el cual, por definición hace

evolucionar una población de individuos generada a partir de una colección de datos de partida, y la somete a diferentes etapas que la modifican de manera aleatoria imitando el proceso evolutivo en la naturaleza (principalmente por mutaciones y recombinaciones genéticas). El objetivo de estas alteraciones es seleccionar a los individuos con mayor capacidad de adaptación a los requisitos del problema y descartar los menos aptos. Al finalizar este proceso, la solución ofrecida se construye en base a la población final. En [KWG01] se describen en mayor detalle los principios de este tipo de estrategias.
El algoritmo genético implementado para deducir la naturaleza de las secuencias de alertas reportadas por el sensor asume como entrada, las salidas de la capa superior del esquema de correlación, es decir, las clasificaciones realizadas a nivel de paquete. Cada individuo es definido como un vector de dimensión $d$ representado por su genotipo $G = [G_1, \cdots, G_d]$, donde cada elemento $G_i$, $0 < i \leq d$ es el gen en su posición $i$. La elección de d tiene un impacto directo en la ejecución del algoritmo: cuanto mayor es el genotipo, mayor es la riqueza genética del individuo, lo que suele causar un efecto positivo en la solución del problema. Esto es debido a que los algoritmos genéticos son una familia de los algoritmos probabilistas, y por lo tanto, cuanto menor es $d$, mayor será el componente aleatorio en su resolución. En el llenado del genotipo de los nuevos individuos se considera el conjunto de clasificaciones de los paquetes de la secuencia. El conjunto de sus posibles clasificaciones puede contener etiquetas repetidas, por lo que aquellas que aparecen con mayor frecuencia deben tener un mayor protagonismo en la población.
Tal y como se ilustra en la Figura 7.5, a cada gen de los nuevos genotipos se le asigna un elemento aleatorio del conjunto de etiquetados de la secuencia, habiendo una mayor probabilidad de insertar las categorías más repetidas. Durante el entrenamiento de APAP se construye una tabla que asocia a las secuencias de alertas una posible amenaza. A partir de ella es posible calcular la función de aptitud de cada individuo. En este proceso se selecciona la mejor distancia Levenshtein entre su genotipo y las secuencias de la tabla [Lev66]. Consecuentemente, la función de aptitud de los mejores individuos devuelve el valor 0, indicando que la secuencia encaja perfectamente con la descripción de una amenaza concreta. El algoritmo ejecuta operaciones de cruce y mutación hasta que alcanza su condición de parada. Las mutaciones son modificaciones aleatorias de genes de cada individuo seleccionados a partir de una probabilidad dada y en un rango concreto. Los cruces seleccionan una pareja de individuos al azar y engendran un nuevo individuo cuyo genotipo es definido mediante el intercambio del material genético de sus ancestros. La condición de parada se produce cuando el algoritmo rebasa el máximo número de iteraciones preestablecido, o cuando un subconjunto significativo de la población devuelve un valor de aptitud óptimo. Una vez finalizada la ejecución, se define la solución: la secuencia a clasificar se añade a los grupos de amenazas a los que pertenecen los individuos resultantes con aptitud óptima. Su distribución en la población final determina la probabilidad de pertenencia a cada grupo. En la Tabla 7.3 se resumen los parámetros de configuración del algoritmo que mejor se adaptaron a los requisitos de APAP.

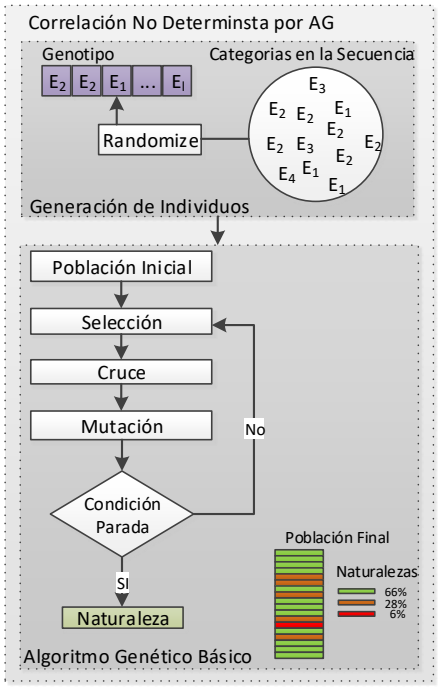



Figura 7.5: Instanciación de ND a nivel de secuencia.

Tabla 7.3: Configuración del algoritmo genético al complementar APAP.

| Parámetro | Valor |
|---|---|
| Longitud de genotipo | 6 |
| Longitud de población inicial | 200 |
| Longitud de población de trabajo | 50 |
| Probabilidad de cruce | 30% |
| Probabilidad de mutation | 10% |
| Función de aptitud | Distancia Levenshtein |
| Método de cruce | Intercambio de genes |
| Método de mutación | Modificación aleatoria de genes |
| Condición de parade | Todos los individuos alcanzan aptitud perfecta o se exceed el máximo número de iteraciones. |

## 7.3 Experimentación

Para evaluar la propuesta se ha utilizado una colección propia de capturas de tráfico real. En un primer enfoque se estudió la posibilidad de adoptar alguna de las diversas colecciones de muestras de dominio público, partiendo de la premisa de que estuviera aceptado como estándar funcional por la comunidad investigadores. Desafortunadamente, ninguno de ellos cumplía esta condición y su uso resultaba controvertido. Además, tal y como se discute en [SSTG12], estas colecciones no tienden a reflejar la actividad real de las

redes actuales, ya que son antiguas, presentan poca consistencia, diversidad y sus muestras a menudo presentan errores de captura o etiquetado. Por otro lado, es importante tener en cuenta que debido a motivos legales, la mayor parte de las colecciones de muestras de trazas de tráfico en redes de comunicaciones proveen de datos anonimizados, por lo que les ha sido sustraída su carga útil y parte de la información del encabezado. La falta de su carga útil afecta directamente al trabajo realizado, ya que se centra en la gestión de incidencias relacionado con sus contenidos anómalos. Además, ésta debe contener una selección variada de malware clasificado tanto a nivel de paquetes como de secuencias. En la extensa bibliografía revisada no se han hallado referencias a colecciones actuales que cumplan estos requisitos, pero constantemente se resalta la falta de coherencia entre las colecciones de muestras consideradas en la evaluación de IDS respecto a su entorno de monitorización real. Dada la dificultad de obtener una colección de muestras apropiada, que contenga datos reales, actuales, con carga útil y una diversidad de malware suficiente para demostrar la eficacia de la instanciación propuesta, se ha optado por utilizar una colección propia.

El conjunto de muestras considerado ha sido proporcionado por el Centro de Cálculo y Procesamiento de Datos de la Universidad Complutense de Madrid (UCM) y se ha obtenido al mismo tiempo que las muestras aplicadas en la evaluación de APAP (ver Sección 6.4 "Evaluación con tráfico HTTP real"). Por lo tanto, contiene muestras de trazas de tráfico con contenido tanto malicioso y legítimo monitorizado en la subred de la facultad de informática de la UCM. Las observaciones maliciosas identificadas han sido clasificadas a nivel de paquete y secuencia, acordes a los criterios establecidos por el marco propuesto. Sus etiquetados son resumidos a continuación (ver Tabla 7.4):

- Las alertas se clasifican en 5 grupos $A_1, \ldots, A_5$ teniendo en cuenta su probabilidad de ser dañinas: ($A_1$) *muy baja*, ($A_2$) *baja*, ($A_3$) *media*, ($A_4$) *alta* y ($A_5$) *muy alta*.
- El contenido anómalo se agrupa en 16 naturalezas diferentes $C_1, \cdots, C_{16}$ a nivel de paquete: ($C_1$) *troyano*, ($C_2$) *escalada de privilegios*, ($C_3$) *código ejecutable*, ($C_4$) *denegación de servicio*, ($C_5$) *software propietario*, ($C_6$) *enumeración poco sospechosa*, ($C_7$) *enumeración*, ($C_8$) *virus*, ($C_9$) *exploit*, ($C_{10}$) *fuga de información*, ($C_{11}$) *control remoto*, ($C_{12}$) *software privativo con acceso autorizado*, ($C_{13}$) *software privativo con acceso no autorizado*, ($C_{14}$) *spyware convencional*, ($C_{15}$) *spyware de gran impacto*, y ($C_{16}$) *otros*. La clasificación se establece en función de su carga útil, por lo que las amenazas menos intuitivas en este campo de información, como ($C_5$) *denegación de servicio* o ($C_{11}$) *control remoto*, deben mostrar contenido reconocible por técnicas de análisis de carga útil.
- Las alertas se dividen en 9 naturalezas diferentes a nivel secuencial: ($E_1$) *botnets*, ($E_2$) *malware ofuscado*, ($E_3$) *gusanos*, ($E_4$) *drive-by*, ($E_5$) *adware*, ($E_6$) *spyware*, ($E_7$) *virus*, ($E_8$) *troyano* y ($E_9$) *otros*. Éstas pueden deducirse a partir de información observable en los paquetes que las integran. Nótese que algunas de ellas están repetidas (i.e. ($E_8$) *troyano a nivel de paquete* y ($C_1$) *troyano a nivel de secuencia*). En estos casos se espera que la mayor parte las alertas en la secuencia sean del mismo tipo que categoriza la secuencia en sí.

Tabla 7.4: Contenido de muestras UCM en la experimentación.

| Grupo | ID | Descripción |
|---|---|---|
| Riesgo | $A_1$ | Muy bajo |
| | $A_2$ | Bajo |
| | $A_3$ | Medio |
| | $A_4$ | Alto |
| | $A_5$ | Muy alto |
| Naturaleza de paquete | $C_1$ | Troyano |
| | $C_2$ | Escalada de privilegios |
| | $C_3$ | Código ejecutable |
| | $C_4$ | Denegación de servicio |
| | $C_5$ | Software propietario |
| | $C_6$ | Enumeración poco sospechosa |
| | $C_7$ | Enumeración |
| | $C_8$ | Virus |
| | $C_9$ | Exploit |
| | $C_{10}$ | Fuga de información |
| | $C_{11}$ | Control remoto |
| | $C_{12}$ | Software privativo con acceso autorizado |
| | $C_{13}$ | Software privativo con acceso no autorizado |
| | $C_{14}$ | Spyware convenciona |
| | $C_{15}$ | Spyware de gran impacto |
| | $C_{16}$ | Otros |
| Naturaleza de secuencia | $E_1$ | Botnet |
| | $E_2$ | Malware ofuscado |
| | $E_3$ | Gusano |
| | $E_4$ | Drive-by |
| | $E_5$ | Adware |
| | $E_6$ | Spyware |
| | $E_7$ | Virus |
| | $E_8$ | Troyano |
| | $E_9$ | Otros |

## 7.4 Resultados

A continuación se describen los resultados obtenidos componente a componente al analizar tráfico real de la UCM.

### 7.4.1 Diagnóstico de anomalías

En la evaluación del componente AD se han realizado dos pruebas. La primera de ellas determina su capacidad de identificar falsos positivos, y por lo tanto consiste en la correlación de los paquetes que han forzado que el sensor emita falsos positivos. Como resultado, el 95.7% de falsos positivos fueron etiquetados exitosamente en los dos grupos de alertas de menor discordancia ($A_1$ y $A_2$), donde el 46% de ellas eran de riesgos muy

bajo ($A_1$) y el 50% de riesgo bajo ($A_2$) (ver Figura 7.6). El 4.3% restante fue agrupado erróneamente en los grupos de riesgo medio. En base a esta clasificación, si el sistema de detección rechazara las alertas con mayor similitud con el modo de uso legítimo de la red protegida (particularmente, los grupos de riesgo muy bajo y bajo), podría afirmarse que el 95.7% de los falsos positivos serían filtrados. Esto sería un comportamiento muy deseable, siempre y cuando las decisiones tomadas basadas en estos datos no tengan un impacto negativo en la tasa de falsos negativos del sistema.

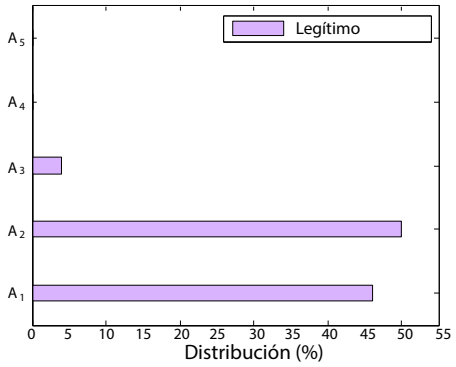



Figura 7.6: Distribución de falsos positivos al correlacionar tráfico en AD.

En la segunda prueba se mide la capacidad de detección de verdaderas amenazas tras tratar de reducir la tasa de falsos positivos, y por lo tanto, las tasas de falsos negativos que genera. Los resultados obtenidos demostraron que el 84.82% de las muestras analizaras fueron correlacionadas correctamente y asignadas al grupo de incidencias correcto. En la Tabla 7.5 y la Figura 7.7 se resume la distribución de alertas resultante. Es importante destacar que todos los errores de clasificación han llevado a asignar la incidencia al grupo inmediatamente más próximo al que corresponden, lo que conlleva que a pesar del error, su impacto no sea tan grande a la hora de tomar decisiones. Los errores potencialmente más peligrosos son los que asocian amenazas de riesgo alto con riesgo medio, ya que acarrean despliegues de contramedidas desproporcionados o insuficientes. Éstos abarcaron el 1.32% de los fallos. Los errores más comunes se concentraban en torno a las incidencias relacionadas con riesgos medios, las cuales podían resultar de riesgo bajo (7.3%) o de riesgo alto (32.7%). En base a estas observaciones es posible concluir que el filtrado de alertas por descarte de los grupos de riesgo más bajos ha sido una estrategia efectiva. Además, su repercusión en la tasa de falsos negativos es escasa, y con poca repercusión en las tareas de toma de decisiones.

Otro aspecto importante a tener en cuenta es el rendimiento del componente. El peor tiempo de procesamiento de la carga útil registrado por el sensor APAP fue de 171.104ms (al tratar secuencias de información de aproximadamente 200KB) El despliegue de la instancia del marco de correlación por medio de únicamente la activación de este componente llevó a un retraso en el peor de los casos de 173.127ms (171.104ms +2.023ms). Esto supone una sobrecarga máxima que se aproxima al 1% del coste de ejecución del sensor, confirmándose la capacidad de la implementación de operar en tiempo real.

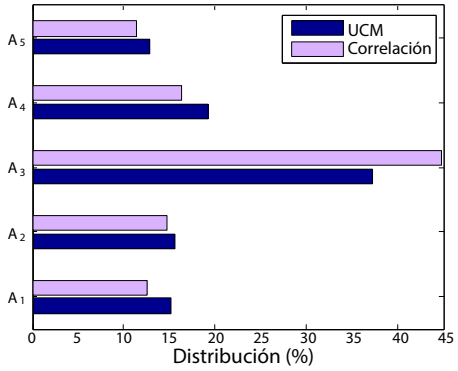



Figura 7.7: Distribución de alertas al correlacionar amenazas en AD.

Tabla 7.5: Distribución en grupos de las alertas correlacionadas.

| Anomalía | UCM 2011 (%) | Correlación (%) |
|---|---|---|
| $A_1$ | 15.1 | 12.3 |
| $A_2$ | 15.6 | 14.8 |
| $A_3$ | 37.2 | 44.8 |
| $A_4$ | 19.2 | 16.4 |
| $A_5$ | 12.9 | 11.4 |

### 7.4.2 Evaluación del ND a nivel de paquete

El componente ND de análisis a nivel de paquete ha sido evaluado considerando las 16 categorías $C_1, \cdots, C_{16}$ descritas en la sección previa. De este modo, todas las muestras de referencia con contenidos maliciosos que no han sido involucradas en el entrenamiento del sistema han servido de conjunto de evaluación. La distribución de las agrupaciones emitidas es resumida en la Figura 7.8 y la Tabla 7.6. El 99.512% de los etiquetados calculados han coincidido con los propuestos por el Centro de Cálculo y Procesamiento de Datos de la UCM, lo que demuestra la buena precisión obtenida. Esto se debe en gran parte al riguroso entrenamiento de la red neuronal en que ha sido instanciado, habiéndose asumido un error del 0.001. También es importante destacar que el error observado ha variado de manera representativa entre las diferentes naturalezas de las amenazas; por ejemplo, algunas de ellas han etiquetado perfectamente todas sus muestras, como es el caso de "troyanos", "escalada de privilegios" o "exploit". Sin embargo, otros grupos, como por ejemplo "denegación de servicio" o "enumeración" han resultado mucho más propensos a fallos. Estos últimos tienden a identificar discordancias asociadas a riesgos que típicamente no son detectados por el análisis de la carga útil, deduciéndose que la información que ofrecen habitualmente es menos significativa a la hora de identificar la intrusión.

En las pruebas de rendimiento se ha observado, que al igual que como sucedía al correlacionar las alertas por el estudio de su grado discordancia, el despliegue realizado facilita con éxito su operatividad en tiempo real. La sobrecarga máxima monitorizada

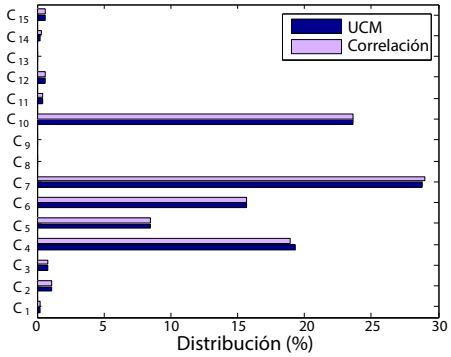



Figura 7.8: Distribución de alertas según el componente ND a nivel de paquete.

Tabla 7.6: Distribución de alertas según el componente ND a nivel de paquete.

| Categoría | UCM 2011 (%) | Correlación (%) |
|---|---|---|
| $C_1$ | 0.25 | 0.25 |
| $C_2$ | 1.13 | 1.13 |
| $C_3$ | 0.78 | 0.8 |
| $C_4$ | 19.27 | 18.93 |
| $C_5$ | 8.44 | 8.5 |
| $C_6$ | 15.7 | 15.7 |
| $C_7$ | 28.8 | 29 |
| $C_8$ | 0.01 | 0.01 |
| $C_9$ | 0.03 | 0.03 |
| $C_{10}$ | 23.61 | 23.62 |
| $C_{11}$ | 0.42 | 0.42 |
| $C_{12}$ | 0.67 | 0.64 |
| $C_{13}$ | 0.04 | 0.03 |
| $C_{14}$ | 0.27 | 0.31 |
| $C_{15}$ | 0.58 | 0.63 |
| $C_{16}$ | 0 | 0 |

ha sido del 2.3%, y el peor tiempo de procesamiento 175.135ms (171.104ms + 4.031ms). Nótese que el tiempo de procesamiento a nivel de paquetes de la instanciación propuesta es el mayor entre los obtenidos por los componentes AD y el ND a nivel de paquete. Esto es debido a que ambos son ejecutados en concurrencia. A la vista de estos resultados, es posible concluir que la penalización máxima de esta etapa de procesamiento de alertas sobre el rendimiento del sensor complementado es del 2.3%, un coste que es compensado con creces al considerar los beneficios que ofrece al gestionar las incidencias y mejora de las tareas de toma de decisiones.

### 7.4.3 Evaluación del ND a nivel de secuencia

Para la mejor comprensión de los resultados ofrecidos por la instanciación del componente ND a nivel de secuencias, es necesario tener en cuenta que fue desarrollada con la capacidad de proveer etiquetados no-deterministas. Al igual que en las pruebas anteriores, todas las alertas que no fueron utilizadas en el entrenamiento del sistema formaron parte del conjunto de muestras de evaluación. El 100% de ellas fueran correlacionadas con éxito en alguna de las posibles soluciones del algoritmo genético básico. En la Figura 7.9 se ilustran la tasa de acierto obtenida por cada naturaleza, y los posibles etiquetados emitidos. Esta información se complementa con la Tabla 7.7, donde se observa la distribución de cada categoría en las secuencias correlacionadas. Los aciertos logrados se han repartido siempre entre las dos opciones más probables, perteneciendo el 75.124% a la más probable, y el 24.876% restante a la segunda más probable. Se trata por lo tanto de una precisión deseada, capaz de proponer soluciones alternativas de gran significancia. De este modo, si por ejemplo un operador identifica una secuencia con un 45% de certeza de que contiene adware y un 55% de que se relaciona con una botnet, las contramedidas a aplicar se centrarán en el despliegue de medidas de mitigación contra botnets. Si éstas resultan poco efectivas, el operador dispone de una segunda opinión, pudiendo rectificar las acciones ejecutadas. Puede decirse que en la experimentación realizada la diferencia entre la primera solución planteada y la segunda representa el grado de homogeneidad de las soluciones alcanzadas. En ciertos contextos podría ser deseable que esta diferencia sea mayor o menor, dando pie a la necesidad de llegar a un calibrado acorde a las necesidades del contexto.

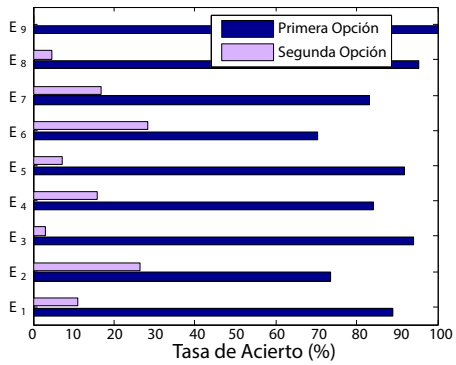



Figura 7.9: Tasa de acierto en las opciones más probables del ND a nivel de secuencias.

En las pruebas de rendimiento los resultados fueron considerablemente peores que en la etapa previa. En particular, se registró un tiempo de procesamiento máximo de 221.362ms. Teniendo en cuenta que 171.104ms es el retraso mayor registrado por APAP, la sobrecarga máxima registrada ha sido del 22.88% (171.104ms+50.258ms), lo que hace que la instanciación de este componente sea apropiada para el análisis forense, pero no para ofrecer correlación en tiempo real. Este hecho no compromete la eficacia de la propuesta, ya que en ningún momento se asumió como requisito de esta instancia la capacidad de operar en tiempo real, habiéndose sin embargo resaltado la importancia de mostrar un comportamiento no determinista.

Tabla 7.7: Tasa de acierto en las opciones más probables del ND a nivel de secuencias.

| Categoría | UCM 2011 (%) | Primera (%) | Segunda (%) |
|---|---|---|---|
| $E_1$ | 0.91 | 89.1 | 10.9 |
| $E_2$ | 0.14 | 73.4 | 26.6 |
| $E_3$ | 0.16 | 94.1 | 2.9 |
| $E_4$ | 23.18 | 84.1 | 15.9 |
| $E_5$ | 30 | 91.7 | 7.3 |
| $E_6$ | 43.2 | 70.5 | 28.5 |
| $E_8$ | 72.8 | 95.2 | 4.8 |
| $E_9$ | 0 | 100 | 0 |

# Capítulo 8

# Mitigación de DDoS en redes de nueva generación

Los ataques de denegación de servicio plantean una amenaza en constante crecimiento. Esto es debido principalmente a la tendencia al incremento en su sofisticación, facilidad de implementación, mejora de su capacidad de ofuscación y la existencia de métodos cada vez más eficaces para ocultar la identidad del atacante. Por otro lado, la evolución de las redes convencionales hacia escenarios auto-organizados, así como la adopción de tecnologías emergentes en su desarrollo (tales como redes definidas por software, virtualización de funciones de red, inteligencia artificial o computación en la nube, han impulsado el diseño de nuevas estrategias defensivas mucho más completas, consistentes, y con capacidad de adaptar los despliegues defensivos al estado actual del entorno protegido. En base a ello, este capítulo propone una estrategia de mitigación de ataques de denegación de servicio basados en inundación, por medio de la construcción de redes de sensores distribuidos con capacidad de adaptación a cambios producidos en el escenario a monitorizar. Cada uno de sus agentes adquiere la capacidad de identificar dichas amenazas, reducir su impacto y facilitar la localización del atacante. Esto se logra por medio de la emulación del comportamiento de los mecanismos defensivos biológicos presentes en la naturaleza, y en particular, en las estrategias inmunitarias propias de los seres humanos. La replicación de las respuestas adaptativas humanas además permite la delimitación de regiones de cuarentena en la red protegida y el mantenimiento de una memoria inmunitaria con capacidad de agilizar la detección de futuras amenazas. La propuesta ha sido evaluada con diferentes colecciones de trazas de tráfico de dominio público (KDD'99, CAIDA'07 and CAIDA'08), y mediante la simulación de redes de diferentes características basadas en muestras de tráfico monitorizado en la subred de la Facultad de Informática de la Universidad Complutense de Madrid.

El contenido del capítulo está organizado de la siguiente manera: en la Sección 8.1 se describen brevemente las características del sistema inmunitario de los seres humanos; en la Sección 8.2 se introduce su adaptación al problema de la mitigación de amenazas de denegación de servicio, haciéndose especial hincapié en la arquitectura de la propuesta, la emulación de las respuestas inmunitarias observadas en la naturaleza, implementación

y la revisión de su principales propiedades; en la Sección 8.3 se describen en detalle los procesos de detección de amenazadas e identificación de su origen llevados a cabo por los agentes inmunitarios; en la Sección 8.4 se detalla la experimentación realizada; finalmente, en la Sección 8.5 se discuten los resultados obtenidos.

## 8.1 El sistema inmunitario de los seres humanos

El sistema inmunitario es el conjunto de estructuras y procesos biológicos de los organismos, que tienen como objetivo protegerlos contra enfermedades, basándose para ello en la identificación de antígenos y su eliminación [Par15]. Los antígenos son los agentes biológicos externos capaces de dañar de alguna manera la anatomía del sistema anfitrión. Pueden formar parte de diferentes microorganismos, como hongos, bacterias o virus. En ocasiones pueden pertenecer al tejido orgánico de otro individuo, lo que añade complejidad a los procesos de detección. A las partes del antígeno que interactúan directamente con el sistema inmunitario se las denomina determinantes antigénicos o epítopos. La mayoría de los antígenos poseen múltiples epítopos, y son sus elementos de unión con los anticuerpos. Las distintas especies biológicas han desarrollado múltiples mecanismos inmunitarios, destacando entre ellos, y por su nivel de sofisticación, los de los animales vertebrados. Dichos sistemas constan de muchos tipos de proteínas, células, órganos y tejidos, los cuales se relacionan en una red elaborada y dinámica. Como parte de esta respuesta inmunitaria más compleja, el sistema inmunitario humano se adapta con el tiempo para reconocer antígenos específicos de manera más eficaz. A este proceso de adaptación se le llama inmunidad adaptativa o inmunidad adquirida capaz de poder crear una memoria inmunitaria [IM10]. A los mecanismos de defensa no adquiridos se les llama inmunidad innata, y habitualmente constituyen la primera línea de defensa frente a agentes externos [MJ02]. A continuación se describe el comportamiento de cada respuesta inmunitaria.

### 8.1.1 Inmunidad innata

La inmunidad innata es un mecanismo inespecífico de defensa, el en que cada agente es capaz de reconocer y/o eliminar diferentes tipos de antígenos [MJ02]. Está compuesto de barreras externas físicas o químicas, y de elementos defensivos internos [AUT06] que actúan tal y como es resumido a continuación.

- *Barreras externas*: La piel y las mucosas de cuerpo constituyen los sistemas defensivos externos, y por lo tanto los primeros en establecer contacto con los antígenos. La piel está compuesta por diferentes células fuertemente adosadas a la capa *epitelial* de la piel (*epidermis*). Esto ofrece una excelente defensa física frente a microorganismos invasores. La capa *epitelial* de las mucosas recubre las cavidades corporales y segrega un líquido denominado *mucus*, que entre otras funcionalidades tiene la de atrapar sustancias extrañas. Las barreras externas también incorporan otro tipo de líquidos con propiedades defensivas, como los producidos por el aparato lagrimal, las glándulas salivales o la secreción sebácea.

- *Defensas internas*: La segunda línea de defensa la componen proteínas antimicrobianas internas, *fagocitos*, células asesinas naturales NK (del inglés *Natural Killers*), inflamación y fiebre; los cuales operan de diferente manera: la sangre y el líquido intersticial contienen diferentes clases de proteínas antimicrobianas que inhiben el crecimiento de los microorganismos, entre las que cabe destacar el sistema de complemento. El sistema de complemento lo forma un grupo de proteínas ubicadas en el plasma sanguíneo y en las membranas plasmáticas, que en condiciones normales permanecen inactivas. Cuando se detecta una amenaza son activadas y aceleran el proceso de reacción inmune mediante la destrucción de determinados microbios, su colaboración en los procesos de *fagocitosis* y su participación en las respuestas inflamatorias. Por otro lado, los fagocitos son células plasmáticas que llevan a cabo el proceso de *fagocitosis*, es decir, de ingestión de microorganismos. Cuando se produce una infección, los fagocitos migran al área infectada, engullendo los antígenos. Esta acción puede destruirlos, debilitarlos o servir de marcado para estimular la actuación de otro tipo de células inmunitarias. Las células NK son linfocitos (pertenecientes al sistema linfático) con capacidad de atacar a las células portadoras de antígenos que presenten en su membrana proteínas anómalas. La inflamación es un intento de eliminar los antígenos presentes en el área comprometida, impedir la diseminación hacia otros tejidos, y preparar el área afectada para los procesos de reparación *tisulares*. Finalmente, la fiebre es la elevación de la temperatura corporal de manera anómala, con el fin de inhibir el crecimiento de determinados microbios e incrementar la velocidad de las reacciones que contribuyen a la reparación de tejidos.

### 8.1.2 Inmunidad adaptativa

La inmunidad adaptativa se caracteriza por presentar especificidad y capacidad de elaboración de una memoria inmunitaria. La primera indica que las respuestas adaptativas únicamente actúan frente a aquellos antígenos que iniciaron este tipo de respuesta [IM10], y la segunda su capacidad de aprendizaje de los resultados de las acciones previamente ejecutadas. Las células más importantes del sistema inmunitario adaptativo son los linfocitos y las células presentadoras del antígeno o CPA (del inglés *Antigen-Presenting Cells*). Los linfocitos son células especializadas fabricadas por células linfoides presentes en la médula ósea y que posteriormente migran a órganos linfoides (timo, ganglios linfáticos, bazo, etc.). Su función es la de capturar antígenos microbianos, exponerlos a otros agentes inmunitarios y colaborar activamente en las tareas para su eliminación. Se clasifican en 3 grupos en función de las moléculas que componen su Cúmulo de Diferenciación (CD) [Mur11]: Linfocitos B, Linfocitos $T$ ($T_h$, $T_c$) y células NK; descritos a continuación:

- *Linfocitos B.* (*bursa-dependientes*, la 'B' proviene del latín *Bursa fabricii*, el órgano en el cual se desarrollan los linfocitos $B$ en las aves). Se encargan de la producción de anticuerpos, en concreto *inmunoglobulinas* que se acoplan al antígeno a través de sus *epítopos* para su identificación de manera unívoca.
- *Linfocitos T.* (*timo-dependientes*, ya que se diferencian en el *Timo*).

Detectan antígenos proteicos asociados a moléculas del Complejo Mayor de Histocompatibilidad (MHC o CMH). El MHC es un conjunto de secuencias de genes alineados en una región grande y continúa del genoma de las células, que permite al sistema inmunitario determinar si se tratan de moléculas propias o invasivas (*antígenos*). Se divide en 3 subgrupos de genes, en base a su funcionalidad: MHC-I (codifican *glicoproteínas*), MHC-II (codifican glicoproteínas), MHC-III (codifican otras proteínas que desempeñan funciones inmunitarias, como las del sistema de complemento). Los Linfocitos $T$ se clasifican en los siguientes dos tipos:

  - *Linfocitos T colaboradores ($T_h$).* También conocidos como linfocitos CD4+. Reconocen antígenos presentados por el MHC-II. Se les denomina colaboradores porque están involucrados en la activación y guía de otras células inmunitarias.
  - *Linfocitos T citotóxicos ($T_c$).* También conocidos como linfocitos CD8+. Reconocen péptidos presentados por MHC-I y tienen capacidad lítica, es decir, de eliminación de las células del huésped del antígeno.

- *Células NK.* También conocidos como linfocitos grandes granulares. No tienen marcadores característicos. También participan en la inmunidad innata, con la capacidad de reconocer *lo propio* y destruir las células invasoras detectadas [VRM+11].

Por otro lado, las células CPA son un grupo diverso de células del sistema inmunitario cuya función es la de captar, procesar y presentar moléculas antigénicas sobre sus membranas para que sean reconocidos por otras células, especialmente linfocitos T. Existe una gran variedad de células que presentan estas propiedades, como las células dendríticas o los macrófagos. Cuando únicamente presentan proteínas del MHC-I son denominadas células CPA *no profesionales*. Cuando presentan proteínas tanto del MHC-I como del MHC-II se denominan células CPA *profesionales*. Hay dos tipos de respuesta inmunitarias adaptativas: respuesta humoral y respuesta intracelular. A continuación se describe brevemente cada una de ellas.

#### 8.1.2.1 RESPUESTA INMUNITARIA HUMORAL

La respuesta humoral del sistema inmunitario identifica y elimina microbios extracelulares y toxinas [Mur11, ALP14]. Los agentes encargados de llevar a cabo dichas acciones son los anticuerpos ($A_c$), un tipo de proteína producida en los linfocitos B. Los anticuerpos reconocen a los antígenos a través de las inmunoglobulinas presentes en su membrana. Es importante destacar que las inmunoglobulinas del cada anticuerpo únicamente son capaces de reconocer un tipo de antígeno. Además, la parte de las inmunoglobulinas encargada de la detección tiene una gran capacidad de mutación. Esto permite que al generarse nuevos anticuerpos en los linfocitos B, se produzca una gran variedad de inmunoglobulinas capaces de detectar diferentes tipos de antígenos. Los anticuerpos pueden estar unidos a linfocitos $B$ o distribuidos en el plasma sanguíneo. Cuando un anticuerpo reconoce un antígeno, se une a sus epítopos. A partir de esta unión pueden suceder dos cosas, En primer lugar, que un macrófago la identifique y fagocite. Los restos del antígeno

se verán reflejados en su complejo MHC-II, pudiendo ejercer la función de célula CPA. Por otro lado, si un linfocito $B$ de los que generan dicho anticuerpo lo identifica en el entorno protegido, neutralizará al invasor mediante endocitosis, es decir, engulléndolo y degradándolo. Al igual que sucede con los macrófagos, los restos de la deglución serán visibles en su membrana, incorporándose al MHC-II.
Cuando un linfocito $T$ colaborador $T_h$ detecta un leucocito $B$ con restos de endocitosis, procede a su activación. De esta manera el leucocito $B$ es transformado en una célula plasmática, que crecerá y se multiplicará produciendo anticuerpos específicos contra el invasor neutralizado. Pasado un cierto periodo de tiempo, los leucocitos $T_h$ y las células plasmáticas desaparecen por un proceso de autorregulación denominado apoptosis, o muerte celular programada. Tan solo una pequeña parte de las células plasmáticas permanecerá, pasando a formar parte de la memoria inmunológica. La memoria inmunológica permite la detección en el futuro de antígenos similares al detectado.

#### 8.1.2.2 Respuesta inmunitaria intracelular

La respuesta inmunitaria intracelular actúa como mecanismo de ataque en contra de los microorganismos intracelulares, como virus y algunas bacterias, capaces de sobrevivir y proliferar en el interior de los fagocitos u otras células del huésped [ALP14]. La respuesta celular principalmente se lleva a cabo mediante las acciones de dos linfocitos: $T_h$ y $T_c$. Cuando un linfocito $T_h$ detecta restos de antígenos en el complejo CHM-II de las células, se procede a su activación celular. Esto desencadena su multiplicación, generándose diferentes tipos de linfocitos $T_h$ denominados efectores, y clasificados en función de las *citoquinas* que producen. La generación de cada tipo de linfocito $T_h$ efector depende de la naturaleza del antígeno detectado. Algunos de los más importantes son $T_h$-1, $T_h$-2 y $T_h$-17; Los cuales son descritos a continuación:

- *$T_h$-1.* Migran a los tejidos infectados y colaboran en la activación de los macrófagos, ya que segregan fundamentalmente interferón $\gamma$. Juegan un papel esencial tanto en la defensa frente a los microorganismos intracelulares como en las inflamaciones.

- *$T_h$-2.* Permanecen principalmente en los tejidos linfoides y colaboran en la activación de los linfocitos B. Segregan mayoritariamente IL-4 (estimulación de la secreción de $I_g$-E, que a su vez activa los *mastocitos*) e $IL$-5 (activación de *eosinófilos*). Son importantes en las reacciones alérgicas y en la defensa frente a parásitos.

- *$T_h$-17.* Segregan $IL$-17 e $IL$-22. Son los principales mediadores en algunas reacciones alérgicas, y parecen estar implicados en el desarrollo de enfermedades como la esclerosis múltiple, la artritis reumatoide y la enfermedad inflamatoria intestinal.

Una segunda reacción inmunitaria celular es llevada a cabo por los linfocitos $T_c$ *citotóxicos*. Los linfocitos $T_c$ reconocen infecciones virales o desarrollos cancerígenos en su complejo CHM-I. Una vez detectados, destruyen las células infectadas mediante la segregación de moléculas (*perforina*, *granzimas*, *FasL*, etc.) que aceleran su proceso de apoptosis. Los restos de su eliminación activan leucocitos $T_c$, estimulando su multiplicación. Esto permite

hacer frente a amenazas similares, con una mayor precisión y eficiencia. Al igual que sucede en la respuesta humoral, pasado un cierto periodo de tiempo, tan solo una pequeña parte de las células linfáticas involucradas en la mitigación del ataque sobreviven a los procesos de apoptosis. Estas células constituyen parte de la memoria inmunitaria del individuo.

## 8.2 Reacciones inmunitarias artificiales en la defensa frente a DDoS

El diseño de sistemas bioinspirados para mitigar ataques basados en la inyección de grandes volúmenes de tráfico requiere tener en cuenta tanto las características de las amenazas a combatir, como los rasgos que mejor describan el entorno de monitorización. Con el fin de establecer las principales limitaciones y principios de diseño de la propuesta, se han asumido las siguientes precondiciones:

- Los ataques de denegación a ser tenidos en cuento pueden tener por origen una o varias fuentes. Se asume su capacidad de adaptación al estado de la red con el fin de alcanzar su objetivo.
- El tráfico malicioso puede dañar la red protegida en distintos puntos: en los nodos víctima y sus proximidades, elementos intermedios de encaminamiento o nodos próximos al origen de la amenaza. Cuando alguno de estos niveles es comprometido, existe una alta probabilidad de que su impacto se propague a los elementos colindantes.
- Cuando una región del entorno protegido se ve afectada por un ataque basado en inundación, se espera que el sistema propuesta sea capaz de reconocer el problema y aislarlo, estableciéndose de esta manera una zona de cuarentena. Ésta será desmantelada al desaparecer cualquier tipo de indicio que desenmascare futuras réplicas.
- A pesar de que una región de la red haya sido declarada en cuarentena, el tráfico legítimo debe ser capaz de atravesarla y transmitir información a sus nodos. En otras palabras, las regiones de cuarentena deben minimizar su impacto en la calidad de servicio del sistema, en caso contrario, logrando los atacantes su objetivo.
- Las acciones de mitigación deben ser ejecutadas por agentes situados lo más cerca posible del origen de la amenaza. De este modo se reduce la dimensión de las áreas de la red protegida afectadas por la incidencia.
- Las comunicaciones entre agentes inmunitarios deben realizarse a través de canales seguros, de esta manera evitando en lo posible que el atacante saque provecho de ellas.
- Los agentes inmunitarios deben poder activarse o desactivarse en función del estado de la red protegida.

- Las acciones defensivas deben ser proporcionales a los riesgos identificados. Consecuentemente, el impacto de las reacciones autoinmunes (habitualmente desencadenadas por falsos positivos) es reducido.
- Con el fin de mejorar las tareas de correlación de incidencias y su comprensión por operadores humanos, se asume la existencia de un registro que almacene la información necesaria para reconstruir el estado de la red en el momento de las incidencias, las acciones defensivas desencadenadas y su eficacia. Nótese que queda fuera del alcance de este capítulo el profundizar el cómo se gestiona dicha información, así como la asociación de eventos descubiertos en la red a la hora de reconstruir el escenario de la amenaza (por ejemplo, el relacionar un ataque de denegación de servicio con una botnet detectada previamente por otros elementos de seguridad). La bibliografía propone diferentes maneras de abordar este problema, destacando por su enfoque hacia las redes de quinta generación el trabajo de Barona et al. [BLMVGV17, BLVCMV$^+$17].

Aunque estas premisas recopilan una parte muy importante de los aspectos a tener en cuenta en el despliegue de la propuesta sobre redes reales, debe resaltarse la existencia de otras características que, por simplicidad y en pro de la mejor comprensión de la investigación realizada, no han sido consideradas. Por ejemplo, no se ha tratado el problema de la ofuscación del origen de las amenazas (Fast-Flux, algoritmos de generación de dominio, redes anónimas, etc.) [ZJT13, ADAH14]. Por otro lado, también se ha omitido la discusión acerca de su adaptación a las estrategias de evasión orientadas a la explotación de la implementación de los métodos de reconocimiento de patrones [OB15], su interoperabilidad con los distintos protocolos de seguridad, políticas de protección de datos o la dificultad de operar sobre encabezados de paquetes cifrados.

### 8.2.1 Arquitectura

El sistema propuesto presenta arquitectura distribuida, en el que cada uno de sus componentes asume varios roles de los esquemas inmunitarios biológicos. Su despliegue se centra en dos tipos de agentes: detectores $H$ ($D_H$) y detectores $A$ ($D_A$). Los primeros participan tanto en las respuestas innatas como adaptativas, y facilitan el mantenimiento de la memoria inmunitaria. Los detectores $D_A$ tienen la capacidad de detectar y mitigar intrusiones previamente identificados por sensores $D_H$, centrándose su participación en la respuesta adaptativa. En la Figura 8.1 se ilustra su despliegue, en el que además de estos dos sensores se distinguen otros elementos, los cuales son brevemente descritos a continuación:

- *Red Protegida.* La red protegida es el entorno en el que opera el sistema inmunitario artificial. Para que éste pueda llegar a ser completamente efectivo, tanto el atacante como la víctima deben formar parte de ella.
- *Nodos intermedios.* Todo nodo ubicado entre el atacante y la víctima sin capacidad de detectar, mitigar o contribuir en las respuestas inmunitarias es denominado nodo intermedio.

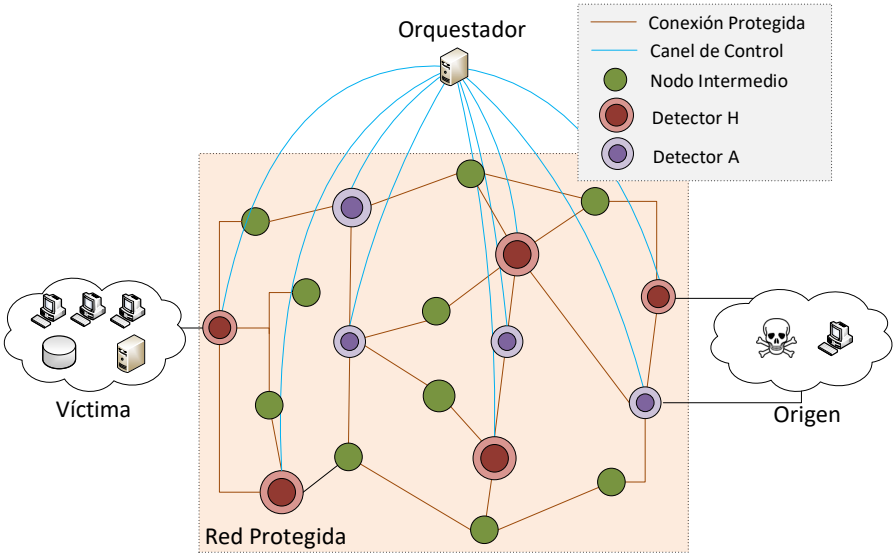



Figura 8.1: Distribución de los distintos componentes de la propuesta.

- *Detectores H*. Los agentes inmunitarios con capacidad de desencadenar respuestas inmunitarias adaptativas son denominados detectores $H$ o $D_H$. Estos nodos hospedan VNFs con capacidad de detectar/mitigar ataques de denegación de servicio.

- *Detectores A*. Los agentes inmunitarios que únicamente forman parte de las respuestas inmunitarias adaptativas son denominados detectores $A$ o $D_A$. Al igual que los sensores los agentes $D_H$, son desplegados en nodos que hospedan VNFs que implementan dichas capacidades.

- *Orquestador*. El orquestador realiza las tareas de gestión y orquestación (NFV M&O) (del inglés *NFV Management and Orchestration*) estandarizadas por la ETSI [ETS14], entre las que se incluye el organizar los recursos tanto físicos como software que dan soporte a la infraestructura de virtualización, o la dirección del ciclo de vida de las funciones VNFs. También hace de mediador entre agentes inmunitarios artificiales, participando en las tareas de delimitación del rango de las señales de activación de las respuestas adaptativas, gestión de las regiones de cuarentena, separación de plano de datos del plano de control y recopilación de información útil para el análisis forense de los incidentes detectados.

- *Conexión Protegida*. Cada conexión entre nodos de la red monitorizada sobre los que el sistema inmunitario artificial tiene la capacidad de actuar es denominada conexión protegida. Canal de Control. El canal de control es el plano de datos que comunica al Orquestador con el resto de elementos del despliegue realizado.

### 8.2.2 Respuestas inmunitarias artificiales

Cada una de las respuestas que es capaz de desencadenar el sistema propuesto emula a su análoga en el sistema inmunitario humano. Por lo tanto, y con inspiración en la naturaleza, se han implementado dos respuestas inmunitarias artificiales: innatas y adaptativas. Éstas son descritas a lo largo de esta subsección.

#### 8.2.2.1 Respuesta innata artificial

Al igual que en la naturaleza, la inmunidad innata que implementa el sistema propuesto actúa como primera línea de defensa. Su objetivo principal es reconocer y mitigar nuevas amenazas y proteger a los detectores $H$ de su denegación de servicio por exceso de información a tratar. El proceso inmunitario innato requiere que los agentes $D_H$ permanezcan activados a lo largo de la red protegida. Éstos implementan VNFs que operan a modo de IDS que monitorizan todo el tráfico que fluye a través de ellos en busca de indicios de comportamientos anómalos sospechosos. Al detectarse una posible amenaza, las acciones de mitigación se basan en la adopción de nuevas directivas que restrinjan las comunicaciones entre nodos, puertos y servicios involucrados en la trayectoria del ataque. En particular, se descartará el tráfico procedente del potencial origen de la amenaza.
La respuesta inmunitaria innata permite la ejecución eficiente de contramedidas, ya que los agentes que la desencadenan no requieren de comunicación con el Orquestador previa a su ejecución. Además, por medio del reconocimiento y eliminación de nuevas amenazas para el entorno protegido se emula el comportamiento de los agentes que participan en la respuesta inmunitaria innata de los seres humanos; se trata de un comportamiento similar al de las barreras externas, y por lo tanto carente de especificidad. Cabe destacar que, en este proceso los agentes inmunitarios artificiales operan como IPS convencionales con capacidad de aplicar contramedidas muy básicas. Pero a pesar de su sencillez, en la experimentación realizada se ha demostrado su eficacia. Nótese que por simplicidad, así como por facilitar la comprensión de este primer enfoque, el diseño e implementación de contramedidas más sofisticadas son delegados a investigaciones futuras.

#### 8.2.2.2 Respuesta adaptativa

La respuesta adaptativa se desencadena cada vez que un agente $D_H$ identifica una nueva amenaza, lo que implica la existencia de una pequeña memoria que almacene los últimos etiquetados realizados. En este contexto, determinar cuándo un ataque no ha sido visto con anterioridad, es decir es NSB (del inglés *Non-Seen-Before*), es el primer paso hacia construir una memoria inmunitaria. La decisión de cómo implementar dicha memoria no es un problema trivial, distinguiéndose dos esquemas de almacenamiento: centralizado y distribuido.
Cuando la memoria inmunitaria se implementa de manera centralizada, depende directamente del Orquestador. En este caso, si algún sensor identifica una posible amenaza, pregunta al Orquestador si es NSB, proceso en el cual se tiene en cuenta la información compartida por todos los sensores desplegados. Por otro lado, cuando la memoria es distribuida cada sensor dispone únicamente de la información que ha adquirido por sí

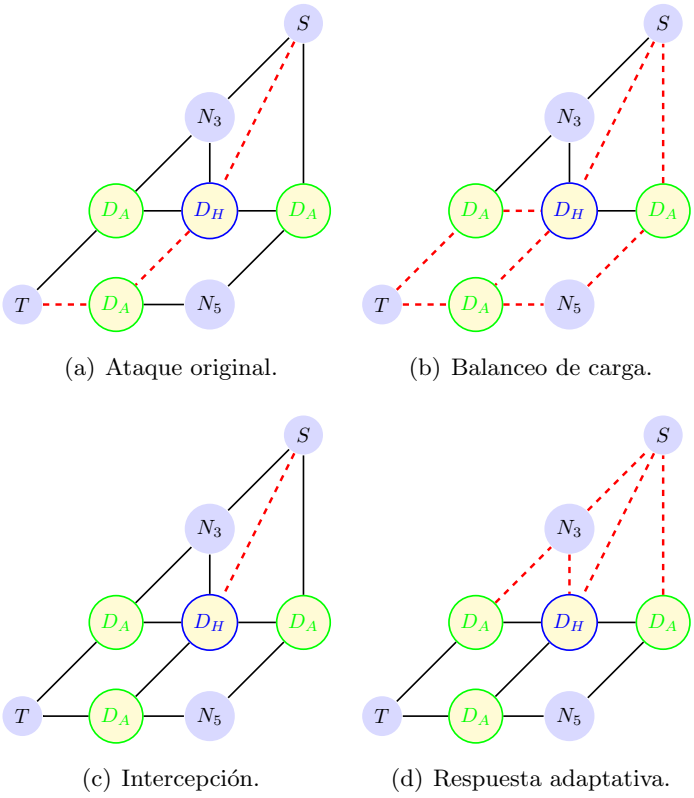
mismo o por el grupo de agentes desplegados en su vecindad. Esto quiere decir que en este caso un incidente puede ser NSB para algunos sensores, y para otros no. Esta segunda aproximación otorga a los agentes mayor autonomía, lo que acaba convirtiéndose en una ventaja en términos de eficiencia, facilidad de diseño y coordinación. Además, elimina el problema de la existencia de un único punto de fallo en términos de almacenamiento. Sin embargo, también acarrea desventajas: las VNFs que implementan las capacidades inmunitarias requieren un mayor coste de recursos computacionales (memoria, energía, hardware, etc.). Por otro lado, dado que ahora el Orquestador no administra la memoria inmunitaria, debe decidirse la estrategia con la que cada agente gestiona el conocimiento adquirido. Finalmente, los sensores ignoran si la información encapsulada en los paquetes monitorizados pertenece a flujos continuos de tráfico, reduciendo la contextualización de la información a tratar, y de este modo comprometiendo la resistencia de los métodos de detección a técnicas de ofuscación. Dada su gran eficiencia y simplicidad, y con el fin de emular con mayor precisión la autonomía de los agentes inmunitarios en la naturaleza, la implementación realizada ha abordado esta segunda opción.
Una vez que los sensores $D_H$ activan los agentes $D_A$ de su proximidad, los segundos instancian VNFs que analizan el tráfico que fluye a través de ellos. A diferencia que los agentes $D_H$, su método de detección varía el nivel de restricción en el que operan. De la manera propuesta es proporcional a la capacidad de inundación de la amenaza que dio pie a su instanciación, siendo normal que actúen de manera mucho más restrictiva que los agentes $D_H$ que detectaron el ataque inicial. Para evitar que esto lleve a la emisión de tasas altas de falsos positivos, se aplica especificidad; es decir, cada VNFs instanciada únicamente opera sobre tráfico originado por el posible ataque que llevó a su inicialización. De este modo, y al igual que en la naturaleza, la respuesta inmunitaria adaptativa artificial tiende a aumentar el número de efectivos capaces de reaccionar contra una posible amenaza, sirviendo los agentes $D_A$ como refuerzo de las habilidades innatas de los $D_H$. Las contramedidas aplicadas contra cada amenaza detectada son eficaces durante un periodo de tiempo limitado: al detectarse réplicas, se extiende su duración; en el caso contario se inicia un periodo de cuarentena, el cual se interrumpirá únicamente si se observan intrusiones similares, en este caso reiniciando el contador que marca su tiempo de expiración. La región de la red protegida cubierta por un conjunto de agentes $D_A$ coordinado por un mismo sensor $D_H$ es la región de cuarentena. Es el agente $D_H$ quien se encarga de su mantenimiento, siendo responsable de su activación/desactivación en función del estado de la red.

### 8.2.3 Implementación

El comportamiento de las respuestas inmunitarias artificiales se ilustra en la Figura 8.2, y es descrito por el siguiente procedimiento:

1. Inicialmente las VNFs gestionadas por los agentes $D_H$ analizan el tráfico que fluyen a través de ellos. Los agentes $D_A$ permanecen inactivos esperando a ser activados.

2. Cuando los agentes $D_H$ identifican tráfico malicioso, las conexiones relacionadas con la incidencia descubierta son bloqueadas a modo de respuesta innata. En este caso




(a) Ataque original. (b) Balanceo de carga.

(c) Intercepción. (d) Respuesta adaptativa.

Figura 8.2: Ejemplo de comportamiento del AIS propuesto.

el Orquestador activa los agentes $D_A$ de sus proximidades, y éstos despliegan las VNFs requeridas por las acciones de análisis/mitigación de la respuesta adaptativa. La notificación que lleva a su activación incluye información acerca del origen de la intrusión y las características del flujo de tráfico malicioso.

3. Los agentes $D_A$ recientemente activados comienzan a analizar el tráfico procedente de los nodos involucrados en las amenazas notificadas por los agentes $D_H$. A diferencia de estos últimos, el nivel de restricción de su umbral de decisión es calibrado acorde a las características de las intrusiones reportadas. Si reconocen alguna réplica de la amenaza original, actúan contra ella por medio del descarte de tráfico comprometido. De este modo si el ataque de inundación trata de tomar rutas alternativas a las que atraviesan los nodos $D_H$ para llegar a su víctima, deberá hacer frente a estos nuevos elementos defensivos.

4. En el caso de que diferentes agentes $D_H$ procedan a la activación de respuestas adaptativas contra un mismo ataque, las VNFs relacionadas con los agentes $D_A$ que tengan en común aplicarán las directivas más restrictivas.

5. Los agentes $D_A$ son desactivados tras transcurrir un periodo de cuarentena preestablecido sin tener noticias de réplicas del ataque original.

En el ejemplo, se muestra el comportamiento del AIS al ser expuesto a un ataque DDoS basado en inundación. En él se parte de la situación mostrada en la Figura 8.2(a), donde

el nodo $S$ es el origen de la amenaza, $T$ es su objetivo, y cada elemento $N_i$ es un nodo intermedio localizado en la posición $i$. Tal y como se muestra en la Figura 8.2(b), en el caso de que el sensor $D_H$ no sea capaz de reconocer la amenaza, ésta se propagara a lo largo de la red protegida siguiendo diferentes rutas según las políticas de balanceo de carga de la red. Pero si el ataque es detectado se ejecuta la respuesta inmunitaria innata. De este modo se procede al descarte de tráfico originado en $S$, ralentizando así su avance a través del entorno protegido, tal y como se muestra en la Figura 8.2(c). El tráfico malicioso tratará de abrirse camino por nuevos enlaces. Pero en este ejemplo la amenaza ha sido considerada NSB, y por lo tanto su descubrimiento también ha conllevado el desencadenamiento de la respuesta inmunitaria adaptativa. Tal y como se muestra en la Figura 8.2(d), los agentes $D_A$ en la vecindad del nodo $D_H$ han sido activados. Esto dificulta el avance de la intrusión por rutas alternativas, reforzando su capacidad de mitigación. En la Figura 8.3 se muestra un diagrama de flujo que resume las distintas etapas de procesamiento de información del sistema propuesto y su relación con las respuestas inmunitarias en que operan. En el Pseudodódigo 3 se resumen las respuestas inmunitarias implementadas.

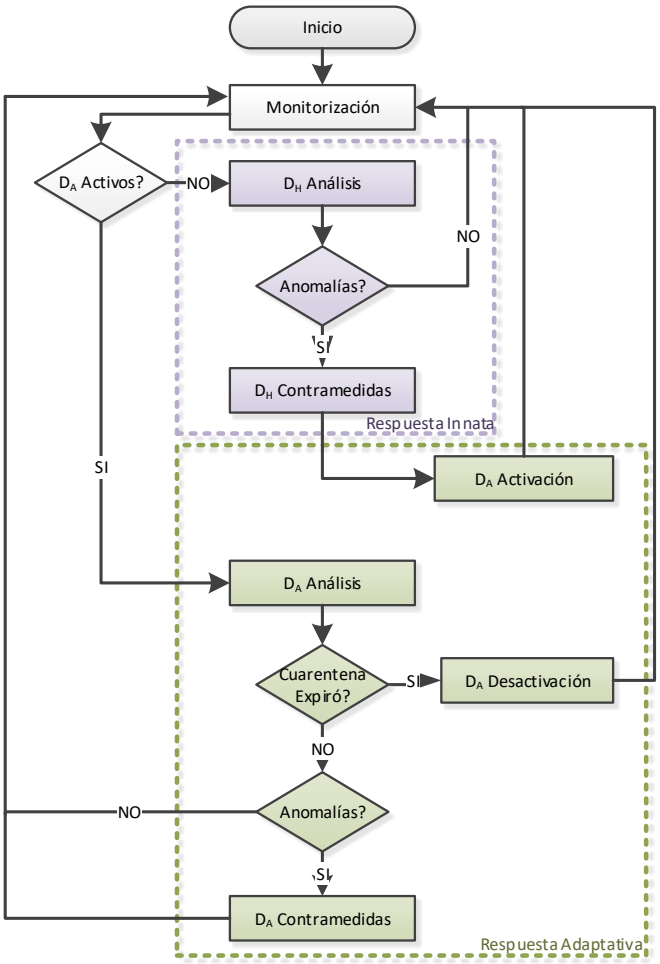



Figura 8.3: Diagrama de flujo que modela el comportamiento del AIS propuesto.

**Algoritmo 3:** Respuestas adaptativas para la mitigación de DDoS.

**Entrada:** El conjunto de observaciones $O = \{o_1, \cdots, o_n\}$. El intervalo de confianza $I$, $o < I \leq 1$.

**Salida :** Notificación de respuestas activadas y riesgos identificados.

```
Mientras el AIS esté activado hacer
    Detección(D_H);
    Si Error(Ô_n, O_n) > I en D_H entonces
        Si Respuesta adaptativa está activada entonces
            Filtrado_Tráfico(D_H);
            Reiniciar_Cuarentena(D_AH);
        Fin
        Activar_D_A();
        Reiniciar_Cuarentena(D_A);
    si no
    Fin
    Si Respuesta adaptativa está activada entonces
        Si Error(Ô_n, O_n) > I en D_A entonces
            Filtrado_Tráfico(D_A);
            Reiniciar_Cuarentena(D_A);
        si no
            Actualizar_Cuarentena(D_H);
        Fin
    si no
    Fin
    Si Exprira_Cuarentena(D_A) entonces
        Replegrar_DA(D_A);
    Fin
Fin
Mostrar respuestas activadas y anomalías descubiertas;
```

### 8.2.4 Propiedades

Una de las principales características de la estrategia propuesta es su capacidad de auto-calibrar el despliegue defensivo. Es habitual que la implementación de elementos de seguridad conlleve una penalización en la calidad de servicio del entorno protegido, debido entre otros factores a su consumo de recursos, retardos relacionados con el procesamiento de información, reducción de escalabilidad e interoperabilidad, o la ejecución de acciones de mitigación contra objetivos incorrectos debida a etiquetados erróneos [BAG15]. Con el fin de elaborar estrategias efectivas para la seguridad de la información, los operadores encargados de su configuración deben tener en cuenta diferentes criterios (objetivos de la organización a proteger, políticas, activos identificados y su valoración, etc.), así como la previa identificación y evaluación de los riesgos a considerar. Por medio de su estudio se facilita la decisión de las mejores contramedidas, las cuales han de ser proporcionales al

valor de los activos protegidos y las amenazas a las que son expuestos [OAGVSO+16]. El esquema propuesto facilita la adaptación del despliegue defensivo a las circunstancias de la red, permitiendo la configuración de la cantidad de sensores instanciados y el ámbito en el que actúan, sus limitaciones, umbrales de decisión y fortalecimiento frente a métodos de evasión. De entre sus ventajas más representativas es importante resaltar: bajo consumo de recursos y escaso consumo de ancho de banda, autoorganización del esquema defensivo y su *modus operandi* en función de la intensidad de las amenazas detectadas, incremento de efectivos en las regiones más afectadas por las intrusiones, y recalibrado de los umbrales de decisión en función del estado de la red. Éstas facilitan su alineamiento con las necesidades reales de los sistemas de gestión de seguridad de la información, y unifican en un único despliegue los elementos encargados de detectar los ataques, mitigarlos e identificar su origen, planteando una solución completa. Además de estas características, el sistema propuesto adopta las propiedades principales de los sistemas inmunitarios en la naturaleza, adaptándolas a la lucha contra la denegación de servicio distribuida de las siguientes maneras:

- *Respuestas innatas y adaptativas.* La propuesta tiene capacidad de reacción frente a ataques DDoS no reconocidos con anterioridad. Una vez detectados, fortalece las medidas defensivas contra futuras réplicas.

- *Especificidad.* En la naturaleza cada célula inmunitaria reacciona contra un único tipo de antígeno. En esta aproximación, la respuesta innata es común ante cualquier tipo de amenaza detectada. Sin embargo, la reacción inmunitaria adaptativa únicamente afecta a la detección/mitigación del ataque de inundación que previamente la ha desencadenado.

- *Clonalidad.* Al detectarse un antígeno desconocido, las respuestas adaptativas biológicas clonan las células que han sido capaces de reconocerlo. De este modo aumentan las posibilidades de identificar sus réplicas. En esta aproximación sucede lo mismo al detectarse ataques DDoS desconocidos.

- *Memoria inmunitaria.* Tanto en la naturaleza como en la propuesta, al detectarse una amenaza las contramedidas adoptadas son conservadas durante un periodo de tiempo. Esto permite reaccionar con mayor eficacia ante futuras réplicas.

- *Autorregulación.* Tras la respuesta adaptativa, el sistema inmunitario humano es regulado por la apoptosis de gran parte las nuevas células. En esta propuesta, las nuevas medidas defensivas también son reducidas tras pasar un determinado periodo de tiempo sin detectarse réplicas de la misma amenaza.

- *Autonomía.* En ambos casos las entidades del esquema defensivo operan sin un control centralizado, y tienen la capacidad de tomar decisiones propias.

- *Diversidad.* En la naturaleza, el conjunto de agentes inmunitarios debe de ser capaz de detectar cualquier tipo de antígeno. El sistema propuesto tiene la capacidad de identificar y mitigar cualquier tipo de ataque DDoS basado en inundación.

## 8.3 Detección de ataques de inundación

A continuación se describen los procesos de detección de amenazas e identificación de sus orígenes llevados a cabo por los agentes que integran el AIS propuesto. Con este fin se revisan las métricas consideradas, métodos de predicción, construcción de umbrales adaptativos y reconocimiento de elementos de red comprometidos.

### 8.3.1 Métricas

El tráfico monitorizado es analizado por medio del estudio de la entropía de su distribución. Con este fin se buscan variaciones inesperadas en el volumen de datos transmitidos que descubran comportamientos discordantes. Por lo tanto, cuando los agentes inmunitarios actúan como sensores, en realidad están operando como sistemas de detección de intrusiones basados en el reconocimiento de anomalías [ZJT13]. El uso de esta técnica permite satisfacer la propiedad biológica de diversidad, según la cual el agente inmunitario es capaz de detectar cualquier tipo de antígeno, incluidos aquellos que no han sido reconocidos con anterioridad. La decisión del estudio de las variaciones en la entropía con dicho fin no plantea de por sí una solución nueva, habiendo sido demostrada su efectividad respecto a otras métricas a lo largo de la bibliografía [OB15]. Esto es debido principalmente a que la precisión que ofrece depende en menor medida que el resto de métricas, del modo de uso de la red protegida, lo que resulta una gran ventaja al operar en redes de alta heterogeneidad como lo son las de nueva generación. La entropía tradicional fue adaptada por primera vez al área de la teoría de la información por Shannon en el año 1948 [Sha48]. Fue considerada una medida de fluctuación en variables cualitativas, a veces descrita informalmente como "el grado de impredictibilidad de las variables observadas". Dada la variable cualitativa $X$, el conjunto finito $x_1, x_2, \cdots, x_n$ y sus probabilidades $p_1, p_2, \cdots, p_n$, la entropía de Shannon se describe por medio de la siguiente expresión:

$$H(X) = \sum_{i=1}^{n} p_i \log_a \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log_a p_i \tag{8.1}$$

donde $\log_a b \times log_b x = log_a x$. Si la variable $X$ es determinista entonces $H(x) = 0$. Esto significa que toda probabilidad $p_i$ es igual a 0 a excepción de una, que tendrá valor 1. En la actualidad se aplican diferentes generalizaciones de esta entropía adaptadas a sus diferentes casos de uso. De entre ellas el sistema propuesto implementa la de Rènyi. Esta decisión fue tomada en base a estudios como el de Bhuyan et al. [BBK15], en los que se ha destacado su eficacia en la detección de ataques DDoS basados en inundación. La entropía de Rènyi viene dada por la siguiente expresión:

$$H(X) = \frac{1}{1-\alpha} \log_2 \sum_{i=1}^{n} p_i^{\alpha} \tag{8.2}$$

donde el parámetro $\alpha$ indica su orden, cumpliéndose $\alpha \geq 0$ y $\alpha \neq 1$. Nótese que la entropía de Shannon es el caso particular de la generalización de Rènyi en la que $\alpha = 1$.
En el sistema propuesto la variable $X$ se define como el volumen de tráfico que fluye

entre dos elementos de red (del nodo $A$ al nodo $B$) destinado al puerto $C$ del segundo. La probabilidad $P$ implica que $p_i$ represente la frecuencia de ocurrencia en el tráfico monitorizado de paquetes enviados desde cierto origen $A$, hacia cierto destino $B$ y su puerto $C$. Por lo tanto, se cumple que:

$$p_i = \frac{a_i}{No.packets} \tag{8.3}$$

donde $a_i$ es el número total de paquetes que cumplen la tupla previamente descrita y $No.packets$ es el total de paquetes observados. A lo largo del proceso de análisis, el sistema propuesto trata las variaciones en la entropía como series temporales univariables de N observaciones expresadas de la siguiente manera:

$$H_\alpha(X)_{t=0}, H_\alpha(X)_{t=1}, \cdots, H_\alpha(X)_{t=N}; (H_\alpha(X)_{t=0}^N) \tag{8.4}$$

### 8.3.2 PREDICCIÓN DE VARIACIONES EN LA ENTROPÍA

La predicción de las variaciones en la entropía del volumen de tráfico monitorizado permite inferir su comportamiento. Cuando los valores observados no se parecen a lo estimado, se ha hallado una situación inesperada que podría ser indicio de algún incidente en la red. Al estimarse la entropía de las observaciones venideras se tiene en cuenta que la serie temporal $H_\alpha(X)_{t=0}^N$ que describe su evolución puede experimentar cambios en su tendencia, periodo y estacionalidad a lo largo del tiempo. También se asumen dos restricciones: los métodos de predicción a implementar han de ser efectivos con pocas observaciones de referencia, y deben operar de manera eficiente en tiempo real. Debido a esto el sistema propuesto implementa el algoritmo de triple alisamiento exponencial propuesto por Holt-Winters. Esta decisión es respaldada por publicaciones como [GD80, Gro73], donde se demuestran estas cualidades y además, que el considerar parámetros como la tendencia o la estacionalidad de la serie temporal cuando éstos son poco representativos, lleva a errores de predicción poco significativos. También cabe destacar su gran eficiencia frente a métodos basados en la construcción de modelos autorregresivos. Nótese que con el fin de evitar la confusión que pueda llevar al lector considerar un parámetro $\alpha$ para definir el rango de la entropía de Rènyi, y otro parámetro $\alpha$ para configurar el factor de alisado del método Holt-Winters, en el resto del capítulo $H_\alpha(X)$ es resumido como $H(X)$, y el resto de símbolos $\alpha$ hace referencia al parámetro de ajuste del alisamiento de la serie temporal.

El método Holt-Winters permite inferir la siguiente observación $H(X)_{t+1}$ por medio del análisis de tres componentes de la serie temporal ($B$, $T$ y $S$) definidos mediante las siguientes expresiones recursivas:

$$B_t = \alpha(H_i - S_{t-N}) + (1-\alpha)(B_{t-1} + T_{t-1}) \tag{8.5}$$

$$T_t = \beta(B_t - B_{t-1}) + (1-\beta)T_{t-1} \tag{8.6}$$

$$S_t = \gamma(H_t - B_t) + (1-\gamma)B_{t-n} \tag{8.7}$$

donde $B_t$ es la estimación base en $t$, $T_t$ es la estimación de la tendencia y $S_t$ es la estimación de su componente estacional. Los parámetros $\alpha$, $\beta$ y $\gamma$ están en el rango $0 < \alpha, \beta, \gamma < 1$, y facilitan el ajuste del alisado. La predicción $H_{t+1}$ habitualmente se calcula mediante la combinación de los componentes de la serie temporal de manera aditiva o multiplicativa. En la experimentación se ha aplicado su versión aditiva, ya que se ha asumido que el patrón estacional es independiente de la tendencia. En consecuencia, la futura observación se calcula de la siguiente manera:

$$H(X)_{t+1} = B_t + T_t + S_t \tag{8.8}$$

Otro importante aspecto a tener en cuenta es la estrategia de inicialización de los casos base $B_0$, $T_0$ y $S_0$ de las expresiones recursivas. En su decisión se ha asumido que cuando no se espera tendencia o estacionalidad en la serie temporal, la inicialización de los estimadores basada en las últimas observaciones realizadas es preferible frente a medidas globales. El método de inicialización implementado se describe en [MWH98], donde además se ha demostrado su eficacia en casos de uso similares. Al igual que en dicha publicación, se han considerado las últimas 24 observaciones, iniciándose los estimadores de la siguiente manera:

$$B_0 = \bar{M_1} \tag{8.9}$$

$$T_0 = \frac{\bar{M_2} - \bar{M_1}}{12} \tag{8.10}$$

$$S_{t-12} = \frac{p_t}{M_1} \tag{8.11}$$

donde $M_1$ resume la primera docena de observaciones y $M_2$ la segunda. El ajuste de $\alpha$, $\beta$ y $\gamma$ se logra por medio del cálculo de los valores que minimicen el error cuadrático medio o SSE (del inglés *Sum of the Squared Errors*) de las predicciones realizadas, definidos como:

$$SSE(\alpha, \beta, \gamma) = \sum_{t=1}^{N} (H(X)_t - \widehat{H(X)}_{t_{|t-1}})^2 \tag{8.12}$$

### 8.3.3 Definición de intervalos de predicción

Para evaluar la variación de la entropía de $X$ en la observación $t$ se construyen dos umbrales adaptativos: el umbral superior $T_h$ y el umbral inferior $T_l$. A partir de ellos se define el intervalo de predicción de los sensores, tal y como es descrito en la bibliografía centrada en el alisamiento exponencial [HKOS05]:

$$Th_t(t) = p_0 + K \times \sqrt{var(E_t)} \tag{8.13}$$

$$Tl_t(t) = p_0 - K \times \sqrt{var(E_t)} \tag{8.14}$$

donde $E_t$ es el error de predicción en $t$ calculado como la diferencia entre la predicción y la observación realizada en $t$ , y $p_0$ es la predicción de la última observación. La varianza $Var(E_t)$ se calcula considerando los errores de predicción de las últimas $t$ observaciones.

Los umbrales se construyen en base al parámetro $K$, el cual permite calibrar el nivel de sensibilidad del sensor. De este modo los agentes $D_H$ pueden modificar el grado de restricción en el que actúan, asignándole por defecto el valor $Z = \frac{\alpha}{2}$. Nótese que relacionar la variación de estos umbrales con la distribución normal es una aproximación frecuente en la bibliografía. Además, tal y como se demuestra en [MWH98], si la serie temporal no converge hacia dicha distribución, el error producido es poco representativo. Además, el margen de error de ambos intervalos es del orden $100(1 - \alpha)$.
Por otro lado, y como parte de la respuesta adaptativa, los agentes $D_A$ tienen la capacidad de modificar la restricción de los umbrales en base a las observaciones realizadas por los agentes $D_H$ que los activan. Su parámetro $K$ es determinado por la siguiente expresión:

$$K(t) = K_{prev}(1 - \frac{Vol_{atk}}{Vol_{leg}}) \tag{8.15}$$

donde $K_{prev}$ es la última configuración de $K$, $V_{atk}$ es el volumen total de tráfico monitorizado durante el ataque y $V_{leg}$ es el volumen total de tráfico monitorizado en la última observación de circunstancias legítimas. En la Figura 8.4 se ilustra un ejemplo de serie temporal que describe la entropía del volumen de tráfico observado y su predicción, en uno de los experimentos realizados. El tráfico legítimo fluye a través del sensor hasta $t = 56$, momento en el cual se observa la inyección de un gran volumen de tráfico. La Figura 8.5 muestra los intervalos de predicción construidos, y el cómo al producirse la fluctuación, la observación supera los límites previamente calculados. En consecuencia, el agente informa del incidente al Orquestador y procede al descarte de paquetes malintencionados como medida de mitigación.

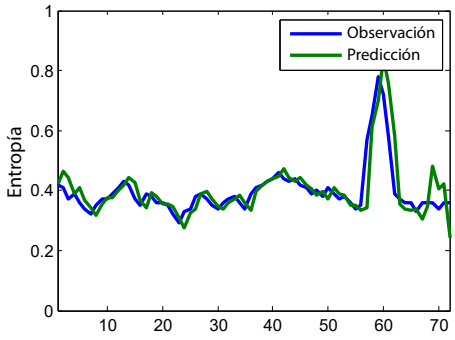



Figura 8.4: Ejemplo de predicción de entropía.

### 8.3.4 IDENTIFICACIÓN DE ORIGEN DEL ATAQUE

Antes de la ejecución de acciones de mitigación, y con el fin de conservar la propiedad inmunitaria de especificidad en la respuesta adaptativa, se requiere identificar el conjunto de posibles orígenes del ataque, facilitando así el actuar únicamente sobre tráfico procedente de ellos. Con este fin se lleva a cabo estudio de las diferentes probabilidades $p$ en el instante $t$ en que se detectó la incidencia. Su análisis asume las siguientes premisas:

- El ataque puede originarse desde diferentes direcciones IP.

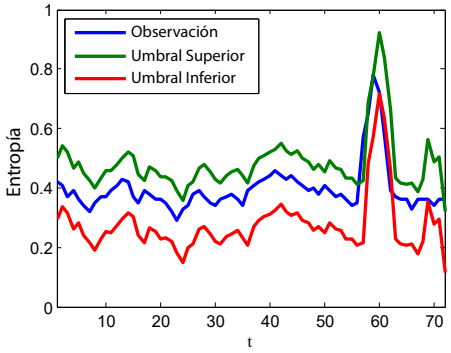



Figura 8.5: Ejemplo de intervalos de predicción.

- El ataque puede dirigirse contra diferentes elementos de red, caracterizados pos sus direcciones IP y puertos destinatarios.
- Las rutas {dirección IP origen, dirección IP destino, puerto destino} con mayor $p$ tienen una mayor probabilidad de verse involucradas en la intrusión.

En base a esto, las diferentes instancias de $X$ son agrupadas tomando como eje de clasificación su $p$. Dada la naturaleza de la información procesada, esta clasificación se lleva a cabo mediante un algoritmo de agrupamiento no supervisado; en concreto, se ha implementado el método k-medias [Jai10] con autoajuste del número de clases $K$ por el método Elbow [XW05]. Se ha elegido K-medias por ser especialmente tolerante a la presencia de discordancias en las observaciones a agrupar y a errores en las distancias de similitud consideradas. El método Elbow compensa su mayor desventaja: la necesidad de predefinir el número de grupos a establecer. Tras la definición de grupos, aquellas muestras agrupadas en las clases con mayor probabilidad $p$ son consideradas sospechosas, y puestas en cuarentena. Es importante destacar el hecho de que con el fin de facilitar la comprensión del esquema propuesto, y tal y como ha sido asumido en sus principios de diseño, se ha pasado por alto el problema de la detección de atacantes ocultos, lo cual requeriría implementar funcionalidades de rastreo a nivel de orquestador y considerar información que únicamente pueden ofrecer elementos desplegados en las conexiones troncales de la red (del inglés *backbone*) [YBV15] (lo que se aleja mucho del objetivo de este capítulo). No obstante, las acciones de mitigación implementadas son efectivas incluso sin conocer con precisión su origen, asumiendo como posible fuente de la inundación el conjunto de nodos comprometidos.

## 8.4 Experimentación

En los sensores desplegados, cada observación realizada está delimitada por las métricas extraídas de un número fijo de paquetes tratados en su orden de llegada. Como alternativa es frecuente su particionado por intervalos de tiempo. Tal y como discuten Ozcelik et al. [OB15] ambos métodos plantean ventajas y desventajas, siendo el primero más eficiente en el análisis de colecciones de capturas de trazas de tráfico, como sucede en el principal caso de uso del AIS propuesto. Por lo tanto, se ha considerado una ventana deslizante de

tamaño N que recorre los últimos paquetes capturados y calcula el volumen de la entropía de cada flujo de datos en base a su contenido. Esta restricción es necesaria para asegurar que los algoritmos sean computables, evitándose situaciones como $N \to \infty$. La propuesta ha sido evaluada en dos etapas: en primer lugar, se ha medido la eficacia de los sensores por separado en la detección de ataques DDoS. Por otro lado, se ha valorado la eficacia de diferentes características del despliegue del AIS. A continuación se describe cada uno de estos pasos experimentales.

### 8.4.1 Evaluación de la precisión de los agentes inmunitarios

Dada la controversia que suscita la elección de método más apropiado para la evaluación de sistemas de detección de intrusiones para la identificación de ataques DDoS, se ha optado por seguir el esquema propuesto por Kumar et al. [KS13]. Con este fin se plantea el uso de dos colecciones de muestras: KDD'99 [KDD99] y CAIDA'07 [CAI07]; y la generación de ataques de inundación con la herramienta DDoSIM [DDo13] en un caso de uso real. Las dos primeras son colecciones de trazas de tráfico de dominio público que han servido a lo largo de los años para comparar los diferentes detectores de ataques DDoS que se han ido publicando. A pesar de su antigüedad, su uso sigue siendo habitual con la finalidad de facilitar la comparación de nuevas propuestas con las publicaciones en la bibliografía. Por otro lado, el análisis de trazas actuales ofrece una visión más realista de la eficacia del AIS en escenarios reales, pero dificulta su comparación. A continuación se describe las características de cada prueba realizada y la colección de muestras de referencia que ha involucrado:

#### 8.4.1.1 Método KDD'99

La colección de muestras KDD'99 [KDD99] es una de las más referenciadas en la bibliografía, y según Bhatia et al. [BSMT14], posiblemente la única que ofrece un etiquetado fiable. Fue creada en el año 1999 en el marco de la competición KDD Cup a partir de trazas de tráfico extraídas de las capturas publicadas en DARPA'98 [Lab98]. En esta competición cada participante debía construir un NIDS capaz de distinguir entre conexiones comprometidas (clase *bad*) y conexiones legítimas (clase *good*). Cada muestra a considerar ofrece 41 características distintas, y los ataques a tener en cuenta se dividieron en cuatro grandes categorías: DoS (ej. clase *syn flood*), R2L (acceso no autorizado a máquina remota, ej. clase *guessing password*), U2R (acceso no autorizado a servicio local con privilegios de superusuario, ej. *clase buffer overflow*) y *probing* (enumeración, ej. clase *port scanning*). Parte de las muestras de la colección eran originalmente consideradas en el entrenamiento de los sensores, y el resto para su evaluación. Sin embargo, dado que el AIS propuesto carece de la primera etapa, todas las muestras (a excepción de las $N$ primeras reservadas para la inicialización de los modelos predictivos) son utilizadas para medir su eficacia. Es importante destacar que tanto la antigüedad de las muestras de DARPA'99 como el descubrimiento de irregularidades en su contenido poco a poco ha llevado a su descredito por parte de la comunidad investigadora. Frecuentemente es tachada de poco representativa y carente de la heterogeneidad que presentan las redes

actuales. También es muy criticado el hecho de que contiene muestras de ataques en desuso y la presencia de errores en el proceso de captura que han llevado a la repetición de muestras o el desequilibrio entre clases. Tal y como subrayaron Viswanathan et al. [VTN13], esto lleva al error de considerar que los resultados que ofrecen son escalables a entornos de monitorización reales. Pero a pesar de las críticas recibidas, sigue siendo uno de los métodos más aplicados, debido a su incuestionable utilidad a la hora de contrastar la eficacia de los nuevos sensores respecto a los propuestos en publicaciones previas, así como a la dificultad administrativa que supone la publicación de colecciones de trazas completas, sin anonimizar y etiquetadas; habiendo muy pocas alternativas que reúnan todos estos requisitos.

#### 8.4.1.2 CAIDA'07/08

La colección de muestras CAIDA'07 [CAI07] provee muestras de trazas de tráfico que forman partes de ataques DDoS basados en inundación (principalmente inundación ICMP, SYN y HTTP) monitorizados en agosto del año 2007. Éstas son distribuidas en diferentes archivos de extensión PCAP separados en intervalos de tiempo de cinco minutos. Tal y como describe su documentación, después del proceso de captura se eliminó gran parte de su contenido no malicioso, dejando un subconjunto de trazas de especial utilidad para calcular las tasas de acierto de los sensores. Sin embargo, para el cálculo de las tasas de falsos positivos se requiere información adicional, lo que a menudo lleva a su complementación por la muestra de capturas de tráfico CAIDA'08 [CAI08], tal y como propusieron Kumar et al. [370]. Ambas colecciones reúnen capturas realizadas en los centros de datos *Equinix* de San José y Chicago, y constituyen el método de evaluación tradicional con mayor similitud con las redes actuales, facilitando además la comparación de los resultados obtenidos con los del resto de trabajos en la bibliografía.

#### 8.4.1.3 DDoSIM y tráfico UCM

Este escenario de evaluación corresponde con un caso de uso real que combina el análisis de tráfico habitual monitorizado en la subred de la Facultad de Informática de la Universidad Complutense de Madrid (UCM) con el análisis de ataques DDoS basados en inundación inyectados desde la herramienta DDoSIM [DDo13]. Por lo tanto, las amenazas generadas actúan contra la capa de aplicación y constituyen intentos de denegación basados en desbordar a la víctima a base de peticiones HTTP y TCP. Durante la ejecución de los ataques, DDoSIM emula el comportamiento de una red de zombis capaces de iniciar sesión en los servidores víctima por medio de la asignación de direcciones IP aleatorias. Una vez establecidas las sesiones, procede al envío masivo de solicitudes. Las muestras de trazas generadas son agrupadas en dos clases: *legítimas* y *maliciosas*, siendo ambas constituidas por 40000 paquetes en formato PCAP; la primera se utiliza en el cálculo de la tasa de falsos positivos de los sensores, y la segunda para determinar su tasa de acierto.

### 8.4.2 Evaluación del sistema inmunitario artificial

Con el fin de evaluar la eficacia del sistema propuesto se ha desarrollado un simulador capaz de generar diferentes redes y distribuciones de tráfico, ubicando los agentes $D_H$ y $D_A$ en distintas localizaciones. La decisión de implementar esta herramienta es consecuencia de las carencias de los métodos de evaluación convencionales a la hora de mostrar información procedente de redes de distinta topología, así como de medir el impacto de los agentes inmunitarios en función de su configuración. Para la generación de diferentes redes se han tenido en cuenta cuatro parámetros: número de nodos, ancho de banda, densidad de enlaces y componente cíclico. Las dos últimas determinan el número medio de enlaces asociados a cada elemento de red y el número de ciclos que la red puede contener cuando su topología es abstraída como un grafo finito. Nótese que la bibliografía ofrece diferentes soluciones al problema de generar topologías de red aleatorias teniendo en cuenta criterios similares. En la experimentación realizada se han considerado aquellas basadas en el método Erdos-Rènyi [ER60] aplicadas como se describe en [DPR08]; por lo tanto se asume como principal atributo la densidad del grafo (i.e. número de enlaces por nodo). En este modelo la densidad $p$ de los nuevos grafos y la cantidad de nodos son seleccionadas aleatoriamente al inicio del proceso de construcción, de manera que cada par de nodo es conectado con una probabilidad $p$. Tal y como indicaron Erdos y Rènyi [ER60], los grafos construidos por este procedimiento se expresan de la siguiente manera: $G(n, p)$; donde $n$ es el número de nodos conectados según $p$. Cuando al construir una topología el número de caminos cerrados excede el valor del componente cíclico, el grafo es descartado y se procederá a construir uno nuevo. Una vez hallado un grafo que cumpla este requisito, se determina el ancho de banda de cada uno de sus enlaces. Dada una red representada por un uno de los grafos previamente descritos, se emula la inyección de tráfico malicioso generado según las pautas establecidas en [BSMT14].

La Figura 8.6 resume las etapas involucradas en la creación de los grafos generados durante la experimentación. En primer lugar, se define la topología de la red y se distribuyen los agentes inmunitarios. La red es el resultado del proceso previamente descrito, donde los vértices del grafo final actúan como nodos de red y las aristas como conexiones. El siguiente paso es establecer el origen y el destino de los ataques que sufrirá. Los agentes inmunitarios son desplegados en función de un algoritmo de coloración de grafos [GH06], donde los colores más repetidos son los nodos que implementan capacidades inmunitarias. En el segundo nivel del proceso de construcción se establece el tráfico que circulará por la red. Las comunicaciones legítimas se deciden aleatoriamente en función de los atributos tenidos en cuenta. El tráfico es generado por la herramienta *hping3*, constituyendo principalmente comunicaciones legítimas en base a diferentes protocolos (FTP, HTTP, ICMP, etc.) y acciones (transferencia de archivos, peticiones, información de gestión de sesiones, etc.). Con toda esta información, la red se traduce a un script que facilite su implementación en un entorno de virtualización.

Para la experimentación se han construido 220 redes diferentes siguiendo este procedimiento, cuyas principales características topológicas son graficadas en la Figura 8.7. El eje X indica la distribución de enlaces por nodo, y el eje Y la cantidad total de

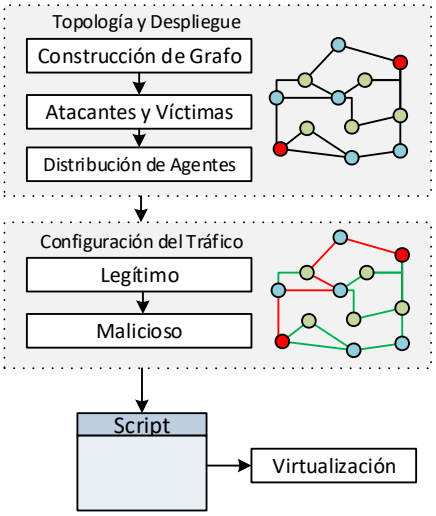



Figura 8.6: Construcción de redes virtuales para la evaluación de la propuesta .

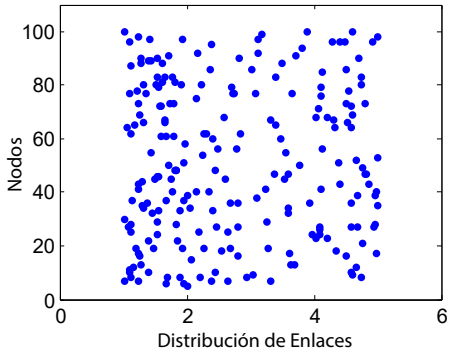



Figura 8.7: Topologías de red en la experimentación.

nodos. La distribución de enlaces por nodo ha variado entre 1 y 5 (media: 2.74, mediana: 2.42, varianza 1.29). En cada red se ha estudiado el comportamiento de ataques DDoS de diferente intensidad, dirigidos entre cada par de posibles nodos origen y destino. A lo largo de la experimentación se ha evaluado la variación de diferentes características: ubicación de los agentes inmunitarios, intensidad del ataque, nivel de congestión de la red o capacidad de mitigación.

## 8.5 Resultados

A continuación se describen los resultados obtenidos al evaluar la capacidad de detección de los agentes inmunitarios por separado, y su eficacia al operar de manera colaborativa.

### 8.5.1 Detección de amenazas

Para evaluar la precisión con la que de los agentes inmunitarios reconocer ataques DDoS se han considerado dos parámetros de ajuste: el orden $\alpha$ de la entropía del volumen de tráfico y el número de paquetes por observación. Las variaciones del primero de ellos

han mostrado un comportamiento similar al analizar las tres colecciones de muestras de referencia: cuanto mayor es $\alpha$, mayor es el nivel de restricción en que opera el sensor, de tal manera que para valores $\alpha > 3$ el despliegue de agentes se vuelve contraproducente; la tasa de falsos positivos es demasiado alta (superior al 20%). Con valores $\alpha$ más bajos, la tasa de falsos positivos es inferior. Dado que la heterogeneidad de las redes actuales propicia la emisión de falsos positivos, se ha considerado $\alpha = 1$, dejando para trabajos futuros el análisis del resto de ajustes del orden de la entropía. En base a esta configuración se han estudiado las colecciones KDD'99, CAIDA'07/08 y el tráfico UCM con la inyección de inundación a través de DDoSIM.

#### 8.5.1.1 KDD'99

La Figura 8.8 muestra los resultados obtenidos al analizar la colección de trazas de tráfico KDD'99. El eje X indica el número de paquetes por observación, y el eje Y las tasas de verdaderos positivos (TPR) y falsos positivos (FPR).

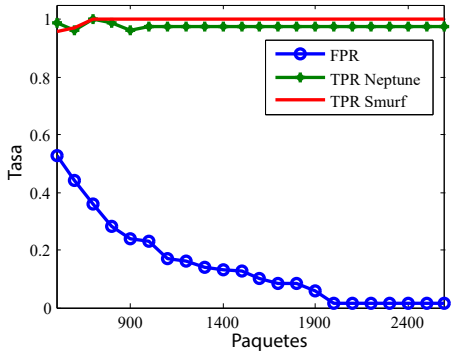



Figura 8.8: Resultados al analizar KDD'99.

La tasa de acierto se ha mantenido constante a lo largo de la experimentación. Sin embargo, los errores al analizar tráfico legítimo han demostrado especial sensibilidad al número de paquetes. Cuando su número es menor, el sistema se comporta de manera más restrictiva. A partir de 2000 observaciones se mantiene constante, operando el sensor en modo de saturación. Se ha estudiado el impacto de este factor en las dos clases de ataques de inundación con más de 100000 paquetes dentro de la colección KDD'99: *smurf* y *neptune*; del resto no se provee la información necesaria para alcanzar dicha saturación. Nótese que los ataques *smurf* tienen por objetivo el denegar el servicio de la víctima a base de peticiones ICMP vía *broadcast*, mientras que los ataques neptune se basan en el envío masivo de peticiones SYN de control de sesión TCP. Las tasas de acierto alcanzadas en saturación son 98.66% (*neptune*) y 100% (*smurf*), en ambos casos con 1.42% de falsos positivos. Teniendo en cuenta que la distribución de muestras es 72.3% (*neptune*) y 27.7% (*smurf*), la tasa media de acierto es 99.03%. Su rendimiento en el espacio ROC se ilustra en la Figura 8.9. En la Tabla 8.1 se comparan los resultados obtenidos con los de propuestas previas que aplican el mismo método de evaluación, demostrándose una precisión similar, y en algunos casos incluso superior. Pero a pesar de la eficacia demostrada, es importante destacar que las muestras analizadas son obsoletas, muy dispares de las actuales.

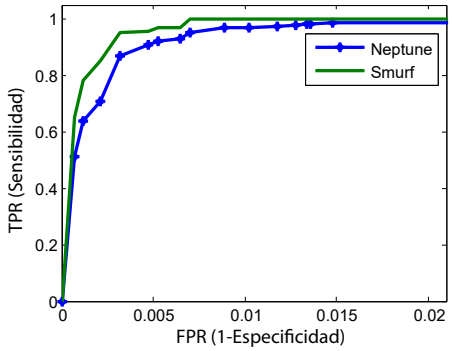



Figura 8.9: Rendimiento en espacio ROC al analizar KDD'99.

Tabla 8.1: Comparación de resultados obtenidos con KDD'99.

| **Propuesta** | **Método** | **TPR (%)** | **FPR (%)** |
|---|---|---|---|
| **Al-Yaseen et al. (2017) [AYON17]** | SVM+ELM | 99.79 | 2.13 |
| **Guo et al. (2016) [GPLL16]** | Híbrida | 97.32 | 0.78 |
| **Liang (2014) [Lia14]** | Neuro-Fuzzy | 99.5 | 1.9 |
| **Kumar y Selvakumar (2013) [KS13]** | Bagging | 91.8 | 6.7 |
| **Kumar y Selvakumar (2013) [KS13]** | NFBoost | 96.1 | 2.8 |
| **Kumar y Selvakumar (2013) [KS13]** | NFBoost+minimización | 98.2 | 1.7 |
| **Ambwani (2003) [Amb03]** | SVM multiclase | 96.8 | 0.43 |
| **Esta propuesta (*Neptune*)** | Anomalía en entropía | 98.66 | 1.42 |
| **Esta propuesta (*Smurf*)** | Anomalía en entropía | 100 | 1.42 |
| **Esta propuesta (*Average*)** | Anomalía en entropía | 99.03 | 1.42 |

#### 8.5.1.2 CAIDA'07/08

La Figura 8.10 muestra los resultados obtenidos al analizar las trazas de CAIDA'07/08. El comportamiento de los agentes inmunitarios ha demostrado ser similar al del experimento anterior, con la excepción de que en esta prueba su estado de saturación ha sido alcanzado con una cantidad menor de paquetes por observación; concretamente 160. En este caso la tasa de acierto ha sido de 98.11% y la de falsos positivos cercana a 1.91%. Las diferencias entre los resultados obtenidos en KDD'99 y CAIDA'07/08 se deben principalmente a la variación de la homogeneidad de las muestras de referencia. En KDD'99 las trazas representan tráfico más antiguo, habiendo sido capturadas en un escenario mucho más sencillo. Por lo tanto, existe una mayor similitud entre ellas, facilitando las tareas de modelado y de esta manera reduciéndose la tendencia a la emisión de falsos positivos. La eficacia en el espacio ROC de los agentes al estudiar CAIDA'07/08 se muestra en la Figura 8.11. En la Tabla 8.2 se comparan los resultados obtenidos con los de propuestas similares, pudiendo observarse cómo en general, son peores que los de las propuestas evaluadas con KDD'99. La tasa de acierto alcanzada ha sido ligeramente superior a la mayoría de las propuestas referenciadas, mientras que la tasa de falsos positivos es parecida. Esta prueba ha facilitado la comparación con estos trabajos, pero como sucedía en el experimento anterior, no ofrece resultados escalables a escenarios de monitorización actuales, llevando a la necesidad de continuar con la experimentación.

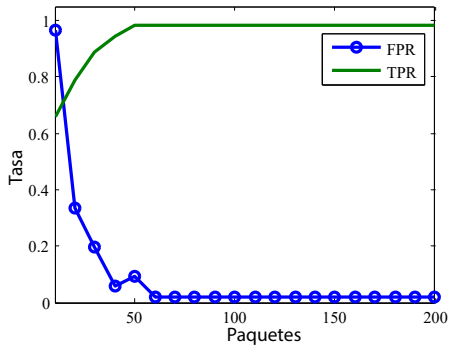



Figura 8.10: Resultados al analizar CAIDA'07/08.

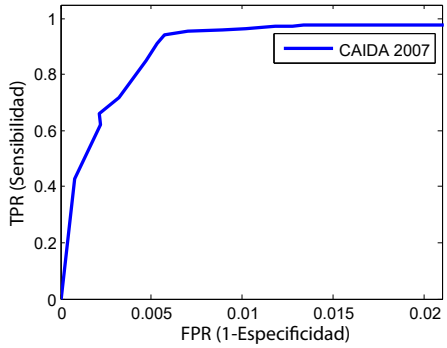



Figura 8.11: Resultados en espacio ROC con CAIDA'07/08.

Tabla 8.2: Comparación de resultados obtenidos con CAIDA'07/08.

| **Propuesta** | **Método** | **TPR (%)** | **FPR (%)** |
|---|---|---|---|
| **Singh et al. (2016) [STD16]** | Red neuronal artificial | 99.62 | 5.61 |
| **Li et al. (2015) [LYW+15]** | SLF | 93.3 | 1.1 |
| **Robinson and Thomas (2015) [RT15]** | Arbol de decisión J48 | 95.5 | 0.1 |
| **Robinson and Thomas (2015) [RT15]** | Arbol de decisión IBK | 97.5 | 0.5 |
| **Luo et al. (2013) [LLZZ13]** | Separaciónn Identifier/Locator | 94.87 | 3.85 |
| **Kumar and Selvakumar (2013)[KS13]** | Bagging | 90.4 | 8.1 |
| **Kumar and Selvakumar (2013)[KS13]** | NFBoost | 97.2 | 4.6 |
| **Kumar and Selvakumar (2013)[KS13]** | NFBoost+minimización | 98.8 | 1.9 |
| **Liu et al. (2011) [LSVK11]** | TrustGuard | 97.68 | 1.0 |
| **Esta propuesta** | Anomalía en entropía | 98.66 | 1.42 |

#### 8.5.1.3 DDoSIM y tráfico UCM

En la Figura 8.12 se muestran los resultados obtenidos al analizar tráfico UCM y ataques DDoS generados por la herramienta DDoSIM. De manera análoga a la experimentación anterior, a partir de un cierto número de paquetes por observación (en ese caso 2000) se satura el impacto de esta métrica en la eficacia del sensor. La precisión alcanzada es peor que en las pruebas anteriores: 92.3% de tasa de acierto y 8.3% de tasa de falsos positivos, lo que demuestra que los buenos resultados observados hasta esta prueba no eran escalables a un escenario real. Esto es debido en su mayor parte al decremento de la homogeneidad en las nuevas muestras, lo que conlleva una mayor tendencia a la emisión

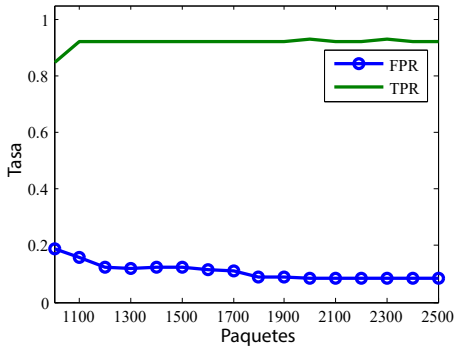



Figura 8.12: Resultados al analizar tráfico UCM con ataques DDoSIM.

de falsos positivos. Como consecuencia empeora la calidad de servicio de la red, ya que las falsas alertas podrían llegar a aplicar acciones de mitigación contra conexiones legítimas. El AIS propuesto reduce este problema aplicando especificidad, de manera que los agentes que se activan sólo operen con mayor restricción sobre flujos de tráfico concretos.
La experimentación realizada también revela otra característica importante del método propuesto: la mayor parte de las amenazas que han sido detectadas han sido reportadas por agentes ubicados en las proximidades del origen y la víctima de la intrusión. Esto es debido a que son las posiciones de la red donde las variaciones en la entropía distinguen con mayor facilidad la inundación del tráfico legítimo. Cuando el ataque presenta intensidad constante (por ejemplo, cuando es de tasa alta), la entropía tiende a estabilizarse pasado un cierto periodo de tiempo, haciéndolo invisible al detector. Este comportamiento se observa con mayor claridad en la Figura 8.13 y en la Figura 8.14, donde se muestra el impacto las variaciones en la entropía del tráfico de uno delos ataques generados por DDoSIM. El eje X indica las observaciones realizadas y el eje Y el valor de la entropía normalizado. El ataque comienza en la observación 60.

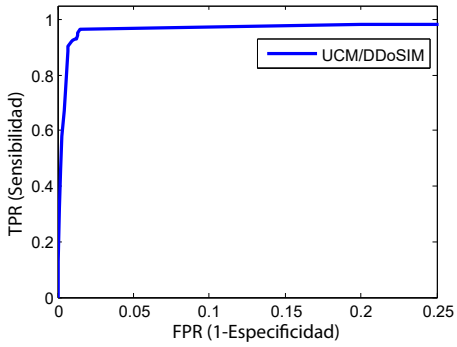



Figura 8.13: Resultados en espacio ROC con UCM y ataques DDoSIM.

Las primeras anomalías en la entropía se observan con mayor claridad en las siguientes 15-20 observaciones, donde la predicción excede los umbrales adaptativos llevando a que el agente reporte una incidencia. Si no se aplican acciones de mitigación (como el descarte de tráfico ilustrado en la Figura 8.2), la entropía se estabilizará poco a poco, pero mostrando valores mucho más altos. A partir de ese momento será muy difícil de detectar. Pasadas varias observaciones se detiene el ataque y comienza a descender hasta su estado inicial. A

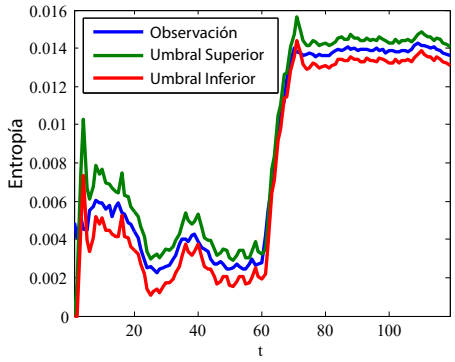



Figura 8.14: Ejemplo de evolución de entropía y umbrales.

la vista de este comportamiento, y con el fin de evitar la evasión del sensor, es recomendable la combinación de la etapa de detección con acciones de mitigación, tal y como describe el sistema propuesto.

#### 8.5.1.4 REACCIONES INMUNITARIAS ARTIFICIALES

La evaluación del comportamiento del sistema propuesto tiene en cuenta la ubicación de los agentes inmunitarios, la intensidad de los ataques de inundación, el nivel de congestión de tráfico legítimo de la red protegida y la capacidad de mitigación del AIS.

#### 8.5.1.5 UBICACIÓN DE LOS AGENTES INMUNITARIOS

En la Figura 8.15 se muestran los resultados obtenidos al desplegar los agentes $D_H$ y $D_A$ en diferentes posiciones de la red. El eje Y indica las tasas TPR/FPR y el eje X la ubicación del agente. Nótese que en esta prueba se considera por ubicación, la posición del sensor en el camino que une el origen del ataque de inundación con la víctima. El eje X la describe de manera normalizada, donde cuando adquiere el valor 0 indica que el sensor está desplegado en la misma fuente del ataque; con valor 1 está desplegado en el nodo víctima. Cuando el ataque es distribuido, cada camino recorrido es estudiado por separado, analizándose trazas que tienen por origen uno de los nodos fuentes y por destino su víctima. También es importante resaltar que, en redes reales la longitud del camino que sigue la inundación puede cambiar. Pero la experimentación ha considerado únicamente caminos estáticos, dejando el estudio del impacto de dichas variaciones para líneas de trabajo futuro.

El TPR promedio alcanzado es 0.85 y el FPR es 0.072. En la figura se muestra una tendencia: cerca de los extremos (origen y destino) la tasa de acierto se aproxima al 100%. En los nodos intermedios la precisión se reduce, llegando a observarse una tasa de acierto de 0.70 en los agentes que equidistan de los extremos. Cuando los agentes $D_A$ se activan como parte de la respuesta inmunitaria adaptativa, se repite este patrón. Sin embargo, su TPR mejora un 7% en los extremos y un 12% en los nodos equidistante, llegando a mostrar un valor de 0.82 en la segunda posición. La respuesta adaptativa prácticamente no afecta a la tasa de falsos positivos, donde se observa un incremento máximo del 0.6%.

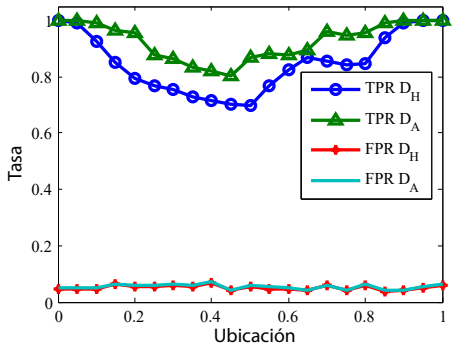



Figura 8.15: Precisión en función de la ubicación de los agentes.

En vista de los resultados observados, y teniendo en cuenta que la mayor parte de los IPS de la bibliografía operan de manera similar a como lo hacen los agentes $D_H$ en la respuesta inmunitaria innata, es posible concluir que el AIS propuesto mejora considerablemente su eficacia en las situaciones en que éstos encuentran mayores dificultades. En concreto, cuando son desplegados en los nodos que equidistan entre el atacante y la víctima, debido principalmente a que el tráfico malicioso puede fluir a través de más nodos intermedios, divergiendo cerca del atacante y convergiendo cerca de la víctima.

#### 8.5.1.6 INTENSIDAD DEL ATAQUE

En la Figura 8.16 se muestra el impacto de la intensidad del ataque en la precisión del AIS propuesto. El eje Y indica las tasas TPR/FPR y el eje X la capacidad de inundación de los ataques DDoS emitidos. En esta prueba la intensidad del ataque es calculada en base al porcentaje del ancho de banda que es capaz de consumir en cada conexión. Se representa con valores que oscilan entre 0 y 1.

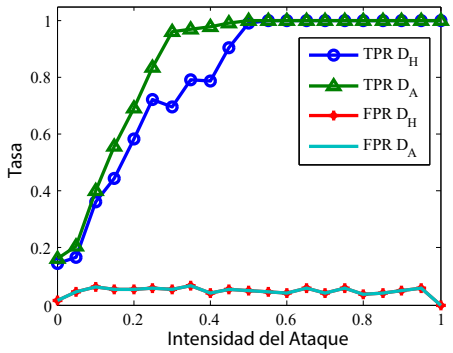



Figura 8.16: Precisión en función de la intensidad del ataque.

Cuando la intensidad tiene valor 0 es descrito el caso en el que no se ha llevado a cabo la inyección de tráfico malicioso; con valor 1 se indica el agotamiento de todo el ancho de banda. Tal y como se observa en la figura, cuando el ataque supera la intensidad de 0.5, ambos tipos de agentes inmunitarios actúan como una precisión similar. Pero cuando los ataques son menos intensos, también son menos ruidosos, situación que dificulta su identificación. Precisamente son en estos casos donde los agentes $D_A$ activados por la

respuesta inmunitaria adaptativa ofrecen una mayor ventaja respecto a los esquemas de detección convencionales representados por los sensores $D_H$. En concreto, la mayor mejora observada es del 26.5% al afrontar ataques de intensidad entre 0.3 y 0.4. Por lo tanto, puede concluirse que, a mayor intensidad del ataque, más eficientemente se comportan los agentes, y que además, cuando la inundación es menos visible gana significancia el papel del esquema inmunitario propuesto frente a IPS convencionales.

#### 8.5.1.7 Congestión en la red

En la Figura 8.17 se muestra la capacidad de detección de los agentes desplegados en función del volumen de tráfico legítimo que circula por la red protegida. El eje X indica las tasas TPR/FPR y el eje Y el volumen de tráfico legítimo. Análogamente a la prueba anterior, en este experimento se ha definido el nivel de congestión en base al porcentaje de ancho de banda ocupado por tráfico legítimo en el instante previo a la ejecución del ataque, y por lo tanto antes de que el tráfico malicioso inyectado agote parte de los recursos de la red. El nivel de congestión es caracterizado por un valor numérico que oscila entre 0 y 1, siendo 0 el caso donde no hay comunicaciones legítimas antes del ataque, y 1 cuando las conexiones estaban completamente saturadas. Los resultados obtenidos muestran un comportamiento parecido al que se ilustró en la Figura 8.16. Sin embargo, cuando en esta prueba la densidad de tráfico es baja, el tráfico inyectado por el atacante se hace mucho más visible, ocupando un porcentaje mayor del ancho de banda. Hasta el nivel de congestión 0.7 no se observan diferencias significativas entre las respuestas inmunitarias implementadas. Pero a partir de este valor el tráfico malicioso pasa desapercibido con mayor facilidad, desencadenando el incremento de la tasa de falsos positivos en los sensores. Cuando actúa la respuesta inmunitaria adaptativa por medio de agentes $D_A$ se observa una mayor precisión; en particular, la tasa de acierto obtenida llega a mejorarse hasta un 13.7% en congestión entre 0.6 y 0.7, lo que supone una importante mejora frente a esquemas de detección convencionales.

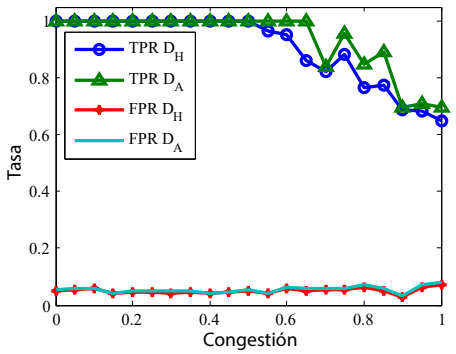



Figura 8.17: Precisión en función de la congestión de la red.

#### 8.5.1.8 MITIGACIÓN

En la Figura 8.18 se muestra la capacidad de mitigación de sistema propuesto en función del número de nodos involucrados en el proceso de intrusión. Por lo tanto, el eje Y indica las tasas TPR/FPR y el eje X el número de nodos en el camino seguido por el tráfico malicioso. En esta prueba se considera que una amenaza ha sido mitigada si no ha alcanzado su destino, y por lo tanto se asume que ha sido neutralizada por alguno de los agentes que actúan en alguna de las respuestas inmunitarias implementadas. Al desencadenarse únicamente la respuesta innata, en el peor de los casos se ha llegado a mitigar el 81.4% de las amenazas emitidas. Sin embargo, la respuesta adaptiva ha llegado a bloquear el 95.5% de los ataques bajo circunstancias similares, lo que ha supuesto una mejora del 14.1%. Por lo general, el AIS tiene una mayor capacidad de mitigación en redes pequeñas, donde el ataque tiene la posibilidad de llegar a la víctima por menos rutas alternativas. En redes grandes los agentes inmunitarios pueden ser evadidos, llegando el ataque por caminos que podrían no estar vigilados. A partir de la figura es posible deducir que el AIS propuesto ofrecerá una mejor solución a los IPS convencionales en los casos en que las redes a proteger tengan mayores dimensiones, donde operan con más dificultad y su capacidad de mitigación es menor.

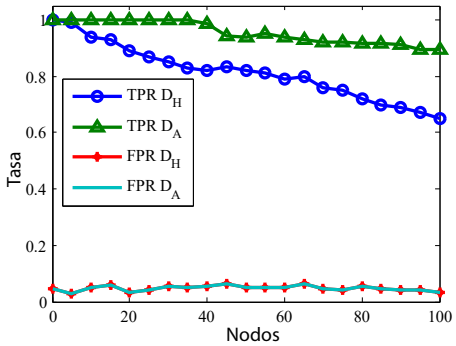



Figura 8.18: Mitigación en función de la cantidad de nodos afectados.

# Capítulo 9

# Conclusiones y trabajo futuro

## 9.1 Conclusiones

La detección de intrusiones basada en el reconocimiento de anomalías se ha convertido en un elemento fundamental de las estrategias para la seguridad de la información actuales. Sin embargo, los avances en la tecnología de la información y la aparición de nuevos modelos de mercado centrados en el delito informático han dado lugar a la evolución de los escenarios de monitorización convencionales. En consecuencia, muchos de los esquemas de detección clásicos han quedado obsoletos, siendo necesaria su actualización, y su evaluación por medio de metodologías acordes a las exigencias de los nuevos despliegues de seguridad. A lo largo de la investigación realizada se ha profundizado en las causas que han llevado a la pérdida de eficacia de estas tecnologías. Con este fin se ha llevado a cabo un estudio exhaustivo de las bases que asientan su diseño e implementación, revisándose normas, marcos, modelos, estrategias, paradigmas y su ámbito desde el punto de vista de la gestión de la seguridad. También se ha revisado el concepto de anomalía, sus interpretaciones más comunes, métodos de detección y la manera en que son considerados en diferentes contextos. Esto ha permitido identificar una serie de dificultades comunes al operar en escenarios actuales, que sin duda marcarán los principales desafíos a tener en cuenta en los años venideros: altas tasas de falsos positivos en entornos de monitorización muy heterogéneos, ausencia de una aproximación universal al problema del reconocimiento de anomalías, fortalecimiento contra métodos de evasión, adaptabilidad a escenarios de características variables, dificultad en hallar metodologías de evaluación y conjuntos de muestras de referencia adecuados a los nuevos casos de uso, discrepancia en la elección de distancias y medidas de similitud, y el consumo de recursos de cómputo, siendo este último un tema especialmente candente en la computación ubicua. Con la motivación de ilustrar con mayor claridad esta problemática, se ha hecho hincapié en cinco casos de uso especialmente sujetos a padecerla: la detección de atacantes enmascarados ocultos bajo métodos de evasión basados en imitación; la adaptación de los sistemas de detección de intrusiones que operan sobre redes de comunicaciones al análisis de tráfico actual; la gestión y correlación de las incidencias reportadas por estos sensores considerando información intrínseca a la naturaleza de las anomalías identificadas; la mitigación de

ataques de denegación de servicio a lo largo del sistema protegido; y la identificación de código malicioso en aplicaciones para dispositivos móviles previa a su instalación. Todos ellos han sido estudiados en detalle, y en el caso de los cuatro primeros se han introducido estrategias para su adaptación.

La primera de estas nuevas aproximaciones tiene como principal objetivo mejorar los esquemas de detección de atacantes enmascarados convencionales. Con este fin se aplican diferentes técnicas orientadas a optimizar su precisión, reducir la tasa de falsos positivos, permitir el análisis de eventos en tiempo real, y fortalecerlo frente a ataques basados en imitación. Con el fin de mejorar su precisión y reducir su tasa de falsos positivos se ha propuesto el análisis de las secuencias de acciones realizadas por el usuario del sistema, mediante técnicas de alineamiento de secuencias local. También se introduce el uso del U-test de Mann-Whitney. Gracias a esta técnica, es posible determinar en tiempo real la naturaleza de la actividad del usuario del sistema, y validarla con el fin de reducir la probabilidad de emitir etiquetados incorrectos. Para la detección de ataques de mimetismo se aplican técnicas de sub-secuenciación y paralelismo. Esto conlleva un análisis no determinista de la actividad del sistema, y facilita su identificación. La experimentación realizada ha mostrado una precisión similar a la de las mejores propuestas de la bibliografía en su evaluación por estándares funcionales. Pero a diferencia de sus predecesoras, también ha demostrado ser capaz de identificar gran parte de los intentos de evasión a los que ha sido sometida, demostrando un comportamiento mucho más eficaz al operar sobre este nuevo escenario.

Por otro lado, se ha propuesto una estrategia de reconocimiento de amenazas desconocidas sobre redes de comunicaciones, basada en la identificación de anomalías en la carga útil del tráfico analizado, y orientada a reducir el problema de las altas tasas de falsos positivos de sus predecesoras al operar sobre redes actuales. Sus diferentes etapas de procesamiento de información involucran el uso de la metodología N-gram y la organización de los datos en filtros Bloom. A partir de ellos es posible la construcción de reglas de detección capaces de decidir en tiempo real la naturaleza de la carga útil analizada. En la experimentación realizada sobre estándares de evaluación funcionales se ha observado una precisión similar, aunque ligeramente mejor en términos de tasa de falsos positivos, que en trabajos previos. Pero su principal contribución fue demostrada en su despliegue sobre escenarios de monitorización reales, probándose su gran coherencia entre los resultados obtenidos en este proceso, respecto a los observados en los estándares, y sobre todo que a diferencia que en aproximaciones anteriores su tasa de falsos positivos no empeora drásticamente.

También se ha introducido un marco para la correlación de alertas emitidas por sistemas de detección de malware en redes basados en la identificación de anomalías en su carga útil. Éste abarca las estrategias de análisis y modelado del sensor a complementar, lo que facilita un mayor nivel de integración. Presenta una arquitectura multinivel capaz de correlacionar tanto alertas como secuencias de ellas, y cuya estrategia de análisis se fundamenta principalmente en dos grandes criterios de similitud: la asociación de incidencias por medio del estudio de la naturaleza de las anomalías que las desencadenan, y su nivel de discordancia. Por lo tanto, el marco propuesto únicamente considera rasgos

inherentes a las anomalías, permitiendo su integración en esquemas de correlación que además tienen en cuenta información extraída directamente del encabezado y la carga útil de los paquetes a analizar. Para su evaluación se ha implementado una instancia que complementa el sistema de detección mencionado en el párrafo anterior. Su eficacia ha sido demostrada al correlacionar alertas emitidas por el sensor al analizar tráfico real, demostrando ser capaz de priorizar el tratamiento de las incidencias, descartar falsos positivos, y agrupar la información recibida teniendo en cuenta las características de los riesgos que preceden.

Con la motivación de contribuir a los esfuerzos defensivos frente a los ataques DDoS basados en inundación se ha descrito una propuesta para su mitigación. Ésta conlleva el despliegue de una red de sensores distribuidos que operan a modo de sistema inmunitario artificial inspirado en los mecanismos defensivos biológicos de los seres humanos. Se ha introducido una combinación de los métodos de detección clásicos basados en el estudio de las variaciones en la entropía del volumen de tráfico analizado con la emulación del comportamiento de los agentes inmunitarios, lo que permite la aplicación de contramedidas en tiempo real, la construcción de una memoria inmunitaria y la definición de regiones de cuarentena, todo ello teniendo en cuenta el estado de la red protegida. Otra contribución importante la constituye la estrategia de reconocimiento de amenazas que implementan los agentes inmunitarios, la cual se basa en la identificación de anomalías a partir de las observaciones realizadas y la predicción del comportamiento de la red. La experimentación realizada a mostrado resultados prometedores, habiéndose evaluado la propuesta tanto con estándares de evaluación funcionales (KDD'99, CAIDA'07/08) como con trazas de tráfico real (DDoSIM y tráfico UCM).

Con el fin de facilitar la comprensión del trabajo de investigación desempeñado, los resultados de la extensa experimentación realizada son acompañados por gran cantidad de tablas, figuras y elementos aclaratorios, destacando la honesta discusión de sus beneficios frente a propuestas similares; nótese que si bien las pruebas realizadas arrojaron resultados muy satisfactorios, también dejaron entrever ciertos aspectos mejorables, sobre los que se ha indagado a lo largo de la documentación con el fin motivar el emprendimiento de líneas de investigación relacionadas.

## 9.2 Trabajo futuro

A continuación se describen las principales líneas de trabajo futuro derivadas de la investigación realizada:

### 9.2.1 Reconocimiento de anomalías en dispositivos móviles

El problema del reconocimiento de anomalías en dispositivos móviles orientado a la identificación del malware descargado de los mercados de distribución de aplicaciones es uno de los casos de uso que ilustran la problemática de adaptar los métodos de detección convencionales a los nuevos escenarios de monitorización. Si bien este objeto de estudio es revisado en profundidad en los primeros capítulos de este documento, el desarrollo y

publicación de una propuesta sólida para su mitigación es una tarea actualmente en curso; la aproximación en la que se está trabajando se basa en los métodos de alineamiento de secuencias y en el estudio de las llamadas al sistema ejecutadas por las aplicaciones monitorizadas, importando de esta manera algunas de las ideas de nuestra propuesta para la detección de enmascarados.

### 9.2.2 RECONOCIMIENTO DE ANOMALÍAS EN LA DETECCIÓN DE RANSOMWARE

El término ransomware es aplicado a cualquier tipo de software malicioso que al ejecutarse bloquee total o parcialmente el sistema de la víctima, y que demande un pago a modo de rescate, tras el que promete restablecer su funcionalidad original. A pesar de su aparente sencillez, este tipo de ataque ha crecido de manera alarmante en los últimos años, estando en el punto de mira de la mayor parte de las organizaciones para la seguridad de la información. El reconocimiento de anomalías juega un papel esencial en la lucha contra esta amenaza, ya que permite descubrir sus procesos de enumeración de la víctima, cifrado de activos, y eliminación. Por lo tanto, se trata de un problema emergente que plantea otro interesante escenario emergente de monitorización.

### 9.2.3 RECONOCIMIENTO DE ANOMALÍAS EN SISTEMAS DE CONTROL DE ACCESO BASADOS EN BIOMETRÍA

Otro posible de interés en el reconocimiento de anomalías es su adaptación a los sistemas de control de acceso basados en biometría. Dada la resistencia de estas tecnologías a intentos de falsificación, el uso de rasgos biométricos es cada vez más frecuente en ciertos escenarios: reconocimiento de firmas manuscritas y sus dinámicas, voz, patrones de uso de teclado y movimientos de razón, etc. Sin embargo, presentan una gran dependencia respecto a las muestras de referencia con los que son entrenados, las cuales a menudo requieren de su constante actualización. Esta característica puede llegar a penalizar representativamente la calidad de experiencia del usuario, a quien continuamente se somete a nuevas pruebas para detectar cambios en sus rasgos biométricos. El estudio en profundidad de este problema, y la elaboración de métodos concretos para su paliación basados en la identificación de características discordantes es un tema de interés que será tratado en trabajos de investigación futuros.

# Capítulo 10

# Lista de publicaciones

1. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2018. "Adaptive artificial immune networks for mitigating DoS flooding attacks". *Swarm and Evolutionary Computation*, Volumen 38, pp. 94-108. (Índice de Impacto 3,893 (Q1 19/133)).

2. Jorge Maestre Vidal, Marco Antonio Sotelo Monge, Luis Javier García Villalba, 2018. "A Novel Pattern Recognition System for Detecting Android Malware by Analyzing Suspicious Boot Sequences". *Knowledge-Based Systems*, DOI: 10.1016/j.knosys.2018.03.018. (Índice de Impacto 4,529 (Q1 13/133)).

3. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2017."Alert Correlation Framework for Malware Detection by Anomaly-based Packet Payload Analysis". *Journal of Network and Computer Applications*, Volumen 29, pp. 11-22. (Índice de Impacto 3,5 (Q1 6/106)).

4. Luis Javier García Villalba, Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, 2017. "Advanced Payload Analyzer Preprocessor". *Future Generation Computer Systems*, Volumen 76, pp. 474-485. (Índice de Impacto 3,997 (Q1 10/104)).

5. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2016. "Online masquerade detection resistant to mimicry". *Expert Systems with Applications*, Volumen 61, No. 1, pp. 162-180. (Índice de Impacto 3,928 (Q1 18/133)).

6. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2015. "Quantitative Criteria for Alert Correlation of Anomalies-based NIDS". *IEEE Latin America Transactions*, Volumen 13, No. 10, pp. 3461-3466. (Índice de Impacto 0,631 (Q3 135/146)).

7. Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Jorge Maestre Vidal, 2015. "Malware Detection System by Payload Analysis of Network Traffic". *IEEE Latin America Transactions*, Volumen 13, No. 3, pp. 850-855. (Índice de Impacto 0,631 (Q3 135/146)).

8. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2015. "Network Intrusion Detection Systems in Data Centers". Handbook on Data Centers, Springer Science+Business Media, New York, US, pp. 1185-1207.

9. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2017. "Malware Detection in Mobile Devices by Analyzing Sequences of System Calls". En Proc. 19th International Conference on Information Technology (ICIT 2017), Paris, France.

10. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2017. "Poster: Mitigation of DDoS Attacks in 5G Networks: a Bio-inspired Approach". En Proc. 2nd IEEE European Symposium on Security and Privacy (EuroS&P), Paris, France.

11. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2015. "Detección de Malware en Dispositivos Móviles mediante el Análisis de Secuencia de Acciones". En Proc. VIII Congreso Iberoamericano de Seguridad Informática (CIBSI), Quito, Ecuador.

12. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2015. "Correlación de alertas en la detección de malware en redes basada en anomalías". En Proc. I Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), León, Spain.

13. Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba, 2015. "Sistema Inmunitario Adaptativo para la mitigación de ataques de Denegación de Servicio". En Proc. I Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), León, Spain.

# Bibliografía


[AAB12] K. Alsubhi, I. Aib, and R. Boutaba, 2012. "FuzMet: a fuzzy logic based alert prioritization engine for intrusion detection systems". *International Journal of Network Managment*, Volumen 22, No. 4, pp. 263-284.

[AAL+03] N. Athanasiades, R. Abler, J. Levine, H. Owen, and G. Riley, 2003. "Intrusion detection testing and benchmarking methodologies". En Proc. 1st IEEE International Workshop on Information Assurance, Darmstadt, Germany.

[AAS15] C. Annachhatre, T.H. Austin, and M. Stamp, 2015. "Hidden Markov models for malware classification". *Journal of Computer Virology and Hacking Techniques*, Volumen 11, Issue 2, pp. 59-73.

[ABC+03] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod, 2003. "Danger theory: the link between AIS and IDS". En Proc. 2nd International Conference on Artificial Immune Systems (ICARIS), Edinburgh, UK, pp. 147-155.

[ABK99] M. Ankerst, M.M. Breuning, and H.P. Kriegel, 1999. "OPTICS: ordering points to identify the clustering structure". En Proc. 1999 ACM SIGMOD International Conference on Management of Data, Philadelphia, PA, USA, Volumen 28, Issue 2, pp. 49-60.

[ACM15] M. Andreolini, M. Colajanni, and M. Marzoetti, 2015. "A collaborative framework for intrusion detection in mobile networks". *Information Sciences*, Volumen 321, pp. 179-192.

[ADAH14] B. Al-Duwairi and A. Al-Hammouri, 2014. "Fast flux watch: a mechanism for online detection of fast-flux networks". *Journal of Advanced Research*, Volumen 5 Issue 4, pp. 473-479.

[ADT95] R. Andrews, J. Diederich, and A.B. Tickle, 1995. "Survey and critique of techniques for extracting rules from trained artificial neural networks". *Knowledge-Based Systems*, Volumen 8, Issue 6, pp. 373-389.

[Agg13] C.C. Aggarwal, 2013. "Outlier analysis". Springer-Verlag New York.

[AJA11] S.H. Ahmadinejad, S. Jalili, and M. Abadi, 2011. "A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs". *Computer Networks*, Volumen 55, Issue 9, pp. 2221-2240.

[AJPS11] M. Albanese, S. Jajodia, A. Pugliese, and V.S. Subrahmanian, 2011. "Scalable analysis of attack scenarios". En Proc. 16th European Symposium on Research in Computer Security (ESORICS), Leuven, Belgium, Volumen 6879, pp. 416-433.

[AKK+13] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, 2013. "DNS amplification attack revisited". *Computers & Security*, Volumen 39, part B, pp. 475-485.

[Ali14] C. Alippi, 2014. "Intelligence for embedded systems". Springer-Verlag, Berlin, Germany.

[ALP14] A. Abbas, A.H. Lichtman, and S. Pillai, 2014. "Cellular and molecular immunology". 8th edition, Saunders Elsevier Philadelphia, PA, US.

[AM16] M. Ahmed and J. Mahmood, A.N.and Hu, 2016. "A survey of network anomaly detection techniques". *Journal of Network and Computer Applications*, Volumen 60, pp. 19-31.

[Amb03] T. Ambwani, 2003. "Multi class support vector machine implementation to intrusion detection". En Proc. International Joint Conference on Neural Networks 2003. Portland, OR, US; pp. 1-6.

[AMZ08] S.O. Al-Mamory and H. Zhang, 2008. "IDS alerts correlation using grammar-based approach". *Journal in Computer Virology*, Volumen 5, Issue 4, pp. 271-282.

[And72] J.P. Anderson, 1972. "Information security in a multi-User computer environment". *Advances in Computers*, Volumen 12, pp. 1-36.

[AP13] R. Azmi and B. Pishgoo, 2013. "SHADuDT:secure hypervisor-based anomaly detection based on danger theory". *Computers & Security*, Volumen 39, part B, pp. 268-288.

[Apv14] A. Apvrille, 2014. "The evolution of mobile malware". *Computer Fraud & Security*, Volumen 8, pp. 18-20.

[AR08] C. Alippi and M. Roveri, 2008. "Just-in-time adaptive classifiers-part II: designing the classifier". *IEEE Transactions on Neural Networks*, Volumen 19, No. 12, pp. 2053-2064.

[AR14] N.M. Alenezi and M.J. Reed, 2014. "Uniform DoS traceback". *Computers & Security*, Volumen 45, No. 1, pp. 17-26.

[ARP14] M.V.O Assis, J.J.P.C Rodrigues, and M.L. Proenca, 2014. "A seven-dimensional flow analysis to help autonomous network management". *Information Sciences*, Volumen 278, pp. 900-913.

[ASH+14] D. Arp, M. Spreitzenbarth, M.H. Hubner, H. Gascon, and K. Rieck, 2014. "Drebin: effective and explainable detection of Android malware in your pocket". En Proc. 21th Annual Symposium on Network and Distributed System Security (NDSS), San Diego, CA, US, pp. 1-12.

[ASW15] S. Aghabozorgi, A.S. Shirkhorshidi, and T.Y Wah, 2015. "Time-series clustering - A decade review". *Information Systems*, Volumen 53, pp. 16-38.

[AT07] A.A.E. Ahmed and I. Traore, 2007. "A new biometric technology based on mouse dynamics". *IEEE Transactions on Dependable and Secure Computing*, Volumen 4, Issue 3, pp. 165-179.

[AT14] A.A. Ahmed and I. Traore, 2014. "Biometric recognition based on free-text keystroke dynamics". *IEEE Transactions on Cybernetics*, Volumen 44, Issue 4, pp. 458-472.

[ATG11] D. Ariu, R. Tronci, and G. Giacinto, 2011. "HMMPayl: An intrusion detection system based on Hidden Markov Models". *Computers & Security*, Volumen 30, Issue 4, pp. 221-241.

[AUT06] S. Akira, S. Uematsu, and O. Takeuchi, 2006. "Pathogen recognition and innate immunity". Cell, Volumen 124, Issue 4, pp. 783-801.

[AW10] H. Abdi and L.J. Williams, 2010. "Principal component analysis". *Wiley Interdisciplinary Reviews: Computational Statistics*, Volumen 2, Issue 4, pp. 433-459.

[AY01] C.C. Aggarwal and P.S. Yu, 2001. "Outlier detection for high dimensional data". En Proc. 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD'01), Santa Barbara, CA, USA, pp. 37-46.

[AYON17] W. Al-Yaseen, Z. Othman, and M. Nazri, 2017. "Real-time multi-agent system for an adaptive intrusion detection system". *Pattern Recognition Letters*, Volumen 85, pp. 56-64.

[AYTF14] A. Almalawi, X. Yu, Z. Tari, and A. Fahad, 2014. "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems". *Computers & Security*, Volumen 46, pp. 94-110.

[AZM+12] A.A. Amaral, B.B. Zarpelao, L.S. Mendez, J.J.P.C. Rodrigues, and M.L.P. Junior, 2012. "Inference of network anomaly propagation using spatio-temporal correlation". *Journal of Network and Computer Applications*, Volumen 35, Issue 6, pp. 1781-1792.

[BAG15] N. Ben-Asher and C. Gonzalez, 2015. "Effects of cyber security knowledge on attack detection". *Computers in Human Behavior*, Volumen 48, pp. 51-61.

[BBK14] M.H. Bhuyan, D.K. Bhattacharyya, and J.K. Kalita, 2014. 'Network anomaly detection: methods, systems and tools". *IEEE Communications Surveys & Tutorials*, Volumen 16, Issue 1, pp. 303-336.

[BBK15] M.H. Bhuyan, D. Bhattacharyya, and J. Kalita, 2015. "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection". *Pattern Recognition Letters*, Volumen 51, No. 1, pp. 1-7.

[BCK08] S. Boriah, V. Chandola, and V. Kumar, 2008. "Similarity measures for categorical data: a comparative evaluation". En Proc. International SIAM Conference of Data Mining (SIAM 08), Atlanta, GA, USA, pp. 243-254.

[BDR06] D. Barbará, C. Domeniconi, and J.P. Rogers, 2006. "Detecting outliers using transduction and statistical testing". En Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), Philadelphia, PA, USA, pp. 55-64.

[BEHZ06] D. Bolzoni, S. Etalle, P. Hartel, and E. Zambon, 2006. "POSEIDON: a 2-tier anomaly-based network intrusion detection system". En Proc. Fourth IEEE International Workshop on Information Assurance (IWIA), London, United Kingdom, pp. 144-156.

[BG15] A.L. Buczak and E. Guven, 2015. "A survey of data mining and machine learning methods for cyber security intrusion detection". *IEEE Communications Surveys & Tutorials*, Volumen 18, Issue 2, pp. 1153-1176.

[BGC15] K. Bhattacharya, A. Ghosh, and S. Chowdhury, 2015. "An affinity-based new local distance function and similarity measure for kNN algorithm". *Pattern Recognition Letters*, Volumen 60-61, pp. 24-31.

[BH10] P. Brooks and B. Hestnes, 2010. "User measures of quality of experience: why being objective and quantitative is important". *IEEE Network*, Volumen 24, Issue 2, pp. 8-13.

[BJ76] G.E.P. Box and G.M. Jenkins, 1976. "Time series analysis: forecasting and control". Holden Dayr, San Francisco, California.

[BkNS00] M.M. Breuning, H.P. kriegel, R.T. Ng, and J. Sander, 2000. "LOF: identifying density-based local outliers". En Proc. 2000 ACM SIGMOD International Conference on Management of Data, Dalles, TX,USA, pp. 93-104, Mayo 2000.

[BL94] V. Barnett and T. Lewis, 1994. "Outliers in statistical data". John Wiley & Sons, 3rd edition.

[BLMVGV17] L. Barona López, J. Maestre Vidal, and L.J. García Villalba, 2017. "An approach to data analysis in 5G networks". *Entropy*, Volumen 19, Issue 2, No. 24.

[BLVCMV+17] L. Barona López, L. Valdivieso Caraguay, J. Maestre Vidal, M.A. Sotelo Monge, and L.J. García Villalba, 2017. "Towards incidence management in 5G based on situational awareness". *Future Internet*, Volumen 9, Issue 1, No. 3.

[BMR15] R. Baldoni, L. Montanari, and M. Rizzuto, 2015. "On-line failure prediction in safety-critical systems". *Future Generation Computer Systems*, Volumen 45, pp. 123-132.

[BOS84] L.F. Breiman, R. Olshen, and C. Stone, 1984. "Classification and regression trees". Wadsworth International, California.

[BP66] L.E. Baum and T. Petrie, 1966. "Statistical inference for probabilistic functions of finite state Markov chains". *The Annals of Mathematical Statistics*, Volumen 37, Issue 6, pp. 1554-1563.

[BS14] D. Brzezinski and J. Stephanowski, 2014. "Reacting to different types of concept drift: The accuracy updated ensemble algorithm". *IEEE Transactions on Neural Networks and Learning Systems*, Volumen 25, No. 1, pp. 81-94.

[BSMT14] S. Bhatia, D. Schmidt, G. Mohay, and A. Tickle, 2014. "A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events". *Computers & Security*, Volumen 40, No. 1, pp. 95-107.

[BZNT11] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, 2011. "Crowdroid: behavior-based malware detection system for Android". En Proc. 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, Chicago, IL, US , pp. 15-26.

[CADZ15] J. Chen, M.H. Alalfi, T.R. Dean, and Y. Zou, 2015. "Detecting Android malware using clone detection". *Journal of Computer Science and Technology*, Volumen 30, Issue 5, pp. 942-956.

[CAI07] CAIDA, 2007. CAIDA UCSD, "DDoS attack 2007 dataset". Available at http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[CAI08] CAIDA, 2008. CAIDA UCSD, "Anonymized Internet traces 2008 (CAIDA 08)". Available at http://www.caida.org/data/passive/passive_2008_dataset.xml.

[CAI18] CAIDA, 2018. The CAIDA Dataset. Available at: https://www.caida.org/data/overview/.

[CALK08] L. Cohen, G. Avrahami, M. Last, and A. Kandel, 2008. "Info-fuzzy algorithms for mining dynamic data streams". *Applied Soft Computing*, Volumen 8, No. 4, pp. 1283-1294.

[Can98] J. Cannady, 1998. "Artificial neural networks for misuse detection". En Proc. 21st National Information Systems Security Conference, Arlington, VA, USA, pp. 443-456.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar, 2009. "Anomaly detection : a Survey". *ACM Computing Surveys*, Volumen 41, Issue 3, No. 15.

[CBK12] V. Chandola, A. Banerjee, and V. Kumar, 2012. "Anomaly detection for discrete sequences: a survey". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 24, Issue 5, pp. 823-839.

[CBMP16] L.F. Carvalho, S. Barbon, L.S. Mendes, and M.L. Proenca, 2016. "Unsupervised learning clustering and self-organized agents applied to help network management". *Expert Systems with Applications*, Volumen 54, pp. 29-47.

[CBSB03] S. Coull, J. Branch, B. Szymanski, and E. Breimer, 2003. "Intrusion detection: a bioinformatics approach". En Proc. 19th IEEE Annual Conference on Computer Security Applications (CSAC), Las Vegas, NV, USA, pp. 24-33.

[CC12] CCN-CERT, 2012. MAGERIT: Risk Analysis and Management Methodology for Information Systems.

[CC16] CCN-CERT, 2016. "Guía de seguridad de las TIC CCN-STIC-817", Series 800 National Security Framework, Cyberincident management. Available: https://www.ccn.cni.es.

[CCT10] S.S. Choi, S.H. Cha, and C.C. Tappert, 2010. "A survey of binary similarity and distance measures". *Journal of Systemics, Cybernetics and Informatics*, Volumen 8, No. 1, pp. 43-48, 2010.

[CE18] CERT-EU, 2018. Available at: https://cert.europa.eu/cert/alertedition/.

[Cen15] CERT Insider Threat Center, 2015. "U.S. state of cybercrime survey 2015". Available at https://www.cert.org/insider-threat.

[CFGH10] Y. Cai, R.M. Franco, and M. García-Herranz, 2010. "Visual latency-based interactive visualization for digital forensics". *Journal of Computational Science*, Volumen 1, No. 2, pp. 115-120.

[CGL+16] L. Caviglione, M. Gaggero, J.F. Lalande, W. Mazurcyk, and M. Urbánski, 2016. "Seeing the unseen: revealing mobile malware hidden communications via energy consumption and artificial intelligence". *IEEE Transactions on Information Forensics and Security*, Volumen 11, Issue 4, pp. 799-810.

[CGPP12] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, 2012. "Wave-cusum: improving cusum performance in network anomaly detection by means of wavelet analysis". *Computers & Security*, Volumen 31, No. 5, pp. 727-735.

[CGR13] I. Coronoa, G. Giacinto, and F. Roli, 2013. "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues". *Information Sciences*, Volumen 239, pp. 201-225.

[CH13a] Y. Cherdantseva and J. Hilton, 2013. "A reference model of information assurance & security". En Proc. 8th IEEE International Conference on Availability, Reliability and Security (ARES), Regensburg, Germany, pp. 546-555.

[CH13b] G. Creech and J. Hu, 2013. "A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns". *IEEE Transactions on Computers* , Volumen 64, Issue 3, pp. 807-819.

[Cha07] S.H. Cha, 2007. "Comprehensive survey on distance/similarity measures between probability density functions". *International Journal of Mathematical Models and Methods in Applied Sciences*, Volumen 1, Issue 4, pp. 300-307.

[Cis17] Cisco, 2017. "Annual cybersecurity report 2017". Available at https://www.cisco.com.

[CK13] E. Cohen and H. Kaplan, 2013. "What you can do with coordinated samples". En Proc. 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization and 17th International Workshop, Randomization and Computation. Berkeley, CA, USA, pp. 452-467.

[CKCY14] C.H. Chen, L.P. Khoo, Y.T. Chong, and X.F. Yin, 2014. "Knowledge discovery using genetic algorithm for maritime situational awareness". *Expert Systems with Applications*, Volumen 41, Issue 6, pp. 2742-2753.

[Cla04] B. Claise, 2004. "Cisco systems NetFlow services export Version 9". IETF RFC 3954.

[Cla08] B. Claise, 2008. "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information". IETF RFC 5101.

[CLL+15] H. Chen, T. Li, C. Luo, S.J. Horng, and G. Wang, 2015. "A decision-theoretic rough set approach for dynamic data mining". *IEEE Transactions on Fuzzy Systems*, Volumen 23, Issue 6, pp. 1958-1970.

[CLLL11] T.H. Cheng, Y.D. Lin, Y.C. Lai, and P.C. Li, 2011. "Evasion techniques: sneaking through your intrusion detection/prevention systems". *IEEE Communications Surveys & Tutorials*, Volumen 14, Issue 4, pp. 1011-1020.

[CMP13] H. Cam, P.A. Mouallem, and R.E. Pino, 2013. "Alert data aggregation and transmission prioritization over mobile networks, network science and cybersecurity". *Advances in Information Security*, Volumen 55, pp. 205-220.

[CMW13] Y. Chen, X. Ma, and X. Wu, 2013. "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory". *IEEE Communications Letters*, Volumen 17, No. 5, pp. 1052-1054.

[CMZS13] R.J.G.B. Campello, D. Moulavi, A. Zimek, and J. Sander, 2013. "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies". *Data Mining and Knowledge Discovery*, Volumen 27, Issue 3, pp. 344-371.

[CnHGMT14] J.B. Carmiña, C. Hernández-Gracidas, R. Monroy, and L. Trejo, 2014. "The windows-users and -intruder simulations logs dataset (WUIL): an experimental framework for masquerade detection mechanisms". *Expert Systems with Applications*, Volumen 41, Issue 3, pp. 919-930.

[CS08] S.E. Coull and B. Szymanski, 2008. "Sequence alignment for masquerade detection". *Journal Computational Statistics & Data Analysis*, Volumen 52, Issue 8, pp. 4116-4131.

[CZ02] L.N. Castro and F.J.V. Zuben, 2002. "Learning and optimization using the clonal selection principle". *IEEE Transactions on Evolutionary Computation*, Volumen 6, No. 3, pp. 239-251.

[CZJK14] T. Chen, X. Zhang, S. Jin, and O. Kim, 2014. "Efficient classification using parallel and scalable compressed model and its application on intrusion detection". *Expert Systems with Applications*, Volumen 41, Issue 13, pp. 5972-5983.

[CZS+16] G.O. Campos, A. Zimek, J. Sander, R.J.G.B. Campello, B. Micenková, E. Schubert, I. Assent, and M.E. Houle, 2016. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". *Data Mining and Knowledge Discovery*, Volumen 30, Issue 4, pp. 891-927.

[DBK+97] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, V. Vapnik, M. Mozer, J. Jordan, and T. Petscbe, 1997. "Support vector regression machines". *Advances in Neural Information Processing Systems*, MIT Press Volumen 9, pp. 155-161.

[DCF07] H. Debar, D. Curry, and B. Feinstein, 2007. "The Intrusion Detection Message Exchange Format (IDMEF)". IETF RFC 4765.

[DDo13] DDoSIM, 2013. DDoSIM Layer 7 DDoS Simulator. Available at http://http://sourceforge.net/projects/ddosim/.

[DDW99] H. Debar, M. Darcier, and A. Wespi, 1999. 'Towards a taxonomy of intrusion-detection systems". *Computer Networks*, Volumen 31, Issue 8, pp. 805-522.

[DH88] B.D. Davison and H. Hirsh, 1988. "Predicting sequences of user actions". En Proc. Workshop on Predicting the Future: AI Approaches to Time-Series Analysis. Madison, WI, USA, pp. 5-12.

[DKPS05] H. Dreger, C. Kreibich, V. Paxson, and R. Sommer, 2005. "Enhancing the accuracy of network-based intrusion detection with host-based context". En Proc. 2nd International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), Vienna, Austria. Volumen 3548, pp. 206-221.

[DLBM14] X. Ding, Y. Li, A. Belatreche, and L.P. Maguire, 2014. "An experimental evaluation of novelty detection methods". *Neurocomputing*, Volumen 135, pp. 313-327.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin, 1977. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society*, Series B, Volumen 39, No. 1, pp. 1-38.

[DMC15] Q. Do, B. Martini, and K.K.R. Choo, 2015. "Exfiltrating data from Android devices". *Computers & Security*, Vol 48, pp. 74-91.

[DP09] M. Djioua and R. Plamondon, 2009. "Studying the variability of handwriting patterns using the Kinematic Theory". *Human Movement Science*, Volumen 28, Issue 5, pp. 588-601.

[DPR08] J. Daudin, F. Picard, and S. Robin, 2008. "A mixture model for random graphs". *Statistics and Computing*, Volumen 18, Issue 2, pp. 173-183.

[DRAP15] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, 2015. "Learning in nonstationary environments: a Survey". *IEEE Computational Intelligence Magazine*, Volumen 10, Issue 4, pp. 12-25.

[DRT11] H. Deng, G. Runger, and E. Tuv, 2011. "Bias of importance measures for multi-valued attributes and solutions". En Proc. 21st international Conference on Artificial Neural Networks (ICANN'11), Espoo, Finland, Volumen 2, pp. 293-300.

[Dum99] W. Dumouchel, 1999. "Computer intrusion detection based on bayes factors for comparing command transition probabilities". *National Institute of Statistical Sciences*, No. 91.

[DYN11] D. Dasgupta, S. Yu, and F. Nino, 2011. "Recent Advances in Artificial Immune Systems: Models and Applications". *Applied Soft Computing*, Volumen 11, Issue 2, pp. 1574-1587.

[EA12] P. Esling and C. Agon, 2012. "Time-series data mining". *ACM Computing Surveys*, Volumen 45, Issue 1, No. 12.

[Edg87] F.T. Edgeworth, 1887. "On discordant observations", *Philosophical Magazine* Volumen 23, No. 5.

[EFB+15] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, 2015. "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems". *Expert Systems with Applications*, Volumen 42, Issue 1, pp. 193.-202.

[EH07] W. Eberle and L. Holder, 2007. "Anomaly detection in data represented as graphs". *Intelligent Data Analysis*, Volumen 11, Issue 6, pp. 663-689.

[End88] M.R. Endsley, 1988. "Design and evaluation for situation awareness enhancement". En Proc. 32nd Annual Meeting on Human Factors Society, Santa Monica, CA,US, pp. 97-101.

[ENI15] ENISA, 2015. "ENISA Threat Landscape 2015". Available at: https://www.enisa.europa.eu/publications/etl2015.

[EO11] H.T. Elshoush and I.M. Osman, 2011. "Alert correlation in collaborative intelligent intrusion detection systems-A survey". *Applied Soft Computing*, Volumen 11, No. 7, pp. 4349-4365.

[EO13] H.T. Elshoush and I.M. Osman, 2013. "Intrusion alert correlation framework: an innovative approach". *IAENG Transactions on Engineering Technologies*. Lecture Notes in Electrical Engineering, Volumen 229, pp. 405-420.

[EP11] R. Elwell and R. Polikar, 2011. "Incremental learning of concept drift in nonstationary environments". *IEEE Transactions on Neural Networks*, Volumen 22, No. 10.

[ER60] P. Erdos and A. Renyi, 1960. "On the evolution of random graphs". *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, Volumen 5, Issue 1, pp. 17-60.

[ETS14] ETSI, 2014. "GS NFV 002 V1.2.1: Network Functions Virtualisation (NFV); architectural framework", Available at http://www.etsi.org/technologies-clusters/technologies/nfv.

[Eur14] Europol, 2014. "Police ransomware threat assessment". Available at https://www.europol.europa.eu/.

[Eur16] Europol, 2016. "The Internet Organised Crime Threat Assessment (iOCTA)".

[FAK11] H. Farhadi, M. AmirHaeri, and M. Khansari, 2011. "Alert correlation and prediction using data mining and HMM". *International Journal of Information Security*, Volumen 3, No. 2, pp. 77-101.

[FASW15] A. Feizollah, N.B. Anuar, R. Salleh, and A.W.A. Wahab, 2015. "A review on feature selection in mobile malware detection". *Digital Investigation*, Volumen 13, pp. 23-37.

[Faw06] T. Fawcett, 2006. "An introduction to ROC analysis". *Pattern Recognition Letters*, Volumen 27, Issue 8, pp. 861-874.

[FBL15] P. Faruki, A. Bharmal, and V. Laxmi, 2015. "Android security: a survey of issues, malware penetration, and defenses". *IEEE Communications Surveys & Tutorials*, Volumen17, Issue 2, pp. 998-1022.

[FHOM09] C. Ferri, J. Hernández-Orallo, and R. Modroiu, 2009. "An experimental comparison of performance measures for classification". *Pattern Recognition Letters*, Volumen 31, Issue 1, pp. 27-38.

[FIR15] FIRST, 2015. CVSS: Common Vulnerability Scoring System. Available: https://www.first.org/cvss/specification-document.

[FMRH16] A. Foss, M. Markatou, B. Ray, and A. Heching, 2016. "A semiparametric method for clustering mixed data". *Machine Learning*, doi:10.1007/s10994-016-5575-7, pp. 1-40.

[FRP15] G. Fernandes, J.J.P.C Rodrigues, and M.L. Proenca, 2015. "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis". *Applied Soft Computing*, Volumen 34, pp. 513-525.

[FV14] B. Frénay and M. Verleysen, 2014. "Classification in the presence of label noise: a survey", *IEEE Transactions on Neural Networks and Learning Systems*. Volumen 25, No, 5, pp.845-869.

[FZ02] A. Foss and O.R. Zaiane, 2002. "A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets". En Proc. IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, pp. 179-186.

[FZFW09] H. Fan, O.R. Zaiane, A. Foss, and J. Wu, 2009. "Resolution-based outlier factor: detecting the top-n most outlying data points in engineering data". *Knowledge and Information Systems*, Volumen 19, Issue 1, pp. 31-51.

[GA13] S. Geravand and M. Ahmadi, 2013. "Bloom filter applications in network security: a state-of-the-art survey". *Computer Networks*, Volumen 57, pp. 4047-4064.

[Gaf13] T. Gaffney, 2013. "Following in the footsteps of Windows: how Android malware development is looking very familiar". *Network Security*, Volumen 2013, Issue 8, pp. 7-10.

[GAT10] J. Greensmith, U. Aickelin, and G. Tedesco, 2010. "Information fusion for anomaly detection with the dendritic cell algorithm". *Information Fusion*, Volumen 11, No. 1, pp. 21-34.

[GD80] E.S. Gardner and D.G. Dannenbring, 1980. "Forecasting with exponential smoothing: some guidelines for model selection". *Decision Sciences*, Volumen 11, No. 2, pp. 370-383.

[GFKS15] D. Geneiatakis, I.N. Fovino, I. Kounelis, and P. Stirparo, 2015. "A permission verification approach for android mobile applications". *Computers & Security*, Volumen 49, pp. 195-205.

[GGI+15] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, and A. Winnicki, 2015. "Cyber-physical systems security: experimental analysis of a vinyl acetate monomer plant". En Proc. 1st ACM Workshop on Cyber-Physical System Security (CPSS), Singapore, pp. 1-12.

[GH06] P. Galinier and A. Hertz, 2006. "A survey of local search methods for graph coloring". *Computers & Operations Research*, Volumen 33, Issue 9, pp.2547-2562.

[GKLD16] D. Gkounis, V. Kotronis, C. Liaskos, and X.L. Dimitropoulos, 2016. "On the interplay of link-flooding attacks and traffic engineering". *ACM SIGCOMM Computer Communication Review*, Volumen 46, Issue 2, pp. 5-11.

[GKRB13] N. Gornitz, M. Kloft, K. Rieck, and U. Brefeld, 2013. "Toward supervised anomaly detection". *Journal of Artificial Intelligence Research*, Volumen 46, Issue 1 , pp. 235-262.

[GMW07] G. Gan, C. Ma, and J. Wu, 2007. "Data clustering: theory, algorithms, and applications". *ASA-SIAM*, Volumen 20, Julio 2007.

[GOKO10] D. Geng, T. Odaka, J. Kuroiwa, and H. Ogura, 2010. "An N-Gram and STFIDF model for masquerade detection in a UNIX environment". *Journal in Computer Virology*, Volumen 7, Issue 2, pp. 133-142.

[GOO14] B.C. Gencosman, H.C. Ozmutlu, and S. Ozmutlu, 2014. "Character n-gram application for automatic new topic identification". *Information Processing & Management*, Volumen 50, Issue 6, pp. 821-856.

[GPLL16] C. Guo, Y. Ping, N. Liu, and S.S. Luo, 2016. "A two-level hybrid approach for intrusion detection". *Neurocomputing*, Volumen 214, pp. 391-400.

[Gro73] G. Groff, 1973. "Empirical comparison of models for short-range forecasting". *Management Sciences* Volumen 21, Issue 1, pp. 22-31.

[Gru69] F.E. Grubbs, 1969. "Procedures for detecting outlying observations in samples", *Technometrics*, Volumen 11, No. 1, pp. 1-21.

[GTDVMFV09] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, 2009. "Anomaly-based network intrusion detection: techniques, systems and challenges". *Computers & Security*, Volumen 25, Issues 1-2, pp. 18-28.

[GTDVTSH15] P. Garcia-Teodoro, J.E. Diaz-Verdejo, J.E. Tapiador, and R. Salazar-Hernandez, 2015. "Automatic generation of HTTP intrusion signatures by selective identification of anomalies". Computers & Security, Volumen 55, pp. 159-174.

[GU16] M. Goldstein and S. Uchida, 2016. "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data". *PloS one*, DOI:10.1371/journal.pone.0152173.

[GVMVSO17] L.J. García Villalba, J. Maestre Vidal, and A.L. Sandoval Orozco, 2017. "Advanced payload analyzer preprocessor". *Future Generation Computer Systems*, Volumen 76, pp.474-485.

[GVSOMV15] L.J. García Villalba, A.L. Sandoval Orozco, and J. Maestre Vidal, 2015. "Malware detection system by payload analysis of network traffic". *IEEE Latin America Transactions*, Volumen 13, No. 3, pp. 850-855.

[Han06] D.J. Hand, 2006. "Classifier technology and the illusion of progress". *Statistical Science*, Volumen 21, No. 1, pp. 1-14.

[Haw80] D. Hawkings, 1980. "Identification of outliers". *Monographs on Applied Probability and Statistics*, Springer Netherland.

[HC02] E. Hung and D.W. Cheung, 2002. "Parallel mining of outliers in large database". *Distributed and Parallel Databases*, Volumen 12, Issue 1, pp. 5-26.

[HCPD+16] K. Haufe, R. Colomo-Palacios, S. Dzombeta, K. Brandis, and V. Stantchev, 2016. "Security management standards: a mapping". *Procedia Computer Science*, Volumen 100, pp. 755-761.

[HDND15] X. He, H. Dai, P. Ning, and R. Dutta, 2015. "Dynamic IDS configuration in the presence of intruder type uncertainty". En Proc. IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, pp. 1-6.

[HKC+15] F. Harrou, F. Kadri, S. Chaabane, C. Tahon, and Y. Sun, 2015. "Improved principal component analysis for anomaly detection: Application to an emergency department". *Computers & Industrial Engineering*, Volumen 88, pp. 63-77.

[HKOS05] R.J. Hyndman, A.B. Koehler, J.K. Ord, and R.D. Snyder, 2005. "Prediction intervals for exponential smoothing state space models". *Journal of Forecasting*, Volumen 24, pp. 17-37.

[HKP11] J. Han, M. Kamber, and J. Pei, 2011. "Data mining: concepts and techniques". Elsevier.

[HM16] D. Hieu and P. Meesad, 2016. "A fast outlier detection algorithm for big datasets". En Proc. 12th International Conference on Computing and Information Technology (IC2IT), Khon Kaen, Thailand, Volumen 463, pp. 159-169.

[HNH13] J. Hoffmann, S. Neumann, and T. Holz, 2013. "Mobile malware detection based on energy fingerprints- a dead end?". En Proc. 16th International Symposium of Research in Attacks, Intrusions, and Defenses (RAID), Rodney Bay, St. Lucia. Lecture Notes in Computer Science, Volumen 8145, pp. 348-368.

[Hol93] R.C. Holte, 1993. "Very simple classification rules perform well on most commonly used datasets". *Machine Learning*, Volumen 11, Issue 1, pp. 63-90.

[HS11] L. Huang and M. Stamp, 2011. "Masquerade detection using profile hidden Markov models". *Computers & Security*, Volumen 30, Issue 8, pp. 732-747.

[HS14] N. Hubballi and V. Suryanarayanan, 2014. "False alarm minimization techniques in signature-based intrusion detection systems: A survey". *Computer Communications*, Volumen 49, pp. 1-17.

[HSB+12] D. Hadziosmanovik, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, 2012. "N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols". En Proc. 15th International Symposium on Recent Advances in Intrusion Detection (RAID), Amsterdam, The Netherlands, Volumen 7462, pp. 59- 81.

[HWS+13] K.E. Heckman, M.J. Walsh, F.J. Stech, T.A. Boyle, S.R. DiCato, and A.F. Herber, 2013. "Active cyber defense with denial and deception: a cyber-wargame experiment". *Computers & Security*, Volumen 37, pp. 72-77.

[IET92] IETF, 1992. "The MD5 Message-Digest Algorithm". RFC 1321, Available at http://www.ietf.org/rfc/rfc1321.txt.

[IM10] A. Iwasaki and R. Medzhitov, 2010. "Regulation of adaptive immunity by the innate immune system". *Science*, Volumen 327, Issue 5963, pp. 291-295.

[ISA12] ISACA, 2012. "Cobit 5", ISBN:1604202378-9781604202373.

[ISA15] ISACA, 2015. "2015 Global cybersecurity status report". Available at http://www.isaca.org/pages/cybersecurity-global-status-report.aspx.

[ISO06] ISO, 2006. ISO/IEC 18043:2006, "Selection, deployment and operations of intrusion detection systems".

[ISO13] ISO, 2013. ISO/IEC 27001:2013: "Information technology – Security techniques – Information security management systems – Requirements", 2013. Available: http://www.iso.org/iso/catalogue_detail?csNo.=54534.

[ISO15] ISO, 2015. ISO/IEC 27039:2015, "Selection, deployment and operations of intrusion detection systems (IDPS)".

[Jai10] A.K. Jain, 2010. "Data clustering: 50 years beyond K-means". En Proc. 19th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey. *Pattern Recognition Letters*, Volumen 31, No. 8, pp. 651-666.

[JBA14] P. Jesus, C. Baquero, and P.S. Almeida, 2014. "A Survey of distributed data aggregation algorithms". *IEEE Communications Surveys & Tutorials*, Volumen 17, Issue 1, pp. 381-404.

[JCNJ90] F.V. Jensen, B. Chamberlain, T. Nordahl, and F. Jensen, 1990. "Analysis in hugin of data conflict". En Proc. 6th Annual Conference on Uncertainty in Artificial Intelligence, Cambridge, MA, USA, pp. 519-528.

[JD07a] G.A. Jacoby and N.J. Davis, 2007. "Mobile host-based intrusion detection and attack identification". *IEEE Wireless Communications*, Volumen 14, Issue 4, pp. 53-60.

[JD07b] Z. Ji and D. Dasgupta, 2007. "Revisiting negative selection algorithms". *Evolutionary Computation*, Volumen 15, Issue 2, pp. 223-251.

[JDC+16] Y. Jiang, Z. Deng, K.S. Choi, F.L. Chung, and S. Wang, 2016. "A novel multi-task TSK fuzzy classifier and its enhanced version for labeling-risk-aware multi-task classification". *Information Sciences*, Volumen 357, pp. 39-60.

[JL14] E. Jeong and B. Lee, 2014. "An IP traceback protocol using a compressed hash table, a sinkhole router and data mining based on network forensics against network attacks". *Future Generation Computer Systems*, Volumen 33, No.1, pp. 42-52.

[JST+07] Z. Jian, H. Shirai, I. Takahashi, J. Kuroiwa, T. Odaka, and H. Ogura, 2007. "Masquerade detection by boosting decision stumps using UNIX commands". *Computers & Security*, Volumen 26, Issue 4, pp. 311-318.

[JTH+13] A. Jamdagni, Z. Tan, X. He, P. Nanda, and R.P. Liu, 2013. "RePIDS: a multi tier realtime payload-based intrusion detection system". *Computer Networks*, Volumen 57, pp. 811-824.

[JTHW06] W. Jin, A.K.H. Tung, J. Han, and W. Wang, 2006. "Ranking outliers using symmetric neighborhood relationship". En Proc. 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06), Singapore, pp.577-593.

[JTJS14] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann, 2014. "The sniper attack: anonymously deanonymizing and disabling the tor network". En Proc. 18th Symposium on Network and Distributed System Security (NDSS), San Diego, Ca, US.

[JV01] H.W. Ju and Y. Vardi, 2001. "A hybrid high-order Markov chain model for computer intrusion detection". *Journal of Computational and Graphical Statistics*, Volumen 10, Issue 2, pp. 277-295.

[JW00] P. Jogalekar and M. Woodside, 2000. "Evaluating the scalability of distributed systems". *IEEE Transactions on Parallel and Distributed Systems*, Volumen 11, Issue 6, pp. 589-603.

[KBH15] H.A. Kholidy, F. Baiardi, and S. Hariri, 2015. "DDSGA: a Data-driven semi-global alignment approach for detecting masquerade attacks". *IEEE Transactions on Dependable and Secure Computing*, Volumen 12, Issue 2, pp. 164-178.

[KD12] A. Kumar and H. Daume, 2012. "Learning task grouping and overlap in multi-task learning". En Proc. 29th International Conference on Machine Learning (ICML'12), pp. 1383-1390, Edinburgh, Scotland.

[KDD99] KDD, 1999. The KDD Cup 1999 Dataset.

[KHA99] M.G. Kelly, D.J. Hand, and N.M. Adams, 1999. "The impact of changing populations on classifier performance". En Proc. 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, USA, pp. 36-371.

[KHC+16] F. Kadri, F. Harrou, S. Chaabane, Y. Sun, and C. Tahon, 2016. "Seasonal ARMA-based SPC charts for anomaly detection: application to emergency department systems". *Neurocomputing*, Volumen 173, Part 3, pp. 2102-2114.

[Kis13] R. Kissel, 2013. "Glossary of key information security terms". National Institute of Standards and Technology (NIST) Interagency or Internal Report 7298r2.

[KJS16] J. Kevric, S. Jukic, and A. Subasi, 2016. "An effective combining classifier approach using tree algorithms for network intrusion detection". *Neural Computing and Applications*, pp. 1-8, doi:10.1007/s00521-016-2418-1.

[KKZ10] H.P. Kriegel, P. Kroger, and A. Zimek, 2010. "Outlier detection techniques". En Proc. 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), Washington, DC.

[Kli04] R. Klinkenberg, 2004. "Learning drifting concepts: example selection vs. example weighting". *Intelligent Data Analysis*, Volumen 8, No. 3, pp. 281-300.

[KN97] E.M. Knorr and R.T. Ng, 1997. "A unified notion of outliers: properties and computation". En Proc. 3th International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, USA, pp. 219-222.

[KN98] E.M. Knorr and R.T. Ng, 1998. "Algorithms for mining distance-based outliers in large datasets". En Proc. 24rd International Conference on Very Large Data Bases (VLDB), New York, NY, USA, pp. 392, 403.

[Koh18] R. Kohavi, 2018. "A study of cross-validation and bootstrap for accuracy estimation and model selection". En Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI'95), Montreal, Quebec, Canada, Volumen 2, pp. 1137-1143.

[Koy00] I. Koychev, 2000. "Gradual forgetting for adaptation to concept drift". En Proc. ECAI Workshop on Current Issues in Spatio-Temporal Reasoning, Berlin, Germany, pp. 101-106.

[KS13] P.A.R. Kumar and S. Selvakumar, 2013. "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems", *Computer Communications*, Volumen 36, No. 3, pp. 303-319.

[KVF+12] S. Khanna, S.S. Venkatesh, O. Fatemieh, F. Khan, and C.A. Gunter, 2012. "Adaptive selective verification: an efficient adaptive countermeasure to thwart DoS attacks". *IEEE/ACM Transactions on Networking*, Volumen 20, No. 3, pp. 715-728.

[KWG01] A. Kamrani, R. Wang, and R. Gonzalez, 2001. "A genetic algorithm methodology for data mining & intelligent knowledge acquisition". *Computers & Industrial Engineering*, Volumen 40, Issue 4, 361-377.

[KYZ15] D.W. Kim, P. Yan, and J. Zhang, 2015. "Detecting fake anti-virus software distribution webpages". *Computers & Security*, Volumen 49, pp. 95-106.

[Lab98] MIT Lincoln Laboratory, 1998. "DARPA intrusion detection evaluation data set 1998" (DARPA'98). Available at https://ll.mit.edu/ideval/data/1998data.html.

[Lab99] MIT Lincoln Laboratory, 1999. "DARPA Intrusion Detection Evaluation Dataset 1999" (DARPA 99).

[Lab16] McAfee Labs, 2016. "McAfee labs threats report: Junio 2016". Available at http://www.mcafee.com/ca/resources/reports/.

[LDSR05] P. Laskov, P. Dussel, C. Schafer, and K. Rieck, 2005. "Learning intrusion detection: supervised or unsupervised?". En Proc. 13th International Conference on Image Analysis and Processing (ICIAP 2005), Cagliari, Italy, pp. 50-57.

[Lev66] V.I. Levenshtein, 1966. "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady*, Volumen 10, pp. 707-710.

[LG07] Y. Li and L. Guo, 2007. "An active learning based TCM-KNN algorithm for supervised network intrusion detection". *Computers & Security*, Volumen 26, Issues 7-8, pp. 459-467.

[LHPW16] L. Li, R.J. Hansman, R. Palacios, and R. Welsch, 2016. "Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring". *Transportation Research Part C: Emerging Technologies*, Volumen 64, pp. 45-57.

[LHWP15] D. Li, G. Hu, Y. Wang, and Z. Pan, 2015. "Network traffic classification via non-convex multi-task feature learning". *Neurocomputing*, Volumen 152, pp. 322-332.

[Lia14] H. Liang, 2014. "An improved intrusion detection based on neural network and fuzzy algorithm". *Journal of Networks*, Volumen 9, Issue 5, pp. 1274-1280.

[LKLP12] S.M. Lee, D.S. Kim, J.H. Lee, and J.S. Park, 2012. "Detection of DDoS attacks using optimized traffic matrix". *Computers & Mathematics with Applications*, Volumen 63, Issue 2, pp. 501-510.

[LKT15] W.C. Lin, S.W. Ke, and C.F. Tsai, 2015. "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors". *Knowledge-Based Systems*, Volumen 78, pp. 13-21.

[LL16] L. Lu and Y. Liu, 2016. "Safeguard: user reauthentication on smartphones via behavioral biometrics". *IEEE Transactions on Computational Social Systems*, Volumen 2, Issue 3, pp. 53-64.

[LLCT13] Y.D. Lin, Y.C. Lai, C.H. Chen, and H.C. Tsai, 2013. "Identifying android malicious repackaged applications by thread-grained system call sequences". *Computers & Security*, Volumen 39 pp. 340-350.

[LLL13] H.J. Liao, C.H.R. Lin, and K.Y. Lin, Y.C.and Tung, 2013. "Intrusion detection system: a comprehensive review". *Journal of Network and Computer Applications*, Volumen 36, Issue 1, pp. 16-24.

[LLZ09] J. Liu, X. Li, and W. Zhong, 2009. "Ambiguous decision trees for mining concept-drifting data streams". *Pattern Recognition Letters*, Volumen 30, No. 15, pp. 1347-1355.

[LLZ16] D. Li, S. Liu, and H. Zhang, 2016. "A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection". *Engineering Applications of Artificial Intelligence*, Volumen 50, pp. 93-105.

[LLZZ13] H. Luo, Y. Lin, H. Zhang, and M. Zukerman, 2013. "Preventing DDoS attacks by identifier/locator separation". *IEEE Network*, Volumen 27, Issue 6, pp. 60-65.

[LMS+02] W. Lee, M. Miller, S. Stolfo, W. Fan, and E. Zadok, 2002. "Toward cost-sensitive modeling for intrusion detection and response". *Journal of Computer Security*, Volumen 10, Issue 1, pp. 5-22.

[LMT+16] Y. Lu, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, 2016. "An efficient and scalable density-based clustering algorithm for datasets with complex structures". *Neurocomputing*, Volumen 171, pp. 9-22.

[LMV14] W. Li, V. Mahadevan, and N. Vasconcelos, 2014. "Anomaly detection and localization in crowded scenes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volumen 36, No. 1.

[LPMS13] M. La Polla, F. Martinelli, and D. Sgandurra, 2013. "A survey on security for mobile devices". *IEEE Communications Surveys & Tutorials*, Volumen 15, Issue 1, pp. 446-471.

[LRB08] M.J. Lesot, M. Rifqi, and H. Benhadda, 2008. "Similarity measures for binary and numerical data: a survey". *International Journal of Knowledge Engineering and Soft Data Paradigms*, Volumen 1, Issue 1, pp. 63-84.

[LSM99] W. Lee, V.S. Stolfo, and W. Mok, 1999. "A data mining framework for building intrusion detection models". En Proc. 1999 IEEE Symposium on Security and Privacy, Oakland, California, USA, pp. 120-132.

[LSVK11] H. Liu, Y. Sun, V. Valgenti, and M. Kim, 2011. "TrustGuard: a low-level reputation-based DDoS defense system". En Proc. IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, US; pp. 287-291.

[LYW13] Y.J. Lee, Y.R. Yeh, and Y.C.F. Wang, 2013. "Anomaly detection via online oversampling principal component analysis". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 25, Issue 7, pp. 1460-1470.

[LYW+15] C. Li, J. Yang, Z. Wang, F. Li, and Y. Yang, 2015. "A lightweight DDoS flooding attack detection algorithm based on synchronous long flows". En Proc. IEEE Global Communications Conference (GLOBECOM), San Diego, CA, US.

[MAC+15] D. Maiorca, D. Ariu, I. Corona, M. Aresu, and G. Giacinto, 2015. "Stealth attacks: an extended insight into the obfuscation effects on Android malware". *Computers & Security*, Volumen 51, pp. 16-31.

[MAJ13] S.A. Mirheidari, S. Arshad, and R. Jalili, 2013. "Alert correlation algorithms: a survey and taxonomy". En Proc. 5th International Symposium on Cyberspace Safety and Security (CSS), Zhangjiajie, China, Volumen 8300, pp. 183-197.

[Man13] D. Manky, 2013. "Cybercrime as a service: a very modern business". *Computer Fraud & Security*, Volumen 2013, Issue 6, pp. 9-13.

[MC15] R. Mitchell and I.R. Chen, 2015. "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems". *IEEE Transactions on Dependable and Secure Computing*, Volumen 12, Issue 1, pp. 16-30.

[McC91] J. McCumber, 1991. "Information systems security: a comprehensive model". En Proc. 14th NIST National Computer Security Conference, Washington, D.C. USA, pp. 328-337.

[MCZS15] H.O. Marques, R.J.G.B. Campello, A. Zimek, and J. Sander, 2015. "On the internal evaluation of unsupervised outlier detection". En Proc. 27th International Conference on Scientific and Statistical Database Management (SSDBM '15), San Diego, CA, USA, No. 7, pp. 1-12.

[MDJ12] Y.B. Mustapha, H. Débar, and G. Jacob, 2012. "Limitation of honeypot/honeynet databases to enhance alert correlation". En Proc. 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, St. Petersburg, Russia, Volumen 7531, pp. 203-217.

[MJ02] R. Medzhitov and C.A. Janeway, 2002. "Decoding the patterns of self and nonself by the innate immune system". *Science*, Volumen 296, Issue 5566, pp. 298-300.

[MJS02] S. Mukkamala, G. Janoski, and A. Sung, 2002. "Intrusion detection using neural networks and support vector machines". En Proc. IEEE International Joint Conference on Neural Networks, Honolulu, HI, USA, pp. 1702-1707.

[MK15] K. Malialis and D. Kudenko, 2015. "Distributed response to network intrusions using multiagent reinforcement learning". *Engineering Applications of Artificial Intelligence*, Volumen 41, pp. 270-284.

[MLK14] W. Meng, W. Li, and L.F. Kwok, 2014. "EFM: Enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism". Computers & Security, Volumen 43, pp. 189-204.

[MNK14] S. Mascaro, A. Nicholson, and K. Korb, 2014. "Anomaly detection in vessel tracks using Bayesian networks". *International Journal of Approximate Reasoning*, Volumen 55, Issue 1, pp. 84-98.

[MPBMOO15] F. Maciá-Pérez, J.V. Berna-Martinez, A.F. Oliva, and A.A. Ortega, 2015. "Algorithm for the detection of outliers based on the theory of rough sets". *Decision Support Systems*, Volumen 75, pp. 53-75.

[MSBF15] R. Murmuria, A. Stavrou, D. Barbará, and D. Fleck, 2015. "Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users". En Proc. 18th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID), Kyoto, Japan, pp. 405-424.

[MT04] R.A. Maxion and T.N. Townsend, 2004. "Masquerade detection augmented with error analysis". *IEEE Transactions on Reliability*, Volumen 53, Issue 1, pp. 124-147.

[Mur11] K. Murphy, 2011. "Janeway's immunobiology". 8th edition, Garland Science, Taylor & Francis LLC, New York, NY, US.

[MVK+15] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B.D. Payne, 2015. "Evaluating computer intrusion detection systems: a survey of common practices". *ACM Computing Surveys*, Volumen 48 Issue 1, No. 12, pp. 1-41.

[MVSMGV18] J. Maestre Vidal, M.A. Sotelo Monge, and L.J. García Villalba, 2018. "A novel pattern recognition system for detecting android malware by analyzing suspicious boot sequences". *Knowledge-Based Systems*, Volumen 150, pp. 198-217.

[MVSOGV14] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2014. "Sistema de detección de atacantes enmascarados basado en técnicas de alineamiento de secuencias". En Proc. XIII Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Alicante, Spain.

[MVSOGV15a] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2015. "Network intrusion detection systems in data centers", Handbook on Data Centers, Springer New York, pp. 1185-1207.

[MVSOGV15b] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2015. "Quantitative criteria for alert correlation of anomalies-based NIDS". *IEEE Latin America Transactions*, Volumen 13, No. 10, pp. 3461-3466.

[MVSOGV15c] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2015. "Correlación de alertas en la detección de malware en redes basada en anomalías". En Proc. 1st Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), León, Spain.

[MVSOGV15d] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2015. "Sistema inmunitario adaptativo para la mitigación de ataques de denegación de servicio". En Proc. 1st Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), León, Spain.

[MVSOGV15e] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J Garcia Villalba, 2015. "Malware detection in mobile devices analyzing sequences of system calls". En Proc. 8th Congreso Iberoamericano de Seguridad Informática (CIBSI), Quito, Ecuador.

[MVSOGV16] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2016. "Online masquerade detection resistant to mimicry". *Expert Systems with Applications*, Volumen 61, pp. 162-180.

[MVSOGV17] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2017. "Alert correlation framework for malware detection by anomaly-based packet payload analysis". *Journal of Network and Computer Applications*, Volumen 29, pp. 11-22.

[MVSOGV18] J. Maestre Vidal, A.L. Sandoval Orozco, and L.J. García Villalba, 2018. "Adaptive artificial immune networks for mitigating DoS flooding attacks". *Swarm and Evolutionary Computation*, Volumen 38, pp. 94-108.

[MW46] H.B. Mann and D.R. Whitney, 1946. "On a test of whether one of two random variables is stochastically larger than the other". *The Annals of Mathematical Statistics*, Volumen 18, No. 1, pp. 50-60.

[MWH98] S. Makridakis, S.C. Wheelwright, and R.J. Hyndman, 1998. "Forecasting: methods and applications". John Wiley & Sons, 1998.

[MWY10] L.L. Minku, A.P. White, and X. Yao, 2010. "The impact of diversity on online ensemble learning in the presence of concept drift". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 22, No. 5, pp. 731-742.

[MY12] L.L. Minku and X. Yao, 2012. "DDD: A new ensemble approach for dealing with concept drift". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 24, No. 4, pp. 619-633.

[ND08] W. Ng and M. Dash, 2008. "A test paradigm for detecting changes in transactional data streams". *Database Systems for Advanced Applications*, Springer-Verlag, Berlin, Germany, pp. 204-219.

[NIS18] NIST, 2018. NIST-SP800 series: special publications on computer security. Available: http://csrc.nist.gov/publications/PubsSPs.html#SP 800.

[NJKH13] H.W. Njogu, L. Jiawei, J.N. Kiere, and D. Hanyurwimfura, 2013. "A comprehensive vulnerability based alert management approach for large networks". *Future Generation Computer Systems*, Volumen 29, Issue 1, pp. 27-45.

[NP97] P.G. Neumann and P.A. Porras, 1997. "EMERALD: Event monitoring enabling responses to anomalous live disturbances". En Proc. 20th NIST National Information Systems Security Conference, Baltimore, USA, pp. 353-365.

[NW70] S.B. Needleman and C.D. Wunsch, 1970. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of molecular biology*, Volumen 48, Issue 3, pp. 443-453.

[OAGVSO+16] R. Oliveira Albuquerque, L.J. García Villalba, A.L. Sandoval Orozco, R. Timóteo de Sousa, and T.H. Kim, 2016. "Leveraging information security and computational trust for cybersecurity". *The Journal of Supercomputing*, Volumen 72, Issue 10, pp- 3729-3763.

[OB15] I. Ozcelik and R.R. Brooks, 2015. "Deceiving entropy based DoS detection". *Computers & Security*, Volumen 48, No. 1, pp. 234-245.

[OGIR14] C. OReilly, A. Gluhak, M.A. Imran, and S. Rajasegarar, 2014. "Anomaly detection in wireless sensor networks in a non-stationary environment". *IEEE Communications Surveys & Tutorials*, Volumen 16, No. 3, pp. 1413-1432.

[OOAK04] M. Oka, Y. Oyama, H. Abe, and K. Kato, 2004. "Anomaly detection using layered networks based on eigen co-occurrence matrix". En Proc. 7th International Symposium on Recent Advances in Intrusion Detection (RAID), Sophia Antipolis, France, Volumen 3224, pp. 223-237.

[Orl98] E. Orlowska, 1998. "Incomplete information: Rough set analysis". *Studies in Fuzziness and Soft Computing*, Volumen 13, Physica-Verlag Heidelberg.

[ORLS14] A. Oza, K. Ross, R.M. Low, and M. Stamp, 2014. "HTTP attack detection using n-gram analysis". *Computers & Security*, Volumen 45, pp. 242-254.

[Ou12] C.M Ou, 2012. "Host-based intrusion detection systems adapted from agent-based artificial immune systems". *Neurocomputing*, Volumen 88, No. 1, pp. 78-86.

[PAF+09] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, 2009. "McPAD: a multiple classifier system for accurate payload-based anomaly detection". *Computer Networks*, Volumen 53, Issue 6, pp. 864-881.

[Par98] D.B. Parker, 1998. "Fighting computer crime: a new framework for protecting information". Wiley & Sons, New York.

[Par15] P. Parham, 2015. "The immune system". 4th edition, Garland Science, Taylor & Francis Group LLC, New York, NY, US. ISBN: 978-0-8153-4466-7.

[Pau16] C. Paulsen, 2016. "Cybersecuring small businesses". *Computer*, Volumen 49, Issue 8, pp. 92-97.

[Paw82] Z. Pawlak, 1982. "Rough sets". *International Journal of Computer & Information Sciences*, Volumen 11, Issue 5, pp. 341-356.

[PDP+15] I. Polakis, M. Diamantaris, T. Petsas, F. Maggi, and S. Ioannidis, 2015. "Powerslave: analyzing the energy consumption of mobile antivirus software". En Proc. 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), Milan, Italy, pp. 165-184.

[Pev16] T. Pevný, 2016. "Loda: lightweight on-line detector of anomalies". *Machine Learning*, Volumen 102, pp. 275-304.

[PFR17] I. Pillai, G. Fumera, and F. Roli, 2017. "Designing multi-label classifiers that maximize F measures: State of the art". *Pattern Recognition*, Volumen 61, pp. 394-404.

[PG14] M. Parodi and J.C. Gómez, 2014. "Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations". *Pattern Recognition*, Volumen 47, Issue 1, pp. 128-140.

[PLGMI16] R. Piltaver, M. Lustrek, M. Gams, and S. Martincic-Ipsic, 2016. "What makes classification trees comprehensible?". *Expert Systems with Applications*, Volumen 62, pp. 333-346.

[POTPL14] S. Pastrana, A. Orfila, J.E. Tapiador, and P. Peris-Lopez, 2014. "Randomized anagram revisited". *Journal of Network and Computer Applications*, Volumen 21, pp. 182-186.

[PSS16] N. Panwar, S. Sharma, and A.K. Singh, 2016. "A survey on 5G: the next generation of mobile communication". *Physical Communication*, Volumen 18, Issue 2, pp. 64-84.

[PSSC+90] P. Porras, D. Schnackenberg, S. Staniford-Chen, M. Stillman, and F. Wu, 1990. "The common intrusion detection framework architecture", CIDF Working Group, Available: http://gost.isi.edu/cidf/.

[PSW16] S. Parkinson, V. Somaraki, and R. Ward, 2016. "Auditing system permissions using association rule mining". *Expert Systems with Applications*, Volumen 55, pp. 274-283.

[PTC+14] M.G. Pérez, J.E. Tapiador, J.A. Clark, G.M. Pérez, and A.F.S. Gómez, 2014. "Trustworthy placements: improving quality and resilience in collaborative attack detection". *Computer Networks*, Volumen 58, pp. 70-86.

[PXY+13] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie, 2013. "WHYPER: towards automating risk assessment of mobile applications". En Proc. 22nd USENIX Conference on Security, Washington, D.C, US. Volumen 13, pp. 527-542.

[PYY14] S. Peng, S. Yu, and A. Yang, 2014. "Smartphone malware and its propagation modeling: a survey". *IEEE Communications Surveys & Tutorials*, Volumen 16, Issue 2, pp. 925-641.

[Qui83] J.R. Quinlan, 1983. "Learning efficient classification procedures and their application to chess end games". *Machine Learning - An Artificial Intelligence Approach*, Tioga Publishing Company, pp. 463-482.

[Qui86] J.R. Quinlan, 1986. "Induction of decision trees". *Machine Learning*, Volumen 1, No. 1, pp. 81-106.

[Qui93] J.R. Quinlan, 1993. "C4.5: programs for machine learning". Morgan Kaufmann Publishers, California.

[RAA15] A.A. Ramaki, M. Amini, and R.E. Atani, 2015. "RTECA: real time episode correlation algorithm for multi-step attack scenarios detection". *Computers & Security*, Volumen 49, pp. 206-219.

[RAAQ11] E. Rendón, I. Abundez, A. Arizmendi, and E.M. Quiroz, 2011. "Internal versus external cluster validation indexes". *International Journal of computers and Communications*, Volumen 5, Issue 1, pp. 27-34.

[RCEC13] B.P. Rocha, M. Conti, S. Etalle, and B. Crispo, 2013. "Hybrid static-runtime information flow and declassification enforcement". *IEEE Transactions on Information Forensics and Security*, Volumen 8, Issue 8, pp. 1294-1305.

[RED12] E. Raftopoulos, M. Egli, and X. Dimitropoulos, 2012. "Shedding light on log correlation in network forensics analysis". En Proc. 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), Crete, Greece, Volumen 7591, pp. 232-241.

[Rey09] D. Reynolds, 2009. "Gaussian mixture models". *Encyclopedia of Biometrics*. Springer US. pp. 659-663.

[Rez19] H. Rezanková, 2019. "Cluster analysis and categorical data". *Statistika*, Volumen 3, pp. 216-232.

[RHL15] S. Rastegari, S. Hingston, and C.P. Lam, 2015. "Evolving statistical rulesets for network intrusion detection". *Applied Soft Computing*, Volumen 33, PP. 348-359.

[RK15] O. Rottenstreich and I. Keslassy, 2015. "The Bloom paradox: when not to use a Bloom filter". *IEEE/ACM Transactions on Networking*, Volumen 23, Issue 3, pp. 703-716.

[RL06] K. Rieck and P. Laskov, 2006. "Detecting unknown network attacks using language models". En Proc. 3rd International Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), Berlin, Germany, Volumen 4064, pp. 74-90.

[RRS00] S. Ramaswamy, R. Rastogi, and K. Shim, 2000. "Efficient algorithms for mining outliers from large data sets". En Proc. 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00), Volumen 29, Issue 2, pp. 427-438, Dallas, TX, USA.

[RT15] R. Robinson and C. Thomas, 2015. "Ranking of machine learning algorithms based on the performance in classifying DDoS attacks". En Proc. IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, India, pp. 185-190.

[SA14] N.A. Seresht and R. Azmi, 2014. "MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach". *Engineering Applications of Artificial Intelligence*, Volumen 35, pp. 286-298.

[SAN15] S. Sheen, R. Anitha, and V. Natarajan, 2015. "Android based malware detection using a multifeature collaborative decision fusion approach". *Neurocomputing*, Volumen 151, Part 2, pp. 905-912.

[SBPC14] S. Stalla-Bourdillon, E. Papadaki, and T. Chown, 2014. "From porn to cybersecurity passing by copyright: How mass surveillance technologies are gaining legitimacy. The case of deep packet inspection technologies". *Computer Law & Security Review*, Volumen 30, Issue 6, pp. 670-686.

[SEA00] SEA, 2000. Schonlau dataset. Available at: http://www.schonlau.net/intrusion.html.

[SEKX98] J. Sander, M. Ester, J.P. Kriegel, and X. Xu, 1998. "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications". *Data Mining and Knowledge Discovery*, Volumen 2, Issue 2, pp. 169-194.

[Sen14] S. Sen, 2014. "Using instance-weighted naive bayes for adapting concept drift in masquerade detection". *International Journal of Information Security*, Volumen 13, Issue 6, pp. 583-590.

[Sen15] S. Sen, 2015. "Sequence-based masquerade detection for different user groups". *Security and Communication Networks*, Volumen 8, Issue 7, pp. 1265-1278.

[SES+14] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, 2014. "AVOIDIT: a cyber attack taxonomy". En Proc. 9th Annual Symposium on Information Assurance (ASIA), Abany, NY, USA.

[SGK12] S.S.S. Sindhu, S. Geetha, and A. Kannan, 2012. "Decision tree based light weight intrusion detection using a wrapper approach". *Expert Systems with Applications*, Volumen 39, Issue 1, pp. 129-141.

[Sha48] C. Shannon, 1948. "A mathematical theory of communication". *Bell Systems Technical Journal*, Volumen 27, Issue 3, pp. 379-423.

[SHGH+15] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, 2015. "Intrusion alert prioritisation and attack detection using post-correlation analysis". *Computers & Security*, Volumen 50, pp. 1-15.

[SHS08] M.B. Salem, S. Hershkop, and S.J. Stolfo, 2008. "A survey of insider attack detection research". Insider Attack and Cyber Security, Springer US. Advances in Information Security, Volumen 39, pp. 69-90.

[SLKK13] S. Shin, S. Lee, H. Kim, and S. Kim, 2013. "Advanced probabilistic approach for network intrusion forecasting and detection". *Expert Systems with Applications*, Volumen 40, No. 1, pp. 315-322.

[SLMB14] O. Salem, Y. Liu, A. Mehaoua, and R. Boutaba, 2014. "Online anomaly detection in wireless body area networks for reliable healthcare monitoring". *IEEE Journal of Biomedical and Health Informatics*, Volumen 18, Issue 5, pp. 1541-1551.

[SMFDV13] S. Salah, G. Maciá-Fernández, and J.E. Díaz-Verdejo, 2013. "A model-based survey of alert correlation techniques". *Computer Networks*, Volumen 57, No. 5, pp. 1289-1317.

[Smi12] R.E. Smith, 2012. "A contemporary look at saltzer and schroeder's 1975 design principles". *IEEE Security & Privacy*, Volumen 10, No. 6, pp. 20-25.

[SMP13] Z.C. Schreuders, T. McGill, and C. Payne, 2013. "The state of the art of application restrictions and sandboxes: a survey of application-oriented access controls and their shortfalls". *Computers & Security*, Volumen 32, pp. 219-241.

[Sno18] Snort, 2018. Available at: https://www.snort.org.

[Sop15] Sophos, 2015. Security Threat Report 2015. Available at http://www.sophos.com.

[SRM11] Z.I. Shaleh, H. Refai, and A. Mashhrou, 2011. "A proposed framework for security risk assessment". *Journal of Information Security*, Volumen 2, Issue 2, pp. 85-90.

[SS11a] M.B. Salem and S.J. Stolfo, 2011. "Decoy document deployment for effective masquerade attack detection". En Proc. of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). Amsterdam, Netherlands; pp. 35-54.

[SS11b] M.B. Salem and S.J. Stolfo, 2011. "Modeling user search behavior for masquerade detection". En Proc. 14th International Symposium on Recent Advances in Intrusion Detection (RAID), Menlo Park, CA, Volumen 6961, pp. 181-200.

[SS11c] M.B. Salem and S.J. Stolfo, 2011. "Modeling user search behavior for masquerade detection". En Proc. 14th international Conference on Recent Advances in Intrusion Detection, Menlo Park, CA, US, pp. 181-200.

[SSA08] S. Sahibudin, M. Sharifi, and M. Ayat, 2008. "Combining ITIL, COBIT and ISO/IEC 27002 in order to design a comprehensive IT framework in organizations". En Proc. 2nd Asia International Conference on Modeling & Simulation (AICMS), Kuala Lumpur, Malaysia, pp. 749-753.

[SSABC16] A. Shameli-Sendi, R. Aghababaei-Barzegar, and M. Cheriet, 2016. "Taxonomy of information security risk assessment (ISRA)". *Computers & Security*, Volumen 57, pp. 14-30.

[SSB13] R.H. Syed, M. Syrame, and J. Bourgeois, 2013. "Protecting grids from cross-domain attacks using security alert sharing mechanisms". *Future Generation Computer Systems*, Volumen 29, Issue 2, pp. 536-547.

[SSS+10] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, 2010. "An overview of IP flow-based intrusion detection". *IEEE Communications Surveys & Tutorials*, Volumen 12, Issue 3, pp. 343-356.

[SSSS13] N. Sengupta, J. Sen, J. Sil, and M. Saha, 2013. "Designing of on line intrusion detection system using rough set theory and Q-learning algorithm". *Neurocomputing*, Volumen 111, pp. 161-168.

[SSTG12] A. Shiravi, H. Shiravi, M. Tavallaee, and A.A. Ghorbani, 2012. "Toward developing a systematic approach to generate benchmark datasets for intrusion detection". *Computers & Security*, Volumen 31, Issue 3, pp. 357-374.

[SSYP16] Z. Sitova, J. Sedenka, Q. Yang, and G. Peng, 2016. "HMOG: new behavioral biometric features for continuous authentication of smartphone users". *IEEE Transactions on Information Forensics and Security*, Volumen 11, Issue 5, pp. 877-892.

[ST00] M. Schonlau and M. Theus, 2000. "Detecting masquerades in intrusion detection based on unpopular commands". *Information Processing Letters*, Volumen 76, Issue 1-2, pp. 33-38.

[STD16] K. Singh, K. Thongam, and T. De, 2016. "Entropy-based application layer DDoS attack detection using articial neural networks". *Entropy* Volumen 18 , Issue 10, No. 350.

[STTPLA14] G. Suarez-Tangil, J.E. Tapiador, P. Peris-Lopez, and J.B. Alis, 2014. "Dendroid: a text mining approach to analyzing and classifying code structures in android malware families". *Expert Systems with Applications*, Volumen 41, Issue 4, pp. 1104-1117.

[STTPLR14] G. Suarez-Tangil, J.E Tapiador, P. Peris-Lopez, and A. Ribagorda, 2014. "Evolution, detection and analysis of malware for smart devices". *IEEE Communications Surveys & Tutorials*, Volumen 16, Issue 2, pp. 961-987.

[SUR18] SURFcent, 2018. SURFcent IDS. Available at http://ids.surfnet.nl.

[SV14] J. Shana and T. Venkatachalam, 2014. "An Improved method for counting frequent itemsets using Bloom filter". *Procedia Computer Science*, Volumen 47, pp. 84-91.

[SVS+14] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, 2014. "Syntactic N-grams as machine learning features for natural language processing". *Expert Systems with Applications*, Volumen 41, Issue 3, pp. 853-560.

[SW81] T.F. Smith and M.S. Waterman, 1981. "Identification of common molecular subsequences". *Journal of molecular biology*, Volumen 147, Issue 1, pp. 195-197.

[SWJR07] X. Song, M. Wu, C. Jermaine, and S. Ranka, 2007. "Conditional anomaly detection". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 19, Issue 5, pp. 631-645.

[SWZ15] E. Schubert, M. Weiler, and A. Zimek, 2015. "Outlier detection and trend detection: two sides of the same coin". En Proc. IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA pp. 40-46.

[SWZK12] E. Schubert, R. Wojdanowski, A. Zimek, and H.P. Kriegel, 2012. "On evaluation of outlier rankings and outlier scores". En Proc. 12th SIAM International Conferene on Data Mining (SDM), Anaheim, CA, USA, pp. 1047-1058.

[Sym16] Symantec, 2016. "2016 Internet Security Threat Report" (ISTR). Available at https://www.symantec.com/security-center/threat-report.

[SYR15] X. Shu, D. Yao, and N. Ramakrishnan, 2015. "Unearthing stealthy program attacks buried in extremely long execution paths". En Proc. 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, CO, USA, pp. 401-413.

[SZ00] E. Spafford and D. Zamboni, 2000. "Intrusion detection using autonomous agents". *Computer Networks*, Volumen 34, pp. 547-570.

[SZ15] T.R.L. Santos and L.E. Zárate, 2015. "Categorical data clustering: What similarity measure to recommend?". *Expert Systems with Applications*, Volumen 42, Issue 3, pp. 1247-1260.

[SZK14] E. Schubert, A. Zimek, and H.P. Kriegel, 2014. "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection". *Data Mining and Knowledge Discovery*, Volumen 28, Issue 1, pp. 190-237.

[SZY+14] D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, 2014. "Anomaly detection in online social networks". *Social Networks*, Volumen 39, pp. 62-70.

[TA05] G. Tedesco and U. Aickelin, 2005. "Strategic alert throttling for intrusion". En Proc. 4th International Conference on Information Security (WSEAS), Tenerife, Canary Islands, Spain, pp. 246-251.

[TC11] J.E. Tapiador and J.A. Clark, 2011. "Masquerade mimicry attack detection: a randomised approach". *Computers & Security*, Volumen 30, Issue 5, pp. 297-310.

[TFF+16] P. Teufl, M. Ferk, A. Fitzek, D. Hein, S. Kraxberger, and C. Orthacker, 2016. "Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play)". *Security and Communication Networks*, Volumen 9. Issue 5, pp. 389-419.

[TFPC10] G.C. Tjhai, S.M. Furnell, M. Papadaki, and N.L. Clarke, 2010. "A preliminary two-stage alarm correlation and ltering system using SOM neural network and K means algorithm". *Computers & Security*, Volumen 29, Issue 6, pp. 712-723.

[TKBK09] S.A. Thorat, A.K. Khandelwal, B. Bruhadeshwar, and K. Kishore, 2009. "Anomalous packet detection using partitioned payload". *Journal of Information Assurance and Security*, Volumen 3, Issue 3, 195-020.

[TKCH04] J. Timmis, T. Knight, L.N. Castro, and E. Hart, 2004. "An overview of artificial immune systems". *Computation in Cells and Tissues*, Springer Berlin Heidelberg, pp. 51-91.

[TLHC14] Y. Tang, X. Luo, Q. Hui, and R.K.C. Chang, 2014. "Modeling the vulnerability of feedback-control based Internet services to low-Rate DoS attacks". *IEEE Transactions on Information Forensics and Security*, Volumen 9, No. 3, pp. 339-353.

[TLL95] I.V. Tetko, D.J. Livingstone, and A.I. Luik, 1995. "Neural network studies. 1. Comparison of overfitting and overtraining". *Journal of Chemical Information and Modeling*, Volumen 35, No. 5, pp. 826-833.

[TRM09] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, 2009. "Intrusion detection using fuzzy association rules". *Applied Soft Computing*, Volumen 9, No. 2, pp. 462-469.

[TY06] J. Takeuchi and K. Yamanishi, 2006. "A unifying framework for detecting outliers and change points from time series". *IEEE Transactions on Knowledge and Data Engineering*, Volumen 18, Issue 4, pp. 482-492.

[UC16] US-CERT, 2016. Home Network Security. Avaibale at https://www.us-cert.gov/Home-Network-Security.

[VJBS15] J. Voris, J. Jermyn, N. Boggs, and S.J. Stolfo, 2015. "Fox in the trap: thwarting masqueraders via automated decoy document deployment". En Proc. 8th European Workshop on System Security (EuroSec), Bordeaux, France, No. 3.

[VKMF15] E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, and M. Fischer, 2015. "Taxonomy and survey of collaborative intrusion detection". *ACM Computing Surveys*, Volumen 47, Issue 4, No. 55.

[VRM+11] E. Vivier, D.H. Raulet, A. Moretta, M.A. Caligiuri, L. Zitvogel, L.L. Lanier, W.M. Yokoyama, and S. Ugolini, 2011. "Innate or adaptive immunity? the example of natural killer cells". *Science*, Volumen331, Issue 6013, pp. 44-49.

[VTN13] A. Viswanathan, K. Tan, and C. Neuman, 2013. "Deconstructing the assessment of anomaly-based intrusion detectors". En Proc. 16th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID), Rodney Bay, St. Lucia, pp. 286-306.

[War70] W.H. Ware, 1970. "Security controls for computer systems (U): report of defense science board task force on computer security". The RAND Corporation, Santa Monica, CA, USA.

[WB10] S.X. Wu and W. Banzhaf, 2010. "The use of computational intelligence in intrusion detection systems: A review". *Applied Soft Computing*, Volumen 10, Issue 1, pp. 1-35.

[WCCQ07] K. Wang, M. Cai, Y. Chen, and M. Qin, 2007. "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes". *IEEE Transactions on Dependable and Secure Computing*, Volumen 4, Issue 1, pp. 41-55.

[WCS05] K. Wang, G. Cretu, and S.J. Stolfo, 2005. "Anomalous payload-based worm detection and signature generation". En Proc. 8th International Symposium on Recent Advances in Intrusion Detection (RAID), Seattle, WA, USA, Volumen 3858, pp. 227-246.

[WCXJ13] W. Wei, F. Chen, Y. Xia, and G. Jin, 2013. "A rank correlation based detection against distributed reflection DoS attacks". *IEEE Communications Letters*, Volumen 17, No. 1, pp. 173-175.

[WDD99] A. Wespi, M. Dacier, and H. Debar, 1999. "An intrusion-detection system based on the Teiresias pattern-discovery algorithm". En Proc. Annual Conference of the European Institute for Computer Anti-Virus Research (EICAR), Aalborg, Denmark, pp. 1-15.

[WF74] R.A Wagner and M.J. Fisher, 1974. "The string-to-string correction problem". *Journal of the ACM*, Volumen 21, Issue 1, pp. 168-173.

[WFBS14] D.J. Weller-Fahy, B.J. Borghetti, and A.A. Sodemann, 2014. "A survey of distance and similarity measures used within network intrusion anomaly detection". *IEEE Communications Surveys & Tutorials*, Volumen 17, Issue 1, pp. 70-91.

[WG03] J.A. White and S.M. Garrett, 2003. "Improved pattern recognition with artificial clonal selection?". En Proc. 2nd International Conference on Artificial Immune Systems, Edinburgh, UK, pp. 181-193.

[WGNF12] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, 2012. "ProfileDroid: multi-layer profiling of android applications". En Proc. 18th Annual International Conference on Mobile Computing and Networking (Mobicom), Istambul, Turkey, pp. 137-148.

[Wil46] F. Wilcoxon, 1946. "Individual comparisons by ranking methods". *Biometrics Bulletin*, pp. Volumen 1, No. 6, 80-83.

[WLJW16] L. Wang, Q. Li, Y. Jiang, and J. Wu, 2016. "Towards mitigating link flooding attack via incremental SDN deployment". En Proc. IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, pp. 27-30.

[WPS06] K. Wang, J.J. Parekh, and S.J. Stolfo, 2006. "Anagram: a content anomaly detector resistant to mimicry attack". En Proc. 9th International Symposium on Recent Advances in Intrusion Detection, Hamburg, Germany, Volumen 4219, pp. 226-248.

[WS03] K. Wang and S. Stolfo, 2003. "One-class training for masquerade detection". En Proc. 3rd IEEE Workshop on Data Mining for Computer Security. Melbourne, FL, USA, pp. 188-201.

[WS04] K. Wang and S.J. Stolfo, 2004. "Anomalous payload-based network intrusion detection". En Proc. 7th International Symposium on Recent Advances in Intrusion Detection (RAID), Sophia Antipolis, France, Volumen 3224, pp. 203-222.

[XPW+14] L. Xing, X. Pan, R. Wang, K. Yuan, and X. Wang, 2014. "Upgrading your Android, elevating my malware: privilege escalation through mobile OS updating". En Proc. 35th IEEE Symposium on Security and Privacy, San Jose, CA, US, pp. 393-408.

[Xu06] X. Xu, 2006. "Adaptive intrusion detection based on machine learning: feature extraction, classifier construction and sequential pattern prediction". *International Journal of Web Services Research*, Volumen 2, No. 1-2, pp. 49-58.

[XW05] R. Xu and D. Wunsch, 2005. "Survey of clustering algorithms". *IEEE Transactions on Neural Networks* , Volumen 16, Issue 3, pp. 645-678.

[XZZ+14] H. Xiong, G. Zeng, Y. Zeng, W. Wang, and C. Wu, 2014. "A novel scalability metric about iso-area of performance for parallel computing". *The Journal of Supercomputing*, Volumen 68, Issue 2, pp. 652-671.

[YAC+07] X. Yue, A. Abraham, Z.X. Chi, Y.Y. Hao, and H. Mo, 2007. "Artificial immune system inspired behavior-based anti-spam filter". *Soft Computing*, Volumen 11, Issue 8, pp. 729-740.

[YAF10] H. Yonaba, F. Anctil, and V. Fortin, 2010. "Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting", Journal of Hydrologic Engineering, Volumen 15 (4), pp. 275-283.

[YBV15] G. Yao, J. Bi, and A.V. Vasilakos, 2015. "Passive IP traceback: disclosing the locations of IP spoofers from path backscatter". *IEEE Transactions on Information Forensics and Security*, Volumen 10, Issue 3, pp. 471-484.

[YHK+15] M. Yampolskiy, P. Horváth, X.D. Koutsoukos, Y. Xue, and J. Sztipanovits, 2015. "A language for describing attacks on cyber-physical systems". *International Journal of Critical Infrastructure Protection*, Volumen 8, pp. 40-52.

[YJJ16] Q. Yu, L. Jibin, and L. Jiang, 2016. "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks". *International Journal of Distributed Sensor Networks*, Volumen 12, No. 1, ID 9653230.

[YLC+02] N. Ye, X. Li, Q. Chen, S.M. Emran, and M. Xu, 2002. "Probabilistic techniques for intrusion detection based on computer audit data". *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Volumen 31, Issue 4, pp. 266-274.

[YSP13] Y. Ye, S. Squartini, and F. Piazza, 2013. "Online sequential extreme learning machine in nonstationary environments". *Neurocomputing*, Volumen 116, pp. 94-101.

[Zad65] L.A. Zadeh, 1965. "Fuzzy Sets". *Information and Control*, Volumen 8, Issue 3, pp. 338-353.

[ZCS13] A. Zimek, R.J.G.B. Campello, and J. Sander, 2013. "Ensembles for unsupervised outlier detection: challenges and research questions". *ACM SIGKDD Explorations Newsletter*, Volumen 15, Issue 1, pp. 11-22.

[ZGTJ16] H. Zheng, X. Geng, D. Tao, and Z. Jin, 2016. "A multi-task model for simultaneous face identification and facial expression recognition". *Neurocomputing*, Volumen 171, pp. 515-523.

[ZHJ09] K. Zhang, M. Hutter, and H. Jin, 2009. "A new local distance-based outlier detection approach for scattered real-world data". *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, Volumen 5476, pp. 813-822.

[ZHT06] H. Zou, T. Hastie, and R. Tibshirani, 2006. "Sparse principal component analysis". *Journal of Computational and Graphical Statistics*, 15, pp.265-286.

[Zim14] A. Zimmermann, 2014. "The data problem in data mining". *ACM SIGKDD Explorations Newsletter*, Volumen 16. Issue 2, pp. 38-45.

[ZJT13] S.T. Zargar, J. Joshi, and D. Tipper, 2013. "A survey of defense mechanisms against distributed Denial of Service (DDoS) flooding attacks". *IEEE Communications Surveys & Tutorials*, Volumen 15, No. 4, pp. 2046-2069.

[ZJW+14] W. Zhou, W. Jia, S. Wen, Y. Xiang, and W. Zhou, 2014. "Detection and defense of application-layer DDoS attacks in backbone web traffic". *Future Generation Computer Systems*, Volumen 38, pp. 36-46.

[ZM97] M. Zait and H. Messatfa, 1997. "A comparative study of clustering methods". *Future Generation Computer Systems*, Volumen 13, Issue 2-3, pp. 149-159.

[ZSL13] M. Zheng, M. Sun, and J.C.S. Lui, 2013. "Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware". En Proc. 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Melbourne, VIC, Australia, pp. 163-171.

[ZSY10] L. Zhaowen, L. Shan, and M. Yan, 2010. "Real-Time intrusion alert correlation system based on prerequisites and consequence". En Proc. 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, Chine, pp. 1-5.

[ZV15] A. Zimek and J. Vreeken, 2015. "The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives". *Machine Learning*, Volumen 98, Issue 1, pp. 121-155.

[ZYB+14] B.B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, 2014. "Captcha as graphical passwords- a new security primitive based on hard AI problems". *IEEE Transactions on Information Forensics and Security*, Vol .9, No. 6, pp. 891-904.

[ZYYG14] Y. Zhang, M. Yang, Z. Yang, and G. Gu, 2014. "Permission use analysis for vetting undesirable behaviors in Android apps". *IEEE Transactions on Information Forensics and Security*, Volumen 9, Issue 11, pp. 1828-1842.

[ZYZ08] T. Zang, X. Yun, and Y. Zhang, 2008. "A Survey of alert fusion techniques for security incident". En Proc. 9th IEEE International Conference on Web-Age Information Management (WAIM), Zhangjiajie, China, pp. 475-481.

# Parte III

# Anexos

# Anexo A

# Métodos de detección de anomalías

Este anexo revisa las principales metodologías de detección de anomalías. Con el fin de facilitar la comprensión del resto del documento, se hará hincapié en las técnicas que serán mencionadas en capítulos posteriores. Se ha organizado en las siguientes secciones: en la Sección A.1 se discuten algunas de las taxonomías más relevantes de la bibliografía; en la Sección A.2 se describen las estrategias de detección basadas en modelado; en la sección A.3 se repasan las estrategias de detección basadas en proximidad; en la Sección A.4 se revisan los métodos basados en agrupamiento; finalmente, en la Sección A.5 se profundiza en las estrategias que asumen hipótesis estadísticas.

## A.1 Clasificación y taxonomías

A lo largo del último siglo, la gran demanda de estrategias para el reconocimiento de anomalías ha dado lugar a una inmensa cantidad de publicaciones, las cuales se han orientado tanto a su aplicación como a la propuesta de métodos de propósito general para su análisis. En consecuencia, sintetizar todos estos enfoques, así como proponer una clasificación que facilite su comprensión y la toma de decisiones, se ha convertido en el objeto de estudio de muchas investigaciones. De entre ellas, la aproximación publicada por V. Chandola [CBK09] es una de las que mejor resume estos avances. En ella se propone una ontología de los métodos de detección de anomalías que tiene por eje de clasificación premisas que involucran relaciones entre los conjuntos de datos normales y anómalos. Esto ha dado lugar a seis grandes grupos de estrategias: aquellas a las que denominaron *clasificación*, *proximidad con los vecinos*, *agrupamiento*, *estadística*, *teoría de la información* y *análisis espectral*. Una versión simplificada de esta clasificación fue propuesta por J. Han et al. [HKP11], donde atendiendo a los mismos criterios se establecen tres grupos: métodos basados en *análisis estadístico*, *proximidad* y *agrupamiento*. De entre las muchas alternativas a la división de V. Chandola cabe destacar la de C.C. Aggarwal et al. [Agg13]. En ella se distinguen diferentes modelos para la detección de anomalías considerando diversas características, como el tipo de datos a procesar, tamaño

de la información, disponibilidad o interpretabilidad de los resultados. Es importante considerar que C.C. Aggarwal et al. ensalzaron la repercusión de la última de ellas a la hora de definir su clasificación, debido principalmente a que la interpretabilidad mide la facilidad con que los analistas determinan las causas de las discordancias, y por lo tanto la capacidad de inferir conocimiento de los algoritmos. En base a estos criterios definieron seis clasificaciones: modelos basados en el *análisis de valores extremos*, *probabilidad y estadística*, *análisis lineal*, *proximidad*, *teoría de la información* y *datos de grandes dimensiones*. Otra taxonomía aún más reciente ha sido propuesta por T. Pevný [Pev16], en la que se introduce una división en base a nuevos criterios, como la orientación de los métodos: *datos* y *modelos*, o el tiempo de procesamiento: *online* y *offline*. En ella también se tiene en cuenta por primera vez la presencia de estrategias que combinan de manera cooperativa diferentes esquemas de detección de anomalías. Éstos analizan distintos tipos de datos y en ocasiones proveen información muy dispar, lo que lleva a la adquisición de un conocimiento mucho más completo del entorno de monitorización [ZCS13, ZV15]. Con el fin de facilitar la comprensión de cada una de estas técnicas, y teniendo en cuenta todas dichas consideraciones, a continuación se introduce una clasificación de estrategias de detección de anomalías que distingue cuatro tipos de aproximaciones: aquellas que principalmente se basan en *modelado*, *proximidad*, *agrupamiento* y *estadística*. Con este fin, y al igual que en [CBK09, CH13b], se considera como eje de clasificación la relación existente entre las observaciones normales y las anómalas. Nótese que la descripción de cada categoría es acompañada de diferentes ejemplos, los cuales han sido seleccionados por su relevancia en la bibliografía relacionada con la seguridad de la información, o bien por asumir un papel esencial en los diferentes capítulos de esta tesis.

## A.2 DETECCIÓN BASADA EN MODELADO

Los métodos de detección de anomalías agrupados en esta colección asumen como principal prerrequisito el hecho de que es posible distinguir observaciones normales de las discordancias a partir de una etapa de aprendizaje previo sobre el espacio muestral. Por lo tanto, estas estrategias tienen en común su dependencia de algoritmos de adquisición de conocimiento inducido, tales como los esquemas supervisados, semi-supervisados o no supervisados previamente descritos en la Sección 3.3 “Adquisición de conocimiento”. A continuación, se describen algunos ejemplos de métodos basados en modelado, en concreto las redes neuronales artificiales, redes bayesianas, modelo oculto de Markov, maquinas vector soporte, sistemas expertos basados en reglas, árboles de decisión algoritmos genéticos y sistemas inmunitarios artificiales. En [GKRB13, BBK14] se resumen muchas otras aproximaciones similares.

### A.2.1 REDES NEURONALES ARTIFICIALES

Inspiradas en los sistemas nerviosos biológicos, las redes neuronales artificiales son esquemas colaborativos que interconectan diferentes nodos con el fin de producir un estímulo deseado de salida en función de la información de entrada. Los nodos son

denominados neuronas y están distribuidos en capas, contando siempre con una capa de entrada, una o varias capas intermedias ocultas, y una capa de salida. Sus conexiones están ponderadas, y estos pesos variarán en función de la información de referencia adquirida, ajustándose de esta manera la estimulación de las salidas deseadas. Por lo tanto, esta etapa de ajuste es su etapa de entrenamiento, en la que se define el modelo de información que es capaz de reconocer. En [WB10] se describe en detalle esta tecnología, haciendo hincapié en su despliegue para la detección de intrusiones. La Figura A.1 muestra un esquema típico de neurona artificial. En ella se observan n estradas $X_1, X_2, \cdots, X_n$, y los pesos $W_i$ asignados a cada una de ellas en la posición i-esima. La salida de la neuronal artificial viene dada por la expresión:

$$y_k = \varphi(\sum_{i=0}^{n} W_{kj} X_j) \tag{A.1}$$

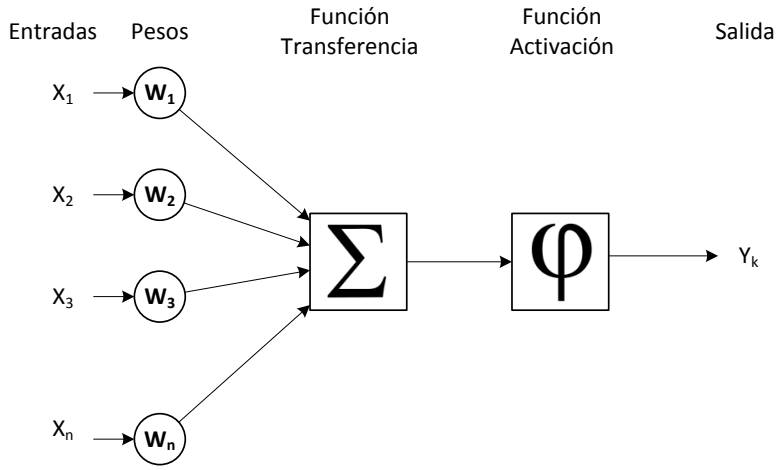



Figura A.1: Ejemplo de neurona artificial.

donde el sumatorio es la función de propagación y $\varphi$ es la función de activación.
Según S. Sindhu et al. [SGK12] existen dos maneras de implementar las redes neuronales artificiales para el reconocimiento de anomalías en la seguridad de la información. La primera de ellas consiste en su incorporación a otros esquemas de detección con el fin de filtrar observaciones, dejando pasar únicamente las que no levanten sospechas. De este modo se mejora el comportamiento del sistema original y se reduce su carga de trabajo. Por otro lado, la red neuronal artificial puede actuar en solitario como un único sistema de detección, reportando al operador las observaciones realizadas que no correspondan con el modelo de información construido durante su entrenamiento. Tal y como indicó J. Cannady [Can98], las principales ventajas de la aplicación de redes neuronales artificiales en la detección de anomalías son su eficiencia, capacidad de analizar datos incompletos o corruptos, e identificación de la probabilidad de éxito de los etiquetados. Sin embargo, tienen una gran dependencia de la calidad de su entrenamiento, para la cual a menudo se requieren muchas muestras debidamente etiquetadas. También son propensas al problema

del sobreentrenamiento [TLL95]. Además, actúan como un modelo de caja negra, lo que dificulta la interpretación de los resultados y la depuración de errores.

### A.2.2 REDES BAYESIANAS

Las redes Bayesianas son la representación gráfica de conjuntos de variables aleatorias y sus relaciones por medio de grafos dirigidos acíclicos, en los que los vértices son los datos observados y las aristas sus probabilidades condicionales. Tal y como se indica en [BG15], estas estructuras pueden ser construidas por expertos o por algoritmos basados en inferencia, lo que constituyen las etapas de modelado del sistema. En la Figura A.2 se muestra un ejemplo de red Bayesiana, en la que los vértices representan variables en el ámbito de la gestión de incidencias en redes (ej. identificación de retraso en comunicaciones, exceso de sesiones activas, problemas de balanceo de carga, etc.) y las aristas la probabilidad de que una incidencia implique una dependencia con otra (ej. la probabilidad de que se esté produciendo un ataque de denegación de servicio, si se ha detectado congestión en la red es de 0.15).

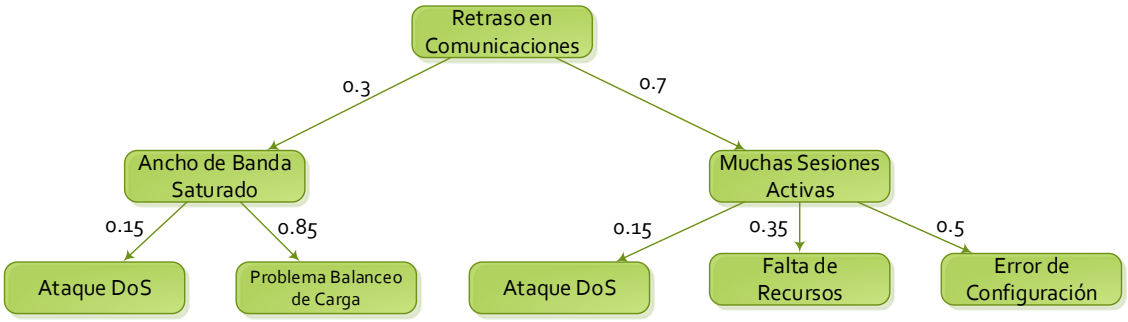



Figura A.2: Ejemplo de red Bayesiana para evaluar incidencias en redes.

Aunque no existe un método general para reconocer anomalías en base a estas estructuras, lo más frecuente es considerar algún tipo de umbral $T_h$, de forma que la observación es considerada anómala al satisfacerse la siguiente inecuación [MNK14]:

$$P\left(e|m\right) < T_h \tag{A.2}$$

Nótese que, en la expresión $P$ es la propiedad condicional que relaciona dos vértices del grafo, $e$ es una observación o evento registrado y $m$ es el modelo. Esta expresión puede ser ampliada a secuencias de eventos, donde para considerar una observación como discordante, se debe satisfacer la siguiente inecuación en $N$ observaciones en el tiempo:

$$\frac{1}{N}\sum_{i} P\left(e_i|m\right) < T_h \tag{A.3}$$

Existen muchas alternativas a estos esquemas. Otro ejemplo fue introducido en [JCNJ90], donde la existencia de anomalías se relaciona con la identificación de conflictos dentro de conjuntos de observaciones. Según S. Mascaro et al. [MNK14], el uso de redes Bayesianas aporta tres principales ventajas respecto a otros métodos. En primer lugar,

los operadores no expertos en el dominio en que son implementadas pueden trabajar con ellas con facilidad. Por otro lado, y dada su estructura en forma de grafo, las redes Bayesianas facilitar la inserción, eliminación y modificación del conocimiento. Finalmente, y dado que representan relaciones de causalidad entre eventos, pueden ser fácilmente verificadas y validadas. En su contra, cabe destacar que sus etapas de aprendizaje suelen ser especialmente costosas computacionalmente, debido principalmente a que requieren operar con grandes matrices de datos.

### A.2.3 Modelo oculto de Markov

Un proceso de Markov es un fenómeno aleatorio dependiente del tiempo que satisface la propiedad de Markov, es decir, que sus probabilidades condicionales sobre el estado presente, futuro y pasado del sistema sean independientes [BP66]. Los modelos de Markov más simples son las cadenas de Markov, las cuales representan conjunto de estados interconectados por transiciones probabilistas, siendo éstas las que definen la topología del modelo. Como caso particular de este tipo de estructuras, cabe destacar el Modelo Oculto de Markov o HMM (del inglés *Hidden Markov Model*), en el que parte de los estados son desconocidos, siendo su desenmascaramiento una de las principales tareas en esté área de investigación. En [HS11] se ilustra un ejemplo claro de HMM, con el que se pretende hacer seguimiento del uso que da un usuario de un sistema de la información. Éste asume que los usuarios pueden conectarse de manera remota al equipo, y que por lo tanto no es posible conocer con certeza quién está accediendo. Sin embargo es posible monitorizar las acciones que está realizando, facilitando la definición de un modelo como el que se muestra en la Figura A.3.

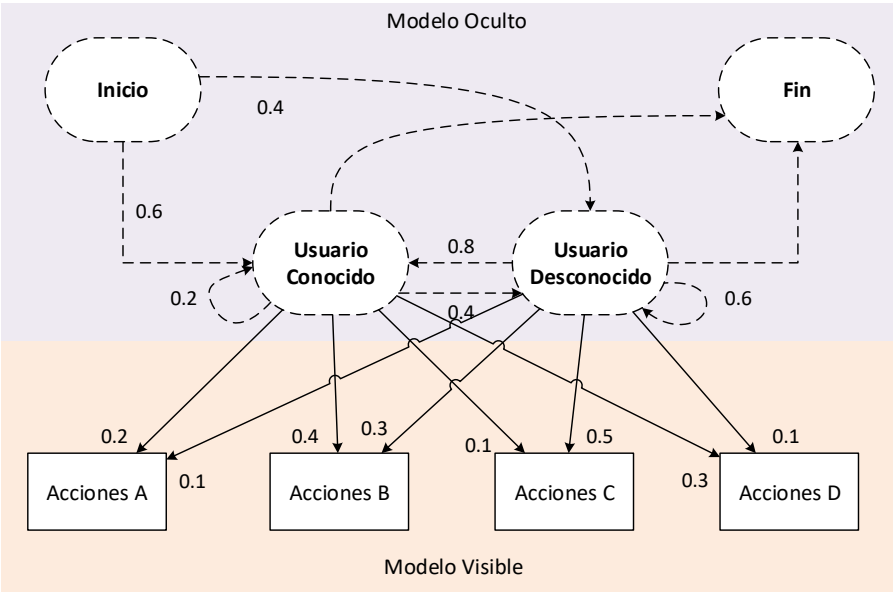



Figura A.3: Ejemplo de HMM para reconocimiento de usuarios.

Donde la matriz $A$ representa las transiciones entre el usuario conocido y desconocidos, y la matriz $B$ las relaciones entre los grupos de comandos y cada uno de los estados ocultos:

$$A = \begin{pmatrix} 0.2 & 0.8 \\ 0.4 & 0.6 \end{pmatrix} \tag{A.4}$$

$$B = \begin{pmatrix} 0.2 & 0.4 & 0.1 & 0.3 \\ 0.1 & 0.3 & 0.5 & 0.1 \end{pmatrix} \tag{A.5}$$

$$\pi = \begin{pmatrix} 0.6 & 0.4 \end{pmatrix} \tag{A.6}$$

Los estados iniciales vienen dados por $\pi$. A partir de estas matrices se define el HMM del ejemplo como $\lambda = (A, B, \pi)$, lo que según C. Annachhatre et al. [AAS15] permite la resolución de tres problemas: (1) el cálculo de la probabilidad $P(O|\lambda)$ de una observación $O$ sobre el modelo, (2) la predicción de las transiciones más probables desde un estado oculto concreto y (3) el entrenamiento del sistema para hallar el $\lambda = (A, B, \pi)$ que maximice la probabilidad de observar $O$. Aunque estos problemas pueden adaptarse de diferentes maneras en su aplicación al reconocimiento de anomalías, el procedimiento más frecuente consiste en basarse en (3) para la elaboración de un modelo del entorno de monitorización, y (1) para valorar la presencia de nuevas observaciones respecto a dicho modelo, identificándose discordancias al superarse cierto umbral $T_h$, tal que $P(O|\lambda) < T_h$. Éste es el caso de [HS11], en el que esta estrategia se implementa en el desenmascaramiento de usuarios no autorizados. Otra aplicación frecuente es la predicción. Esto se lleva a cabo por medio de la construcción de un modelo según (3) y la predicción de una secuencia de transiciones (2), de este modo siendo posible determinar cambios de estado no esperados, y por lo tanto discordantes. Un ejemplo de esto se muestra en [BMR15], donde HMM se aplican a la predicción de errores críticos de seguridad.

La aportación de los HMM a la detección de anomalías varía mucho en función de los tipos de datos a analizar, jugando un papel especialmente relevante en el estudio de secuencias de observaciones en el tiempo. En términos generales, HMM permiten el modelado de procesos estocásticos de gran complejidad y facilitan la gestión de variaciones en el escenario de monitorización. Sin embargo, y al igual que sucede con las redes Bayesianas, habitualmente requieren de la construcción y procesamiento de grandes matrices de datos, lo que implica un importante consumo de recursos de cómputo durante su entrenamiento. Por otro lado, además de heredar las dificultades inherentes a asumir las bases de los procesos Markovianos, los HMM tienden a necesitan colecciones de muestras de referencia de gran extensión, siendo esta otra importante desventaja al tratar casos de uso en los que no se dispone de tanta información.

### A.2.4 MÁQUINAS DE VECTOR SOPORTE

Las máquinas de vector soporte son una colección de algoritmos de aprendizaje automático supervisados basados en la transformación del espacio de entrada en otro de dimensión superior e infinita, donde el problema a tratar es resuelto a partir del cálculo del hiperplano óptimo de separación muestral al que se llama vector soporte [MJS02]. Por lo tanto, las

máquinas de vector soporte son clasificadores que dado un conjunto de datos de referencia (plano) representados como un vector ordenado $p$-dimensional, buscan el hiperplano que los separe en clases de manera óptima, definiendo de esta manera los conjuntos de pertenencia. Por ejemplo, al tratar datos de 2-dimensiones, son representados en función de $x$ e $y$, tal y como muestra en la Figura A.4.

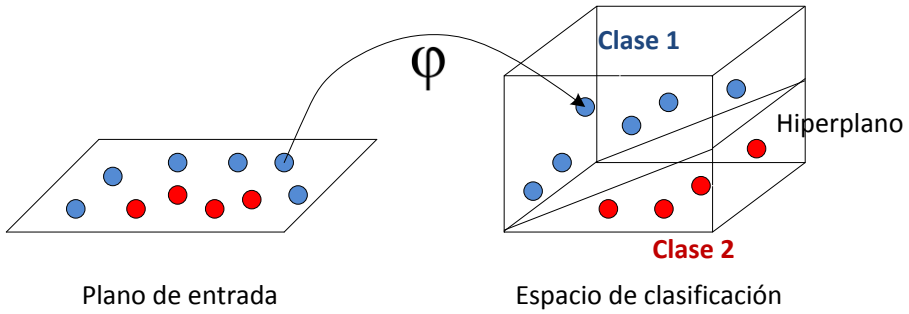



Figura A.4: Ejemplo de transformación del espacio de entrada en función de $\varphi$.

El hiperplano ideal para delimitar las clases es un vector de 1-dimension que maximice la distancia entre los miembros de cada clase al que se denomina *hiperplano de margen máximo*. Sin embargo, no siempre es posible de hallar, debido principalmente a la compensación de los errores registrados durante el entrenamiento y a la limitación de recursos de cómputo. Esto último lleva a la implementación de funciones *kernel* predefinidas, la cuales proyectan la información a un espacio de mayor dimensionalidad sobre el que es posible operar de manera más eficiente. Algunas de ellas son la función *polinomial homogénea* $k(x_i, x_j) = (x_i \times x_j)^d$, *polinomial no homogénea* $k(x_i, x_j) = (x_i \times x_j + 1)^d$, *perceptrón* $k(x_i, x_j) = \| x_i - x_j \|$, *base radial gaussiana* $k(x_i, x_j) = exp(-\gamma) \| x_i - x_j \|^2$ para $\gamma > 0$ o *hiperbólica tangente* $k(x_i, x_j) = \tanh(kx_i \times x_j + c)$ para $k, c > 0$, pudiéndose encontrar en [DBK+97] su descripción y algunas más. Además de resolver problemas de clasificación, a partir de la aproximación publicada por H. Drucker et al., las máquinas de vector soporte también son aplicadas en problemas de regresión lineal.

Tal y como se muestra en [CBK09], el despliegue de máquinas de vector soporte en la detección de anomalías a menudo se centra en la construcción de un modelo normal por medio de la representación en el plano de un conjunto de clases construidas desde observaciones consideradas normales. Cuando las muestras a analizar no pertenecen a alguna de estas clases, son etiquetadas como anómalas. En [BG15] se recopila una gran cantidad de propuestas basadas en este método. Un ejemplo muy claro de su aplicación en la seguridad de la información se ilustra en [SS11c], donde se trata el problema de la identificación de usuarios no autorizados. En esta aproximación la máquina de vector soporte es entrenada con datos pertenecientes a las actividades habituales perpetradas por los usuarios legítimos en el sistema a proteger, emitiéndose alertas cuando las observaciones a analizar disten considerablemente de ellas. Tal y como se indica en [MJS02], las dos principales ventajas de las máquinas de vector soporte respecto a estrategias similares son su velocidad y escalabilidad. A éstas hay que añadirles su gran capacidad de generalización gracias a la adopción del paradigma de la minimización del riesgo estructurado o SRM (del inglés *Structural Risk Minimization*), y su facilidad de configuración dados sus pocos

parámetros de ajuste. Sin embargo, también presentan desventajas, como las restricciones inherentes al uso de funciones *kernel*, o su complejidad computacional, la cual lleva a un elevado consumo de memoria.

### A.2.5 SISTEMAS EXPERTOS BASADOS EN REGLAS

El principal objetivo de los sistemas expertos basados en reglas es generar conocimiento a partir de datos previamente conocidos. Sus elementos básicos son los *objetos*, *hechos* y *reglas*. Los *objetos* son las unidades básicas de información, y su función es especificar los datos sobre los que el sistema es capaz de operar (ej. temperatura, nivel de congestión, número de sesiones iniciadas, etc...); los *hechos* son el conocimiento inferido o establecido en la etapa de modelado del sistema (ej. *temperatura*=35, *nivel de congestión*=72.5%, *número de sesiones iniciadas*=21); por último, las reglas son las directivas de razonamiento que permiten que el sistema alcance conclusiones. Por ejemplo, supongamos que un sistema experto opera sobre dos tipos de objetos (*temperatura*, *nivel de riesgo*) y asume siguientes reglas:

$$Regla1 : temperatura > 50 \longrightarrow (nivelderiesgo = alto); \tag{A.7}$$

$$Regla2 : temperatura \leq 50 \longrightarrow (nivelderiesgo = bajo). \tag{A.8}$$

Si el sistema registra el hecho de que *temperatura*=68 se activará la Regla 1, generándose el conocimiento *nivel de riesgo*=alto. En la Figura A.5 se muestra la arquitectura básica de los sistemas expertos basados en reglas, en la que se distinguen tres componentes principales: *base de conocimiento*, *motor de inferencia* y *memoria de trabajo*. En la *base del conocimiento* se gestionan las reglas y objetos; la *memoria detrabajo* almacena de manera temporal los hechos con los que el sistema debe trabajar; el *motor de inferencia* es la parte más compleja del sistema, ya que en él se aplican las reglas y se infiere el nuevo conocimiento.

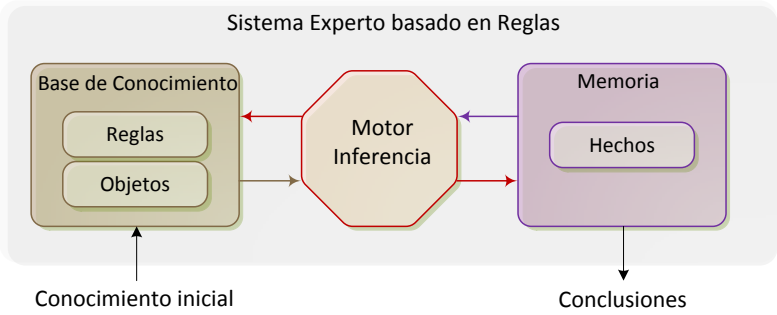



Figura A.5: Arquitectura clásica de sistema experto basado en reglas.

Tal y como indican V. Chandola et al. [CBK09], la aplicación de estos sistemas a la detección de anomalías pasa por una etapa de generación de reglas a partir de la cual se construye un modelo del entorno de monitorización. Su implementación más habitual parte de la premisa de que al registrarse un evento, si no es cubierto por ninguna de las

reglas aplicadas a los hechos conocidos, es de naturaleza anómala. En [MC15] se muestra un ejemplo de la aplicación de esta idea al modelado del comportamiento legítimo en infraestructura crítica sanitaria. Como alternativa a este esquema, es frecuente elaborar reglas de tal manera que cuando la variable asignada a un hecho supere cierto umbral, sea considerada anómala. Esto se aplica en [RHL15] para la identificación de comportamientos discordantes en red. En [BG15] se recopila una gran variedad de algoritmos basados en este tipo de sistemas expertos, así como de métodos de generación de reglas. En términos generales, las principales ventajas de los sistemas expertos basados en reglas para la detección de anomalías son su escalabilidad, capacidad de tener en cuenta gran cantidad de información y tolerancia a errores en los datos adquiridos. Sin embargo, cuando las colecciones de reglas y hechos son demasiado largas, su tratamiento puede llegar a ser un proceso muy costoso computacionalmente. Otra importante desventaja es la dificultad de generar conjuntos de reglas suficientemente representativos.

### A.2.6 Árboles de decisión

Los árboles de decisión son diagramas de construcciones lógicas que modelan los sucesos que pueden derivar de una observación realizada. Éstos son construidos de tal manera que cada nodo representa el asumir una premisa acerca de un atributo, cada rama su valoración, y cada hoja una clasificación, también consideradas decisiones. En [Qui86] se discute en detalle la teoría que está detrás de este paradigma. La Figura A.6 muestra un sencillo ejemplo de construcción de un árbol de decisión a partir de los datos recopilados en la Tabla A.1. En ella se indica la tolerancia a tres parámetros del agua (*potencial Hidrógeno pH*, *Magnesio Mg* y *Alcalinidad Kh*) de diferentes seres vivos dentro de un acuario. Como puede observarse, el árbol de decisión permite clasificarlos en siete familias $(A, B, C, D, E, FyG)$.

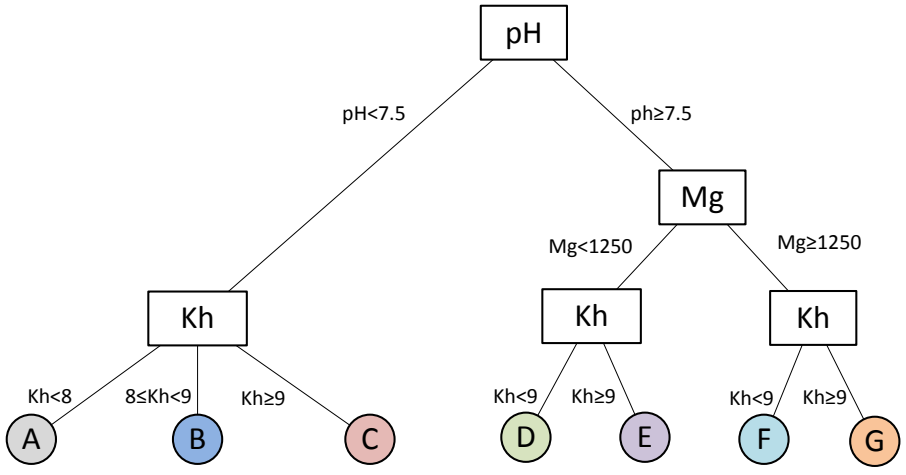



Figura A.6: Ejemplo de árbol de decisión.

Cuando estos modelos tienen por variables destino, conjuntos finitos de etiquetas, son denominados árboles de clasificación; éste es el caso del ejemplo anterior. Pero si sus hojas representan datos continuos, son denominados árboles de predicción. A lo largo de

Tabla A.1: Datos de ejemplo sobre tolerancia a parámetros en distintos especímenes.

| Individuo | pH | Mg | Kh | Familia |
|---|---|---|---|---|
| 1 | 7.2 | 1310 | 6.9 | A |
| 2 | 7.1 | 1230 | 9.2 | C |
| 3 | 7.6 | 1260 | 8.2 | F |
| 4 | 7.6 | 1215 | 9.1 | E |
| 5 | 7.4 | 1260 | 8.2 | B |
| 6 | 7.3 | 1245 | 7.7 | A |
| 7 | 7.5 | 1225 | 8.3 | D |
| 8 | 7.6 | 1220 | 9.3 | E |
| 9 | 7.1 | 1300 | 8.5 | B |
| 10 | 7.7 | 1270 | 9.1 | G |
| 11 | 7.5 | 1256 | 9.4 | G |
| 12 | 7.4 | 1300 | 9.3 | C |

los años se han planeado diferentes algoritmos para su construcción, como ID3 [Qui83], C4.5 [Qui93] o CART [BOS84]. En [PLGMI16] se revisan algunas de estas técnicas y se exploran métricas para facilitar la comprensión del modelo realizado. La manera más habitual de implementar los árboles de decisión en la detección de anomalías consiste en añadir clases de datos discordantes entre sus hojas. Un ejemplo de esta técnica se observa en [KJS16], donde se analiza la colección NSL-KDD de muestras de tráfico de red en busca de amenazas. De manera similar, en [SAN15] se propone un esquema colaborativo para reconocer aplicaciones maliciosas en dispositivos móviles. Tal y como se indica en [BG15], las principales ventajas de los árboles de decisión son su manera intuitiva de representar el conocimiento, gran precisión y simplicidad de implementación. Sin embargo, también presentan inconvenientes, destacando de entre ellos las dificultades a la hora de tratar con datos categóricos con un número variable de valores [DRT11] y su gran sensibilidad a pequeñas variaciones en ellos.

### A.2.7 ALGORITMOS GENÉTICOS

Los algoritmos genéticos son métodos de clasificación y minería de datos probabilistas, inspirados en la evolución biológica y su base genético-molecular. Se diferencian de otras estrategias de propósito similar, por hacen evolucionar una población inicial de individuos (observaciones) generada a partir de la información de entrada, sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas). Cada uno de estos individuos es representado mediante un genotipo, el cual habitualmente viene dado por un vector de atributos. Las mutaciones y recombinaciones genéticas son operaciones sobre dichos datos, entre las que se incluye la sustitución de elementos, inserción, eliminación etc. Tras completarse estas modificaciones, se lleva a cabo una selección de acuerdo a algún criterio preestablecido, en función de la cual se deciden los individuos más adaptados que van a prevalecer, y los menos aptos que son descartados. El algoritmo iterara sobre modificaciones y selecciones de datos hasta

que la población resultante provea la solución óptima al problema a resolver, o hasta que se supere un cierto número de ciclos. En función de las características de esos individuos se construye la solución definitiva. La Figura A.7 muestra el esquema del algoritmo genético básico, donde se seleccionan los $n$ mejores individuos de una población inicial de $N$. A partir de ellos selecciona parejas de individuos, los cruza y somete a mutaciones a su descendencia. En [KWG01] se revisa esta metodología con más detalle y se discute su adaptación a la clasificación y la adquisición de conocimiento inteligente.

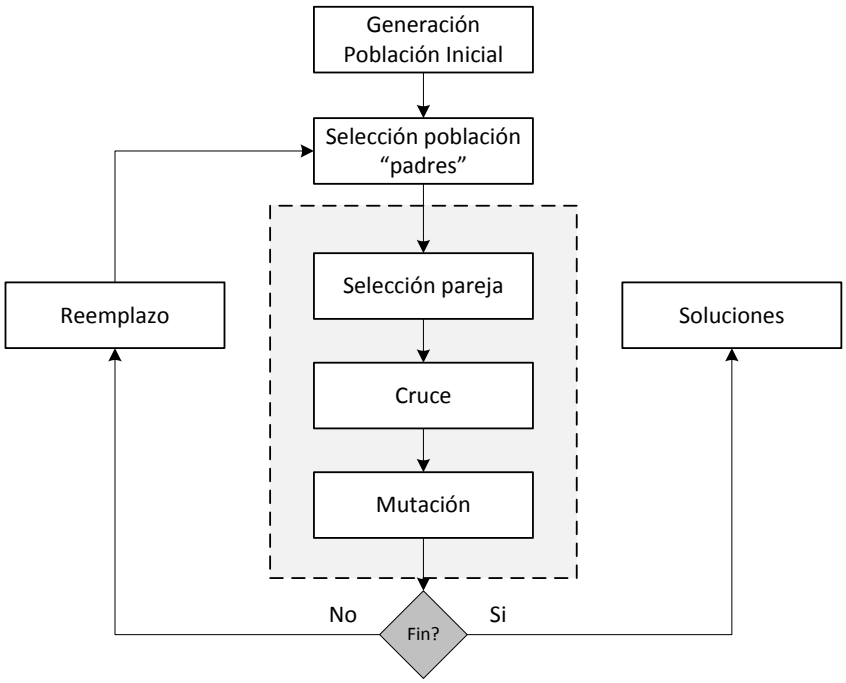



Figura A.7: Algoritmo genético básico.

La aplicación más frecuente de los algoritmos genéticos en el reconocimiento de anomalías es la inducción de reglas de detección a partir de datos de referencia, o colecciones de reglas conocidas *a priori*. Por ejemplo, esta estrategia es implementada en [RHL15] para la identificación de amenazas en redes o en [CKCY14] para detectar discordancias relacionadas con la seguridad marítima. Los algoritmos genéticos a menudo también se despliegan como complemento de los sistemas de detección con el fin de optimizar tareas relacionadas con la selección de características y optimización de métricas. Un ejemplo de esto se ilustra en [LKLP12], donde por medio de un algoritmo genético se mejoran las características de las matrices de información que modelan el tráfico de las redes protegidas. En términos generales, sus principales ventajas son la capacidad de siempre proveer una solución aproximada a la óptima, configuración modular, y fácil paralelización, de manera que su despliegue distribuido resulta muy intuitivo. Sus mayores desventajas son el riesgo de sobreentrenamiento, no garantizan el hallar soluciones óptimas, gran complejidad computacional y dependencia de muchos parámetros de ajuste (operaciones de cruce, mutación, función de aptitud, etc.).

### A.2.8 Sistemas inmunitarios artificiales

La detección de anomalías basada en sistemas inmunitarios artificiales emula las actividades llevadas a cabo por las células inmunitarias de los organismos vivos en la naturaleza. Tal y como se indica en [Ou12], estas aproximaciones a menudo se implementan mediante sistemas multi-agente y no se limitan al desempeño de tareas de identificación; también es habitual que implementen funcionalidades relacionadas con la prevención y la ejecución de acciones de mitigación. Los cuatro algoritmos inmunitarios artificiales más habituales en la bibliografía son: selección negativa, selección clonal, redes inmunitarias artificiales y teoría del peligro. En [DYN11] se profundiza en cada uno de ellos y se recopilan aproximaciones similares. A continuación se describen brevemente sus principales características.

#### A.2.8.1 Selección negativa

Los procesos de selección negativa son aquellos en que los agentes inmunitarios (linfocitos) aprenden a distinguir elementos ajenos al organismo (antígenos) de los que sí lo son. Para ello construyen un modelo de las actividades normales dentro del sistema, basándose en la generación iterativa de vectores de detección aleatorios y en el descarte de aquellos que se parecen demasiado a comportamientos normales. Por lo tanto, las observaciones realizadas no reconocidas como propias, son consideradas anómalas. Un claro ejemplo de la implementación de este método puede observarse en [LLZ16], donde su eficacia es evaluada a partir de estándares funcionales en el ámbito de la biomedicina. En [JD07b] se discute el proceso completo de selección negativa y sus principales algoritmos. A partir de ellos se concluye en que las principales ventajas de esta técnica son su naturaleza distribuida y el hecho de que no necesitan conocer previamente las características de los elementos internos y ajenos al sistema. Sin embargo, tienden a la emisión de tasas altas de falsos positivos, situación que frecuentemente lleva a su refinamiento por otras estrategias inmunitarias, como por ejemplo selección clonal.

#### A.2.8.2 Selección clonal

La teoría de la selección clonal postula que cuando un antígeno es detectado por el sistema inmunitario, tan solo los agentes que son capaces de identificarlo proliferan. En consecuencia, se reproducen por clonación y los nuevos agentes son sometidos a un proceso de mutación, mejorando de esta forma su capacidad de detectar el elemento potencialmente nocivo. Pasado un cierto periodo de cuarentena el sistema elimina gran parte de los nuevos agentes defensivos, autorregulándose y reduciendo el riesgo de que deriven en reacciones autoinmunes. La adaptación de este teorema al aprendizaje automático y la clasificación se describe en detalle en [CZ02], donde las características de los antígenos vienen dadas por los atributos de las observaciones y los agentes inmunitarios se presentan como vectores de umbrales que faciliten su identificación. En [SA14] se muestra un ejemplo de su implementación en la seguridad de la información, donde la selección clonal permite la generación y mejora de las reglas para la detección de anomalías. Tal y como se indica en [WG03], en términos generales la selección clonal destaca por su flexibilidad y gran

tolerancia a ruido. Su principal desventaja es su importante consumo de recursos de cómputo; a pesar de generar muy buenos resultados al analizar conjuntos de muestras pequeños, requiere de la construcción de una gran cantidad de generaciones al mutar.

#### A.2.8.3 Redes inmunitarias artificialesl

La teoría de las redes inmunitarias postula que, en la naturaleza, los diferentes agentes inmunitarios están interconectados a través de una compleja red de información. Por lo tanto, son capaces de reconocerse entre sí y de actuar de manera cooperativa. En su implementación para el reconocimiento de patrones, las redes inmunitarias artificiales parten de una colección de detectores especializados en un tipo concreto de antígenos. En la etapa de modelado, los detectores que mejor se comportan ganan prioridad frente a los que no lo hacen. Muchos de los algoritmos que aplican este paradigma combinan las funciones que modelan la conexión de agentes inmunitarios en la naturaleza con otras estrategias bio-inspiradas, como las selecciones negativa y clonal. En la actualidad no existe una estrategia generalizada de su implementación, encontrándose en [TKCH04] una recopilación de las técnicas más utilizadas. En [SA14] se muestra un ejemplo de su aplicación en la identificación de anomalías para la prevención del spam en el correo electrónico [YAC+07]. En general, las redes inmunitarias artificiales son muy eficientes, flexibles, y tolerantes a errores en los datos de referencia. Sin embargo, a menudo requieren de muchos parámetros de configuración, lo que implica complejidad de diseño.

#### A.2.8.4 Teoría del peligro

El nuevo paradigma planteado por la teoría del peligro surge como alternativa a las teorías que postulan que los sistemas inmunitarios de los seres vivos se centran en la discriminación de elementos que no pertenecen a sí mismo de los que si lo hacen. En su lugar plantea una serie de premisas que tienen como eje central la idea de que las respuestas inmunitarias son desencadenadas a partir de señales emitidas por los tejidos dañados directamente por los antígenos. De este modo distinguen los elementos que son nocivos de aquellos que no lo son. En [ABC+03] se repasan estas ideas más detalladamente y se discute su adaptación al reconocimiento de anomalías. Para ello se proponen metáforas de los procesos de autorregulación por muerte celular natural (apoptosis) y muerte celular anómala (necrosis) como señales básicas. El reconocimiento de intrusiones se lleva a cabo por medio de su correlación y la delimitación de zonas de peligro. Los métodos de detección más frecuentes derivados de la teoría del peligro son el Algoritmo de las Células Dendríticas o DCA (del inglés *Dendritic Cell Algorithm*) y el de los receptores de tipo Toll o TLR (del ingés *Toll-like Receptor algorithm*). El primero de ellos emula el procesamiento de señales múltiple llevado a cabo por las células CPA dendríticas [GAT10] y el segundo imita las interacciones entre dichas células y los agentes inmunitarios [AP13]. Las principales ventajas de estos métodos en la detección de anomalías son su bajo consumo de recursos de cómputo y facilidad de despliegue distribuido. Sin embargo heredan las desventajas de los sensores colaborativos multi-agente, siendo además frecuente la necesidad de un gran número de agentes para operar de manera satisfactoria.

## A.3 DETECCIÓN BASADA EN PROXIMIDAD

El grupo de estrategias de reconocimiento de anomalías basado en proximidad está presente en prácticamente todas las taxonomías de la bibliografía, tal y como se observa en [HKP11, Agg13]. Los algoritmos que lo componen comparten la asunción enunciada por V. Chandola et al. [CBK09], en la que se indica que los datos relacionados con observaciones normales tienden a agruparse en regiones concretas y muy densas, mientras que aquellos relacionados con anomalías se ubican lejos de dichas agrupaciones. Es decir, si la proximidad de una observación a analizar respecto a la colección de datos de referencia difiere considerablemente de la del resto de observaciones, es una observación anómala. Siguiendo la división propuesta por [HKP11], la manera más intuitiva de separar las estrategias de detección basadas en proximidad es en métodos basados en distancias y en densidad. A continuación, se describe cada uno de ellos.

### A.3.1 PROXIMIDAD BASADA EN DISTANCIA

Los métodos de detección de anomalías basados en distancias estudian el vecindario de los datos a analizar. Para ello tienen en cuenta su distancia respecto al k-esimo elemento más próximo de su vecindario. En la actualidad existen diferentes estrategias que aplican este método, siendo buenos ejemplos de ellos las publicaciones derivadas del modelo $DB(r, \pi) - anomal\acute{i}a$, distancia local (LDOF) y resolución (ROF). A continuación, se describe brevemente cada uno de ellos.

#### A.3.1.1 $DB(r, \pi) - anomalías$

Los métodos agrupados en esta categoría parten de la formalización del concepto de anomalía enunciado por E.M. Knorr et al. [KN97], en la que éstas son especificadas como$DB(r, \pi) - anomal\acute{i}as$. Según su definición, dada una distancia umbral $r \geq 0$, una distancia $dist(o, \acute{o})$ entre dos puntos (ver Sección 3.4 "Distancias y medidas de similitud"), y una fracción umbral $0 < \pi \leq 1$, una observación o es discordante si se cumple la siguiente inecuación:

$$\frac{\parallel \{\grave{o} \mid dist(o, \grave{o}) \leq r\} \parallel}{\parallel D \parallel} \leq n \tag{A.9}$$

De manera equivalente, E.M. Knorr et al. indicaron que era posible reconocer este tipo de anomalías por medio del análisis de la distancia entre $o$ y su vecino más próximo $o_k$, donde $k = \lceil \pi \parallel D \parallel \rceil$. En este caso o es anómalo si $dist(o, \grave{o}_k) > r$. En [KKZ10] se profundiza en este paradigma de detección, recopilándose una gran cantidad de propuestas que lo implementan. De entre ellas, la aplicación más directa de la definición de $DB(r, \pi)$-$anomal\acute{i}a$ también fue publicada por E.M. Knorr et. al [KN98], y fue denominada método del bucle anidado o NL (del inglés *Nested Loop algorithm*). La idea básica detrás de este algoritmo es que para cada observación $o_i$, $0 \leq i \leq n$ se calcule la distancia entre $o_i$ y cada uno de los otros datos, contándose en cada iteración el número de elementos en su vecindario, y sy naturaleza según Pseudocódigo 4.

**Algoritmo 4:** Algoritmo NL para hallar anomalías.

**Input :** El conjunto de objetos $D = \{o_1, \cdots, o_n\}$, umbral $r$, $(r > 0)$ y $p$, $(0 < \pi \leq 1)$.
**Output:** $DB(r, \pi)$ anomalías halladas en $D$.

```
while i=1 hasta n do
    contador ← 0;
    while j=1 hasta n do
        if i ≠ j Y dist(o_i, o_j) ≤ r then
            contador ← contador + 1;
        else
            if contador ≥ π × n then
                Salida {o_j no es discordancia en DB(r, π)};
            else
            end
        end
    end
    Mostrar o_i{o_i es anomalía en DB(r, π)};
end
```

A pesar de la sencillez de este algoritmo, en [HC02] se demostró que su alto coste computacional lo hace impracticable en casos de uso complejos. Como alternativa, en [KN98] se introducen otras aproximaciones similares, como el indexado multinivel, derivación del conocimiento intencional o la comparación de vecinos a partir de una matriz de adyacencia, de tal manera que el espacio muestral quede particionado en una matriz. Todas implican también un elevado coste computacional. A estos inconvenientes se le suma la crítica publicada por S. Ramaswamy et al. [RRS00], en la que se discute la dificultad de establecer una distancia $d$ apropiada y el hecho de que no permiten establecer un ranking de anomalías en función del nivel de discordancia. Todo esto dificulta su despliegue en casos de usos reales que requieran la gestión de una gran cantidad de información, aunque siguen siendo una buena solución a problemas de poca complejidad, que requieran analizar datos unidimensionales

#### A.3.1.2 Anomalías por distancia local

Como alternativa a los métodos derivados de la definición de $DB(r, \pi) - anomal\acute{i}as$, S. Ramaswamy et al. [RRS00] introdujeron una nueva línea de estrategias basadas en la valoración de la distancia local respecto al resto de datos en el vecindario o LDOF (del inglés *Local Distance-Based Outlier Detection Factor*) y en la construcción de un ranking de discordancias. El factor LDOF de cada anomalía se construye a partir de la distancia $\bar{d}_{x_p}$ de la observación $x_p$ a analizar respecto a sus k-vecinos más cercanos o K-NN (del ingés *K-nearest neighbors*) definida como:

$$\bar{d}_{x_p} = \frac{1}{k} \sum_{x_i \in N_p} dist(x_i, x_p) \tag{A.10}$$

y de su distancia interna expresa de la siguiente manera:

$$\bar{D}_{x_p} = \frac{1}{k(k-1)} \sum_{x_i, x_j \in N_p, i \neq j} dist(x_i, x_j) \tag{A.11}$$

Definiéndose la distancia local para $x_p$ a partir de la expresión:

$$LDOF_K(x_p) = \frac{\bar{d}_{x_p}}{\bar{D}_{x_p}} \tag{A.12}$$

donde $N_p$ es el conjunto de vecinos próximos a $x_p$ y $dist(x_i, x_j) > 0$ es la distancia elegida para valorar la diferencia entre $x_i$ y $x_p$ (ver Sección 3.4 "Distancias y medidas de similitud"). En la Figura A.8 se ilustran gráficamente cada uno de sus componentes.

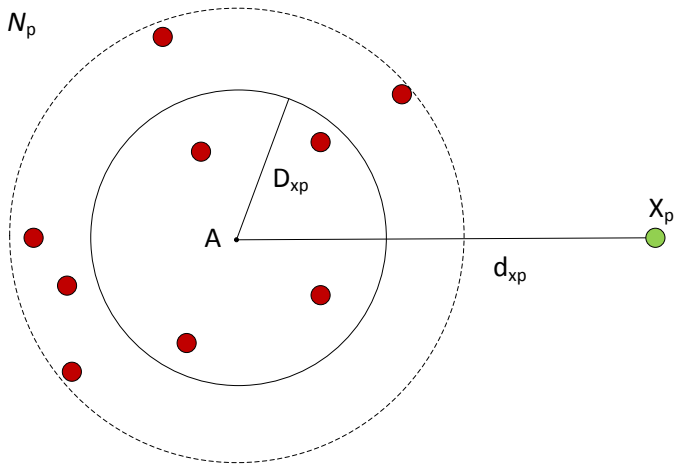



Figura A.8: Componentes LDOF en el plano.

En [ZHJ09] se repasan los principales algoritmos para la detección de anomalías inspirados en el cálculo de la LDOF. Sus autores demuestran que de entre ellos, el algoritmo al que denominaron top-$n$ LDOF es uno de los más prácticos para casos de uso reales, debido principalmente a su gran eficiencia a la hora de tratar información poco homogénea. Dado un conjunto de muestras $D$, el algoritmo top-$n$ ejecuta las siguientes acciones: (1) para cada objeto $p$ en $D$ se obtienen sus vecinos más próximos; (2) se calcula la LDOF para cada uno de ellos; (3) se devuelven los vecinos ordenados en función de su LDOF. Puede observarse como su complejidad es menor que la de los métodos basados en $DB(r, \pi) - anomalía$ (antes era de $\theta(N)^2$ y ahora es de $\theta(N \log N)$), y que además top-$n$ provee un ranking de anomalías. Sin embargo, persiste el problema de decidir la medida de distancia óptima.

#### A.3.1.3 ANOMALÍAS POR RESOLUCIÓN

Las estrategias basadas en resolución parten de la aproximación al reconocimiento de anomalías publicada por A. Foss et al. [FZ02], a la que denominaron TURN*. En ella cada vez que una nueva observación era etiquetada, se redistribuían las fronteras

de las agrupaciones de información. El algoritmo que propusieron agrupaba en la misma categoría todos los datos de proximidad baja, mientras que cada dato de proximidad alta representaba un único grupo. Por lo tanto, el agrupamiento óptimo se hallaba entre ambos extremos. A partir de estas categorías era posible identificar discordancias y crear un ranking, teniendo por principales ventajas su independencia de las características de los datos a analizar, y el hecho de que no requería parámetros de ajuste.

Posteriormente H. Fan et al. [FZFW09] resumieron la idea común en las técnicas de detección basadas en resolución, estableciendo la asunción de que dado un conjunto de muestras, los datos etiquetados como normales o anómalos pueden cambiar su clasificación al introducirse nueva información o al variar la distancia y el umbral de decisión. A la solución generada tras cada cambio de configuración la llamaron resolución. En base a esta premisa, definieron un factor análogo al LDOF para la valoración del nivel de anomalía de una muestra, pero adaptado a cambios de resolución, al que denominaron factor de discordancia basado en resolución o ROF (del inglés *Resolution-Based Outlier Factor*). En el cálculo de dicho umbral, se considera máxima resolución o *Rmax* al conjunto de datos que presentan una distancia mínima respecto al umbral de decisión, y que por lo tanto componen el grupo de observaciones anómalas. Por otro lado, los datos con mínima resolución o *Rmin* son los más lejanos al umbral de decisión, componiendo de esta manera el grupo de observaciones normales. El valor ROF viene dado por la expresión:

$$ROF\,(o) = \sum_{i=1}^{R} \frac{ClusterSize\,(o, r_{i-1}) - 1}{ClusterSize\,(o, r_i)} \tag{A.13}$$

donde $r_i, \cdots, r_R$ es la resolución para cada cambio, $R$ es el número total de cambios de resolución entre $Rmax$ y $Rmin$, y $ClusterSize(o, r_i)$ es el tamaño del grupo que contiene a la observación o tras la resolución $r$. Al igual que en LDOF, la complejidad de la generación del ranking de valoraciones es $\theta(N \log N)$, lo que demuestra su eficiencia. Su cálculo es un proceso sencillo y de especial eficacia en entornos de monitorización heterogéneos. Además, ha demostrado ser muy preciso a la hora de identificar anomalías locales teniendo en cuenta rasgos globales, tal y como se lleva a cabo en muchos de los escenarios de monitorización actuales.

### A.3.2 Proximidad basada en densidad

Los métodos de detección de anomalías basados en densidad tienen la característica común de que llevan a cabo la estimación de la densidad de elementos dentro de cada vecindario. Consideran anómalas a las muestras pertenecientes a vecindarios de baja densidad y normales a las que integran vecindarios de alta densidad. Por lo tanto, la densidad del vecindario al que pertenecen las anomalías es su principal medida de puntuación de discordancias. En la actualidad existe una gran cantidad de aproximaciones basadas en esta idea, como por ejemplo aquellas que se basan en el cálculo del factor de anomalía local o el nivel de anormalidad influido. A continuación se describen algunos ejemplos de propuestas que adoptan este paradigma.

### A.3.2.1 FACTOR DE ANOMALÍA LOCAL

El Factor de anomalía local o LOF (del inglés *Local Outlier Factor*) fue propuesto por M.M. Breuning et al. [BkNS00]. Con ello se pretendía mejorar la eficacia de la detección basada en proximidad aplicando distancias de similitud convencionales al tratar puntos sobre regiones de diferente densidad en el espacio. Este problema fue explicado con un ejemplo análogo al que se muestra en la Figura A.9, donde se observa un conjunto de datos bidimensional. En él la distancia entre cada elemento $q$ del agrupamiento $C_1$ es mayor que la de $p_2$ respecto al agrupamiento $C_2$. Esto puede llevar al error de etiquetar $p_2$ como no anómalo, ya que $p_2$ es considerado parte del agrupamiento $C_1$. Sin embargo, la discordancia $p_1$ probablemente sea identificada correctamente.

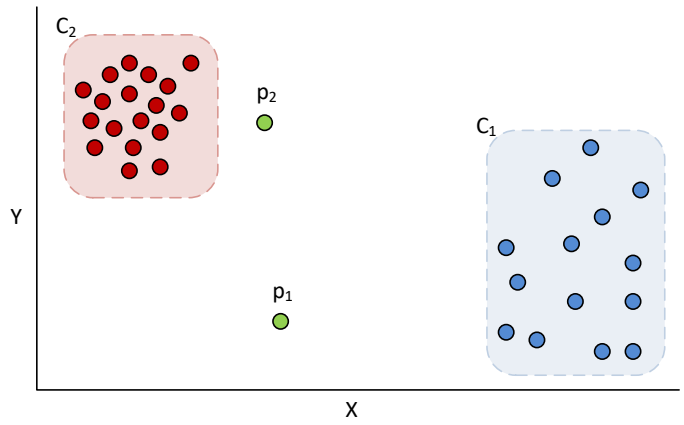



Figura A.9: Ejemplo de grupos de observaciones con diferente densidad.

Para M.M. Breuning et al., la solución a este problema se hallaba en considerar distancias relativas. En base a esto propusieron la detección de anomalías por medio de la comparación de la distancia local de una observación con la distancia local de sus vecinos. El valor de la distancia local es estimado por la distancia típica desde la que una observación puede ser alcanzada desde sus vecinos. En base a esta idea, la alcanzabilidad de un dato $o$ desde su vecino $p$ viene dada por la expresión $reach\_dist_k(p, o)$, y se representa de la siguiente manera:

$$reach\_dist_k(p, o) = \max\{k\_dist(o), dist(p, o)\} \tag{A.14}$$

donde $k_dist$ es la distancia entre la observación o a analizar y su k-esimo vecino. A partir de $reach\_dist_k$ es posible calcular su alcanzabilidad local o $lrd(o)$, tal que:

$$lrd(o) = \frac{1}{\frac{\sum_{p \in N_k(o)} reach\_dist_k(p,o)}{|N_k(o)|}} \tag{A.15}$$

donde $N_k$ es el conjunto de vecinos cercanos. Nótese que $lrd(o)$ es la inversa de la alcanzabilidad media entre $o$ y sus vecinos. El factor de anomalía local se calcula a partir de la comparación de este valor con el de sus vecinos, de tal manera que:

$$LOF_k(o) = \frac{\sum_{p \in N_k(o)} \frac{lrd(p)}{lrd(o)}}{\mid N_k(o) \mid} \tag{A.16}$$

donde si $LOF_k(o) \approx 1$, la observación $o$ es comparable con sus vecinos y por lo tanto normal. Cuanto más se aleje de 1, mayor será el nivel de discordancia.
Además de solucionar el problema previamente propuesto, este método tiene la ventaja de ser fácilmente adaptable a diferentes contextos. En [SZK14] se revisa en profundidad y se recopilan sus principales variaciones, discutiéndose su adaptación a varios escenarios. Sin embargo, tal y como señalan sus autores, su mayor desventaja es la dificultad de interpretar los resultados obtenidos, y por lo tanto de decidir cuándo a partir de un cierto valor el dato debe ser considerado discordante. Por ejemplo, $LOF_k(o) \approx 1.2$ puede interpretarse como un indicio de anomalía en un caso de uso concreto, mientras que en otros puede considerarse normal.

#### A.3.2.2 Nivel de anormalidad influido

Los métodos basados en el estudio del nivel de anormalidad influido parten de la propuesta de W. Jin et al. [JTHW06] para solucionar un importante problema relacionado con el uso del factor de anomalía local. En particular, W. Jin et al. demostraron que el LOF original no era un método demasiado eficaz para tratar casos de uso en los que los miembros de agrupamientos con distintas densidades se entremezclan. Como solución introdujeron la idea de considerar las relaciones entre vecindarios dentro del proceso de análisis. La llevaron a cabo definiendo la colección de vecinos más próximos del punto $p$ a estudiar $NN_k(p)$ y su inversa $RNN_k(p)$, siendo esta última el conjunto de observaciones que tienen a $p$ entre sus vecinos más próximos. Por lo tanto, en el conjunto muestral $D$ se cumple $RNN_k(p) = q \mid q \in D, p \in NN_k(q)$. Denominaron densidad de una observación $den(p)$ a la inversa de su distancia respecto a su k-esimo vecino. A partir de estos niveles es posible calcular el nivel de anomalía influido de dicha observación $INFLO_k(p)$, expresado de la siguiente manera:

$$INFLO_k(p) = \frac{\frac{\sum_{o \in kIS(p)} o}{\|kIS(p)\|}}{den(p)} \tag{A.17}$$

donde $kIS(p)$ es el radio de influencia de $p$, tal que $kIS(p) = NN_k(p) \cup RNN_k(p)$. Al igual que sucedía con LOF, a medida que $INFLO_k(p) \approx 1$, menor es la discordancia en $p$; cuando se aleja de 1 aumentan su posibilidad de ser anómalo. Además, este método también padece el problema de decidir a partir de qué valores un dato es interpretado como anómalo o no. Sin embargo, tal y como indican Y. Lu et al. [LMT+16], es una de las propuestas más precisas en la actualidad, aunque especialmente sensible a cambios de densidad. En [BGC15] se recopilan algunas propuestas similares, y se compara INFLO con cada una de ellas. En [HM16] se critican los algoritmos de elaboración de rankings basados en INFLO, resaltando su ineficiencia al tratar grandes cantidades de datos.

## A.4 Detección basada en agrupamiento

Los métodos de detección basada en agrupamiento asumen la premisa de que las observaciones que no pueden ser clasificadas en algún grupo de datos son anómalas. Asimismo, también son consideradas discordantes las observaciones que pertenecen a grupos especialmente dispersos y pequeños, o que previamente han sido etiquetados como grupos de datos anómalos. En [HKP11, BG15, BBK14] se recopila una gran cantidad de algoritmos basados en esta idea. Ejemplos de ello son la consideración de estas premisas tras aplicar DBSCAN, K-medias, lógica difusa o conjuntos aproximados; métodos que son descritos a continuación.

### A.4.1 DBSCAN

El algoritmo DBSCAN (del inglés *Density-Based Clustering Based on Connected Regions with High Density*) es un método de agrupamiento basado en densidad que construye grupos de observaciones teniendo en cuenta su cantidad de vecinos próximos [HKP11]. Para ello distingue tres tipos de datos sobre el plano: puntos núcleo, puntos densamente alcanzables y anomalías. Dado un punto $p$ se dice que pertenece al conjunto de puntos núcleo si una cantidad mínima de observaciones $minPts$ se halla a menos de una distancia $\varepsilon$ del mismo. Dichos datos son referidos como directamente alcanzables. Por otro lado, un punto $q$ es alcanzable por $p$ si existe un camino $p_1, \cdots, p_n$, donde $p_1 = q$, $p_n = q$ y cada $p_{i+1}$ es directamente alcanzable por $p_{i+1}$ considerando $\varepsilon$ y $minPts$, $0 \leq i \leq n$, $p_i \epsilon D$. Cada agrupamiento al menos contiene un punto núcleo, siendo el resto de puntos su periferia; a partir de ellos no es posible alcanzar más puntos. En consecuencia, todos los puntos de un agrupamiento están interconectados entre sí, y si un punto $p$ es densamente alcanzable por otro punto $q$, entonces también forma parte de su grupo. Los puntos que no son alcanzables desde ninguno otro son considerados anomalías.
DBSCAN ofrece varias ventajas, como su gran eficiencia, el no necesitar la especificación del número de agrupamientos para operar correctamente, o tolerancia al ruido. Sin embargo, su eficacia depende mucho de la distancia aplicada, y puede no ser demasiado preciso al tratar grupos de datos con grandes diferencias de densidades. Además, no es un algoritmo determinista, ya que los puntos alcanzables desde más de una agrupación pueden etiquetarse como pertenecientes a cualquiera de ellas. En consecuencia, se han publicado diferentes modificaciones de DBSCAN enfocadas a cubrir estas deficiencias, como GDBSCAN [SEKX98], OPTICS [ABK99] o DBSCAN* [CMZS13].

### A.4.2 K-medias

A pesar de su antigüedad, el algoritmo k-medias y sus variaciones siguen siendo una solución frecuente al problema de la detección de anomalías. En [Jai10] se discute su evolución y el cómo se ha ido adaptando a los nuevos escenarios de monitorización. Su versión original permite clasificar una colección de observaciones $X = x_1, \cdots, x_n$ en un número determinado de clusters $C = c_1, \cdots, c_K$. Para ello define una partición de las observaciones en $K$ basada en minimizar el error cuadrático entre la media empírica

de cada grupo y los datos a tener en consideración. Dado el error cuadrático entre la observación $\mu_i$ y los miembros del grupo $c_k$ definido tal que:

$$J(c_i) = \sum_{xj \in c_k} \parallel x_j - \mu_j \parallel \tag{A.18}$$

El objetivo de k-medias es minimizar el valor arrojado por la siguiente expresión:

$$J(c) = \sum_{k=1}^{K} \sum_{xj \in c_k} \parallel x_j - \mu_j \parallel \tag{A.19}$$

Tal y como se ilustra en [HKP11], una manera clásica implementar k-medias en el reconocimiento de anomalías consiste en asignar una puntuación de discordancia a la distancia $dist(o, c_o)$ entre cada observación $o$ respecto del valor central de su grupo más cercano $c_o$. Cuando $dist(o, c_o)$ supera un umbral predefinido $T_h$, tal que $dist(o, c_o) > T_h$, entonces $o$ se considera anómala. Las principales ventajas de este método son su sencillez y eficiencia, permitiendo procesar grandes cantidades de datos en poco tiempo, siempre y cuando el valor $K$ no sea demasiado alto. Sin embargo, su funcionamiento óptimo depende de la correcta elección de $K$. Además, el orden de procesamiento de datos puede hacer variar la distribución de los grupos generados, llevando a la necesidad de adoptar métodos de refinamiento que lo complementen.

### A.4.3 Lógica difusa

Los métodos de detección de anomalías basados en lógica difusa se basan en la construcción de grupos de datos cuya pertenencia, a diferencia de los algoritmos de agrupamiento convencionales, es definida por el intervalo cerrado $[0, 1]$ en vez de por el conjunto de valores estáticos $\{0, 1\}$. Según la propuesta original de L. A. Zadeh [Zad65], los sistemas basados en lógica difusa o LFS (del inglés *Fuzzy Logic Systems*) se componen de cuatro elementos: fuzificador, motor de inferencia, base de reglas y defuzificador. En el primero de ellos se calcula el grado de membresía de los datos de entrada respecto a cada grupo. El motor de inferencia y la base de reglas actúan de manera similar a los de los sistemas expertos basados en reglas (ver Anexo A.2.6 "Sistemas expertos basados en reglas"). Finalmente, el defuzificador transforma los datos difusos inferidos en información adecuada al problema a tratar (ver Figura A.10).

Tal y como se muestra en [TRM09], las reglas difusas pueden construirse para describir grupos de datos normales y anómalos. También permiten establecer la probabilidad de que una conclusión sea correcta, pudiendo adaptarse fácilmente a la evaluación de amenazas en la seguridad de la información. Nótese que la definición de este tipo de reglas a menudo no es una tarea trivial, lo que implica la necesidad de importar estrategias de aprendizaje automático, como árboles de decisión o algoritmos genéticos [EFB+15]. La principal ventaja de los FLS es su capacidad de valorar variables lingüísticas (ex. poco, mucho, suficiente, etc.), las cuales son adaptadas a valores difusos en la etapa de fuzificación. Por lo tanto, facilitan su implementación tanto en procesos lineales como en contextos de difícil modelado. En contra, cabe destacar su gran dependencia de la etapa

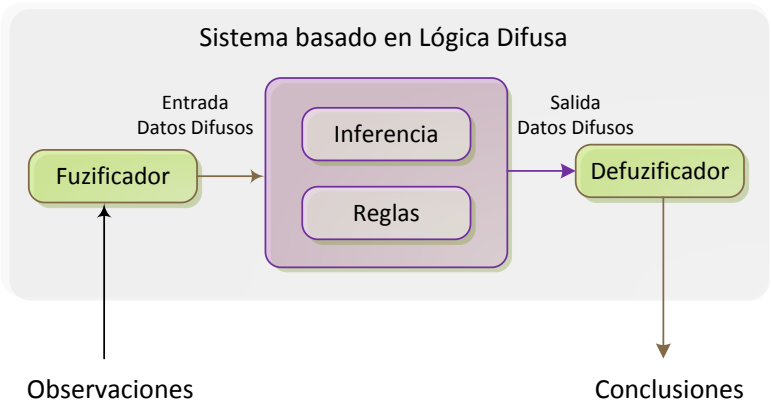



Figura A.10: Arquitectura genérica de los sistemas basados en lógica difusa.

de entrenamiento, la cual a menudo es lenta y costosa computacionalmente. Además, es posible hallar dificultades en la elección de la función de membresía óptima.

### A.4.4 CONJUNTOS APROXIMADOS

Los conjuntos aproximados (del inglés *rough*) sets fueron introducidos por Z. Pawlak [Paw82] con el propósito inicial de transformar información proveniente de observaciones en conocimiento. Según esta publicación, un conjunto aproximado es un agrupamiento de objetos que no pueden ser clasificados de manera certeza en función de sus atributos. Según esta teoría, se denomina relación de indiscernibilidad a la relación entre objetos que presentan un subconjunto predefinido de atributos con valores similares. En función de una relación de indiscernibilidad, un conjunto aproximado se divide en dos subconjuntos: aproximación superior (región negativa) e inferior (región positiva). El primero contiene las observaciones que podrían pertenecer a dicho grupo, y el segundo reúne aquellas que seguro que lo hacen. Ambos subconjuntos están separados por la región de indiscernibilidad (ver Figura A.11).

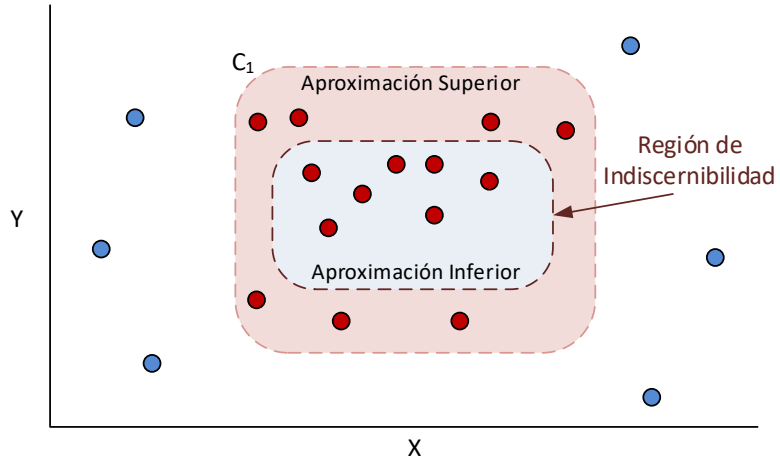



Figura A.11: Ejemplo de conjunto aproximado.

Si un conjunto aproximado no tiene elementos en su región negativa, se dice que es exactamente definible. Si un conjunto de atributos $A$ permite definir un agrupamiento exactamente definible, es denominado reducción $R(A)^*$. A la intersección de todas las reducciones se la denomina núcleo. En base a la distancia que separa los datos a analizar de las regiones positivas de los grupos de similitud más cercanos es posible identificar elementos discordantes. En las últimas décadas se han publicado diferentes algoritmos para la definición de reglas de pertenencia y extracción de atributos exactamente definibles. En [Orl98] se recopilan muchos de ellos. En [MPBMOO15, CLL$^+$15] se observan ejemplos más recientes de su adaptación a la detección de anomalías. Las principales ventajas de esta aproximación son su capacidad de reducir el número de atributos a procesar sin perder demasiada calidad, eliminación de datos redundantes, escasa dependencia de parámetros de ajuste y tolerancia a ambigüedades entre muestras. Sin embargo, para su correcto funcionamiento las tablas que representan dicha información deben estar completamente especificadas, situación que difícilmente se satisface en casos de uso complejos.

## A.5 Detección basada en estadística

Las estrategias de detección de anomalías basadas en estadística asumen la premisa planteada por V. Chandola et al. [CBK09], según la cual, las observaciones discordantes son aquellas que suscitan la sospecha de ser parcialmente o completamente irrelevantes desde un punto de vista estadístico, debido a que no han sido generadas a partir del modelo estocástico que previamente se haya asumido. Es decir, las observaciones normales se encuentran en las regiones de mayor probabilidad dentro de un modelo estocástico, mientras que las discordancias lo hacen en las zonas que localizan las probabilidades más bajas. El resto de esta subsección describe ejemplos de métodos que adoptan este paradigma.

### A.5.1 Pruebas estadísticas

Tal y como es descrito en [KKZ10], la estadística provee una gran cantidad de pruebas que facilitan la decisión de considerar una observación como normal o anómala. Su *modus operandi* es el siguiente: se establece una asunción acerca del conjunto de datos denominada hipótesis nula $H_o$. A lo largo de la prueba se considera la hipótesis alternativa $H_\alpha$, donde $\alpha$ indica el nivel de significancia de adoptar esta decisión. Al finalizar la prueba se determina la validez de $H_o$. Por lo tanto, se trata de una metodología intuitiva, que no suele requerir cálculos especialmente complejos. Sin embargo, tiende a ser muy dependiente de las características de la información a tratar, lo que frecuentemente lleva a la elección de pruebas no paramétricas. En la actualidad existen diferentes maneras de aplicar este método a la detección de anomalías, siendo habituales en la bibliografía, aquellas que permiten determinar a partir de $H_o$ si un dato pertenece a una población específica, resultando ser discordante en el caso contrario. Ejemplos claros de esta implementación se observan en las pruebas de Mann-Whitney [MW46] y la prueba de los rangos con signo de Wilcoxon [Wil46], las cuales son descritas a continuación.

#### A.5.1.1 U-TEST

La prueba de contraste de hipótesis propuesta por Mann-Whitney, conocida como U-test [MW46] es una extensión de la prueba T-Student no paramétrica adaptada al análisis de dos muestras independientes. Su objetivo es la comprobación de que dos muestras simétricas hayan sido extraídas de la misma población ($H_o$). Los datos a analizar deben estar medidos en una escala ordinal, lo que implica la necesidad de ordenar las puntuaciones obtenidas. El cálculo del estadístico $U$, parte de los valores $U_1$ y $U_2$, calculados mediante las fórmulas:

$$U_1 = n_1 n_2 \frac{n_1(n_1+1)}{2} - R_1 \tag{A.20}$$

$$U_2 = n_1 n_2 \frac{n_2(n_2+1)}{2} - R_2 \tag{A.21}$$

donde $n_1$ y $n_2$ son la longitud de los vectores de las puntuaciones ordenadas y $R_1$ y $R_2$ las sumas de los rangos de las observaciones. El estadístico de contraste U se define como $U = \min\{U_1, U_2\}$. Dado que el número de muestras es grande, el estadístico $U$ tiende a parecerse a la distribución normal, de manera que:

$$z = \frac{U - M_u}{\sigma_u} \tag{A.22}$$

donde $m_u$ y $\sigma_U$ son la media y la desviación estándar de $U$, formuladas de la siguiente manera:

$$m_u = \frac{n_1 n_2}{2} \tag{A.23}$$

$$\sigma_u = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \tag{A.24}$$

Una vez obtenida la probabilidad de pertenencia, se comprueba que la cota de porcentaje de error sea admisible. Si el conjunto de muestras de referencia presenta naturaleza normal, pero la prueba no es superada, se ha identificado una posible discordancia. Además de las ventajas inherentes a las pruebas de contraste de hipótesis no paramétricas, el U-test permite el análisis de dos conjuntos de muestras de diferente tamaño de una manera sencilla e intuitiva. Sin embargo, pierde precisión al tratar con colecciones pequeñas. Un claro ejemplo de su aplicación al reconocimiento de anomalías se observa en [MVSOGV16], donde facilita la comparación entre vectores de puntuaciones relacionados con secuencias de actividades legítimas dentro de un sistema de la información, respecto a métricas asociadas a posibles ataques.

#### A.5.1.2 PRUEBA DE LOS RANGOS CON SIGNO DE WILCOXON

La prueba de los rangos con signo de Wilcoxon [Wil46] opera sobre vectores de datos apareados, de manera que son comparados los elementos con el mismo índice. Para ello parte de la suposición de que se dispone de $l$ pares de observaciones, denominadas $(x_i, y_i)$. El objetivo del test es determinar que los valores $x_i$ e $y_i$ son equivalentes ($H_o$). En el

reconocimiento de anomalías éstos corresponden con las distancias entre los elementos a analizar o su grado de pertenencia al conjunto de datos normal. Para verificar la asunción inicial, ordena los valores absolutos $\mid z_1 \mid, \cdots, \mid z_n \mid$, tal que $z_i = z_i$ y les asigna un rango $\mathcal{R}_i$. A continuación, se calcula la suma de los rangos con diferencias positivas $T+$, y la de las diferencias negativas $T$-. Esto permite hallar el estadístico de contraste, que viene dado por la expresión $T = min\{T+, T-\}$. A partir del estadístico $T$ es posible calcular un $p$ valor. En el caso de que $l$ sea pequeño (normalmente $l < 20$) esto se hace a partir de la tabla estadística de la distribución de Wilcoson. Si $l$ es suficientemente grande, se calcula en función de la distribución normal, tal que:

$$Z = \frac{\frac{W - n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{2}}} \tag{A.25}$$

La prueba es superada si $p < I$, siendo $I$ el intervalo de confianza asignado. Esto es interpretado como que la diferencia entre las poblaciones es significativa, y por lo tanto no se debe al azar. Cuando una muestra pertenece a observaciones normales y la otra tiene un origen desconocido, se considera que los datos a analizar son discordantes. Este método presenta las ventajas y desventajas inherentes a las pruebas estadísticas no paramétricas, siendo una alternativa a la prueba T-Student en los casos en que no se puede asumir que la distribución de datos sea normal, y en que la comparación se lleve a cabo considerando parejas de datos con el mismo índice dentro del vector que los representa. En [MVSOGV15e, MVSMGV18] se muestra un ejemplo de su implementación en la detección de anomalías en las secuencias de arranque de las aplicaciones de dispositivos móviles.

### A.5.2 Modelos de Mezclas Gaussianas

Los Modelos de Mezclas Gaussianas o GMM (del inglés *Gaussian Mixture Model*) son modelos estadísticos que representan la distribución de probabilidad conjunta, entre los vectores de características, y la población a la que pertenecen las muestras [Rey09]. Este tipo de representaciones se denominan modelos generativos. Los modelos generativos se basan en el conocimiento de la función de probabilidad conjunta entre los datos de entrenamiento, y la clase a la que pertenecen, permitiendo la generación de datos nuevos, análogos a la función de distribución de probabilidad de su modelado. Según, la probabilidad de que un vector de características pertenezca a un modelo estadístico $\lambda$ específico, se representa a través de una combinación lineal de distribuciones de probabilidad gaussianas D-dimensionales, con $D$ igual al tamaño del vector de características:

$$P(\vec{y}_t \mid \lambda) = \sum_{i=1}^{M} w_i p_i(\vec{y}_t) \tag{A.26}$$

donde $\vec{y}_t$ es el vector de características, $M$ el número de componentes gaussianas, $w_i$ sus pesos, y $p_i$ viene dado por la siguiente expresión:

$$p_i(\vec{y}_t) = \frac{1}{(2\pi)^{\frac{D}{2}} \mid (\sum i)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x}_t - \vec{\mu}_i)' \sum i^{-1} (\vec{x}_t - \vec{\mu}_i)} \tag{A.27}$$

donde $\sum_i^M w_i = 1$, $\int_{-inf}^{+inf} p_{i(y)dy} = 1$ y $\mu_i$ es la observación realizada. Es decir, a cada componente gaussiana le corresponde un vector de medias y una matriz de covarianza. Dadas estas condiciones, el modelo se puede representar de la siguiente manera:

$$\lambda = w_i \mu_i, \sum i,\ 1 \leq i \leq M \tag{A.28}$$

A partir de la distancia entre las nuevas observaciones y $\lambda$, es posible determinar si pertenecen o no a dicha población. En la detección de anomalías $\lambda$ habitualmente se construye a partir de datos normales, discriminando aquellos que difieren de manera representativa. Sus principales ventajas son su flexibilidad a la hora de construir grupos, y la posibilidad de que un dato pertenezca a varios de éstos. Sin embargo, es complejo y costoso computacionalmente, lo que ha dado lugar a nuevas aproximaciones enfocadas a su optimización, como por ejemplo, aquellas que incorporan algoritmos de esperanza-maximización o EM (del inglés *Expectation–maximization*) [DLR77]. En [LHPW16, LMV14] se observan ejemplos de su aplicación en la detección de anomalías. En [LHPW16] permite la identificación de vuelos sospechosos, y en [LMV14] dirige la detección de anomalías temporales en grandes aglomeraciones de personas.

### A.5.3 MODELOS BASADOS EN REGRESIÓN

Tal y como se indica en [CBK09], la detección de anomalías basada en regresión se ha centrado principalmente en el estudio de series temporales de observaciones. En esta adaptación se consideran dos etapas de procesamiento de información fundamentales: la construcción de los modelos regresivos y su análisis en base a la predicción de su evolución. En la primera se construye un modelo de regresión adaptado a los datos de referencia. A partir de ello, para cada nueva observación se determinará una puntuación de discordancia en función de su componente residual. El nivel de significancia de dicho valor frecuentemente se determina a partir de pruebas estadísticas de contraste de hipótesis (ver Anexo A.5.1 "Pruebas estadísticas"). A partir de los datos iniciales, el modelo permite inferir las próximas observaciones. Dicha predicción facilita la construcción de un umbral adaptativo, el cual servirá de referencia a la hora de decidir si una variación del comportamiento observado es normal o discordante. De entre los métodos basados en regresión cabe destacar por su relevancia en la bibliografía la familia de modelos ARIMA (del inglés *Auto Regressive Integrated Moving Average*), también conocidos como modelos de Box-Jenkins [BJ76] y el alisamiento exponencial. A continuación, se describe brevemente cada uno de ellos.

#### A.5.3.1 FAMILIA DE MODELOS AUTORREGRESIVOS

El componente más básico de este conjunto de técnicas de análisis de información es el modelo autorregresivo AR, a partir del cual es posible estudiar el comportamiento de una

serie temporal en base a su pasado. Por lo tanto, es una regresión de la variable estadística en sí misma. Un modelo AR(p) define la serie temporal de la siguiente forma:

$$X_t = \mu + \theta_1 X_{t-1} + \ldots + \theta_p X_{t-p} + a_t \tag{A.29}$$

siendo $\mu$ un valor constante, $\theta_i$ variables, $a_t$ ruido blanco (es decir, ruido con media cero) y $p$ el orden de la regresión. La familia de modelos de medias móviles MA además tiene en cuenta su error (media móvil de la serie de errores), definiéndose $MA(q)$ tal que:

$$X_t = \mu + \theta_1 a_{t-1} + \ldots + \theta_p a_{t-p} + a_t \tag{A.30}$$

La combinación de AR y MA se denomina ARMA, expresándose los modelos ARMA(p,q) a partir de la siguiente expresión:

$$X_t = \mu + \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \ldots - \theta_p a_{t-q} \tag{A.31}$$

que equivale a:

$$(1 - \phi_1 B - \ldots - \phi_p B^p) X_t = \mu + (1 - \theta_1 B - \ldots - \phi_q B^q)\ a_t \tag{A.32}$$

simplificado como:

$$\Phi_p\,(B)\,X_t = \mu + \Theta(B) a_t \tag{A.33}$$

Los procesos integrados son aquellos que precisan de la realización del operador diferencia para ser estacionarios. Por lo tanto, un proceso integrado de orden $d$ verifica:

$$(1 - B)^d X_t = a_t \tag{A.34}$$

donde $a_t$ es un proceso de ruido blanco. Decimos que si $X_t$ es un proceso ARIMA(p,d,q) entonces $(1 - B)^d X_t$ es un ARMA(p,q). La principal ventaja de esta familia de métodos son su eficiencia y capacidad de considerar directamente intervalos de confianza. Sin embargo, dado que son estrategias de modelado lineal, pueden ser difíciles de construir sobre casos de uso complejos. Un ejemplo de su aplicación se ilustra en [KHC+16], donde una variante de ARMA se adapta al reconocimiento de anomalías en las cadenas de montaje de productos industriales. En [YJJ16] la implementación de ARIMA permite detectar tráfico sospechoso en los nodos de una red de sensores inalámbrica.

#### A.5.3.2 Alisado exponencial

Los métodos de modelado basados en alisado exponencial hacen uso de datos históricos para obtener una serie temporal mucho más suave, es decir, libre de ruido y elementos discordantes, a partir de la cual hacer la predicción. Para ello se consideran todos los datos previos al periodo de previsión, y se les asigna pesos decrecientes exponencialmente a medida que se alejan de dicho instante de tiempo. La técnica de alisamiento exponencial se caracteriza por actualizar tras cada periodo de observación, hasta tres parámetros de las series de datos: nivel medio (alisado simple), nivel medio y tendencia (alisado doble

o modelo Holt), y nivel medio, tendencia y estacionalidad (alisado triple o Holt-Winters) [GD80]. El primero de estos modelos puede construirse a partir de la siguiente expresión:

$$S_t = \alpha x_t + (1 - \alpha) S_{t-1} \tag{A.35}$$

siendo $S_t$ el valor calculado para el instante $t$, $x_t$ la observación realizada en $t$ y $\alpha$ el factor de alisado, $0 < \alpha < 1$.
Por otro lado, el alisado de tendencia y estacionalidad propuesto por Holt es descrito en la siguiente ecuación recursiva que tiene como casos base $s_1 = x_1$ en $t \leq 1$ y $b_1 = x_1 - x_0$ en $t > 2$:

$$S_t = \alpha x_t + (1 - \alpha)(S_{t-1} + b_{t-1}) \tag{A.36}$$

$$b_t = \beta\,(S_t - S_{t-1}) + (1 - \beta) b_{t-1}) \tag{A.37}$$

donde $\beta$ es el factor de alisado de la tendencia. A partir de este modelo es posible predecir valores en un intervalo $m$ de observaciones a partir de $F_{t+m} = S_t + m b_t$.
Finalmente, el modelo de Holt-Winters se expresa de la siguiente manera:

$$B_t = \alpha(H_i - S_{t-N}) + (1 - \alpha)(B_{t-1} + T_{t-1}) \tag{A.38}$$

$$T_t = \beta(B_t - B_{t-1}) + (1 - \beta) T_{t-1} \tag{A.39}$$

$$S_t = \gamma(H_t - B_t) + (1 - \gamma) B_{t-n} \tag{A.40}$$

siendo $B_t$ la estimación del nivel medio en $t$ y $T_t$ la tendencia. Los parámetros $\alpha$, $\beta$ y $\gamma$ facilitan el ajuste de cada uno de estos valores. La predicción en la observación se construye mediante el siguiente cálculo:

$$F_{t+1} = B_t + T_t + S_t \tag{A.41}$$

En términos generales, el alisado exponencial es sencillo, eficiente y no requiere de un conjunto de datos demasiado grande para su inicialización. Sin embargo, puede ser difícil de ajustar a entornos de monitorización complejos, y la estrategia de inicialización de sus parámetros tiene un gran impacto en la calidad de las predicciones. En [ARP14] se muestra un ejemplo de su implementación en el modelado de tráfico de redes, a partir del cual es posible deducir las anomalías. En [SLMB14] se aplica para identificar problemas de salud a partir de información monitorizada por sensores inalámbricos en pacientes.

### A.5.4 ANÁLISIS DE COMPONENTES PRINCIPALES

Los métodos de identificación de anomalías basados en algoritmos de análisis de componentes principales o PCA (del inglés *Principal Component Analysis*) facilitan la reducción de la dimensionalidad del conjunto de datos a estudiar de manera no supervisada [AW10]. Esto permite distinguir los rasgos característicos de las muestras normales de aquellas que no lo son. Tal y como se indica en [LYW13], la deducción de los componentes principales a menudo implica la construcción de una matriz de covarianza de datos y

el cálculo de sus vectores propios dominantes. Se asume que estos vectores representan la información más relevante de la matriz, y por lo tanto son considerados como sus direcciones principales. Dado $A = [{X_1}^T, {X_2}^T, \cdots, {X_n}^T] \in \mathbb{R}^{n \times p}$ donde $X_i$ es una muestra en un espacio de dimensión $p$, y $n$ es el total de muestras, según [LYW13] un ejemplo clásico de PCA es definido por el siguiente problema de optimización:

$$max \sum_{I=1}^{N} U^T (x_i - \mu)(x_i - \mu)^T U \tag{A.42}$$

donde la matriz $U$ está definida por $k$ vectores propios. Por lo tanto, este problema se formula como un ejercicio típico de identificación del sub-espacio de $U$ en el que los datos representados muestran una mayor variación. De manera alternativa, el PCA puede plantearse como la búsqueda de sub-espacios en los que, al proyectar información, los errores mínimos cuadrados de los residuos sean mínimos [Agg13], tal que:

$$min\ J(U) = \sum_{I=1}^{N} \parallel (x_i - \mu) - UU^T (x_i - \mu) \parallel^2 \tag{A.43}$$

Según se muestra [Agg13], una de las maneras más habituales de aplicar PCA al reconocimiento de anomalías consiste en el estudio de la dimensión de los elementos residuales; las observaciones con una mayor cantidad de componentes residuales muestran un mayor grado de discordancia. Esto es debido a que no parecen formar parte del sub-espacio al que pertenecen las muestras de referencia. Las ventajas más representativas de este método es que eliminan la redundancia entre muestras de referencia y reduce el ruido, permitiendo discriminar discordancias en las colecciones de datos normales. Sin embargo, tal y como indicaron H. Zou et al. [ZHT06], dado que cada componente principal es una combinación lineal de las variables de referencia, resulta muy difícil valorar su relevancia a la hora de construir subespacios. En [FRP15] se ilustra un ejemplo de su implementación para la construcción de perfiles del modo de uso de redes, y la identificación de anomalías a partir de ellos. Otro ejemplo se observa en [HKC+15], en el que PCA permite la detección de incidencias en infraestructura crítica.