Adaptación y calibrado de algoritmos de predicción para la identificación de ataques DDoS en redes de quinta generación

Autores

Andrés Herranz González

Borja Lorenzo Fernández

Guillermo Rius García

Director

Luis Javier García Villalba

Codirector

Jorge Maestre Vidal

Trabajo de Fin de Grado

Facultad de Informática

Universidad Complutense de Madrid

Madrid, Junio de 2018

Agradecimientos

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado SELFNET aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, `http://gass.ucm.es`, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

En primer lugar, queremos agradecer la labor de nuestro cotutor Jorge Maestre Vidal y de Marco Sotelo Mongue, por creer en nosotros desde el primer momento, por todo el esfuerzo realizado y por todo lo que nos han enseñado y hemos podido aprender a su lado. Estaremos eternamente agradecidos.

Por otro lado, a toda la gente por el apoyo recibido, no sólo en el desarrollo de este trabajo sino durante toda la carrera, pieza fundamental para llegar dónde hemos llegado.

Agradecer en especial a mis padres, que han estado siempre a mi lado, a mis abuelos, que me enseñaron a ser como soy, a mis amigos que siempre están ahí, y a Lore por apoyarme y ayudarme a llegar hasta aquí. Andrés

Quiero agradecer a mi padres, hermanos y abuelos, a mis amigos y en especial a María, por haberme enseñado a ser y haberme ayudado a llegar tan lejos. Borja.

A mis padres, a mi hermano y a toda mi familia, gracias por apoyarme en todo momento, creer en mí y ayudarme a superar todas las dificultades. Guillermo.

Abstract

The advances of wireless mobile networking towards its fifth generation, popularly known as 5G, arrive hand in hand with a collection of emerging technologies that provide important improvements in terms of key performance indicators related among others, with effectiveness, efficiency, energy consumption and mobility. They also facilitate the development of self-organization capacities based on studying observations on the monitoring environment, thus bringing cognitive and holistic solutions to their incident response mechanisms. In order to contribute to their development, the performed work focuses on the problem of anticipating network events. With this purpose, a novel adaptive prediction strategy that takes into account the great heterogeneity of data sources and the non-stationarity inherent in the forthcoming landscape, has been developed. This has been achieved through the implementation of machine learning methods for selecting the best algorithms according to the context, and by making evolve their calibration based on variations at traffic observations. The proposed approach has been evaluated in the grounds of the functional evaluation standard M3-Competition. In addition, it was deployed on a specific use case: the detection of distributed denial of service attacks. For the latter, a collection of network traffic samples captured from devices of different nature has been gathered, from which classical indicators of this kind of threats have been extracted and analyzed. It is worth to highlight that the extensive experimentation displayed very promising results, thus pointing out interesting lines of future research.

Keywords

Communication Networks, Denial of Service, Machine Learning, Pattern Recognition, Prediction.

Resumen

El avance de las redes de telefonía móvil hacia su quinta generación, popularmente conocida como 5G, viene de la mano de una colección de tecnologías emergentes que brinda importantes mejoras en sus principales indicadores de desempeño, como su rendimiento, eficiencia, ahorro energético o movilidad. También permiten desarrollar capacidades de autoorganización basadas en el estudio de observaciones en el entorno de monitorización, dando un enfoque cognitivo y holístico a sus mecanismos de respuesta a incidencias. Con el fin de contribuir a su desarrollo, el trabajo realizado se centra en la anticipación de eventos en red, habiéndose desarrollado una estrategia de predicción adaptativa que tiene en cuenta la gran heterogeneidad de fuentes de información y la no estacionariedad, inherentes a los escenarios de red venideros. Esto se ha logrado mediante la implementación de estrategias de aprendizaje automático para la selección de los mejores algoritmos según el contexto, y la evolución de su calibrado acorde a las variaciones de las observaciones. El método propuesto ha sido evaluado a partir del estándar de evaluación funcional M3-Competition y en un caso de uso específico: la detección de ataques de denegación de servicio distribuidos. Para esto último se ha recopilado una colección de muestras de tráfico de red capturados en dispositivos de diferente naturaleza, a partir de las cuales se han extraído y analizado indicadores propios de este tipo de amenazas. La amplia experimentación realizada ha arrojado resultados muy prometedores, indicando interesantes líneas de trabajo futuro.

Palabras Clave

Aprendizaje Automático, Denegación de Servicio, Predicción, Reconocimiento de Patrones, Redes de Comunicaciones.

Índice General

Índice de Tablas

Índice de Figuras

Capítulo 1

Introducción

En los últimos años la sociedad ha incrementado su necesidad de crear, distribuir y manipular información, llegando a jugar un papel esencial en prácticamente todos los ámbitos de la vida de las personas, que abarcan desde aspectos relacionados con cultura y ocio, hasta economía, salud o aprovisionamiento de recursos fundamentales como el agua o energía. Dentro del panorama actual, la tecnología móvil se ha convertido en uno de los pilares que sostienen la sociedad de la información, la cual hoy en día dispone de sus avances hasta una cuarta generación (comúnmente conocida como 4G) de soluciones tecnológicas para la comunicación inalámbrica entre dispositivos. Pero a pesar de que este conjunto de herramientas lleva vigente desde hace casi una década, el incremento en la demanda de información ha dado pie al desarrollo de una nueva generación (5G), centrada en la mejora de sus los indicadores de rendimiento o KPIs (del inglés Key Performace Indicators) de sus predecesoras. Estos incluyen entre otras, importantes mejoras en términos de calidad de experiencia, eficiencia, movilidad, conectividad o consumo. El difícil desafío de estos requisitos ha incentivado la aparición de las denominadas nuevas tecnologías de redes de comunicación o tecnologías emergentes de comunicación, como por ejemplo las redes definidas por software o SDN (del inglés *Software-Defined Networking*), virtualización de funciones de red o NFV (del inglés *Network Function Virtualization*), computación en la nube, redes autoorganizadas o SON (del inglés *Self-Organizing Networks*), análisis de grandes datos (del inglés Big Data) o su adaptación a procesos de aprendizaje profundo (del inglés *Deep Learning*). Estas forman parte de un complejo y sofisticado ecosistema que dificulta la gestión y el tratamiento de los datos, caracterizado por su heterogeneidad y no estacionariedad. En consecuencia, muchas de las estrategias analíticas convencionales deben de ser adaptadas a las características inherentes a las nuevas fuentes de información.

Con el fin de contribuir a su desarrollo, el trabajo realizado ha abordado el problema de inferencia de conocimiento por medio de la proyección del estado actual de la red. Esto ha requerido de la realización de una revisión en profundidad de las principales métodos de predicción adaptados al análisis de tráfico de redes, y la definición de un sistema experto capaz de tomar decisiones proactivas en función del estado actual de la red y la eficacia de las decisiones tomadas con anterioridad, lo que implica llevar a cabo tareas como la decisión del mejor algoritmo en función del contexto, calibración de sus parámetros de ajuste, establecimiento de umbrales de predicción o el análisis de los datos obtenidos. El trabajo realizado se enmarca en el proyecto de financiación europea SELFNET - Framework for Self-Organized Network Management in Virtualized and Software Defined Networks (Convocatoria: H2020-ICT-2014-2, Acción de Investigación e Innovación (RIA), Número de Propuesta: 671672)[1], habiendo sido integrado como parte de la solución análitica enmarcada en su tarea T4.3 "Analyzer Module". Una vez concluido su desarrollo, y tras probarse su eficacia en base al estándar funcional de evaluación M3-Competition para estrategias de predicción, se ha llevado a cabo su evaluación en un caso de uso real, cuya elección ha sido motivada por los hechos descritos a continuación:

1.1. Caso de uso: Detección de Ataques DDoS

El gran aumento de ataques de denegación de servicio distribuido (DDoS) registrados en los últimos años ha advertido a las principales organizaciones de seguridad [2]. Un ejemplo significativo de este problema se observó en octubre de 2016, cuando los servidores DNS del proveedor Dyn registraron una de las campañas DDoS más complejas y mediáticas [3]. Su consecuencia fue la desactivación de docenas de servicios, páginas web y redes sociales, algunas de ellas relacionadas con soluciones de gran difusión, por ejemplo, Twitter, Reddit, Github, Amazon o Spotify. Esto se logró explotando una vulnerabilidad presente en millones de dispositivos de diferente naturaleza conectados al Internet de las Cosas o IoT (del inglés Internet of Things)[4]. La amenaza fue orquestada desde una *botnet* gestionada por el malware *Mirai* [5][6], y el ataque sirvió para agravar la incertidumbre de muchos usuarios sobre la seguridad de sus dispositivos de red. Como resultado de este incidente o ataques similares, los usuarios se preguntaron: ¿mis dispositivos forman parte de campañas maliciosas coordinadas? en este caso, ¿cuáles son sus propósitos? ¿En qué medida están contribuyendo? o, ¿cómo puedo evitar tales situaciones? Pero a pesar de la importancia de combatir estas amenazas mediante el análisis del tráfico entrante/saliente de los dis-

positivos protegidos ha sido estudiado por la comunidad investigadora desde el punto de vista de las redes de comunicación[7][8], cuyos esfuerzos generalmente apuntaban a analizar el tráfico de red en los extremos intermedios/víctimas de la intrusión, o en la identificación de dispositivos comprometidos por malware de control remoto [9]. Prácticamente no se ha realizado a partir de datos generados desde el origen de la amenaza, estudiando cada dispositivo individualmente.

Con el fin de contribuir al desarrollo de soluciones capaces de hacer frente a los problemas antes mencionados, la estrategia de predicción desarrollada y sus capacidades de proyectar el estado de la red, ha sido instanciada para reconocer anomalías en el tráfico que fluye a través de ellas. En concreto, la solución planteada aborda el desafío de analizar los flujos de tráfico en busca de rasgos de actividades maliciosas, en particular los relacionados con la participación de un dispositivo como origen de ataques DDoS. El descubrimiento de actividades sospechosas se centra en la estimación del comportamiento del tráfico monitorizado basado en el estudio de métricas agregadas y la elaboración de intervalos de predicción. Cuando la observación excede los umbrales que delimitan las actividades normales y legítimas, la discordancia se etiqueta como anómala y se informa de una situación sospechosa. Con fines experimentales, la instanciación del esquema de predicción propuesto dió lugar a la herramienta DroidSentinel, originalmente creada como caso de uso específico para sistemas Android (de ahí el nombre). Este método es escalable para tecnologías IoT alternativas cuando se adopta la implementación adecuada. La primera implementación de DroidSentinel planteó una solución portátil, donde todo el análisis se realizaba en el dispositivo, lo que permitía a los usuarios instalar y ejecutar una aplicación defensiva que ejecutaba cada etapa de procesamiento de datos [10]. Pero debido a la heterogeneidad y no estacionariedades inherentes a la salida de tráfico de un único dispositivo móvil, que generalmente depende del comportamiento del usuario, los procesos analíticos se adaptaron a los cambios en la distribución de los datos monitorizados, de esta manera ganando sofisticación. Nótese que, a pesar de su eficacia, estas modificaciones implicaron importantes penalizaciones en términos de calidad de la experiencia del usuario (CPU, memoria y consumo de batería).

1.2. Objetivos

El principal objetivo de este trabajo es desarrollar una estrategia de predicción para la proyección del estado de redes de comunicaciones de nueva generación,

adaptada a su gran heterogeneidad de fuentes de información y a los procesos no estacionarios inherentes a los datos que gestionan. Además, debe de ser capaz de detectar, en base a predicciones adaptativas, anomalías en el tráfico de red. Con este propósito se analizan los flujos de tráfico generando series temporales. El elevado tráfico de red condiciona al sistema a ser lo más eficiente posible con la finalidad de funcionar en tiempo real. Esto ha dado lugar al desarrollo de un modelo capaz de decidir de manera inteligente cuál será el mejor algoritmo de predicción de la batería disponible, lo que reduce considerablemente el tiempo de procesamiento y mejora su efectividad.

Cabe destacar que, dado el gran crecimiento y notoriedad de los nuevos escenarios y tecnologías de comunicación, se pretende crear un marco de predicción compatible con estos mismos y capaz de adaptarse a las series temporales, tanto estacionarios como no estacionarios, inferidas a partir de los flujos de tráfico de red.

Además, con el objetivo de integrar el sistema en las redes de quinta generación, su instanciación se ha separado en dos casos de uso. El primero de ellos es su despliegue en una infraestructura adaptada a 5G. Para ello, se ha colaborado con el proyecto SELFNET (H2020-ICT-2014-2/671672) y el grupo de investigación GASS de la Universidad Complutense de Madrid, habiéndose evaluado a partir del estándar funcional M3-Competition. En segundo lugar, se ha instanciado como solución a un problema real, que es la defensa frente a los ataques de denegación de servicio distribuidos. Esto último a dado lugar a la propuesta DroidSentinel, capaz de reconocer este tipo de intrusiones en los extremos origen del ataque.

1.3. Organización del proyecto

El presente documento se divide en ocho capítulos, en los cuales se explica el contenido de la investigación, la experimentación realizada, así como los resultados obtenidos durante su transcurso:

En el Capítulo 2 se hace una revisión del estado del arte relacionado con las nuevas redes de comunicación 5G, las tecnologías que las sustentan, y el proyecto europeo SELFNET.

En el Capítulo 3 se revisan el estado del arte acerca de los ataques de denegación

de servicio distribuidos, su evolución, y las diferentes contramedidas planteadas por la comunidad investigadora para su mitigación.

En el Capítulo 4 se propone un marco para la predicción adaptativa orientado al estudio de indicadores del estado de la red en escenarios de comunicación emergentes.

En el Capítulo 5 se introduce la herramienta DroidSentinel, un caso de uso de la estrategia de predicción propuesta, que la instancia para la detección de rasgos de actividades maliciosas en el tráfico de red. A partir del análisis realizado es posible reconocer la implicación de dispositivos finales en ataque de DDoS orquestados de manera remota.

En el Capítulo 6 se detallan la metodología de evaluación adoptada, las colecciones de muestras analizadas y las diferentes pruebas realizadas.

En el Capítulo 7 se presentan y se discuten los resultados obtenidos.

Finalmente, en el Capítulo 8 se lleva a cabo una breve reflexión sobre el esfuerzo realizado y sus diferentes líneas de trabajo futuro.

Introduction

In recent years, the information society prompted the need for creating, distributing and manipulating more and more data, which plays an essential role in virtually all areas of people's lives, ranging from aspects related to culture and leisure, to economy, health or provisioning fundamental resources like water or energy. At the current communication landscape, the mobile technologies become one of the principal pillars that sustain the information society, which advanced up to a fourth generation (commonly known as 4G) of technological solutions for wireless communication between devices. But despite the fact that this set of tools were available and enhanced for almost a decade, the increase in the demand for information has motivated the development of a new generation (5G), focused on significantly improving the Key Performance Indicators (KPI) of their predecessors. They include, among others, important improvements in terms of Quality of Experience (QoE), efficiency, mobility, connectivity or resource consumption. The difficult challenges that reaching these requirements poses, have encouraged the emergence of the so-called emerging communication technologies, like Software-Defined Networking (SDN), Network Function Virtualization (NFV), Cloud Computing, Self-Organized Networks (SON), Big Data analytics or their adaptability through deep learning processes. They take part of a heterogeneous and sophisticated ecosystem that makes difficult to manage and process data, being characterized by complexity and high presence of non-stationary processes. Consequently, many of the conventional analytical strategies must be adapted to the characteristics inherent in the forthcoming sources of information.

In order to contribute to their design and development, the performed work has addressed the problem of providing knowledge inference capabilities through projection of the current state of the network. This has required an in-depth review of the main prediction methods adapted to the analysis of network traffic in the bibliography, as well as the definition of an expert system capable of making proactive decisions based on network situations or the effectiveness of the decisions

previously made as feedback, thus implying to carry out tasks such as the decision of the best algorithm depending on the context, calibration of its adjustment parameters, establishment of prediction thresholds or analysis of the obtained data. The performed work was framed by the European project SELFNET - "Framework for Self-Organized Network Management in Virtualized and Software Defined Networks"(Call: H2020-ICT-2014-2 for Research and Innovation Action (RIA), with Proposal Number: 671672) [1], being integrated as part of the analytical solution framed in its task "T4.3 Analyzer Module". Once its development concluded, and after testing its effectiveness based on the functional evaluation standard proposed at the M3-Competition for prediction strategies, it has been deployed on a real use case, which choice was motivated according to the facts described in the next subsection:

Use case: Detection of DDoS threats

The alarming increase in the number of Distributed Denial of Service (DDoS) attacks recorded in the last years has warned the different organizations for cyber-defense [2]. A clear example of this problem was observed in October 2016, when the DNS servers of the Dyn provider suffered one of the most complex and mediatic DDoS campaigns [3] ever seen. Its consequence was the deactivation of dozens of services, websites and social networks, some of them related to widely distributed and popular products, among them Twitter, Reddit, Github, Amazon or Spotify. This was achieved by exploiting a vulnerability present in millions of devices of different nature connected to the Internet of Things (IoT) [4]. The threat was orchestrated from a botnet managed by the malware Mirai [5][6], and the attack served to aggravate the uncertainty of many users about the security of their data and network devices. As a result of this incident or similar attacks, many users asked themselves: are my devices taking part of remotely coordinated malicious campaigns?. In this case, what are their purposes?; to what extent are they contributing?; or, how can I avoid such situations?. But despite the fact that to combat the attacks by analyzing the incoming/outgoing traffic of the protected devices has been widely studied by the research community [7] [8], (which efforts generally aimed on analyzing network traffic at the intermediate/victims edges of the intrusion, or by identifying devices compromised by remote control malware [9], there has practically not been conducted from studying data generated from the origin of the threat (source-side), in this way studying the end-points separately.

In order to contribute to the development of solutions capable of dealing with the aforementioned problems, the prediction strategy developed, as well as its capabilities to project the state of the network, have been instantiated for anomaly recognition. In particular, the proposed solution addresses the challenge of analyzing traffic flows looking for traits of malicious activities, in particular those that may be related to the participation of the end-point as sources of DDoS attacks. The discovery of suspicious activities lies on estimating monitored traffic behaviors based on studying aggregated metrics and the construction of prediction thresholds. When the observation exceeds the thresholds that delimits normal and legitimate activities, the mismatch is labeled as discordant (suspicious), being properly reported. With experimental purpose, the instantiation of the proposal became the DroidSentinel monitoring tool, originally developed for Android systems (hence the name). However, the solution was scalable for alternative IoT technologies when properly configured. The first implementation of DroidSentinel raised a portable solution, where all the analytics were executed on the device, thus allowing users to install and run a defensive application that performed each data processing stage locally [10]. But due to the heterogeneity and non-stationarity inherent in the traffic output of a single mobile device, which generally highly depends on the user behavior, the analytical processes were adapted to the changes in the distribution of the monitored data, thus gaining sophistication. Note that, in spite of their effectiveness, these modifications implied important penalties in terms of quality of the user experience (CPU, memory and battery consumption).

Objetives

The principal objective of this work is to develop a prediction strategy for projecting the state of new generation communications networks, which must be adapted to their great heterogeneity of data sources and to the non-stationary processes inherent in the information they manage. In addition, it must be able to detect anomalies on network traffic based on adaptive predictions. For this purpose, traffic flows are analyzed, which metrics facilitated the definition of time series. In order to operate in real time, the huge network traffic volume conditions the proposal to be as efficient as possible. This has led to the development of a model capable of intelligently deciding which could be the best available battery prediction algorithm, that considerably reduces processing time and improves effectiveness. It should be noted that, given the great growth and notoriety of the new communication scenarios and technologies, it is intended to create a prediction framework compatible

with them able to adapt to time series inferred from the network traffic flows, both with stationary and non-stationary nature.

On the other hand, with the aim of integrating the system into the fifth-generation networks, its instantiation has been separated into two use cases. The first one is its deployment on a real infrastructure adapted to 5G. For this, we have collaborated with the SELFNET project (H2020-ICT-2014-2 / 671672) and the GASS research group of the Complutense University of Madrid; where the proposal was evaluated based on the M3-Competition functional standard. Secondly, it has been instantiated as a solution to a real security challenge, in particular, the defense against distributed denial of service attacks. This last action gave rise to the approach DroidSentinel, being able to recognize this type of intrusions at source-side.

Project organization

This document is divided into eight chapters, where the performed work and experimentation are described in-depth:

In Chapter 1 the introduction, objectives and organization of the document are detailed.

In Chapter 2 the state of the art related with the emergent communication technologies and the 5G landscape are reviewed, emphasizing their role and the main features of the SELFNET project.

In Chapter 3 the problems and challenges posed by the DDoS threat are explained, where its evolution and main defensive solutions are detailed.

In Chapter 4 a framework for adaptive prediction based on studying network indicators is proposed.

In Chapter 5 the DroidSentinel tool is introduced, a use case of the prediction framework for discovering malicious traits on network traffic. From them it is possible to recognize end-points related with DDoS attacks remotely orchestrated.

In Chapter 3 the problems and challenges posed by the DDoS threat are explained, where its evolution and main defensive solutions are detailed.

In Chapter 4 a framework for adaptive prediction based on studying network indicators is proposed.

In Chapter 5 the DroidSentinel tool is introduced, a use case of the prediction framework for discovering malicious traits on network traffic. From them it is possible to recognize end-points related with DDoS attacks remotely orchestrated.

In Chapter 6 the applied methodology and the dataset involved in the experimental validation of the proposals are described.

In Chapter 7 the obtained results are discussed.

Finally, in Chapter 8 the conclusions and future work are presented.

Capítulo 2

Escenarios de comunicación emergentes

En la actualidad, las redes móviles son conjuntos complejos de equipos heterogéneos que utilizan aplicaciones de administración patentadas, lo que supone un gran gasto y esfuerzo, así como un proceso lento para gestionar todos los elementos de la red mediante enfoques actualmente manuales o semiautomáticos. Con el crecimiento de las nuevas tecnologías, tales como las redes definidas por software (SDN, del inglés *Software Defined Network*), la virtualización de funciones de red (NFV, del inglés *Network Function Virtualization*) y la computación en la nube, nos acercamos cada vez más al concepto de redes autónomas las cuales pueden programarse y son fácilmente administrables.

En este capítulo se presentan las capacidades de un nuevo marco propuesto por el proyecto SELFNET que permite desplegar funcionalidades de administración autónomas en una red 5G. El trabajo propuesto se centra en el caso de uso de autoprotección, que se puede aplicar para tratar de forma reactiva o preventiva las anomalías de red detectadas o predichas. SELFNET puede proporcionar un marco de gestión autónomo que reduce notablemente los costes operativos y mejora sustancialmente la calidad de experiencia (QoE, del inglés *Quality-of-Experience*) en términos de fiabilidad, disponibilidad, continuidad del servicio y seguridad.

2.1. Redes de telefonía móvil de quinta generación

Hoy en día, los operadores de red se encuentran con problemas, tales como, fallos de enlace, ataques de seguridad, degradación de calidad de servicio (QoS), errores de software y fallos hardware, y deben resolverlos de forma manual o semiautomática. La solución de estos problemas generalmente requiere la reconfiguración manual de los equipos y, en algunos casos, la instalación de nuevos sistemas y funcionalidades como enrutadores, traductores de direcciones de red, cortafuegos y balanceadores de carga, que no pueden ejecutarse sin interrumpir el funcionamiento normal de la red. Lo que supone interrupciones en los servicios y violaciones en los acuerdos a nivel de servicio, además de incurrir en mayores costes operacionales y de capital y comprometer la calidad de experiencia (QoE) de los usuarios finales [11]. Es por eso que el gasto operativo (OPEX) de los operadores móviles es actualmente tres veces mayor que el gasto de capital [10].

La evolución de las generaciones de redes móviles está influenciada principalmente por el crecimiento exponencial del uso de dispositivos inalámbricos, el uso de datos y la necesidad de una mejor calidad de experiencia (QoE). Se espera que más de cincuenta mil millones de dispositivos conectados utilicen los servicios de red 5G para fines del año 2020, lo que supondría un gran incremento en el tráfico de datos en comparación con el año 2014. Sin embargo, las soluciones de vanguardia no son suficientes para los desafíos mencionados anteriormente. En resumen, el aumento de 3D (Dispositivos, Datos y tasa de transferencia de Datos) fomenta el desarrollo de redes 5G.

Específicamente, la quinta generación (5G) de las redes móviles destacará y abordará tres vistas amplias:

- **Centrada en el usuario:** Para mejorar la experiencia de usuario, las redes de quinta generación tienen que poder proveer a los usuarios de servicio ininterrumpido y sin problemas o caídas.

- **Centrado en el proveedor de servicios:** Se espera recibir una gran demanda de diferentes servicios, por lo tanto, la red debe estar preparada para balancear la carga o ajustarse, de manera que no colapse.

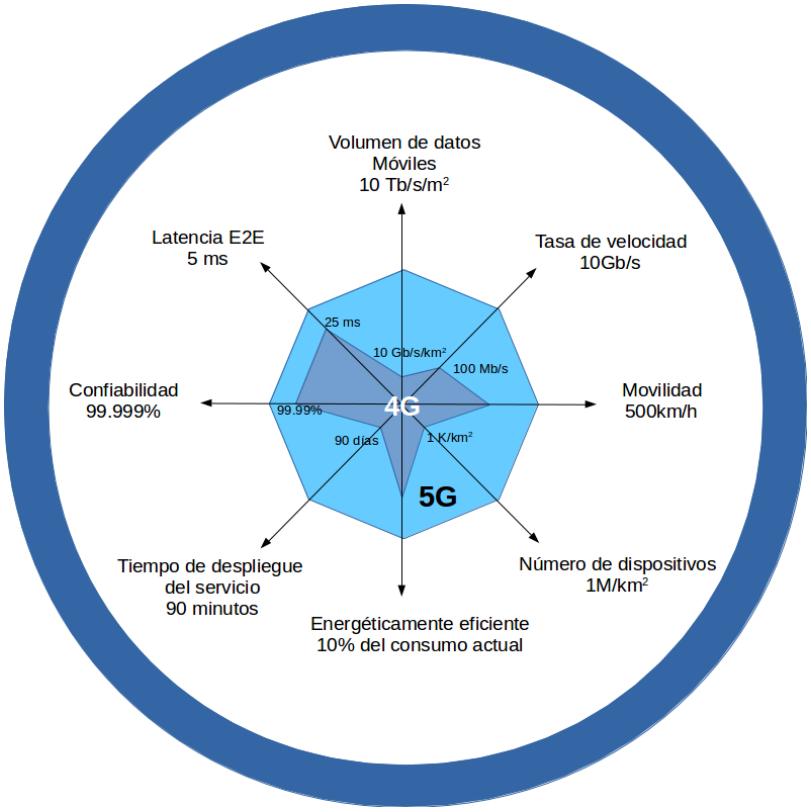


Figura 2.1: Indicadores clave de desempeño 5G

- **Operador de red:** Se intentará dar el mejor servicio a las operadoras de red, proporcionando un servicio de bajo costo y energéticamente eficiente, mejorando su escalable e infraestructura.

2.1.1. Indicadores clave de desempeño

Los parámetros que se enumeran a continuación y que se pueden ver resumidos en la Fig.2.1, son los factores de desempeño que caracterizarán a las redes 5G a fin de cumplir los requerimientos establecidos por esta tecnología. Estos se detallan a continuación:

- **Tasa de transferencia (10 Gbps).** A diferencia de las plataformas LTE predecesoras, se espera que las redes 5G alcancen tasas de transferencia de hasta 10Gbps para un acceso ubicuo a una amplia variedad de servicios [12].

- **Movilidad (500Km/h).** Este indicador de desempeño permite que las redes 5G mantengan sus niveles de operatividad incluso en condiciones de movilidad

de hasta 500Km/h que pueden encontrarse, por ejemplo, en trenes de alta velocidad.

- **Número de dispositivos (1M/Km2).** Las redes 5G estarán capacitadas para albergar una cantidad de dispositivos conectados hasta mil veces superior a la ofrecida por 4G. Esto la constituye en una plataforma idónea para el despliegue de soluciones IoT, redes de sensores, entre otras [13].

- **Eficiencia energética (10 % de la actual).** Uno de los principales objetivos de 5G es reducir significativamente el consumo energético en hasta un 90 %. El consumo energético se adaptará así a las fluctuaciones de tráfico y nuevos despliegues de infraestructura de menor consumo.

- **Tiempo de despliegue de servicios (90 minutos).** Se espera que el tiempo estimado de despliegue de nuevos servicios se reduzca hasta alcanzar los 90 minutos. Esto es posible a través de las tecnologías de virtualización y provisión de recursos bajo demanda [14].

- **Fiabilidad (99.999 %).** La infraestructura 5G estará capacitada para ofrecer una fiabilidad promedio de 99.999 %, alcanzando tiempos mínimos de inactividad.

- **Latencia (5ms).** Se espera contar con niveles de latencia de extremo a extremo (E2E) inferiores a los 5ms. Este nivel de mínimo retardo en las comunicaciones 5G es también referido como "cero latencia". Este factor introduce mejoras significativas de la calidad percibida por el usuario, y permite también desplegar infraestructuras críticas más eficientes.

- **Volumen de datos móviles (10Tb/s/Km2).** La alta densidad y concentración de dispositivos conectados por área geográfica generará volúmenes de datos móviles del orden de 10Tb/s/Km2. Se estima por lo tanto que millones de usuarios puedan interactuar incluso en situaciones de máxima demanda conservando los requisitos operacionales establecidos por 5G.

2.1.2. Tecnologías relacionadas

Con la aparición de los nuevos escenarios de comunicación, han tomado relevancia nuevas tecnologías que han posibilitado la creación de nuevos paradigmas de gestión adaptados a las redes emergentes. Estos se caracterizan por un alto grado de agilidad, dependencia mínima del hardware y capacidades de gestión automática. En esta

sección se hace una breve revisión algunas las tecnologías que han establecido las bases para el diseño de las plataformas de red 5G [14].

SDN

Las redes definidas por software (SDN, del inglés *Software Defined Networking*) presentan un nuevo paradigma en el diseño y administración de los elementos de red, al separar los planos de datos y control de los dispositivos de comunicación (routers, switches, etc.). En una red SDN se identifican entonces las siguientes capas:

- **Plano de datos,** en esta capa se procesan los paquetes de red a través de conmutadores. Estos dispositivos son administrados desde el controlador SDN, para lo cual es necesario que estos cuenten con una interfaz de configuración, comúnmente implementada por el protocolo *OpenFlow*.

- **Plano de control,** se encarga de gestionar los elementos de la red en función de sus necesidades a través de los controladores SDN. En el nivel inferior de este plano (*SouthBound interface*), el controlador configura las tablas de encaminamiento de los dispositivos *OpenFlow*, mientras que, en el nivel superior, el controlador implementa una interfaz de alto nivel (*NorthBound interface*) que posibilita la interacción de esta capa con aplicaciones que configuran el comportamiento de la red.

- **Capa de aplicación,** en esta capa se encuentra el software capaz de automatizar tareas de configuración, despliegue de servicios y gestión del tráfico de red. Las aplicaciones en esta capa se implementan en lenguajes de programación de alto nivel, según los requisitos del controlador SDN implementado.

SDN implementa las prácticas de ingeniería de software, dando lugar así al concepto de redes programables. Esto supone un cambio de paradigma importante que ha abierto un nuevo modelo de prestación de servicios de red para los operadores, y ha sido una característica fundamental para establecer los fundamentos de las redes 5G.

NFV

La virtualización de funciones de red (NFV, del inglés *Network Function Virtualization*) ofrece una nueva forma de diseñar, implementar y administrar servicios de red.

NFV implementa funciones de red, comúnmente soportadas sobre hardware de comunicaciones, en aplicaciones de software cuya instanciación se produce bajo demanda según las necesidades del proveedor de servicios. Estas funciones comprenden la traducción de direcciones de red (NAT), cortafuegos (*firewall*), detección de intrusos (IDS), servicio de nombres de dominio (DNS), almacenamiento en caché, entre otros.

Las tecnologías de virtualización han facilitado el surgimiento de esta tecnología. La creación de plataformas de computación en la nube (*cloud computing*) ha fortalecido aún más las capacidades para el despliegue automático de estas funciones.

En el año 2012, un grupo de operadores miembros del Instituto de Estándares de Telecomunicaciones de Europa (ETSI), propuso un marco para la gestión de funciones de red virtuales, en la que se identifican los siguientes módulos principales:

- *Network Functions Virtualized Infrastructure* **(NFVI):** Esta capa abstrae los componentes de hardware de la infraestructura de red, categorizándolos en elementos de cómputo, red y almacenamiento necesarios para instanciar funciones de red.

- *Virtualized Network Function* **(VNF):** consiste en una función de red implementada en software que se despliega sobre la plataforma NFVI.

- NFV *Management and Orchestration* (NFV M & O): Esta capa gestiona y orquesta el ciclo de vida de los recursos hardware y/o software, que soportan la virtualización de la infraestructura y de los módulos VNF.

Aunque SDN y NFV son tecnologías que pueden implementarse independientemente, su despliegue conjunto ha posibilitado la provisión de servicios de red más eficientes.

Cloud Computing

La computación en la nube consiste en un modelo para la provisión automática y bajo demanda de recursos de cómputo que son accesibles para el usuario desde cualquier lugar. Los proveedores de este tipo de servicios ponen a disposición de los clientes herramientas y aplicaciones web para facilitar la administración de los recursos administrados.

La computación en la nube otorga la capacidad de compartir recursos físicos (cómputo, red y almacenamiento) por múltiples usuarios de manera transparente. Las plataformas de virtualización son utilizadas para crear entornos aislados que los proveedores ofrecen a los usuarios. En dichos entornos los usuarios pueden administrar los recursos cómputo, red o almacenamiento contratados; con independencia de las tareas de administración llevadas a cabo en otros entornos. Este paradigma se conoce comúnmente como *multi-tenancy*, y es un aspecto distintivo en las plataformas de computación en la nube [15].

Desde el punto de vista de los servicios ofrecidos, la computación en la nube introduce tres conceptos ampliamente extendidos en la literatura. Cada uno de ellos está relacionado con un nivel de la arquitectura, y son los que se enumeran a contracción:

- **Infraestructura como servicio** (IaaS, del inglés *Infrastructure as a Service*). Los recursos de infraestructura, comúnmente alojados en centros de datos, son abstraídos por la capa de virtualización y provistos a los usuarios bajo demanda. Sobre la infraestructura asignada el usuario puede desplegar los sistemas operativos, software de aplicación o configuración de red preferida que se adapte a sus requerimientos.
- **Plataforma como servicio** (PaaS, del inglés *Platform as a Service*). En este nivel de la arquitectura, el proveedor pone a disposición de los usuarios marcos para el desarrollo y ejecución de aplicaciones adaptadas a los entornos en la nube.
- **Software como servicio** (SaaS, del inglés *Software as a Service*). Es el nivel más alto de la arquitectura y con el que los usuarios finales interactúan directamente. En este modelo sólo se necesita un navegador web o un cliente ligero para acceder a la aplicación. Google Docs es un ejemplo de este tipo de servicios.

Desde el punto de vista de las redes 5G, las plataformas en la nube facilitan el despliegue de servicios en menor tiempo, contribuyendo de esta manera a alcanzar uno de los indicadores de desempeño claves establecidos en 5G.

Inteligencia artificial

La inteligencia artificial se concibe como una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos en base a dos de sus características primordiales: el

razonamiento y la conducta [16].

La inteligencia artificial ha tenido un gran auge en los últimos tiempos y cada vez se aplica en más campos. En el caso de los nuevos escenarios de comunicación tiene diversos usos tales como: redes neuronales capaces de aprender y tomar decisiones de manera autónoma, clasificadores capaces de seleccionar la mejor contramedida en caso de un ataque, algoritmos de predicción para prever el flujo de tráfico de la red, y otras aplicaciones.

Redes autoorganizadas

Las funciones autoorganizadas (del inglés *Self Organized Networks*) son responsables de administrar automáticamente una plataforma de red. Este modelo se enfoca en el logro de tres objetivos principales: la autoconfiguración, autooptimización y autocuración de la red. Este modelo facilita el logro de altos niveles de rendimiento de la red al identificar continuamente situaciones de mejora [17]. Esto se hace mediante el despliegue de sensores que se encargan de recopilar datos de rendimiento de la red. Asimismo, se ponen en marcha diversos actuadores que ejecutan modificaciones sobre los parámetros de red existentes.

Con un modelo autoorganizado, las operaciones de administración se pueden realizar de manera eficiente debido a la disponibilidad de modelos estadísticos formados por gran cantidad de indicadores clave de rendimiento (KPI) y sus dependencias entre sí.

2.2. SELFNET

SELFNET es un proyecto financiado por la Comisión Europea, parte del programa Horizonte 2020, que proporciona un sistema de gestión de red 5G escalable, extensible e inteligente mediante la integración de nuevas tecnologías (SDN, NFV, computación en la nube, inteligencia artificial, etc.). Por lo tanto, SELFNET tiene como objetivo ayudar a los gestores de una red 5G a realizar tareas de administración como la instanciación automática de aplicaciones SDN/NFV con el propósito de supervisar y mantener la red de manera autónoma. Es necesario para ello definir medidas tácticas de alto nivel y habilitar acciones preventivas y correctivas de manera automática para mitigar fallos de red existentes o potenciales [17].

2.2.1. Arquitectura de SELFNET

Para una mayor comprensión de la arquitectura SELFNET, se expone a continuación una breve descripción de los niveles funcionales diferenciados, los cuales se muestran en la Fig.2.2. Estas son: capa de infraestructura, capa de red virtualizada, capa de control SON, capa autónoma SON, capa de interfaz SON y un módulo de orquestación y gestión NFV [17] [18].

- **Capa de Infraestructura:** Esta capa provee los recursos necesarios para la instanciación de funciones virtuales y soporta los mecanismos necesarios para ese fin. Esta a su vez se subdivide en dos subcapas: subcapa física y subcapa de virtualización. La subcapa física incluye los recursos físicos requeridos para proveer capacidades computacionales, de red y de almacenamiento sobre hardware, y la subcapa de virtualización posibilita la compartición de los recursos disponibles entre distintos usuarios o servicios.

- **Capa de Datos SON:** En esta capa, las distintas funciones de red son situadas e interconectadas bajo una topología diseñada. Las funciones de red virtuales (VNF) incluyen las instancias requeridas para la normal operatividad de la infraestructura virtual, así como aquellas creadas por SELFNET como parte de la implementación SON. Esta capa también proporciona soporte *multi-tenancy* que posibilita la compartición de recursos entre distintos usuarios, cada uno de los cuales es capaz de administrar su propio dominio.

- **Capa de Control SON:** La capa de control SON está formada por sensores y actuadores orientados a desempeñar funciones de autoorganización. Por un lado, los sensores recopilan información de los servicios que se ejecutan en la infraestructura de red. Por otro lado, los actuadores aplicarán las acciones pertinentes sobre la red.

- **Los sensores y actuadores SON** se administran desde la Capa Autonómica SON, la cual otorga inteligencia a la red. De forma similar, esta capa interactúa con el plano de control en arquitecturas SDN.

- **Capa Autónoma SON:** La Capa Autónoma SON proporciona la inteligencia de la red. La información recolectada por los sensores es usada para diagnosticar el estado de la red. Luego, las acciones para alcanzar los objetivos del sistema desde la perspectiva de la gestión autonómica. En primer lugar, el módulo de monitorización y análisis recoge y almacena todos los datos procedentes de los sensores, constituyéndose en el primer nivel de recolección de

datos. A continuación, algoritmos de extracción de datos, reconocimiento de patrones, predicción, entre otros, permiten un análisis detallado de la información recibida. En este punto, se realiza el reconocimiento de comportamientos anómalos o sospechosos basados en las métricas definidas. Los resultados del diagnóstico obtenido sirven para decidir acciones sobre los problemas de red existentes y/o potenciales tanto de manera reactiva como proactiva.

- **Orquestador NFV y Capa de Gestión:** Esta capa es responsable de controlar y concatenación las distintas funciones de red en la infraestructura virtualizada. Las decisiones tomadas por la capa autonómica permiten al Orquestador la administración y configuración de las aplicaciones SDN y NFV. Este componente resuelve la dependencia, el orden de ejecución y la prioridad de diferentes acciones y asegura que las aplicaciones SDN / NFV dispongan de recursos suficientes para realizar sus tareas.

- **Capa de Acceso SON:** Esta capa proporciona una interfaz que proporciona diferentes capacidades de monitorización y gestión de la configuración al operador de red, dependiendo de los niveles de privilegios asignados. De esta forma, los administradores pueden comprobar el estado actual de las operaciones en SELFNET.

2.2.2. Casos de uso

SELFNET se centrará principalmente en tres aspectos en lo referente a la administración de red: proporcionar capacidades de autoprotección contra ataques de red distribuidos, funcionalidades de autorreparación en caso de fallos de red detectados o previstos, y características de autooptimización para mejorar dinámicamente el rendimiento de la red y la calidad de experiencia (QoE) de los usuarios. Estas capacidades suponen un avance significativo para los operadores de red porque supone la implementación de servicios *multi-tenancy* distribuidos en las redes 5G y generan nuevas oportunidades de negocio para los proveedores de servicios. Los tres casos de uso se explican a continuación:

Self-protection Use Case

El objetivo principal de este caso de uso es detectar y mitigar ciberataques y amenazas de red. Para ello se utilizarán tecnologías emergentes como sensores y actuadores VNF (monitores virtuales de tráfico, *honeypots*, etc) desplegadas en

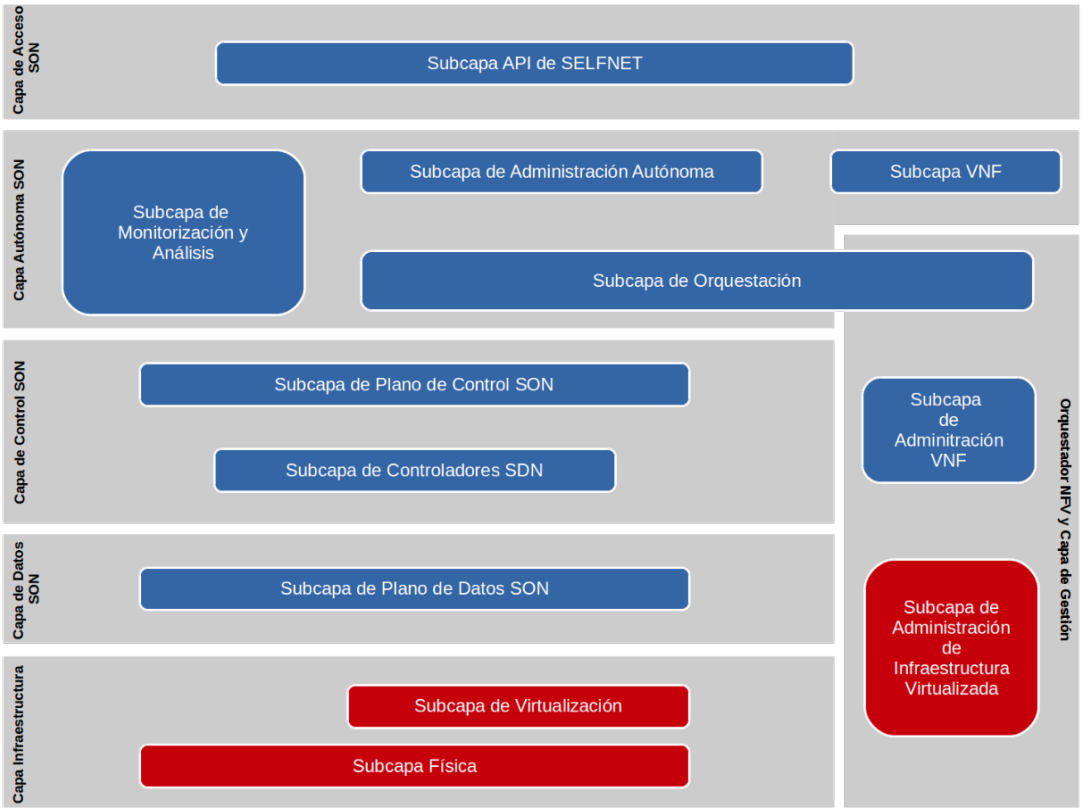


Figura 2.2: Arquitectura SELFNET

diferentes puntos de la red, por ejemplo, en el acceso móvil o puntos de acceso de la red [17].

Self-healing Use Case

Este caso de uso consiste en detectar y predecir fallos en la infraestructura de la red 5G tanto a nivel de hardware, software, vulnerabilidades de la infraestructura, interrupciones en el suministro de energía, etc. para aplicar acciones de recuperación reactiva y preventiva. El sistema consiste en un analizar de red con función de autorreparación capaz de inferir las métricas indicativas del estado de la red para detectar problemas potenciales en el sistema. Además, debe tener la capacidad de tomar decisiones para autorepararse [15].

Self-optimisation Use Case

Este caso de uso se centra en crear un sistema autónomo capaz de adaptarse a las necesidades de los usuarios manteniendo así su calidad de usuario esperada. Para ello SELFNET dispone de sistemas de monitorización y análisis para observar o predecir cargas masivas de tráfico que alertarán a unos mecanismos de gestión autoajustables que serán capaces de gestionar debidamente este tráfico reduciendo los retrasos en la red.

Capítulo 3

Denegación del servicio

Este capítulo aborda la problemática relacionada con la prevención, detección, mitigación e identificación del origen de los ataques de Denegación de Servicio en los escenarios de red actuales. Con este fin se profundizará en mayor medida en el subgrupo de los ataques distribuidos basados en inundación, por ser la amenaza por identificar en el caso de uso de la estrategia de predicción propuesta.

3.1. Ataques de Denegación de Servicio

Los ataques de Denegación de Servicio o DoS (del inglés *Denial of Service*) constituyen una amenaza creciente en los últimos años, habiendo llegado a convertirse en un auténtico desafío en el área de la seguridad en sistemas de la información. Entre sus objetivos se encuentran empresas, organismos gubernamentales, bancos, ejércitos o servicios como universidades, hospitales y aeropuertos.

Los ataques de denegación de servicio consisten en bloquear los servicios de red a los usuarios legítimos. Se han convertido en un gran problema actual en lo referente a la seguridad de Internet de acuerdo con el informe publicado en 2016 por el Instituto Ponemon. Este tipo de ataques representa la primera causa de pérdidas en las empresas de Reino Unido y la segunda en Estados Unidos dentro del ámbito del cibercrimen [19].

Los ataques DoS más efectivos y utilizados son de naturaleza distribuida y se conocen como DDoS (del inglés *Distributed Denial of Service*). En este caso, el atacante hace uso de servidores maestros (del inglés *botmaster*) que controlan a esclavos o *zombies* previamente infectados con malware, para entre otros, atacar de manera coordinada a sus víctimas. Este esquema se muestra en la Fig.3.1.

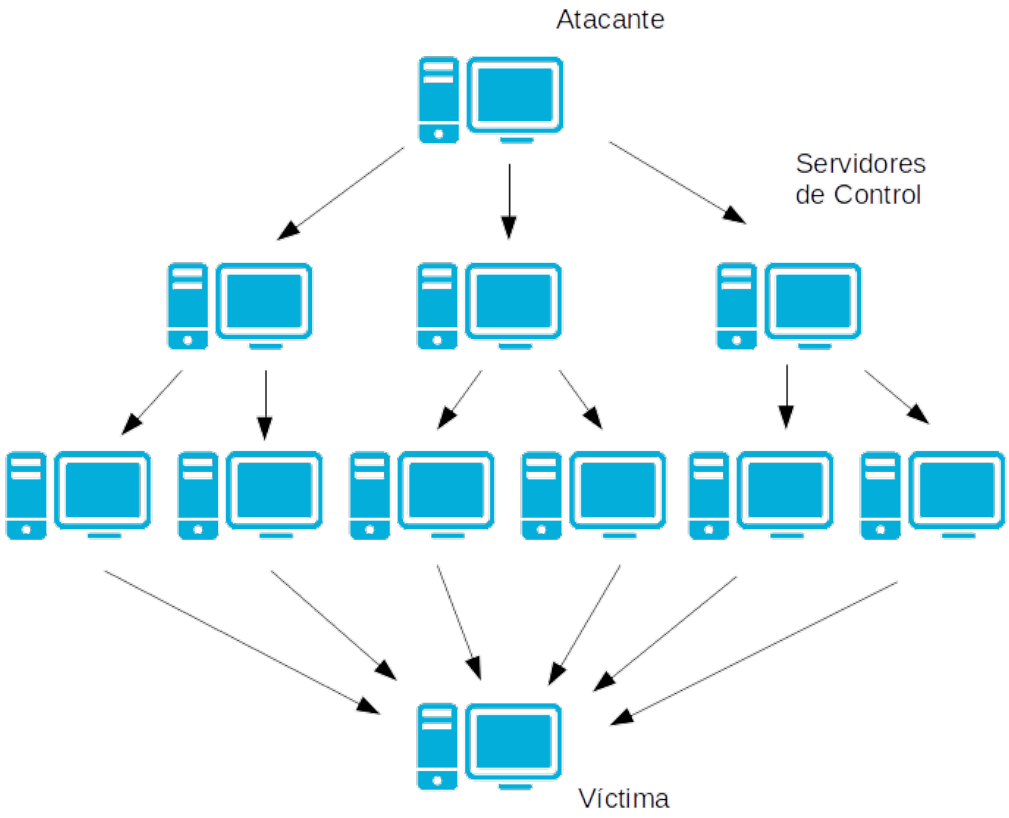


Figura 3.1: Escenario ataque DDoS

3.2. Motivaciones

Los ataques DDoS pueden estar motivados por causas muy diversas, aunque principalmente se deben a los siguientes motivos: económicos (la víctima se encuentra ante la disyuntiva de pagar o tener sus servidores colapsados), como distracción para otro ataque mayor, o circunstancias políticas. El gran aumento de ataques DDoS registrado en los últimos años se debe, entre otros factores, a la facilidad para conseguir las herramientas, los escasos conocimientos necesarios para poder hacer un ataque de este tipo, la ausencia de mecanismos efectivos de defensa y la aparición de individuos o grupos que venden sus servicios (*botnets*, servidores infectados, asesoramiento, etc.) para facilitar la labor al atacante y poner a su alcance todas las herramientas necesarias [21]. Por este motivo se prevé que se habrán registrado más de 10 millones de ataques DDoS al concluir el año 2018, con un tamaño medio de 1.5Gb (según las expectativas publicadas a finales del 2017 por la empresa Deloitte, en su informe "*Global Technology, Media and Telecommunications Predictions* 2017"). Por lo que tal y como ha señalado la Comisión Europea, cada

vez resulta más importante encontrar herramientas que permitan a sus ciudadanos y sus diferentes organizaciones, defenderse frente a estos ataques.

3.3. Ataques de Denegación de servicio Distribuidos

Con el fin de maximizar su efecto, los ataques DoS han evolucionado a una técnica mucho más eficaz: los ataques de Denegación de Servicio Distribuidos o DDoS (del *inglés Distributed Denial of Service*). Los ataques DDoS se definen como: "ataques que tienen como objetivo el agotamiento de los recursos críticos de un sistema/red y que provienen de fuentes múltiples distribuidas a lo largo de ella" [22]. Esto permite el envío de grandes cantidades de información (actualmente, en el orden de Gbs/Tbs) y presenta una serie de ventajas frente a los ataques DoS convencionales, entre las que destacan:

- **Origen múltiple:** La distribución del origen del ataque dificulta su identificación.
- **Ataques combinados:** El tráfico inyectado con cada fuente habitualmente no es considerado como una amenaza por separado.
- **Facilidad de ocultación:** Es posible su ofuscación mediante técnicas de mimetismo, entre las que destaca asemejar su comportamiento a intentos de acceso masivos de usuarios legítimos del sistema (del inglés *flash crowds*).

Asimismo, el rápido crecimiento de las redes de ordenadores *zombis* o *botnets* supone un importante refuerzo a la hora de elaborar este tipo de ataques. En este contexto, el controlador de la red puede ejecutar de manera rápida y efectiva el ataque desde todos los equipos infectados. Esto facilita la ejecución de determinados tipos de DDoS que requieren de técnicas de suplantación de identidad, popularmente conocidas como *Spoofing*. Además de estas ventajas, las propias características de los protocolos de red actuales facilitan la utilización de este tipo de ataques. Entre ellas cabe destacar:

- **Recursos compartidos:** El uso compartido de recursos y su gestión dinámica permite a cualquier usuario agotar los recursos del canal compartido. Asimismo, cualquier ataque es potencialmente capaz de afectar a cualquiera de los usuarios.

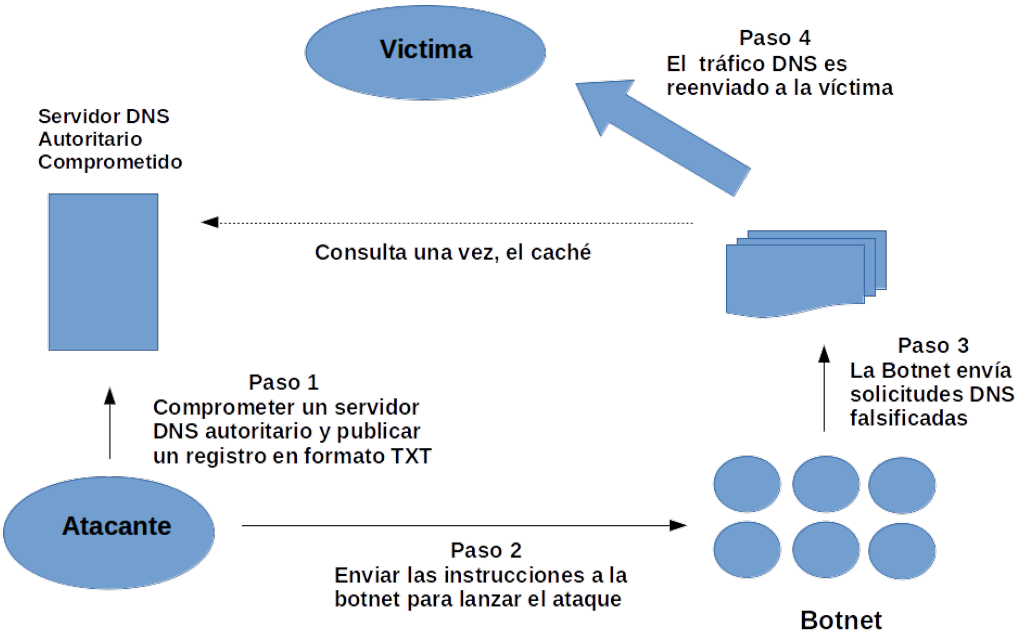


Figura 3.2: Ejemplo de RDDoS

- **Características de dispositivos de encaminamiento:** Los dispositivos de encaminamiento con mayor nivel de proximidad al núcleo de Internet (subredes de interconexión de los ISP) habitualmente presentan menor complejidad. Esto dificulta las tareas de rastreo del origen del ataque, y la identificación de robos de identidad.

- **Encaminamiento múltiple:** Dado que en los protocolos de encaminamiento actuales cada paquete puede recorrer diferentes caminos para llegar a su destino, la reconstrucción de escenario del ataque es mucho más compleja.

- **Gestión descentralizada:** La gestión descentralizada de Internet impide la aplicación de medidas de detección comunes.

En la actualidad es frecuente una nueva variedad de ataque DDoS que se beneficia de cada una de estas facilidades: los DDoS reflectantes o RDDoS (del inglés *Reflected Distributed Denial of Service*). Los ataques RDDoS se aprovechan de elementos neutrales de la red, que actúan como elementos reflectantes y en ocasiones amplificadores del ataque. Esto presenta una importante ventaja a la hora de ocultar el origen del ataque y es capaz de maximizar el daño causado.

En la Fig.3.2 se muestra un ejemplo de RDDoS conocido como Amplificación DNS. La Amplificación DNS consiste en explotar servidores DNS neutrales para la denegación de servicio de un sistema. Se basa en que cuando un nodo realiza una petición al servidor, recibe diferentes datos sobre el dominio y la dirección solicitados. El servidor DNS almacena la información de cada dominio en estructuras de datos denominadas Registros de Recursos o RR (del inglés *Resource Records*). El campo de los registros que mayor impacto produce es el TXT, que tiene longitud variable, y contiene información variada que el propietario del dominio quiera mostrar.

Durante el ataque, el atacante inunda al servidor DNS con peticiones que incluyen el RR del dominio solicitado. En una primera etapa, el propio atacante puede comprometer el servidor DNS mediante la creación de un TXT de longitud muy grande en el RR a solicitar. No obstante, este paso a veces es ignorado, dado que algunos dominios públicos ya tienen su campo TXT lo suficientemente grande. En la segunda etapa, el atacante ordena a los diferentes nodos de origen (habitualmente nodos infectados por *botnets*) que soliciten el RR del dominio con el TXT grande. Las peticiones presentan una dirección de origen falsa, que coindice con la de la víctima (*Spoofing*). En consecuencia, la víctima es inundada por las respuestas del servidor. La amplificación se produce cuando el tamaño de la respuesta es mayor que el de la solicitud (motivo por el cual se eligen RR con el campo TXT grande).

3.3.1. Técnicas de ofuscación

Para dificultar los procesos de detección, los ataques DDoS habitualmente incorporan de técnicas de ofuscación. Las técnicas más frecuentes se basan en el engaño a sistemas de detección basados en el estudio de fluctuaciones en el volumen de tráfico de la red protegida. Éste es el caso del análisis de su entropía o el establecimiento de umbrales adaptativos de rechazo. Los dos ataques más representativos son el incremento paulatino del volumen de tráfico (del inglés *Slowly Increasing Attack*) y la inundación en intervalos (del inglés *Low-rate Attack*).

La Fig.3.3 muestra un ejemplo de *Slowly Increasing Attack*. En ella el volumen de tráfico inyectado por el atacante aumenta lentamente en función del tiempo. Esto evita que los sistemas de detección identifiquen grandes variaciones en intervalos de tiempo cortos. La Fig.3.4 muestra un ejemplo de *Low-rate Attack*. En ella la inyección de grandes volúmenes de tráfico se divide en intervalos de tiempo. De esta manera es posible evadir estrategias de detección que aplican técnicas para el reconocimiento de falsos positivos. La no continuidad de la amenaza puede llevar al detector a identificarlo como un falso positivo, y cancelar su etiquetado.

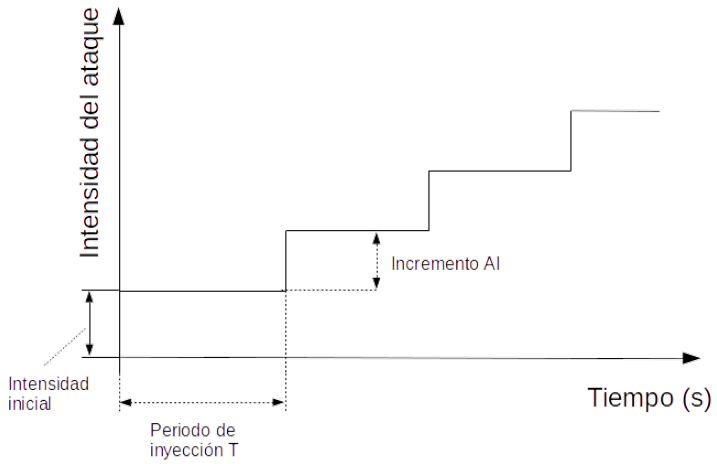


Figura 3.3: Ejemplo de ofuscación mediante incremento paulatino del volumen de tráfico

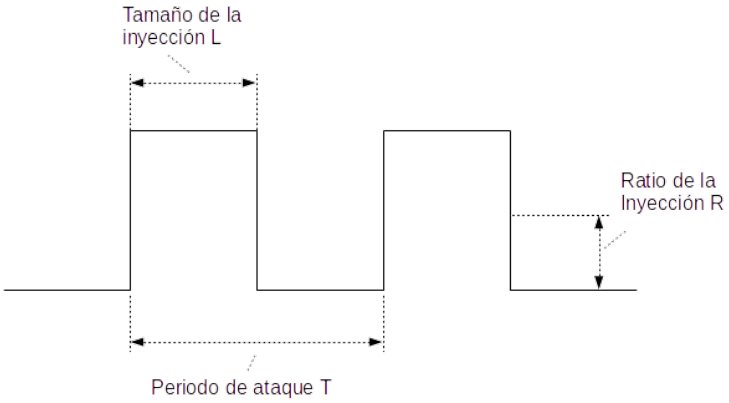


Figura 3.4: Ejemplo de ofuscación mediante inundación a intervalos

También es frecuente la aplicación de técnicas de mimetismo (del inglés *mimicry*). Los ataques de mimetismo se basan en el modelado de las características del ataque, de manera que conserve un alto grado de similitud con el modo de uso habitual y legítimo de la red. Esta medida es especialmente eficaz frente a sistemas de detección de intrusiones que basan sus estrategias de detección en la identificación de anomalías en el uso de la red. No obstante, presentan una mayor dificultad de ejecución, ya que requieren conocimiento previo sobre los modos de uso de la red a atacar.

3.3.2. Detección y mitigación de ataques DDoS

Las medidas de detección y mitigación de ataques DDoS puede ubicarse en diferentes lugares de la topología de la red: en el extremo más próximo al origen del ataque (del inglés *Source-end defense*), distribuida entre el origen del ataque y la víctima (del inglés *Core-network defense*), o inmediatamente antes de la víctima (del inglés *Victim-end defense*) [23] [24]. Cada ubicación aporta características diferentes en la defensa. La proximidad al origen del ataque permite la aplicación de contramedidas de una manera más eficiente y antes de su propagación. La ubicación distribuida entre los extremos del ataque facilita el proceso de reconstrucción del escenario de ataque, y la aplicación de contramedidas antes de su llegada a la víctima. Finalmente, la ubicación en el extremo víctima aporta una mayor precisión en los procesos de detección. Esto es debido a que es el punto en el que el ataque ha sido completamente ensamblado.

En la actualidad la tendencia es aprovechar las ventajas de cada una de las ubicaciones, en la denominada defensa colaborativa (del inglés *Collaborative defense*). La defensa colaborativa distribuye la actividad defensiva en tres etapas: detección, identificación del origen y mitigación. A continuación, se describe cada una de ellas.

Detección

En la defensa colaborativa la detección de las amenazas se desempeña en el extremo más próximo al sistema a proteger. De esta manera se consigue una mayor precisión. Inicialmente la identificación de ataques DDoS se realizaba de manera selectiva: cada sensor reconocía las características de determinados ataques previamente conocidos. Pero en la actualidad las redes presentan patrones de tráfico dinámicos y el atacante puede aplicar técnicas de ofuscación o suplantación de identidad para traspasar las medidas defensivas tradicionales. El atacante también puede valerse de elementos reflectantes para ocultarse y amplificar el daño causado. Esto ha llevado a que muchas aproximaciones lleven a la aplicación de técnicas de detección basadas en anomalías.

La detección basada en anomalías se centra en el modelado del uso habitual y legítimo de la red o en la elaboración de umbrales adaptativos. Ambas tienen en común el estudio de las tendencias del volumen de tráfico analizado, el estudio de series temporales y la aplicación de técnicas como el análisis espectral, la estimación de las variables cíclicas irregulares o la detección de puntos de inflexión. Un ejemplo claro de este paradigma es el estudio de la entropía del tráfico monitorizado, el cual se

basa en la medición del grado de aleatoriedad que presenta. Su uso se fundamenta en que, en condiciones legítimas, la entropía del tráfico tiende a ser estable. Sin embargo, cuando comienza un ataque DDoS la entropía fluctúa drásticamente en un periodo corto de tiempo. Esto es debido a que, durante los ataques, el tráfico tiende a seguir patrones lineales.

Identificación del origen y mitigación

En la defensa colaborativa la identificación del origen del ataque se lleva a cabo mediante diferentes técnicas de marcado. Esto requiere la colaboración de diferentes elementos de encaminamiento distribuidos a lo largo de la red. Las primeras técnicas para localizar el origen se basaban en el intercambio de mensajes entre los dispositivos. Una vez detectado, se comienza una cadena de mensajes que recorre todos los nodos de la red hasta llegar al atacante. Sin embargo, en la actualidad esta estrategia resulta inviable dada la sobrecarga que produce en la red y la dificultad de rastreo cuando se han practicado robos de identidad. En consecuencia, lo habitual es aplicar técnicas de marcado, entre las que destacan: Marcado de Paquetes Determinista o DPM (del inglés *Deterministic Packet Marking*), Marcado de Paquetes Probabilista o PPM (del inglés *Probabilistic Packet Marking*) y Marcado en Demanda MoD (del inglés *paquet Marking on Demand*).

El Marcado de Paquetes Determinista se basa en almacenar en alguno de sus campos la lista que indica todos los nodos que ha recorrido hasta llegar a la víctima. De esta manera la víctima es capaz de reconstruir de manera precisa el escenario del ataque.

No obstante, su aplicación conlleva un incremento del consumo de recursos computacionales, disminución de la calidad de servicio de la red y habitualmente presenta problemas de escalabilidad.

El Marcado de Paquetes Probabilista busca solucionar los problemas que presenta DPM, a costa de penalizar su precisión. A diferencia de DPM, únicamente conserva información sobre algunos de los nodos por los que ha pasado. Por ejemplo, en el marcado por muestreo de nodos (*node sampling*) únicamente se conserva información de uno de los nodos del recorrido. Cada vez que el paquete atraviesa un nuevo nodo, existe una probabilidad p de que su información sea la marcada. De esta manera, la víctima recibe una mayor cantidad de paquetes marcados de los nodos más próximos. Un ejemplo de esta aproximación se ilustra en el marcado de

bordes (del inglés *edge sampling*), donde cada nodo recorrido por el paquete tiene una probabilidad p de ser el primer nodo marcado, o nodo de inicio. Una vez establecido el nodo de inicio, cada nodo recorrido tiene una probabilidad q de ser el nodo final. A partir de los nodos marcados la víctima es capaz de generar un árbol T con las rutas marcadas y sus probabilidades, entre las que se encuentra el origen del ataque. PPM es menos preciso que DPM, pero su sobrecarga en la red es menor.

Finalmente, el Marcado en Demanda es una alternativa a las propuestas anteriores. Se trate de una estrategia cooperativa entre distintos nodos distribuidos a lo largo de la red. Cada vez que un sensor detecta tráfico sospechoso le asigna una marca única (ID) y lo envía a un servidor central. El servidor central tiene la capacidad de reconstruir el escenario del ataque a partir de todas las marcas recibidas. MoD es la alternativa más rápida, más escalable y que causa menor sobrecarga. Sin embargo, requiere de infraestructura adicional que encarece los costes de implementación.

La mitigación habitualmente consiste en la propagación de filtros hacia los nodos que componen el escenario del ataque. Los filtros contienen reglas que añaden restricciones a los dispositivos de encaminamiento cercanos. Su eficacia depende de la precisión con que se ha identificado el origen de la amenaza y con la que se ha trazado su recorrido. El proceso de propagación es análogo a las estrategias de marcado de paquetes, pero en orden inverso.

3.4. Botnets

El aumento de los ataques DDoS, tal y como se ha comentado anteriormente, está directamente relacionado con el aumento y sofisticación de las *botnets*. Debido a la relevancia de estas estructuras, el resto de esta sección se centrará en su descripción y evolución.

Las *botnets* son redes compuestas por sistemas comprometidos o bots, gestionados de manera remota por el atacante, también conocido como *botmaster*. Habitualmente son diseñadas para perpetrar acciones malintencionadas a gran escala, como el envío de *spam*, ataques de denegación de servicio distribuidos, propagación de *edge sampling* malware o la manipulación de votaciones y sistemas basados en reputaciones. Su uso ha evolucionado adaptándose a las nuevas tendencias. De este modo, en la actualidad es frecuente su presencia en fraudes tales como la minería de Bitcoins desde equipos comprometidos, o la ruptura del anonimato ofrecido por PETs (del

inglés *Privacy-Enhancing Technologies*).

Las principales agencias para la ciberdefensa, así como las empresas líderes en el área de la seguridad de la información advierten de que este problema aumenta con el paso de los años. Las *botnets* actuales tienden a ser menos extensas que sus predecesoras [25]. Sin embargo, han crecido en sofisticación, aplicando técnicas cada vez más efectivas para evadir los sistemas de detección, entre las que se incluye la ocultación de sus servidores C&C (del inglés *Command & Control*) mediante esquemas de anonimato (con este fin, cabe destacar el uso de la red TOR), o la aplicación de técnicas de ofuscación de código, cada vez más avanzadas [26].

Recientemente también se viene dando un problema inherente a su popularidad: el aumento de la cantidad de productos relacionados con *botnets* a modo de CaaS (del inglés *Crimeware-as-a-Service*) provisto por el mercado negro. Esta oferta además varía en función de las necesidades del atacante, pudiendo adquirirse su código fuente, frameworks para su personalización, servicios de soporte técnico para su instalación y colecciones de *bots*. También se alquilan para la ejecución de delitos informáticos determinados, como *phishing* o denegación de servicio [27]. Su adquisición es cada vez más fácil y más barata.

La mayor parte de las fuentes coinciden con que el resultado de estas nuevas tendencias es difícil de cuantificar. Sin embargo, sí que se dispone de algunos datos esclarecedores. Por ejemplo, tal y como anunció la Agencia Europea de Seguridad de las Redes y de la Información (ENISA) en su informe anual del 2014, el 34 % de los ataques reportados se han basado en el uso de *botnets*. Además, en determinadas actividades tienen una presencia todavía mayor. Por ejemplo, la empresa Symantec estimó que el 76 % del spam enviado en ese mismo año, tuvo su origen en este tipo de malware [25].

3.4.1. Origen

El origen de las *botnets* tuvo lugar en el protocolo IRC (del inglés *Internet Relay Chat*), un protocolo de comunicación en tiempo real que permitía a los participantes organizarse en diferentes canales de conversación. Los primeros *bots* no tenían uso malicioso. Se trataban de pequeños scripts capaces de automatizar tareas, o hacerse pasar por usuarios reales. Sin embargo, poco a poco fueron ganando funcionalidad, hasta convertirse en herramientas capaces de perpetrar ataques de denegación de servicio contra otros usuarios, o incluso servidores. De entre ellas cabe destacar el

espécimen "GTbot", el cual causó un mayor impacto a lo largo del año 2000, y se basaba en el cliente mIRC de dicho protocolo. Este además tenía la capacidad de escanear sistemas infectados por troyanos de aquella época, como por ejemplo, "Sub7", y transformarlos en sus propios *bots*.

3.4.2. Técnicas de ocultación

El mayor problema a afrontar a la hora de identificar las *botnets* actuales, es la gran sofisticación de los métodos de evasión que incorporan. Tal y como apuntan las diferentes organizaciones, las *botnets* son cada vez más silenciosas y difíciles de trazar [25] [26][27]. De este modo, aún en el caso de que se consigan desactivar, resulta muy difícil de señalar a sus propietarios.

Para la ocultación de los dominios asociados a la infraestructura C&C, es habitual el uso de dos técnicas: Algoritmos de Generación de Dominios (DGA) y *Fast-Flux*.

Los DGA permiten la generación de grandes cantidades de dominios únicos con nombres prácticamente aleatorios, permitiendo que los bots se conecten a partir de ellos. El *botmaster* conoce el algoritmo implementado en el *malware*, y es capaz de predecir a qué dominios se conectará. Esto dificulta considerablemente la traza del origen de las amenazas, y obliga a los analistas de seguridad a realizar ingeniería inversa sobre ellos. En el mercado negro puede encontrarse una gran variedad de DGAs. Algunos están incluidos en kits de desarrollo de *botnets*, como es el caso de "ZeuS".

Por otro lado, el *Fast-Flux* consiste en asignar diferentes direcciones IP a un mismo dominio. De este modo, cada vez que se haga una consulta al servidor DNS sobre él, devolverá una dirección IP distinta. Las redes *Fast-Flux* son una versión más sofisticada, en las que participan equipos comprometidos a los que apuntan los registros DNS de un determinado dominio, y que actúan como proxy entre los clientes y los servidores donde se almacena el contenido. Por lo tanto, permiten intercambiar en pequeños intervalos de tiempo, las direcciones IP asociadas al dominio.

3.4.3. Estrategias de detección

- **Tarros de miel:** Se denomina tarros de miel a los sistemas o redes de computadores (en este caso, *honeynets*) desplegados con el propósito de atraer ataques

con el fin de alertar a los operadores y desenmascarar su modus operandi.

- **Reconocimiento de firmas:** Los métodos de detección basados en el reconocimiento de firmas consisten en contrastar las características auditadas en el entorno de monitorización, con los patrones de amenazas previamente conocidos, y habitualmente almacenados en bases de datos en forma de reglas.
- **Reconocimiento de anomalías:** Esta estrategia consiste en la construcción de modelos que representan los modos de uso habituales y legítimos de la red y los sistemas protegidos. A partir de ellos es posible identificar casos de usos anómalos, es decir, que no corresponden con su modo de uso habitual; muchas de estas fluctuaciones coinciden con intrusiones reales.

3.4.4. Mitigación

Cuando las *botnets* son detectadas, el siguiente paso a realizar es su neutralización y/o la desinfección de sus bots. Las diferentes técnicas propuestas con este fin, se centran principalmente en dos acciones: la lucha contra la infección y por lo tanto, anexión de nuevos *bots* a la red, y el bloqueo de las comunicaciones entre los equipos comprometidos con el *botmaster*. Estas deben de realizarse a diferentes niveles, que abarcan desde las contramedidas aplicadas por los propios usuarios, hasta acciones tomadas por administradores de redes, e incluso Proveedores de Servicios de Internet (ISP).

El Fig.3.5 se resumen las estrategias de mitigación de *botnets*. Estas son clasificadas en dos grandes grupos: reducción de capacidad de propagación y desmantelamiento. Cada uno de ellos contiene a su vez, diferentes procedimientos. En el primero se consideran prevención, desinfección y contención. En el segundo participan acciones de bloqueo de la red maliciosa, y técnicas para localizar su origen.

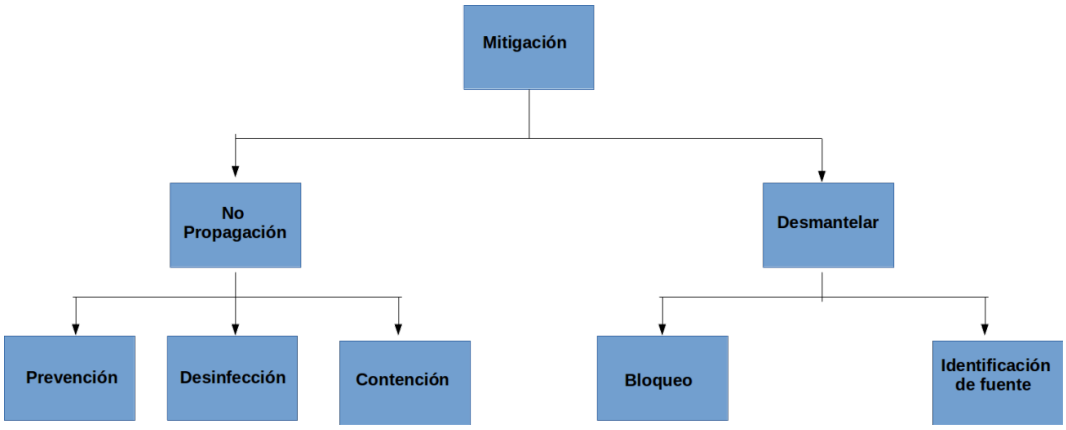


Figura 3.5: Estrategias de mitigación de *botnets*

Capítulo 4

Modelos predictivos en escenarios de red

En este capítulo se describe la estrategia propuesta para la predicción de series temporales de una sola variable, siendo posible a partir de ellas la proyección del estado de la red mediante la estimación de sus indicadores. Con este fin, se distinguen dos grandes etapas de procesamiento de datos: entrenamiento y predicción adaptativa (ver Fig.4.1). La etapa de entrenamiento, realizada previamente al análisis de una serie temporal, tiene como objetivo crear, a partir de una colección de muestras de referencia (del inglés *dataset*), un clasificador que permita elegir el mejor algoritmo predictivo para un conjunto de observaciones en concreto. En la experimentación llevada a cabo, se ha utilizado el conjunto M3-Competition [28]. Por otro lado, en la etapa de predicción adaptativa, una vez seleccionado el mejor algoritmo de predicción, se realiza el calibrado del mismo que permite realizar la predicción correspondiente. Además, las características extraídas de la serie temporal a analizar definen el modelo de predicción que va a ser usado para las siguientes observaciones.

4.1. Entrenamiento

El objetivo principal de esta etapa es adaptar la estrategia de predicción a cualquier conjunto de datos a analizar, para ello, antes de calcular la proyección se decide el modelo que mejor se ajuste a dicho conjunto. Este proceso, se puede subdividir en dos etapas claramente diferenciadas (ver Fig.4.2): en primer lugar, el etiquetado de las muestras proporcionadas por el dataset de referencia, que consiste en extraer las características de las series temporales, realizar una aproximación con la batería de algoritmos con un calibrado aleatorio, clasificar dichos resultados y definir la clase que representa el algoritmo que mejor se adapta para esas características. En se-

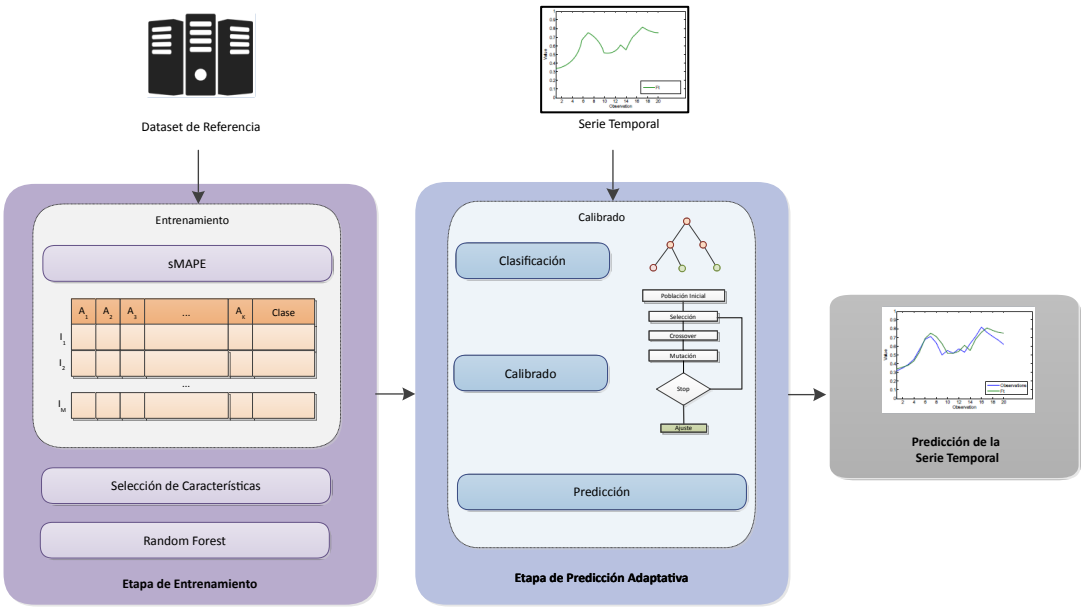


Figura 4.1: Etapas de la propuesta

gundo lugar, consiste en la creación de un clasificador que decide cuál es el mejor algoritmo para un conjunto de características en particular. Para ello, se ha adaptado el proceso de clasificación Random Forest presentado por Breiman [29] que será detallado a continuación.

4.1.1. Extracción de características y etiquetado de las muestras

Con el fin de facilitar la comprensión de la primera etapa que compone la fase de entrenamiento, su explicación se va a dividir en dos subsecciones, por un lado, la que define la extracción de características y por otro lado, la explicación del etiquetado de las muestras.

Extracción de características

Con el fin de entrenar el sistema para que sea capaz de elegir el mejor algoritmo de predicción para una serie temporal cualquiera, se parte de un conjunto de observaciones, en la experimentación habiéndose utilizado el dataset M3-Competition

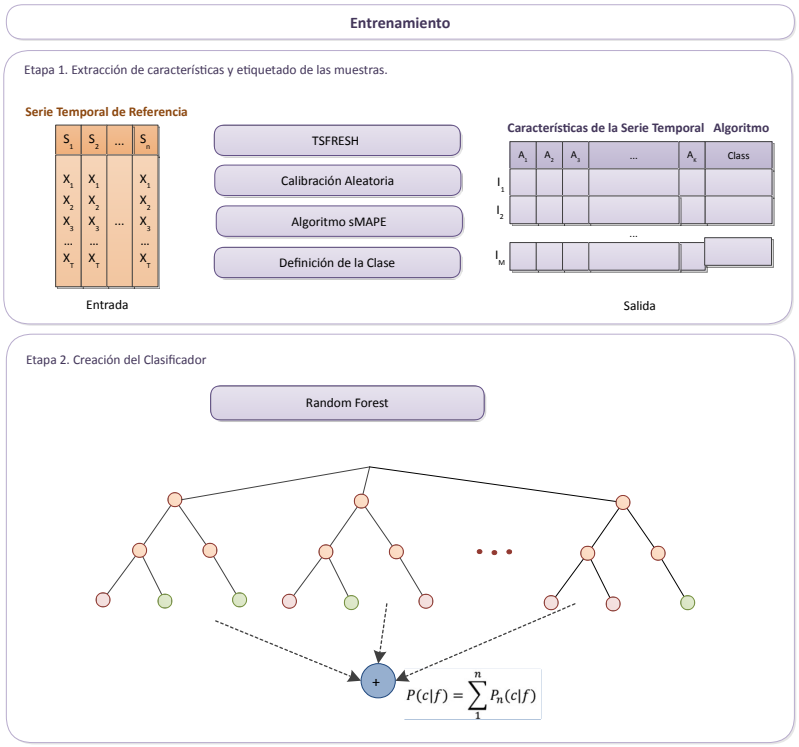


Figura 4.2: Etapa de Entrenamiento

[28], que provee una colección de series temporales de referencia. El primer paso a realizar es la extracción de las características de cada serie temporal. Para ello, se ha utilizado la herramienta TSFRESH, la cual ha sido desarrollada en el marco del proyecto alemán iPRODICT [30]. Esta herramienta permite analizar una serie temporal y extraer una batería de características que definen la serie temporal. Esta batería tiene en cuenta desde atributos estadísticos básicos (picos, valores máximos, mínimos. etc.) hasta medidas de corrección y evolución de una serie temporal (ruido blanco, tendencia, estacionalidad, autocorrelación. etc.).

Etiquetado de las muestras

Una vez definida la serie temporal en función de sus características, el siguiente paso es extraer la clase correspondiente a cada instancia de dicho conjunto. Para ello, se realiza una predicción de cada serie temporal con la batería de algoritmos de predicción detallados a lo largo de esta subsección. Nótese que la mayor parte de ellos necesitan parámetros de ajuste, los cuales son definidos aleatoriamente o recorriendo el espacio completo de posibilidades. Para definir la clase, se elige el algoritmo de predicción que mejor se haya adaptado a ese tipo de características,

es decir, la clase corresponde al algoritmo de predicción que haya obtenido menor Error de Porcentaje Absoluto Simétrico Medio o sMAPE (del inglés *Symmetric Mean Absolute Percentage Error*). Se ha considerado esta medida de evaluación ya que ha sido adoptada, entre otros, por el M3-Competition [28] y permite evaluar la efectividad de los pronósticos. La fórmula del sMAPE se puede expresar formalmente de la siguiente manera:

$$sMAPE = 200\,\% \sum_{t=1}^{n} \frac{|x_t - \ x_t|}{|x_t| + |x_t|} \tag{4.1}$$

dónde n es la observación más reciente correspondiente a las métricas agregadas de la serie temporal $x_1, x_2, \ldots, x_n$ a pronosticar.

Batería de Algoritmos de Predicción

En la experimentación se han implementado las familias de métodos de predicción originalmente provistas por el componente de análisis de SELFNET, entre las que se encuentran modelos basados en medias móviles, autorregresión y alisamiento, los cuales se van a detallar a continuación.

- **Medias Móviles**

  - **Medias móviles acumulativas o CMA** (del inglés *Cumulative Moving Average*) [31]: Su objetivo es calcular la media del conjunto de datos, desde la primera observación hasta el elemento i-ésimo. Esto se puede representar con la siguiente expresión recursiva:

    $$CMA_n = \frac{\sum_{i=0}^{n} x_i}{n} \tag{4.2}$$

    donde el CMA para el elemento n+1 se expresa de la siguiente forma:

    $$CMA_n = \frac{\sum_{i=0}^{n} x_i}{n} \tag{4.3}$$

  - **Medias móviles simples o SMA** (del inglés *Simple Moving Average*) [32]: Este algoritmo es una variación del método CMA, basada en suavizar en base a la media las últimas n observaciones que forman la serie temporal a analizar. Sea m la longitud de la subsecuencia a tener

en cuenta, la fórmula del SMA se puede definir de la siguiente manera:

$$SMA = \frac{P_m + P_{m-1} + \ldots + P_{m-(n-1)}}{n} = \frac{1}{n}\sum_{i=0}^{n-1} P_{m-i} \tag{4.4}$$

y la siguiente observación con la predicción en base al algoritmo SMA es:

$$SMA_{t+1} = SMA_t + \frac{P_m}{n} - \frac{P_{m-n}}{n} \tag{4.5}$$

- **Medias móviles dobles o DMA** (del inglés *Double Moving Avergage*) [32][33][34]: Esta técnica fue presentada por Mullony con el objetivo de reducir el tiempo de ejecución de los algoritmos de medias móviles tradicionales. El valor M_t para un instante de tiempo concreto viene dado por la siguiente expresión:

$$M_t = \frac{Y_t + Y_{t-1} + \ldots + P_{t-(n+1)}}{n} \tag{4.6}$$

donde siguiente fórmula de $M_t^{'}$ es construida a partir del suavizado:

$$M_t^{'} = \frac{M_t + M_{t-1} + \ldots + M_{t-(n+1)}}{n} \tag{4.7}$$

definiéndose DMA a partir de la siguiente fórmula:

$$DMA_t = 2M_t - M_t^{'} \tag{4.8}$$

para la predicción de futuras observaciones, DMA se basa en el parámetro b_t expresado como:

$$b_t = \frac{2}{n-1}\left(M_t - M_t^{'}\right) \tag{4.9}$$

el cual permite definir la observación Y en t+p de la siguiente forma:

$$Y = DMA_t + b_t p \tag{4.10}$$

- **Medias móviles ponderadas o WMA** (del inglés *Weighted Moving Average*) [35]: A diferencia que medias móviles anteriores, WMA considera diferentes ponderaciones multiplicativas a las observaciones en diferentes puntos de la serie temporal, dando más importancia a eventos recientes, lo que permite una mayor reacción a los cambios recientes. La

fórmula es la siguiente:

$$WMA_t = \frac{w_t x_t + w_{t-1} x_{t-1} + \ldots + w_{t-(n+1)} x_{t-(n+1)}}{n + (n-1) + \ldots + 2 + 1} = \frac{\sum_{t=1}^{n} w_t x_t}{\sum_{t=1}^{n} w_t} \tag{4.11}$$

dónde $w_i, 1 <= i <= n$ es la ponderación para la i-ésima observación. Nótese que esta implementación del algoritmo WMA asume la poderación clásica $w_i = i$.

- **Medias móviles simples exponenciales o EWMA** (del inglés *Exponentially Weighted Moving Average*) [36]: Este algoritmo proporciona una rápida respuesta a los cambios más recientes. A diferencia que WMA, este método reduce los factores de ponderación exponencialmente, por lo que podemos asumir, que EMA es un caso específico de WMA. Normalmente se define a partir de la siguiente expresión recursiva:

$$EMA_1 = x_1 \tag{4.12}$$

$$EMA_t = \alpha x_t + (1 - \alpha)\, EMA_{t-1} \tag{4.13}$$

donde $\alpha, 0 <= \alpha <= 1$ es el parámetro de ajuste que determina el grado de disminución de la ponderación. A mayor α, más importancia adquieren las nuevas observaciones.

- **Medias móviles dobles exponenciales o DEMA** (del inglés *Double Exponential Moving Average*): Normalmente, en "dominios financieros"se requiere el cálculo de diferentes variaciones de EMA para diferentes periodos de tiempo y diferentes grados de disminución de la ponderación. Esto supone un elevado coste computacional, lo que ha motivado a la búsqueda de nuevos algoritmos que agilicen el proceso, siendo este el caso de DEMA, propuesto por P.G. Mulloy [37]. DEMA propone un nivel extra de alisamiento para las predicciones. Es calculado de manera análoga a DMA, para una serie temporal de observaciones expresado la siguiente manera:

$$EMA_1 = x_1 \tag{4.14}$$

$$EMA_t = \alpha x_t + (1 - \alpha)\, EMA_{t-1} \tag{4.15}$$

Dónde la siguiente fórmula de $EMA_t^{'}$ es construida a partir del suavizado:

$$EMA_1^{'} = x_1 \tag{4.16}$$

$$EMA_t^{'} = \alpha x_t + (1 - \alpha)\, EMA_{t-1} \tag{4.17}$$

pudiéndose definir DEMA a partir de la siguiente fórmula:

$$DEMA_t = 2EMA_t - EMA_t^{'} \tag{4.18}$$

Para la predicción de futuras observaciones, DEMA se basa en el parámetro b_t expresado como:

$$b_t = \frac{2}{n-1}\left(EMA_t - EMA_t^{'}\right) \tag{4.19}$$

que permite definir la observación Y en $t + p$ de la siguiente forma:

$$Y = DEMA_t + b_t p \tag{4.20}$$

- **Medias móviles triples exponenciales o TEMA** (del inglés *Triple Exponential Moving Average)*) [33]: Fue propuesto por P.G. Mulloy como una alternativa a DEMA. Este algoritmo proporciona, a su vez, un nivel de alisamiento adicional, que se calcula de la fórmula de EMA como:

$$EMA_1 = x_1 \tag{4.21}$$

$$EMA_t = \alpha x_t + (1 - \alpha)\, EMA_{t-1} \tag{4.22}$$

donde la siguiente fórmula de $EMA_t^{'}$ construida a partir del suavizado:

$$EMA_1^{'} = x_1 \tag{4.23}$$

$$EMA_t^{'} = \alpha x_t + (1 - \alpha)\, EMA_{t-1} \tag{4.24}$$

y la fórmula de $EMA_t^{''}$ que considera la base definida previamente:

$$EMA_1^{''} = x_1 \tag{4.25}$$

$$EMA_t^{''} = \alpha x_t + (1-\alpha)\, EMA_{t-1}^{'} \tag{4.26}$$

por lo que TEMA se resume de la siguiente manera:

$$TEMA_t = 3EMA_t - 3EMA_t^{'} + EMA_t^{''} \tag{4.27}$$

- **Alisamiento**

  - **Alisamiento exponencial simple o SES** (del inglés *Simple Exponential Smoothing)*): Este método fue originalmente propuesto por R.G. Brown [38] y extendido por C.C. Holt [39], es una extensión del enfoque analítico atribuido a Poisson. Se considera una variación de EMA cuyo objetivo es predecir observaciones en series temporales sin tendencia o series temporales no estacionarias. Se representa según la siguiente expresión recursiva:

$$S_t = \alpha y_{t-1} + (1-\alpha)\, S_{t-1} \tag{4.28}$$

donde $0 < \alpha < 1, t >= 3, y_i$ es la observación en el instante i, y α es la constante de alisamiento. Para solucionar el enfoque del caso base para esta expresión, se ha considerado el enfoque de predicción clásico, posponiendo la exploración de estrategias alternativas para trabajo futuro. El ajuste del parámetro se obtiene calculando los valores minimizados de la suma de errores cuadráticos medias de la predicción o SSE (del inglés *Sum of the Squared Errors*), representada por la siguiente fórmula:

$$SSE\,(\alpha,) = \sum_{t=1}^{N} \left(H_\alpha\,(X)_t - H_\alpha\,(X)_{t_{|t-1|}} \right)^2 \tag{4.29}$$

Sobre esta base, los valores pronosticados se calculan de la siguiente manera:

$$S_{t+1} = \alpha y_t + (1-\alpha)\, S_t \tag{4.30}$$

que también se puede expresar como:

$$S_{t+p} = S_t + \alpha \epsilon_t \tag{4.31}$$

dónde ϵ_t es el error de la predicción observado en un instante t.

  - **Alisamiento exponencial doble o DES** (del inglés *Double Exponential*

Smoothing)): Por definición, SES resulta inefectivo cuando la serie temporal a analizar presenta una tendencia significativa. Para solventar este problema, se propuso el algoritmo DES [40]. Este introduce una nueva constante que se adapta al nivel de tendencia de la serie temporal, e incluye una segunda ecuación para su generación. Las ecuaciones recursivas se detallan a continuación:

$$S_t = \alpha y_{t-1} + (1-\alpha)\left(S_{t-1} + b_{t-1}\right) \tag{4.32}$$

$$b_t = \gamma\left(S_t - S_{t-1}\right) + (1-\gamma)\, b_{t-1} \tag{4.33}$$

donde $0 <= \alpha >= 1, 0 <= \gamma <= 1$. Como es frecuente en la bibliografía, los casos base se inicializan de la siguiente manera: $S_1 = S_1$ y b_1 debe ser:

$$b_1 = y_2 - y_1 \tag{4.34}$$

$$b_1 = \frac{1}{3}\left[(y_2 - y_1) + (y_3 - y_2) + (y_4 - y_3)\right] \tag{4.35}$$

$$b_1 = \frac{y_n - y_1}{n-1} \tag{4.36}$$

El componente de análisis de SELFNET implementa todos ellos, aunque la ecuación considerada en este trabajo es la segunda. En consecuencia, la predicción basada en este método ha sido calculada de la siguiente manera:

$$y_{t+1} = S_t + b_t \tag{4.37}$$

$$y_{t+m} = S_t + m b_t \tag{4.38}$$

- **Alisamiento Exponencial Triple o TES** (del inglés *Triple Exponential Smoothing)*) [41]: A diferencia que el algoritmo anterior, TES tiene en cuenta los cambios estacionales de las series temporales, lo que supone la introducción de un nuevo parámetro de ajuste β que se relaciona con el grado estacional y una expresión recursiva de suma. Se puede calcular

mediante la siguiente expresión recursiva:

$$b_t = \alpha\left(y_t - S_{t-N}\right) + (1-\alpha)\left(b_{t-1} + T_{t-1}\right) \tag{4.39}$$

$$T_t = \beta\left(b_t - b_{t-1}\right) + (1-\beta)\, T_{t-1} \tag{4.40}$$

$$S_t = \gamma\left(y_{t_t} - b_t\right) + (1-\gamma)\, b_{t-N} \tag{4.41}$$

donde b_t es la estimación base en un instante t, la estimación de la tendencia es denominada por T_t y la estimación del factor estacional es S_t. Por otro lado, los parámetros α, β, γ se definen dentro del rango $0 < \alpha, \beta, \gamma < 1$. La predicción y_{t+m} se puede calcular de dos formas: De forma aditiva:

$$y_{t+m} = mb_t + T_{t-m} + S_t\,(Additive) \tag{4.42}$$

De forma multiplicativa:

$$y_{t+m} = \left(S_t + mb_t\right) T_{t-m}\,(Multiplicative) \tag{4.43}$$

Se han implementado ambas, ya que, en primer lugar, la operación aditiva es recomendada para analizar series temporales con tendencia significativa y un componente estacional aditivo, mientras que la segunda operación es más adecuada para observaciones con un componente estacional multiplicativo. Otro aspecto a tener en cuenta es la inicialización de los estimadores b_0, T_0, S_0. Es preferible, cuando no se espera tendencia ni estacionaridad, que la inicialización de los estimadores se base en las últimas observaciones. El método implementado se define en [43] ya que se ha demostrado que funciona correctamente en casos similares. Es decir, se consideran las últimas veinticuatro observaciones y las operaciones realizadas son las siguientes:

$$b_0 = M_1 \tag{4.44}$$

$$T_0 = \frac{M_2 - M_1}{12} \tag{4.45}$$

$$S_{t-12} = \frac{p_t}{M_1} \tag{4.46}$$

donde M_1 recoge las primeras doce observaciones y M_2 las últimas doce. El ajuste de los parámetros α, β, γ se obtiene una vez más en base a la suma de errores cuadráticos medios en la predicción.

- **Autorregresión**
  A diferencia de la familia de algoritmos de alisamiento exponencial, los modelos autorregresivos no se basan en la descomposición de las observaciones en factores, sino que las observaciones determinadas dependen linealmente de observaciones previas en términos estocásticos.

  - **Modelo clásico Autorregresivo AR(p)** [44] se define como:

$$Y_t = \mu + \phi_1 Y_{T-1} + \ldots + \phi_P Y_{T-P} + \epsilon_t = \mu + \sum_{i=1}^{N} \phi_i Y_{T-i} + \epsilon_t \tag{4.47}$$

    donde $epsilon_t$ significa el ruido blanco (del inglés *white noise*), $phi_1 \ldots \phi_p$ son los parámetros proporcionados por el modelo, μ es un valor constante y p es el orden (número de retrasos temporales) de la autorregresión.
  - **Modelo de medias móviles MA(q)** [45] define un enfoque diferente, donde las observaciones determinadas dependen linealmente del valor actual y una serie de observaciones anteriores, lo que hace posible el aprendizaje de errores previos. Nótese que q es el orden del modelo de medias-móviles. MA es definido por la siguiente expresión:

$$Y_t = \mu\phi_1 Y_{T-1} + \ldots + \phi_P Y_{T-P} + a_t - \theta_1 a_{T-1} - \ldots - \theta_p a_{T-q} \tag{4.48}$$

    que es equivalente a:

$$\left(1 - \phi_1 B - \ldots - \phi_p B^p\right) Y_t = \mu + \left(1 - \phi_1 B - \ldots + \phi_p B^p\right) a_t \tag{4.49}$$

    y sintetizados como:

$$\phi_p\left(B\right) Y_t = \mu + \Theta\left(B\right) a_t \tag{4.50}$$

    Cabe destacar que estos modelos típicamente no son capaces de tratar los datos no estacionales.
  - **Modelos autorregresivos integrados de medias móviles ARIMA(p,d,q)** [45] son una generalización de los modelos ARMA comen-

tados anteriormente, que son capaces de superar la inoperatividad con observaciones no estacionales donde d es el grado de diferenciación, es decir, la resta de todas las observaciones pasadas con el objetivo de convertirlas en un valor estacional. El modelo clásico de ARIMA es expresado como:

$$Y_{T-1} - a_1 Y_{T-1} - \ldots - a_{p'} Y_{T-p'} = \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} \tag{4.51}$$

donde a_i representa los parámetros de la parte autorregresiva, θ_i son los parámetros relacionados con la parte de medias móviles y ϵ_t es el ruido blanco. El ajuste de los parámetros p, d, q puede equivaler a otros algoritmos de predicción. Por ejemplo, para ARIMA (1,1,0) equivale a simple random walk, ARIMA (1,0,0) es un modelo AR, ARIMA (0,01) es un modelo. MA, ARIMA (0,0,0) corresponde a ruido blanco, ARIMA(0,1,1) alisamiento exponencial simple o ARIMA(0,2,2) es alisamiento exponencial doble. Las predicciones en el modelo ARIMA se generan por una generalización del método autorregresivo de predicción, donde:

$$Y_t = \mu + \phi_1 Y_{T-1} + \ldots + \phi_P Y_{T-P} - \theta_1 \epsilon_{t-1} - \ldots - \theta_q \epsilon_{t-q} \tag{4.52}$$

4.1.2. Creación del clasificador

Como se ha comentado en la introducción de este capítulo, para la construcción del clasificador que permita elegir el algoritmo que mejor se adapte a cualquier conjunto de datos a analizar, se ha utilizado el procedimiento denominado Random Forest presentado por Breiman [47] por su precisión, eficacia al estudiar grandes cantidades de muestras y capacidad de operar con eficiencia listas con gran cantidad de atributos. En este caso, cada muestra considerada para este fin es representada con los 100 atributos extraídos de una serie temporal de referencia, la cual pertenece a la clase que identifica el algoritmo de predicción que mejor ha sido capaz de operar sobre ella.

Random Forest

Random Forest es un algoritmo de predicción y regresión, derivado de los Árboles de Clasificación. Este método se basa en una colección de árboles de decisión, es decir, una colección de clasifcadores estructurados en forma de árbol. Estos clasificadores son construidos con valores de un vector aleatorio que ha sido muestreado de forma independiente y que aplica la misma distribución para todos los árboles

que componen el "bosquez que posteriormente promedia. Como otros muchos clasificadores y métodos de regresión, Random Forest se construye sobre la base de entrenamiento de muestras adecuadas para cada caso de uso.

En particular, la versión original de Random Forest implementa Árboles de Clasificación y Regresión o CART (del inglés *Classification And Regression Trees*) [48] y determina qué variable de ajuste utilizar a través de un algoritmo voraz (del inglés *Greedy Algorithm*). Este facilita la reducción del error de predicción mediante un calibrado mucho más eficaz. Para completar esta tarea, es necesario, definir algunos parámetros de ajuste como, por ejemplo, el número máximo de iteraciones que se realizarán si no se cumple la condición de parada, el número de árboles CART a construir o la profundidad máxima. Aunque, como remarcó Breiman, el único parámetro de ajuste realmente significativo es el valor m que determina la cantidad de atributos seleccionados aleatoriamente (se ha asumido que no existen ninguna limitación computacional, ya que los módulos de análisis de SELFNET escalan horizontalmente). Este valor, determina la correlación existente entre cada par de árboles y la "fuerza"de cada árbol individual. Al aumentar este parámetro, tanto la correlación como la fuerza aumenta, esto implica que, si la correlación crece, la tasa de error aumenta, mientras que si la fuerza aumenta la tasa de error disminuye. Por lo tanto, es necesario que exista cierto equilibrio entre ambas características. En el trabajo descrito en este documento, esta problemática ha sido abordada mediante la solución propuesta por Breiman (es decir, m = log M + 1, donde M es el número de características de las muestras del conjunto de datos de referencia). Por lo tanto, queda para trabajos futuro el implementar estrategias de calibrado diferentes.

En la Fig.4.3 se ilustra un ejemplo del algoritmo Random Forest, el cual es entrenado en base a un conjunto de muestras de referencia. En particular, y tal y como se ha indicado anteriormente, está compuesto por los atributos extraídos de las series temporales con la herramienta TSFRESH [32] y la clase que corresponde al mejor algoritmo de predicción, que es el que registró un menor error de pronosticado cuando fue analizado en la etapa anterior. Finalmente, cabe destacar que una de las principales desventajas de los clasificadores basados en Random Forest es la tendencia al sobreajuste (del inglés *overfitting*). Para reducir este problema, se ha complementado con una fase selección previa conducida por un algoritmo voraz de discriminación de características [49] y su evaluación en base a su significancia en procesos predictivos [50].

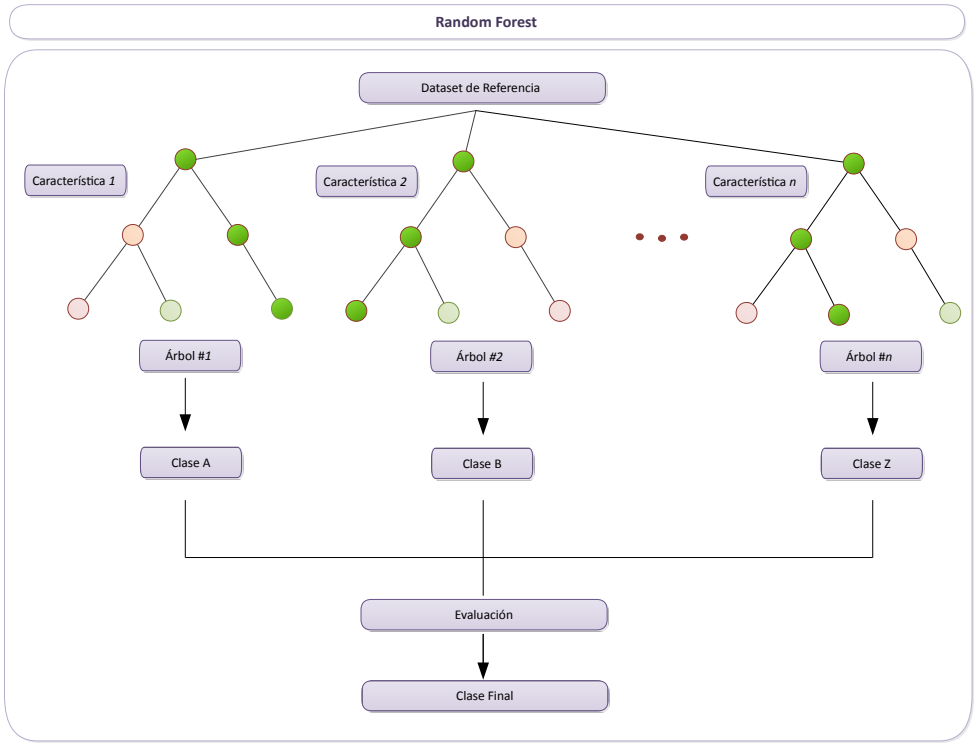


Figura 4.3: Random Forest Simple

4.2. Predicción Adaptativa

R.C. Holte [48] indicó que, en el área del reconocimiento de patrones, es frecuente asumir que el conjunto de muestras de referencia aplicados durante el entrenamiento es representativo de las observaciones esperadas en el escenario de monitorización. La presencia de cambios graduales a lo largo del tiempo en las características estadísticas de la clase a la que pertenece una observación es acaecida por las fluctuaciones no estacionarias de la misma. Esto supone, entre otros, el problema comúnmente denominado concept drift [51], que se observa cuando los modelos construidos en la etapa de entrenamiento dejan de ser representativos de las características construidas durante el entrenamiento de los clasificadores. Partiendo de esta premisa, es decir, de que tras cada observación a analizar la distribución de la información monitorizada puede mostrar cambios representativos, debe asumirse el despliegue de una estrategia de predicción adaptativa.

Para la solución de este problema O'Reilly et al. [52] distinguieron dos grandes paradigmas: adaptación activa y adaptación pasiva. Las soluciones activas requieren del reconocimiento previo de los puntos de inflexión que han derivado en cambios

relevantes en el entorno monitorizado, lo que habitualmente implica la actualización de los modelos construidos previamente. Debido a este comportamiento, estás técnicas habitualmente son conocidas como métodos de detección y respuesta. Por otro lado, las soluciones pasivas asumen que las observaciones monitorizadas varían a lo largo del tiempo, por lo que exige una recalibración continua de las capacidades analíticas. Por lo tanto, si bien las soluciones activas se centran en la distinción puntual de la fluctuación de la observación, el enfoque pasivo demuestra mayor eficacia al pronosticar la fluctuación gradual y los conceptos recurrentes [53]. Durante el trabajo realizado, se ha considerado que el segundo paradigma encaja mejor con el caso de uso a implementar, es decir, el reconocimiento de amenazas DDoS. Nótese que algunas de las técnicas de ofuscación citadas en capítulos anteriores pueden evadir con bastante éxito los procesos de adaptación activa, ya que dificultan la identificación de puntos de cambio en la distribución de datos. En consecuencia, se ha desarrollado una solución pasiva, quedando la exploración de alternativas activas o híbridas para trabajos futuros.

La adaptación pasiva a la no estacionariedad inherente a los escenarios emergentes de red se ha resuelto en dos fases, tal y como se muestra en Fig.4.4: Selección del algoritmo de predicción y Calibrado. Una vez realizadas se lleva a cabo la predicción en sí.

4.2.1. Selección del Algoritmo de Predicción

El enfoque de predicción adaptativa propuesto decide el algoritmo de pronóstico más adecuado basado en el estudio de las características de TSFRESH extraídas de las series temporales de referencia. Como se ilustra Fig.4.4, este conjunto de características sirve como entrada del clasificador Random Forest construido previamente, en la etapa de Entrenamiento. La clase resultante representa el mejor algoritmo de predicción, que estima el comportamiento esperado de la serie temporal a analizar. Este procedimiento se repite en cada observación, por lo que el método de predicción variará a medida que cambie la distribución de las observaciones. Por ejemplo, para una serie temporal concreta el clasificador decide, inicialmente, cual el algoritmo más adecuado de acuerdo con las características obtenidas por la herramienta TSFRESH en T_s es el alisamiento exponencial simple (SES) [39]. Pero en las próximas m observaciones $T_(s+m)$ aumenta significativamente la tendencia y la estacionaridad. En este caso, la probabilidad de pasar de SES a un alisado exponencial triple (TES) [44] aumenta, ya que TES se comportó con mayor precisión que SES en circunstancias similares en la etapa de Entrenamiento.

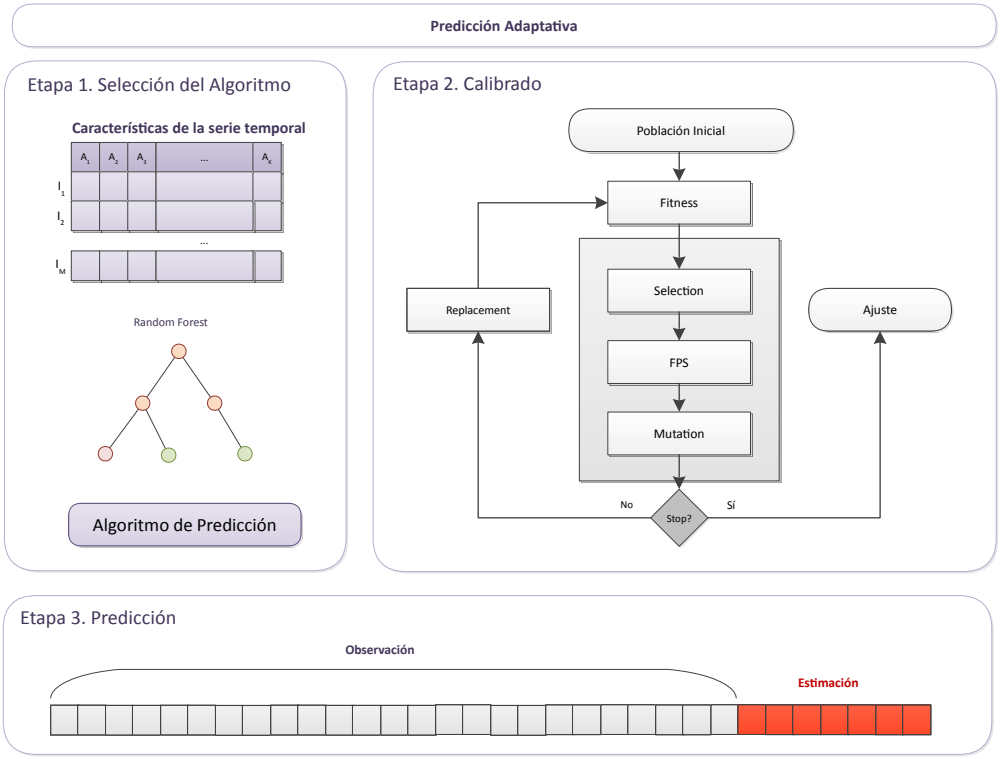


Figura 4.4: Predicción Adaptativa

4.2.2. Calibrado

Partiendo de la base de que la mayoría de los métodos que componen la batería de algoritmos que se han implementado requieren configuración previa, la calibración adecuada de sus parámetros de ajuste desempeña un papel importante para lograr un mayor rendimiento y mejorar la predicción deseada. Debido a esto, una vez completada la etapa anterior y seleccionado el método de predicción adecuado, la siguiente etapa exige el calibrado de dicho algoritmo a través de un Algoritmo Genético Básico (GA) [54].

Algoritmo Genético

Los algoritmos genéticos están basados en teorías biológicas evolutivas y parte de la base de la genética molecular. Es un algoritmo probabilístico que se basa en la evolución de una población inicial de individuos (observaciones) generada a partir de conocimiento factual inicial, que, a través de acciones con resultados aleatorios (es decir, mutaciones genéticas y recombinación genética) tratan de acercarse a la solución óptima en cada iteración. Este proceso se asemeja a los procesos de evolución biológica.

Tabla 4.1: Descripción del algoritmo genético implementado

Etapa	Acción Principal
Población Inicial	En la primera ejecución, la población inicial es generada de manera aleatoria mientras que en la siguiente observación la población resultante da lugar a la nueva población inicial.
Función de aptitud (Fitness)	El sMAPE [28] obtenido para una calibración específica.
Selección (Selection)	Selección por ruleta (del inglés *Fitness Proportionate Selection* [56].
Combinación (Crossover)	Intercambio de genes en un punto aleatorio.
Mutación (Mutation)	Mutación de un gen aleatorio.
Condición de parada	Se ha alcanzado en número de máximo de iteraciones o se ha encontrado la solución óptima.

Los principales inconvenientes de este tipo de algoritmos están relacionados con el alto consumo de recursos y la falta de garantía en encontrar una solución óptima, ambos problemas discutidos en la bibliografía [55]. Tanto la discusión como la mitigación de los mismos están fuera del alcance de este proyecto.

En este proyecto, se ha utilizado el GA como solución al problema relacionado con la calibración del algoritmo de predicción debido a diferentes motivos, entre los que se puede destacar: el hecho de que los GA ya han sido utilizados, previamente, como solución a problemas de optimización con el objetivo de realizar un calibrado [56], que son capaces de operar con vectores de diferente naturaleza (en este caso los diferentes parámetros de ajuste) y que, su funcionamiento se adapta a la perfección al nivel de detalle en el que se deben calcular las diferentes calibraciones. Este último es especialmente importante cuando se trabaja en escenarios en tiempo real, lo que permite equilibrar la precisión con el rendimiento necesario para cualquier caso.

Otro aspecto a tener en cuenta en el algoritmo genético es que permite implementar diferentes parámetros de ajuste, como por ejemplo el tamaño de la población inicial, la probabilidad de mutación, el número máximo de iteraciones... lo que hace que sea más versátil y permita adaptarse a cualquier tipo de escenario. En la Tabla 4.1 se describen sus características más relevantes.

- **Población Inicial**

  Se ha considerado como población evolutiva el conjunto de posibles soluciones donde cada individuo representa una posible configuración de los parámetros de ajuste para el algoritmo a calibrar. La población está formada por genotipos que representan un vector de genes. En cada una de sus posiciones, el gen representa uno de los parámetros de ajusto del método de predicción. Por ejemplo, en el caso del algoritmo de predicción TES se construiría a partir de una colección de tres características: factor de suavizado de datos (α), factor de suavizado de tendencia (β), factor de suavizado de cambio estacional (γ) [41]. Además, para cada genotipo se añade otro parámetro más que representa el horizonte de pronóstico (τ).

  Uno de los aspectos más importantes relacionados con la población del algoritmo genético es la inicialización de la misma, es decir, la población inicial de la que se va a partir y de la que se va a evolucionar. En el marco de este trabajo se han planteado dos escenarios diferentes: 1) en el caso de no tener registros previos, la población inicial se calcula completamente aleatoria teniendo en cuenta que algunos de los parámetros de ajuste no pueden ser mayores (o menores) a ciertos valores de referencia. 2) Por otro lado, con el objetivo de reducir recursos y remarcar la predicción adaptativa, la población final para una observación en concreto se convierte en la población inicial de la siguiente observación, lo que permite mejorar las predicciones futuras además de reducir el coste computacional. Este último escenario se basa en el hecho de que la mayoría de las series temporales presentarán pequeñas variaciones a lo largo de un periodo de tiempo, por lo que no se esperan grandes cambios en los parámetros de ajuste.

- **Función de aptitud**

  Unos de los principios básicos para el correcto funcionamiento del algoritmo genético es la idea de que sólo los individuos más adaptados tienen la posibilidad de persistir en las futuras generaciones. Hay que tener en cuenta que, como en la naturaleza, la aptitud de un individuo define su capacidad de adaptación al entorno y, por tanto, la probabilidad de reproducción. Para ello, es necesario definir una función de aptitud (fase de fitness) que permita evaluar cada individuo de la población y permita clasificar dicha población.

  Para ello, se ha utilizado el Error de Porcentaje Absoluto Simétrico Medio,

del (sMAPE) [28]. Cada genotipo es evaluado con esta métrica que permite evaluar la posible efectividad del pronóstico con el algoritmo de predicción en cuestión. Por lo tanto, la población evaluada será aquella con los genotipos valorados con el mejor sMAPE.

Una vez definida la función de aptitud, la población inicial es evaluada en base a ella, lo que proporciona una nueva población donde se ha tenido en cuenta los cromosomas que mejor se han adaptado al medio, es decir, los que hayan sido evaluados con un menor sMAPE. Este paso se repite a lo largo de todas las iteraciones del GA. Una vez creada dicha población se seleccionan los cromosomas padres, que serán combinados para la producción, y por lo tanto generación de nuevos individuos.

- **Selección y Combinación**

  Para que la población pueda evolucionar, es necesario que los mejores rasgos de cada cromosoma se transmitan a lo largo de ésta. Para ello deben ser seleccionados vatios cromosomas, que participarán en procesos de reproducción (cruce) y dan lugar a nuevos cromosomas. Este proceso permite explorar las diferentes posibles soluciones al problema a solucionar. Para la selección de los candidatos al cruce se ha implementado el método de la ruleta (del inglés *Fitness Proportionate Selection*) [57]. Se trata de un método elitista que otorga a cada cromosoma una posibilidad de selección proporcional a su adaptación, la cual determina la porción de la ruleta que ocupa. Para su proporción se calcula la siguiente expresión [58]:

c_i	sMAPE asociado a cada cromosoma.
$f_i = f(c_i)$	Fitness del elemento i.
$P_i = P(c_i)$	Probabilidad de selección de elemento-i.
N	Tamaño de la población.

$$F(c_i) = \frac{\sum_{j=0}^{n} c_j}{c_i} \tag{4.53}$$

$$P(c_i) = \frac{F(c_i)}{\sum_{j=0}^{n} F(c_j)} * 360 \tag{4.54}$$

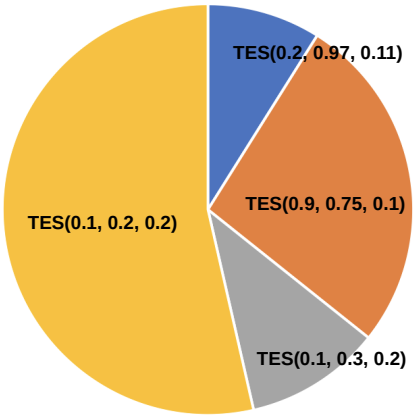


Figura 4.5: Ejemplo de Algoritmo de Ruleta

Se puede observar un ejemplo de la implementación del algoritmo de selección de Ruleta en la Fig.4.5, donde se ha considera el cromosoma para el algoritmo de triple alisamiento exponencial TES(α, β, γ) [41] y una población de 4 individuos. El cromosoma TES(0.1,0.2,02) tendrá más posibilidad de ser elegido a la hora de elegir los cromosomas para la selección. Una vez definida la probabilidad de que cada cromosoma sea seleccionado, se "hace girar"la ruleta dos veces, para elegir a una pareja de cromosomas. Se ha elegido este método frente a otros (por ejemplo, selección por torneo, selección por rango, selección por estado estacionario, etc.) ya que permite que los cromosomas con mejor función de adaptación aparezcan con más frecuencia. Además, permite seleccionar dos cromosomas iguales, es decir que sean el mismo progenitor, algo que perpetuará los mejores parámetros de ajuste y que luego se resolverá en la etapa de mutación para dar suficiente variedad a la población. Uno de los inconvenientes de este tipo de algoritmo de selección es que cuando las probabilidades de los algoritmos difieren con bastante notoriedad, predomina la selección de ciertas características frente al resto, lo que reduce la diversidad genética de la población. Pero tanto la implementación de otros algoritmos de

selección, como la problemática comentada anteriormente, se ha dejado para el estudio en futuras líneas de trabajo.

Para la combinación de la pareja de cromosomas seleccionados anteriormente se ha elegido la combinación cruce de un punto (del inglés *One-Point Crossover*). Esta consiste en elegir un punto aleatorio donde se cortan cada cromosoma progenitor, se copia la información genética del punto elegido de un padre a otro y viceversa lo que origina dos nuevos cromosomas resultado de dicha combinación. Para esta propuesta, los cromosomas no tienen una gran cantidad de genes, por lo que la forma de proceder ha sido la siguiente: se decide aleatoriamente un gen, denominado punto de intercambio y se intercambian los contenidos genéticos pivotando dicho punto entre ambos progenitores. Como consecuencia, el cromosoma descendiente sustituye al cromosoma padre con una función de adaptación menor. Esto permite mantener el mismo número de cromosomas en la población evolutiva.

- **Mutación**

  Tras el cruce o combinación, se produce la mutación de los cromosomas. En esta etapa, un gen aleatorio de los descendientes de la etapa anterior se reemplaza por un valor aleatorio. Esta etapa, en relación con términos evolutivos, sólo sucede de manera extraordinaria; en este caso, la probabilidad de que un gen mute viene definido por un parámetro de ajuste que se definirá en los siguientes apartados y permitirá decidir si un gen se muta o no, evitando que la búsqueda de la solución óptima sea una mera búsqueda aleatoria. El objetivo de esta etapa es dotar a la población evolutiva de diversidad genética. Tanto en esta etapa, como en la etapa anterior, se ha tenido en cuenta que los genes pueden presentar naturaleza diferente lo que limita las acciones a realizar sobre un mismo cromosoma. Esto límites son establecidos por el rango de datos del parámetro de ajuste. Por ejemplo, para el parámetro α en el algoritmo de predicción TES, el cual puede variar entre $0 \ldots 1$, las mutaciones aleatorias sobre este parámetro deben limitarse entre $0 \ldots 1$.

- **Condición de parada**

  Se han considerado dos posibles condiciones de parada para el GA:

  - Por un lado, y en el peor de los casos, cuando se alcanza a un número máximo y predefinido, de iteraciones. Nótese que este parámetro variará

en función a las prestaciones computacionales o la rapidez con la que se necesite una solución.

- Por otro lado, y en el mejor de los casos, cuando un individuo alcanza su estado "físico"óptimo, es decir, cuando el sMAPE asociado para el algoritmo de predicción a calibrar es 0.

- **Parámetros**

  En este apartado, se van a enumerar los diferentes parámetros configurables que dispone el GA, como se ha comentado anteriormente:

  - **Tamaño de la población inicial y de la población evolutiva (N):** Corresponde con el número de individuos (genotipos) que componen la población, este debe ser lo suficientemente rica como para garantizar la diversidad de todas las soluciones. En la implementación, la población evolutiva se queda con un número x (previamente determinado), de los mejores cromosomas de la población inicial o de la población resultante, donde x ¡N. En cada iteración se completa la población evolutiva con los x mejores y con una serie de cromosomas aleatorios hasta llegar al número N. Esto convierte la población en una selección elitista además de favorecer a la exploración de diferentes posibles soluciones de manera aleatoria.
  - **Porcentaje de combinación:** Este atributo corresponde con la probabilidad de que un par de cromosomas realicen la etapa de cruce o combinación. Este puede ser fijo o variable a lo largo de las iteraciones. Este parámetro además emula la evolución natural donde existe una probabilidad de que los individuos se crucen entre sí.
  - **Porcentaje de mutación:** Del mismo modo que en el atributo anterior, este parámetro determina la posibilidad de que un individuo mute o adquiera parámetros de ajuste de manera aleatoria en la fase de mutación.
  - **sMAPE mínimo:** Aunque anteriormente se ha definido el sMAPE ideal como sMAPE = 0 este se puede modificar, cuando no se necesite valores perfectos, sino que sirva con valores orientativos y aproximaciones lo más precisas posible.
  - **Número de iteraciones:** Como se ha comentado anteriormente, este parámetro de ajuste del GA dependerá de los recursos computacionales, así como de la rapidez con la que se necesite la solución óptima para la resolución del problema de calibrado.

Capítulo 5

Detección de DDoS mediante el estudio de comportamientos inesperados

Este capítulo describe la adaptación de la estrategia de predicción desarrollada a un caso de uso concreto: la detección de amenazas de denegación de servicio en escenarios de red. El esfuerzo realizado ha concluido en la propuesta de la herramienta DroidSentinel, cuyos principios de diseño, arquitectura, métricas, proceso de análisis y criterios de decisión son presentados a continuación.

5.1. Principios de diseño

Tal y como se ha descrito en el Capítulo 3, la defensa frente a los ataques DDoS puede abordarse desde diferentes perspectivas, que abarcan desde la prevención hasta la identificación del origen de las amenazas [61]. Además, dada la complejidad de los escenarios emergentes de red, pueden plantear una gran cantidad de desafíos, como la decisión del lugar de actuación de las medidas defensivas [59], la naturaleza de la información a modelar [60] o la implementación de políticas de gestión de seguridad [63]. Con el fin de facilitar la comprensión del trabajo realizado, DroidSentinel considera por objetivo principal el desarrollo de una estrategia de detección de ataques DDoS en el extremo origen adaptable a procesos no estacionarios en la información a analizar. La solución desarrollada ha de incorporar la estrategia de predicción adaptativa descrita en capítulos anteriores. Nótese que a diferencia de propuestas similares hacia la defensa frente a DDoS, los procesos analíticos implementados sólo han de considerar una única fuente de información, que es el dispositivo protegido [62].

5.2. Asunciones

Con el fin de delimitar y asentar las bases de la investigación realizada, se han asumido las siguientes premisas:

- La detección de la participación de un usuario final o de dispositivos IoT en ataques DDoS en base al estudio de las métricas agregadas de su tráfico entrante/saliente es posible. Esta es la *hipótesis alternativa* de la investigación, siendo su opuesto la *hipótesis nula*.

- Los ataques DoS basados en inundación principalmente se distinguen de la actividad normal en sus distribuciones de número de peticiones y volumen observado en los flujos de tráfico inyectados. En los ataques DDoS además varía el número de clientes involucrados [64].

- El estudio basado en el análisis de discordancia en métricas agregadas a nivel de flujo permite el reconocimiento de actividades DDoS en escenarios convencionales [65].

- La extracción de métricas avanzadas y su análisis en un servidor dedicado reduce considerablemente su impacto en el sistema protegido.

- Se asume la no estacionalidad de la información inferida a partir de flujos de tráfico entrante/saliente de los dispositivos de la red, ya que ésta depende en su mayor parte de los hábitos del usuario.

5.3. Limitaciones

El ámbito del trabajo realizado ha sido delimitado por las siguientes restricciones, la mayoría de ellas pospuestas para futuras investigaciones:

- No se ha tenido en cuenta la protección de los canales de comunicación frente a ataques hacia la integridad, disponibilidad y confidencialidad de la información que transmiten [66]. En consecuencia, durante la investigación se asume que estos canales no han sido comprometidos.

- A pesar de que SELFNET ofrece capacidades avanzadas de correlación de incidencias y actuación, su aprovechamiento queda fuera del alcance de esta contribución. Esto supone una interesante línea de trabajo futuro.

- Aunque en la actualidad existen diferentes estrategias para la evasión de métodos de detección similares a los implementados, no se ha profundizado en los mecanismos adoptados para su prevención [65]. Sin embargo, dada la complejidad que a menudo implica su desarrollo, y con el objetivo de facilitar la comprensión de la principal contribución de nuestra investigación, su adopción está fuera del alcance de esta publicación. Esto incluye enfoques obstaculizadores como la suplantación de direcciones de red, suponiendo que se implementan soluciones similares a las descritas en [72].

- No se ha considerado el problema de la protección de información sensible inherente a las actividades de red compartidas por los usuarios. Tampoco se ha tenido en cuenta la implementación del reciente reglamento general europeo de protección de datos o GDPR (del inglés *EU General Data Protection Regulation*). En consecuencia, se supone que DroidSentinel tiene permiso para monitorizar el tráfico entrante/saliente de los dispositivos de red con fines puramente analíticos.

- No se profundiza en la representación del conocimiento ni en los modelos de datos implementados para la gestión y almacenamiento de la información recolectada.

5.4. Arquitectura

En la Fig. 5.1 se muestran los componentes de la arquitectura DroidSentinel, cuya estructura de capas adopta los principios de SELFNET. Los dispositivos de usuario llevan a cabo la extracción de métricas agregadas que son enviadas a través de una interfaz de alto nivel a la Capa de Análisis. Esta capa lleva a cabo el proceso de reconocimiento de posibles amenazas DDoS, desarrollado en tres etapas: Monitorización, Predicción y Detección. Debido a que la Capa de Análisis centraliza la labor de detección, su despliegue debe ser escalable a múltiples instancias. Por otra parte, la Capa de Entrenamiento actúa como módulo auxiliar para la generación del modelo de clasificación, que es utilizado en la selección del algoritmo predictivo. Finalmente, los resultados de la detección son notificados a los dispositivos como respuesta a los envíos de métricas monitorizadas, completando así el ciclo de detección.

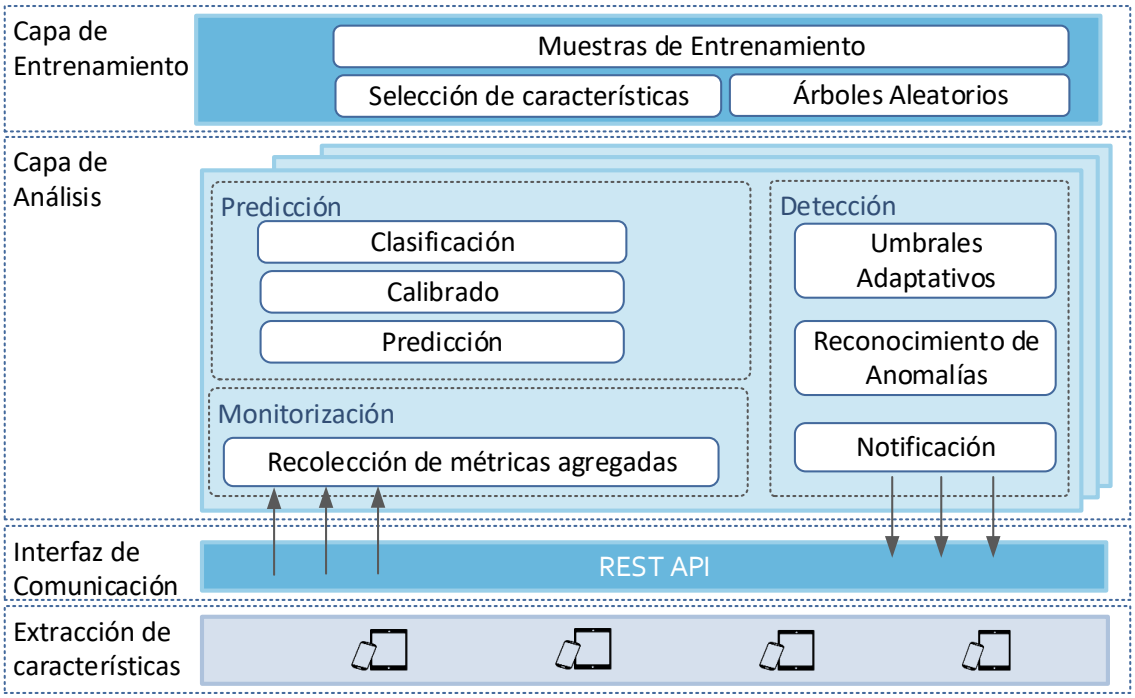


Figura 5.1: Arquitectura de DroidSentinel

5.5. Indicadores DDoS

A lo largo de la investigación realizada se han estudiado diferentes niveles de procesamiento de la información, lo que conlleva la necesidad de extraer características heterogéneas que faciliten el análisis del conocimiento adquirido de los dispositivos monitorizados, que se analiza en forma de serie temporal. Estos son resumidos en la Tabla 5.1 y descritos a lo largo de esta sección.

5.5.1. Características de las series temporales

La primera etapa analítica de DroidSentinel se centra en la extracción de características que permiten definir modelos de uso adaptables a cambios en el entorno monitorizado. Para facilitar la tarea de decidir las estrategias de modelado y predicción más adecuadas, se ha utilizado la herramienta TSFRESH, desarrollada bajo el proyecto iPRODICT [67], el cual construye más de 100 características por serie temporal. Este programa tiene en cuenta desde atributos estadísticos básicos (picos, observaciones máximas/mínimas, modo, etc.) hasta medidas de correlación relacionadas con la evolución de las series temporales (ruido blanco, tendencia, estacionalidad, coeficientes de autocorrelación, etc.). Estas características se aplicaron directamente sobre el M3-Competition [68] en la etapa de entrenamiento del sistema.

Los flujos de tráfico entrantes/salientes del dispositivo protegido se supervisan y estructuran en formato IPFIX [69], según el cual cada flujo de tráfico es un conjunto de paquetes capturados en un cierto intervalo de tiempo t. Comparten las siguientes propiedades: misma dirección IP de origen, dirección IP destino, y protocolo. Los intervalos de tiempo que delimitan los flujos de tráfico establecen la granularidad de las tareas analíticas que se realizarán, de este modo sirven como parámetros de ajuste para configurar el nivel de sensibilidad de los métodos de detección. Por ejemplo, cuando la granularidad es alta, la información a procesar apenas se filtra o suaviza, ya que generalmente se toman menos instancias (paquetes) por cada intervalo de tiempo t (observación). Como resultado, estas observaciones son más propensas a presentar valores atípicos o ruido. Sin embargo, cuando la granularidad es demasiado baja, es posible que las tareas analíticas pasen por alto situaciones relevantes. El primero de estos escenarios da como resultado un ajuste más restrictivo, donde se prioriza la detección de amenazas en oposición a la generación de falsos positivos. En el segundo caso, se prioriza la calidad de la experiencia del usuario a expensas de disminuir el nivel de protección ofrecido. El siguiente par de mediciones se toma por flujo de tráfico: cantidad de paquetes transferidos y cantidad total de información transferida (bytes). De ellos se infiere las métricas agregadas que se describen en la siguiente subsección. Dado que la comparativa entre las características del tráfico saliente y entrante ha sido objeto de estudio de una gran parte de los trabajos de la bibliografía, DroidSentinel también la ha tenido en cuenta por medio de su error cuadrático medio normalizado o $nMSE$ (del inglés *normalized Median Square Error*), expresado de la siguiente manera:

$$nMSE = \frac{\frac{1}{n}\sum_{i=1}(x(a)_i - \hat{x}(b)_i)^2}{\sigma^2} \tag{5.1}$$

donde X es el rasgo a analizar, n es el número total de flujos de tráfico de pares IP origen e IP destino (es decir, el tráfico entrante / saliente entre a y b) $x(a)_i$, es la métrica registrada en el tráfico entrante agrupada en el flujo a, y $x(b)_i$ la métrica registrada en el tráfico saliente en b. Un claro ejemplo se ilustra en la relación que describe la diferencia entre los paquetes entrantes $E_\tau(nP_{in},\ nP_{out})$ y los paquetes salientes $X_{in}(a) = nP_{out}(b)$ capturados en el intervalo de tiempo t.

Por otro lado, el grado de desorden de las observaciones se mide en base a la entropía normalizada de Shannon. Esta decisión está respaldada por trabajos de

Tabla 5.1: Métricas

Nivel	Clase	Expresión	Descripción
Serie Temporal	Características	Ts[...]	Generado con TSFRESH
Flujos de tráfico	Total	nP	Número total de paquetes
		nPin	Número total de paquetes entrantes
		nPout	Número total de paquetes salientes
		nB	Número total de bytes
		nBin	Número total de bytes entrantes
		nBout	Número total de bytes salientes
Agregación	Desorden	H(nP)	Entropía del número de paquetes por flujo
		H(nPin)	Entropía del número de paquetes entrantes por flujo
		H(nPout)	Entropía del número de paquetes salientes por flujo
		H(nB)	Entropía del número de bytes por flujo
		H(nBin)	Entropía del número de bytes entrantes por flujo
		H(nBout)	Entropía del número de bytes salientes por flujo
	Distancia	nMSE(nP)	Diferencia de paquetes entrantes y salientes
		nMSE(nB)	Diferencia de bytes entrantes y salientes

investigación previos relacionados con el reconocimiento DDoS, que abordaron con éxito problemas similares [60]. Asumimos que esta métrica también es válida para la detección en dispositivos IoT desde el lado de la fuente. Como en el caso de la bibliografía, la entropía implementada por DroidSentinel se deduce de la siguiente expresión:

$$H(X) = \frac{-\sum_{i=1}^{n} p_i \log_a p_i}{\log_a n} \tag{5.2}$$

donde n es el número total de flujos monitorizados capturados en el intervalo de tiempo t, y τ, and $p_1, p_2, \ldots, p_n$ son las probabilidades de las instancias $x_1, x_2, \ldots, x_n$ de la variable aleatoria X, la última construida a partir de las métricas de nivel de flujo básico. Por ejemplo, existe un desorden de bytes por flujo $H(nB)_T$ en el intervalo de tiempo T si para $H(nB)_T = 0$ es posible afirmar que X_T es determinista. En el caso opuesto, se produce $H(nB)_T = 1$ para X_T, cuando se registra el grado máximo de desorden.

5.6. Estimación de la evaluación de las métricas agregadas

Las evoluciones de las métricas extraídas del tráfico saliente/entrante de los dispositivos monitorizados permiten reconocer situaciones inesperadas, y, por lo tanto, anómalas. Con este fin, estas son analizadas por medio del marco de predicción propuesto en el Capítulo anterior. Esta herramienta recibe como datos de entrada las series temporales compuestas por las métricas de detección, y devuelve su estimación en un horizonte de tiempo determinado; en particular, el horizonte en el que el algoritmo genético determine una mayor precisión. Nótese que la herramienta de predicción ha sido previamente entrenada a partir de la colección de muestras *M3-Competition* y la batería de algoritmos previamente descrita. Las predicciones realizadas permitirán que la etapa de Clasificación determine el nivel de discordancia de las observaciones realizadas, y, por lo tanto, su similitud con el modo de uso normal y legítimo del dispositivo.

5.7. Clasificación

En la etapa de clasificación de DroidSentinel, se decide la naturaleza de la serie temporal basándose en métricas agregadas construidas a partir de los flujos de tráfico monitorizados. En este contexto, se supone que una observación es un valor atípico si coincide con un comportamiento inesperado, es decir, cuando la variación entre un pronóstico en cierto horizonte de tiempo y el valor observado difieren significativamente. Debido a que la proyección de valores continuos en el tiempo tiende a producir errores, el principal desafío de este proceso es definir su relevancia, que se gestiona mediante la definición de umbrales adaptativos. A continuación, los valores atípicos se etiquetan como posibles comportamientos maliciosos y las situaciones normales se clasifican como legítimas, por lo que la implementación actual de DroidSentinel actúa como un clasificador binario.

El marco del Analizador SELFNET [70] proporciona capacidades analíticas avanzadas relacionadas con la construcción de intervalos de predicción, la mayoría de ellos ampliamente aceptados por la comunidad de investigación para el estudio del tráfico de red. De entre ellos, DroidSentinel integra la metodología de umbrales adaptativos descrita en [71], donde son definidos por las siguientes expresiones:

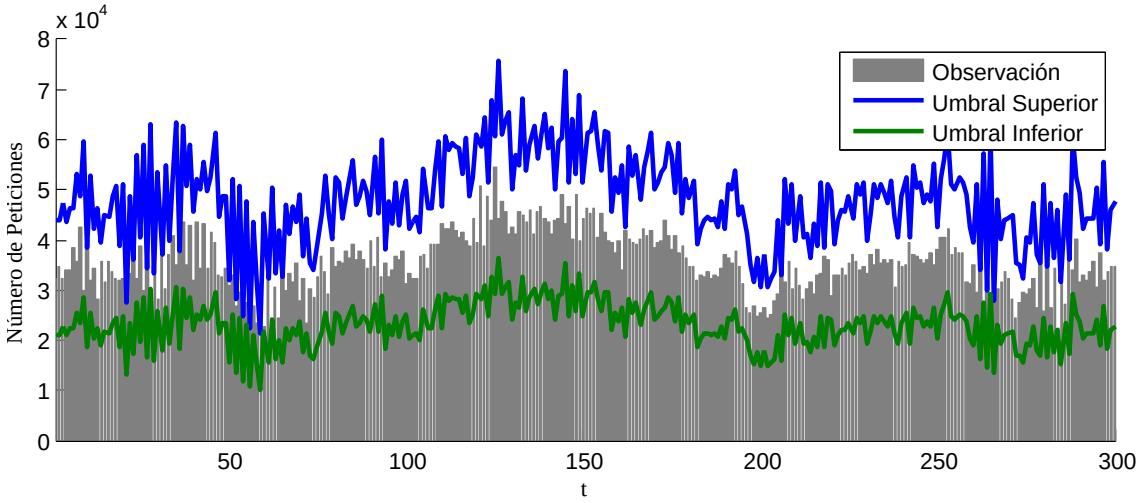


Figura 5.2: Ejemplo de identificación de valores atípicos

$$Ath_{up} = \hat{x}_{n+1} + K\sqrt{\sigma^2(E_t)} \tag{5.3}$$

$$Ath_{down} = \hat{x}_{n+1} - K\sqrt{\sigma^2(E_t)} \tag{5.4}$$

donde $\hat{x}_{n+1}$ es la predicción de cierta métrica agregada x en el horizonte $n+1$, E_t es la distancia euclidiana entre $\hat{x}_{n+1}$ y x_{n+1}, y K es el parámetro de ajuste que configura la restrictividad del sensor. Las ecuaciones distinguen un umbral superior y un umbral inferior Ath_{low},ambos adaptados a t. Se espera que a mayor valor de K, mayor tolerancia al ruido, ya que esta situación expande el margen de error entre $\hat{x}_{n+1}$ y x_{n+1}. En el caso opuesto, DroidSentinel aumenta el nivel de protección, que generalmente ocurre a expensas de penalizar la tasa de falsos positivos. La Fig. 5.2 ilustra un ejemplo de valor atípico inducido por un ataque basado en inundaciones DDoS, donde en $T = 41$ se genera un punto comprometido puesto que se inyectan una gran cantidad de solicitudes HTTP. Durante el ataque, tanto Ath_{lup} como Ath_{low} se exceden repetidamente, lo que lleva a etiquetar el tráfico como potencialmente malicioso.

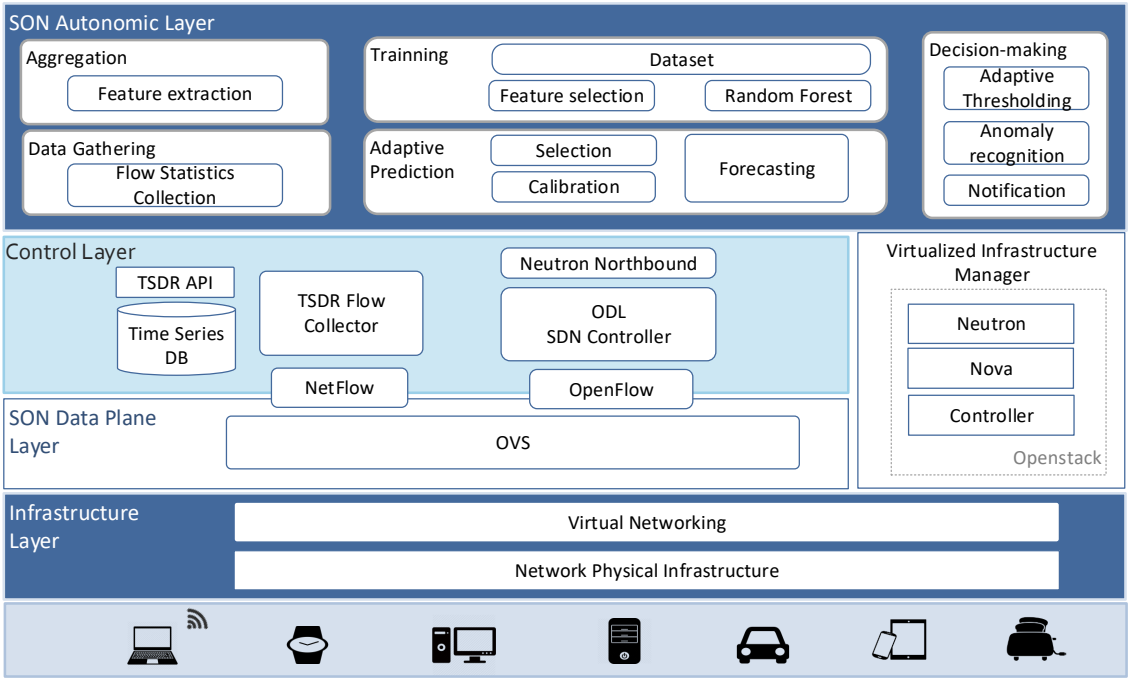


Figura 5.3: Arquitectura DroidSentinel adaptada a los escenarios de comunicación emergentes

5.8. Despliegue en escenarios 5G

Finalmente, cabe destacar que, con el objetivo de probar la estrategia desarrollada en un escenario de red emergente, y a modo de contribución adicional a los esfuerzos realizados durante el desarrollo de este trabajo, se ha llevado a cabo el despliegue de DroidSentinel sobre un entorno auto-organizativo adaptado al advenimiento de las redes 5G: el proyecto SELFNET. Esta tarea ha sido posible gracias a la colaboración directa de integrantes del consorcio SELFNET, quienes ha contribuido de manera activa y directa en su instanciación. El resultado se ilustra en Fig. 5.3, donde la tecnología SDN permite el desacoplamiento de las capas de control y plano de datos, siendo esta una característica notable de las redes de próxima generación [73]. El principal beneficio de este modelo es la inclusión de tareas complejas de procesamiento de datos en la Capa Autónoma SON, encargada de la gestión inteligente y autoorganizadas de incidencias en redes [74].

Capítulo 6

Experimentación

Este Capítulo describe cómo se han evaluado las propuestas realizadas, tanto la estrategia de predicción adaptativa, como su instanciación para la detección de amenazas DDoS en DroidSentinel. Para ello, se ha utilizado una metodología de evaluación experimental. Con este objetivo, se han implementado diferentes conjuntos de pruebas adaptados a las peculiaridades de los distintos enfoques. Por otro lado, se han considerado estándares funcionales que han permitido comparar los resultados preliminares con otros proyectos relacionados.

6.1. Evaluación de la estrategia de predicción

La eficacia del método de predicción adaptativa propuesto ha sido probada en base a la colección de muestras M3-Competition, tal y como se define a continuación.

6.2. M3-Competition

Uno de los objetivos principales de este proyecto es la implementación de una propuesta de predicción adaptativa capaz de ser implementada en entornos de redes 5G, en concreto capaz de ser integrado en proyecto SELFNET. Actualmente no existe ninguna metodología estándar capaz de evaluar la efectividad de los algoritmos de predicción en dichos entornos. Por ello, la forma más fiable capaz de demostrar la capacidad de la propuesta es evaluarlo a partir de metodologías de propósito general adaptadas a la predicción de series temporales, así como de la primera versión del marco de predicción integrado en el componente de análisis de SELFNET. Para ello, se ha utilizado el esquema M3-Competition cuyo dataset y metodología de evaluación son detallados en los siguientes apartados.

Tabla 6.1: Resumen de las muestras en M3-Competition

	Naturaleza de los datos						
	Micro.	Ind.	Macro.	Finanzas	Demo.	Otras	Total
Anual	146	102	83	58	245	11	645
Trimestral	204	83	336	76	57		756
Mensual	474	334	312	145	111	141	1428
Otras	4			29		141	174
Total	828	519	731	308	413	204	3003

6.2.1. Dataset

La colección M3-Competition está formada por un total de 3003 series temporales de diferente naturaleza, de las que se puede destacar, ámbitos financieros, industriales, macroeconómicos, etc. (ver Tabla 6.1). Para asegurar que los algoritmos de predicción tuvieran la capacidad de procesar el conjunto de datos, se definió como longitud mínima para cada tipo de observación: un total de 14 observaciones para series anuales (donde la media por observación es en torno a 19 muestras), 16 para series trimestrales (la media es de 44 observaciones), 48 para series temporales mensuales (la media es de 115 observaciones) y 60 para otras series (donde la media es 63 observaciones). Por todo ello, se ha considerado únicamente tres bloques de datos: anual, trimestral y mensual. Además, todas las series temporales consideradas son positivas, para evitar problemas en las metodologías de evaluación. En la Tabla 6.1 se muestra la clasificación que se ha tenido en cuenta para las diferentes series temporales.

6.2.2. Metodología de Evaluación

Siguiendo la metodología utilizada en la M3-Competition, se han ejecutado los algoritmos de predicción considerando diferentes horizontes (es decir, periodos de predicción). En concreto, para datos anuales: $t+1$ a $t+6$, para datos trimestrales: $t+1$ a $t+8$ y, por último, para datos mensuales: $t+1$ a $t+18$.

Para la evaluación de cada conjunto de datos se ha utilizado las cinco métricas definidas en M3-Competition: error de porcentaje absoluto medio o MAPE (del inglés *Symmetric MAPE*), clasificación media, media simétrica, el mejor porcentaje y error absoluto relativo. Por otro lado, para la evaluación de cada algoritmo de predicción en función a cada conjunto de observaciones se ha usado sMAPE.

6.2.3. Experimentación

Para la experimentación se ha considerado la metodología seguida por el banco de pruebas de evaluación de SEFNET, que consiste en la evaluación de la herramienta para el conjunto de datos proporcionado por el conjunto M3-Competition. El objetivo principal de esta experimentación es comparar la propuesta presentada anteriormente con otras metodologías de predicción como Naive, Holt, Dampen, pero, sobre todo, comparar la efectividad del trabajo realizado frente al Framework de predicción de SELFNET original, el cual carece de la capacidad de adaptación a procesos no estacionarios.

6.3. Evaluación de DroidSentinel en escenarios de red convencionales

Para esta prueba se ha considerado la versión original de la propuesta. Su objetivo es evaluar la estrategia adaptativa de predicción, instanciada para detectar comportamientos inesperados que desenmascaren ataques de denegación de servicio en el extremo origen. Es importante resaltar que es este escenario, la extracción y posterior análisis de las series temporales se hacen en el propio dispositivo a proteger. La extracción de los datasets utilizados para la experimentación, así como la metodología de evaluación y las pruebas realizadas se comentan a continuación.

6.3.1. Dataset

La colección de pruebas reúne muestras de tráfico legítimo saliente capturado en 35 dispositivos pertenecientes a usuarios distintos, todos ellos alumnos de la facultad de Informática de la Universidad de Madrid. Para la extracción de los dataset se implementó la herramienta Varys, que se detallará en las siguientes secciones, y que permitió la generación de los primeros datasets para su posterior análisis. Estas muestras fueron tomadas en distintos periodos de monitorización, separados en intervalos de 1,3 y 5 días en diferente franja horaria. Por motivos de privacidad, cada muestra publicada contiene únicamente las métricas básicas y agregadas consideradas en el estudio realizado. Se ha considerado una granularidad de 3 minutos por observación y una longitud de 120 observaciones por serie temporal, resultando una colección de 210 muestras. Principalmente recopilan actividades normales de un usuario, como búsquedas en internet, uso de servicios de vídeo y audio en streaming, subida de ficheros a la nube. etc. Además, se ha implementado la herramienta

TrafficGenerator, capaz de simular navegación HTTP (del inglés *web scraping*).

Para obtener muestras de tráfico anómalo, se pidió a los usuarios que periódicamente ejecutan ataques de denegación de servicio de 12 minutos de tipo mediante la inyección de tráfico TCP o UDP (4 observaciones) contra extremos de red virtualizados ubicados en un entorno aislado. En total se generaron 70 muestras de ataque. El tráfico era indistintamente dirigido contra un único punto (DoS) o contra varios de ellos (DDoS) para lo cual se valieron de varias herramientas de código abierto como Warchild o TCP Attack [75]. Cabe resaltar con el fin de garantizar la replicabilidad de la experimentación, el banco de pruebas, dataset y las herramientas utilizadas para su gestión están plenamente disponibles en [10].

Varys

Varys [10] es una herramienta de Código Abierto (GPL) desarrollada en Java que se divide en dos módulos principales:

- **Varys Sensor:** En primer lugar, Varys implementa un sensor que monitoriza el tráfico de la red y guarda todos los paquetes compactados en opcode, en una carpeta que los paquetes de red generados/recibidos por el dispositivo. Desde este es posible aplicar reglas de filtrados basadas en [76]. Este módulo implementa la librería de código abierto pcap4j [77].
- **Varys Dataset:** Por otro lado, para el procesado de los paquetes recopilados con el módulo anterior, se ha implementado un módulo que permite analizar el tráfico generado y extraer tanto las métricas básicas como las métricas agregadas. En concreto permite analizar el número de paquetes enviados y recibidos, el número de bytes enviados y recibidos, la diferencia entre el número de paquetes enviados y recibidos, la diferencia entre el número de bytes enviados y recibidos y la entropía de las métricas anteriores.

La herramienta ha sido desarrollada en java, con el objetivo de ser ejecutada en cualquier tipo de dispositivo.

TrafficGenerator

TrafficGenerator [10] es una herramienta de código abierto y licencia GPL desarrollada en Python con el fin de dotar a la comunidad investigadora de una herramienta capaz de generar tráfico legítimo a través de la navegación en diferentes servicios HTTP. El objetivo de esta herramienta es emular el comportamiento de

usuarios reales, lo que permite automatizar algunas tareas y crear datasets con una mayor cantidad de muestras. Es importante destacar que algunas de las funciones implementadas se inspiraron en el curso "Show Me The Data"[78] impartido en la Universidad Complutense de Madrid.

Esta herramienta ha sido desarrollada como un Script de Python que utiliza principalmente las librerías Selenium [78], BeautifulSoup [80] y Request [81] así como otras librerías como WebDriverWait, TimeoutException y Random. La herramienta lanza una sesión de Mozilla Firefox dónde se produce la emulación del comportamiento del usuario.

Su modus operandi implementa un bucle, que se detiene una vez decida la actividad a realizar. Estas pueden ser: visualización de tráiler de películas aleatorias en www.filmin.es, visualización de recetas de cocina aleatorias en la página www.yummly.com o visualización de libros ofrecidos por www.books.toscrape.com.

6.3.2. Metodología de Evaluación

Tal y como se indicó previamente, para evaluar la efectividad de la propuesta se ha utilizado una metodología de evaluación experimental. Ésta consiste en medir el impacto sobre la efectividad de DroidSentinel producto de la variación de los siguientes parámetros de ajuste: métricas básicas, métricas agregadas y ajuste del nivel de restricción de los umbrales predictivos.

Las actividades monitorizadas se han etiquetado de manera dicotómica: muestras legítimas (normales) y maliciosas (discordantes). En analogía con publicaciones anteriores, se ha tenido en cuenta DroidSentinel como un clasificador binario, por lo que, se basa en observar la sensibilidad, que determina la capacidad de señalar correctamente las anomalías como maliciosas frente a la especificidad, que mide la capacidad de reconocer las actividades normales como legítimas [71]. Para ello, se han representado los resultados para las distintas métricas analizadas sobre el espacio ROC (del inglés *Receiver Operating Characteristic*). En base a ello, se ha determinado varios indicadores de efectividad, entre los que destacan según su relevancia: Área Bajo la Curva (AUC) (del inglés *Area Under the Cruve*), Tasa de Positivos Reales (TPR) (del inglés *True Positive Rate*) y Tasa de Falsos Positivos (FPR) (del inglés *False Positive Rate*) en función del mejor ajuste del sensor en términos de K. Como es frecuente en la bibliografía, el ajuste óptimo coincide con la posición en la curva ROC que muestre el mejor índice de Youden (Y) [82] cuyo

rango oscila entre -1 (el peor ajuste) y 1 (el ajuste óptimo).

6.3.3. Experimentación

En base a los criterios comentados anteriormente, se han realizado los siguientes experimentos:

- Evaluación de la eficacia de las métricas básicas, donde se ha considerado las siguientes métricas: número de paquetes enviados, número de bytes enviados y número de IP's destino diferentes.

- Evaluación de la eficacia de las métricas agregadas. Para la realización de esta prueba se ha considerado la entropía de Shannon [87] de las métricas comentados anteriormente: número de paquetes enviados y número de bytes enviados. El objetivo de esta prueba es reducir la tasa de falsos positivos acaecidos por la no estacionalidad de las series temporales a analizar. Además, se ha añadido otras métricas para el mismo fin, como la distancia euclidiana entre el número de paquetes recibidos y el número de paquetes enviados y la distancia euclidiana entre el número de bytes recibidos y el número de paquetes enviados. Tanto la fórmula de la Entropía de Shannon como la fórmula de la distancia euclidiana se detallan a continuación:

  Entropía de Shannon:

$$(X) = \sum_{i=1}^{n} P(X_i)\, I(X_i) = -\sum_{i=1}^{n} P(X_i)\, log_b P(X_i) \tag{6.1}$$

  Distancia Euclidiana:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2} \tag{6.2}$$

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{6.3}$$

6.4. DrodiSentinel en escenarios 5G

Para la segunda experimentación, se ha considerado la arquitectura de DroidSentinel basado en una solución SON dentro del marco del proyecto SELFNET. Cabe destacar que esta prueba es una contribución adicional al trabajo realizado, y que ha contado con el apoyo directo de algunos de los miembros que participan en el proyecto SELFNET.

Durante su transcurso se han realizado diferentes pruebas sobre trazas de tráfico monitorizadas en dispositivos finales (del inglés *end-points*) de diferente naturaleza. Como en la experimentación anterior se ha considerado una metodología de evaluación experimental, donde a continuación, se va a detallar el dataset utilizado, la metodología de avaluación y las pruebas llevas a cabo.

6.4.1. Dataset

A continuación, se va a detallar la colección de muestras recopiladas que han sido utilizadas para su posterior evaluación y análisis. Esta colección de pruebas se ha dividido en dos: muestras obtenidas del tráfico normal de usuarios (tráfico legítimo) y tráfico perteneciente a actividades maliciosas que han formado parte de ataques de denegación de servicio (tráfico DoS). Además, estos datasets se han clasificación en función de la actividad que han realizado y de la familia a la que pertenece el dispositivo end-point utilizado para la creación de los mismos. En el Anexo 1 se incluye información adicional sobre los dispositivos considerados y sus actividades habituales. Por último, con el objetivo de fomentar la investigación de este tema, se ha de destacar que el conjunto de muestras generado está también disponible en [10] para su futuro análisis y replicación de la experimentación practicada.

- **Tráfico Legítimo:** Para generar el dataset de tráfico legítimo, se han obtenido capturas de tráfico saliente de 58 dispositivos diferentes. Cada muestra se ha creado a partir de monitorizaciones de tráficos divididos en tres periodos de tiempo: 1, 3 y 5 días lo que ha supuesto una cantidad definitiva de 150 instancias de 3 horas por dispositivo, lo que implica que el conjunto de datos contenga 8,700 muestras de tráfico normal.

- **Tráfico DoS:** Al final de cada captura de tráfico normal, se lanzaron diversos ataques de DDoS a través de las herramientas descritas en [83] [84]. Estos ataques consisten en inyecciones de tráfico basadas en inundaciones de UDP, HTTP o TCP con intensidades: baja, media y alta. Por ello, el conjunto de

Tabla 6.2: Clasificación en función de la actividad

Actividad	Dispositivos	Muestras	p-ADF
Actividades diarias de un usuario.	18	2,7	0.103
Navegación sintética	22	1202,1	0.0225
Streaming	18	901,95	0.0262

datos proporcionado consta de 78,300 muestras con contenido malicioso, es decir, 26,100 por cada intensidad.

- **Actividades:** Las actividades que representan el dataset se han resumido en la Tabla 6.2. Entre ellas destacan cuatro grandes grupos de actividades diarias del usuario, navegación web sintética con varias herramientas de automatización, y transmisión multimedia (audio y video). Además, en la Tabla 6.2 se muestra el valor promedio de P en la prueba de Dickey-Fuller (ADF) [85], que sirve para determinar el grado de no estacionariedad de cada perfil de tráfico analizado. Los valores de P inferiores a 0.05 determinan observaciones estacionarias, lo que nos lleva a asumir que la mayoría de los dispositivos analizados se comportan como fuentes de datos no estacionarios.

  - Actividades diarias de un Usuario: Representan actividades de propósito general, coinciden con trabajo de oficina misceláneo, ya sea búsquedas de artículos en internet, realización de traducciones esporádicas, consulta de redes sociales, periódicos, almacenamiento en la nube etc. Estas muestras han sido recogidas por voluntarios en el día a día de sus dispositivos.
  - Navegación web sintética: Estos grupos contienen principalmente tráfico web generado de navegación web aleatoria a través de subprocesos que simulan navegación HTTP (del inglés *web scraping*). Este tráfico ha sido generado por robots concretamente las herramientas: "Internet Noise"[86],"Noiszy"[87] y "TrackMeNot"[88].
  - Trasmisión multimedia: Los grupos de transmisión representan el tráfico generado de dispositivos utilizando los principales servicios de transmisión multimedia, enfatizando aquellos relacionados con los contenidos de audio (Spotify, Apple Music, etc.) y de video (Youtube, Twitch, etc.).

- **Familias de dispositivos:** Las 6 familias consideradas durante la fase de experimentación se han recogido en la Tabla 6.3. A estas familias, pertenecen ordenadores de sobremesa, ordenadores portátiles o laptops, teléfonos móviles (del inglés *smartphones*), tabletas, relojes inteligentes (del inglés *smartwatches*) y televisores inteligentes (del inglés *smart TVs*). Dado que, en términos

Tabla 6.3: Clasificación de los dispositivos en función a su familia

End-point	End-points	Normal	Ataque
Ordenador de Sobremesa	24	3,6	32,4
Notebook	18	2,7	24,3
Smartphone	8	1,2	10,8
Tableta	5	750	6,75
Smartwatch	2	300	2,7
Smart TV	1	150	1,35

de modelado de tráfico, el tipo de end-point tiene menos impacto que su modelo de uso, el estudio llevado a cabo se ha centrado primordialmente en su comportamiento. B.

6.4.2. Metodología de Evaluación

De manera análoga a la experimentación de la sección anterior, se ha utilizado de manera similar la metodología de evaluación, donde se ha medido el impacto sobra la efectividad de los siguientes parámetros de ajuste: métrica, nivel de restricción, granularidad e intensidad del ataque. Del mismo modo, las actividades monitorizadas se han etiquetado como normales (legítimas) y anómalas (sospechosas) por lo que se ha considerado la herramienta como un clasificador binario, lo cual es llevado a cabo desde la perspectiva que ofrecen las métricas de sensibilidad y especificidad y su relación, tal y como es frecuente en la bibliografía, estimada mediante el índice de Youden [82] comentado anteriormente.

6.4.3. Experimentación

En base a los criterios comentados anteriormente, se han realizado experimentos que miden las siguientes características: impacto de la granularidad de los datos, impacto de la activad de los dispositivos, e impacto de la intensidad del ataque.

- **Impacto de la Granularidad:** Para analizar la precisión del sensor, en este experimento se han considerado los intervalos de tiempo que se midieron al estudiar los flujos de tráfico capturados: 7,5 segundos, 15 segundos, 30 segundos, 1 minuto, 2 minutos y 3 minutos. Por lo tanto, sólo se ha enfocado en el intervalo de monitorización y el ajuste del parámetro K para la calibración del umbral adaptativo.

- **Impacto de la Actividad de los dispositivos:** Para este experimento, se han considerado para las actividades descritas en la tabla de actividades la

mejor granularidad obtenida en la prueba anterior. Por lo tanto, este experimento analiza la precisión de DroidSentinel en función de las actividades que normalmente realizan los dispositivos comprometidos.

- **Impacto de la Intensidad del ataque:** Por último, para la realización de esta prueba se ha considerado la intensidad con la que ha sido lanzado la amenaza de DoS basada en inundación. El objetivo de esta prueba es probar la dificultad que tiene la herramienta en detectar la existencia de un ataque en función al protocolo (HTTP, TCP, UDP) y su capacidad de inundación (baja, media o alta).

Capítulo 7

Resultados

Este Capítulo describe y discute los resultados obtenidos durante la experimentación realizada. Por lo tanto, se profundizará en la eficacia de la propuesta al ser evaluada bajo el estándar funcional de evaluación M3-Competition. También se demostrará su capacidad de instanciación para la predicción de indicadores propios de amenazas DDoS, a partir de la cual es posible el descubrimiento de comportamientos discordantes que permitan su detección.

7.1. M3-Competition

Como se ha comentado en el capítulo anterior, la experimentación basada en el estándar M3-Competition, se ha dividido en diferentes series temporales, en concreto, de manera anual, trimestral, mensual y otras clasificaciones. Los resultados de diferentes métodos de predicción se muestran a continuación. Por cada uno, se ha calculado la media de los valores de sMAPE para un horizonte de pronóstico dado: de $t+1$ hasta $t+18$ en función de la naturaleza de serie temporal. A diferencia que, en las otras pruebas realizadas con este dataset, para esta propuesta, se ha considerado la aleatoriedad del algoritmo genético, realizándose un total de 100 pruebas para cada serie temporal y analizado en promedio de la totalidad.

7.1.1. Observaciones anuales

Los resultados obtenidos en la evaluación de la propuesta sobre el dataset anual se detallan en la Tabla 7.1. Han sido un total de 645 series temporales diferentes cuyos resultados para la propuesta oscilan entre 6,3 y 7,9, observando así una mejor precisión, tanto en los resultados ofrecidos por M3-Competition como los resultados proporcionados por el Framework de predicción de Selfnet. En consecuencia, se ha

Tabla 7.1: SMAPE para el dataset anual de M3-Competition

Método	Horizonte de Predicción						Promedio		#Obs
	T+1	T+2	T+3	T+4	T+5	T+6	1 a 4	1 a 6	
Naive	8.5	13.2	17.8	19.9	23	24.9	14.85	17.88	645
Single	8.5	13.3	17.6	19.8	22.8	24.8	14.82	17.82	645
Holt	8.3	13.7	19	22	25.2	27.3	15.77	19.27	645
Dampen	8	12.4	17	19.3	22.3	24	14.19	17.18	645
Winter	8.3	13.7	19	20	25.2	27.3	15.77	19.27	645
Comb S-H-D	7.9	12.4	16.9	24.1	22.2	23.7	14.11	17.07	645
B-J automatic	8.6	13	17.5	18.2	22.8	24.5	14.78	17.73	645
Autobox 1	10.1	15.2	20.8	22.5	28.1	31.2	17.57	21.59	645
Autobox 2	8	12.2	16.2	19	21.2	23.3	13.65	16.52	645
Autobox 3	10.7	15.1	20	20.4	25.7	28.1	17.09	20.36	645
Robust-Trend	7.6	11.8	16.6	20.3	22.1	23.5	13.75	16.78	645
ARARMA	9	13.4	17.9	19.1	23.8	25.7	15.17	18.36	645
Automat ANN	9.2	13.2	17.5	19.7	23.2	25.4	15.04	18.13	645
Flores/Pearce 1	8.4	12.5	16.9	19.1	22.2	24.2	14.22	17.21	645
Flores/Peace 2	10.3	13.6	17.6	19.7	21.9	23.9	15.31	17.84	645
PP-autocast	8	12.3	16.9	19.1	22.1	23.9	14.08	17.05	645
ForecastPro	8.3	12.2	16.8	19.3	22.2	24.1	14.15	17.14	645
SmartFcs	9.5	13	17.5	19.9	22.1	24.1	14.95	17.68	645
Theta-sm	8	12.6	17.5	20.2	13.4	25.4	14.6	17.87	645
Theta	8	12.2	16.7	19.2	21.7	23.6	14.02	16.9	645
RBF	8.2	12.1	16.4	18.3	20.8	22.7	13.75	16.42	645
ForecastX	8.6	12.4	16.1	18.2	21	22.7	13.8	16.48	645
Selfnet	6.9	6.6	7.6	7.2	8.5	9.4	7.1	7.7	645
Propuesta	**6.3**	**6.5**	**7.9**	**6.9**	**7.0**	**7.5**	**6.9**	**7.0**	**645**

Tabla 7.2: SMAPE para el dataset trimestral de M3-Competition

Método	Horizonte de Predicción							Promedio			#Obs
	T+1	T+2	T+3	T+4	T+5	T+6	T+8	1 a 4	1 a 6	1 a 8	
Naive	5.4	7.4	8.1	9.2	10.4	12.4	13.7	7.55	8.82	9.95	756
Single	5.3	7.2	7.8	9.2	10.2	12	13.4	7.38	8.63	9.72	756
Holt	5	6.9	8.3	10.4	11.5	13.1	15.6	7.67	9.21	10.67	756
Dampen	5.1	6.8	7.7	9.1	9.7	11.3	12.8	7.18	8.29	9.33	756
Winter	5	7.1	8.3	10.2	11.4	13.2	15.3	7.65	9.21	10.61	756
Comb S-H-D	5	6.7	7.5	8.9	9.7	11.2	12.8	7.03	8.16	9.22	756
B-J automatic	5.5	7.4	8.4	9.9	10.9	12.5	14.2	7.79	9.1	10.26	756
Autobox 1	5.4	7.3	8.7	10.4	11.6	13.7	15.7	7.95	9.52	10.96	756
Autobox 2	5.7	7.5	8.1	9.6	10.4	12.1	13.4	7.73	8.89	9.9	756
Autobox 3	5.5	7.5	8.8	10.7	11.8	13.4	15.4	8.1	9.6	10.93	756
Robust-Trend	5.7	7.7	8.2	8.9	10.5	12.2	12.7	7.63	8.86	9.79	756
ARARMA	5.7	7.7	8.6	9.8	10.6	12.2	13.5	7.96	9.09	10.12	756
Automat ANN	5.5	7.6	8.3	9.8	10.9	12.5	14.1	7.8	9.1	10.2	756
Flores/Pearce 1	5.3	7	8	9.7	10.6	12.2	13.8	7.48	8.78	9.95	756
Flores/Peace 2	6.7	8.5	9	10	10.8	12.2	13.5	8.57	9.54	10.43	756
PP-autocast	4.8	6.6	7.8	9.3	9.9	11.3	13	7.12	8.28	9.36	756
ForecastPro	4.9	6.8	7.9	9.6	10.5	11.9	13.9	7.28	8.57	9.77	756
SmartFcs	5.9	7.7	8.6	10	10.7	12.2	13.5	8.02	9.16	10.15	756
Theta-sm	7.7	8.9	9.1	9.7	10.2	11.3	12.1	8.86	9.49	10.07	756
Theta	5	6.7	7.4	8.8	9.4	10.9	12	7	8.04	8.96	756
RBF	5.7	7.4	8.3	9.3	9.9	11.4	12.6	7.69	8.67	9.57	756
ForecastX	4.8	6.7	7.7	9.2	10	11.6	13.6	7.12	8.35	9.54	756
AAM1	5.5	7.3	8.4	9.7	10.9	12.5	13.8	7.71	9.05	10.16	756
AAM2	5.5	7.3	8.4	9.9	11.1	12.7	14	7.75	9.13	10.26	756
Selfnet	5.3	5.2	4.5	4.7	4.4	4.8	4.9	6.0	4.9	4.8	756
Propuesta	**4,3**	**4,5**	**4,2**	**5**	**4,4**	**4,6**	**5,1**	**4,5**	**4,5**	**4,6**	**756**

obtenido un 6,9 de sMAPE promedio para los horizontes de 1 a 4 y un valor de 7,0 para los horizontes de 1 a 6, exponiendo una mejor precisión global en comparación con los métodos existentes. Además de las mejoras sustanciales, es necesario resaltar la mejora en el tiempo computacional que ofrece la propuesta frente a la herramienta de predicción de Selfnet

7.1.2. Observaciones trimestrales

Los resultados trimestrales se pueden observar en la Tabla 7.2. Los valores promedio de la propuesta se han calculado en un total de 756 series temporales, y se ha obtenido un intervalo entre 4.2 y 5,1 exponiendo una mejor precisión para la mayoría de los horizontes de predicción evaluados en entre $t + 1$ y $t + 8$. Además del coste computacional comentado anteriormente, se ha mejorado los valores, observándose así en el sMAPE promedio de 4,5 de 1 a 4 y de 1 a 6 y de 4,6 para los horizontes de pronóstico de 1 a 8, lo que suponen una mejora considerable, particularmente cuando el horizonte incrementa.

Tabla 7.3: SMAPE para el dataset mensual de M3-Competition

Método	Horizonte de Predicción										Promedio						#Obs
	T+1	T+2	T+3	T+4	T+5	T+6	T+8	T+12	T+15	T+18	1 a 4	1 a 6	1 a 8	1 a 12	1 a 15	1 a 18	
Naive	15	13.5	15.7	17	14.9	14.7	15.6	15	19.3	20.47	15.3	15.13	15.29	15.57	16.18	16.91	1428
Single	13	12.1	12.1	15.1	13.5	13.1	13.8	14.5	18.3	19.4	13.53	13.44	13.6	13.83	14.51	15.32	1428
Holt	12.2	11.6	13.4	14.6	13.6	13.3	13.7	14.8	18.8	20.2	12.95	13.11	13.33	13.77	15.51	15.36	1428
Dampen	11.9	11.4	13	14.2	12.9	12.6	13	13.9	17.5	18.9	12.63	12.67	12.85	13.1	13.77	14.59	1428
Winter	12.5	11.7	13.7	14.7	13.6	13.4	14.1	14.6	18.9	20.2	13.17	13.28	13.52	13.88	14.62	15.44	1428
Comb S-H-D	12.3	11.5	13.2	14.3	12.9	12.5	13	13.6	17.3	18.3	12.83	12.79	12.92	13.11	13.75	14.48	1428
B-J automatic	12.3	11.4	12.8	14.3	12.7	12.6	13	14.1	17.8	19.3	12.78	12.74	12.89	13.21	13.96	14.81	1428
Autobox 1	13	12.2	13	14.5	14.1	13.4	14.3	15.4	19.1	20.4	13.27	13.42	13.71	14.1	14.93	15.83	1428
Autobox 2	13.1	12.1	13.5	15.3	13.3	13.8	13.9	15.2	18.2	19.9	13.51	13.52	13.76	14.16	14.86	15.69	1428
Autobox 3	12.3	12.3	13	14.4	14.6	14.2	14.8	16.1	19.2	21.2	12.99	13.47	13.89	14.43	15.2	16.18	1428
Robust-Trend	15.3	13.8	15.5	17	15.3	15.6	17.4	17.5	22.2	24.3	15.39	15.42	15.89	16.58	17.47	18.4	1428
ARARMA	13.1	12.4	13.4	14.9	13.7	14.2	15	15.2	18.5	20.3	13.42	13.59	14	14.41	15.08	15.84	1428
Automat ANN	11.6	11.6	12	14.1	12.2	13.9	13.8	14.6	17.3	19.6	12.31	12.55	12.92	13.42	14.13	14.93	1428
Flores/Pearce 1	12.4	12.3	14.2	16.1	14.6	14	14.6	14.4	19.1	20.8	13.74	13.93	14.22	14.29	15.02	15.96	1428
Flores/Peace 2	12.6	12.1	13.7	14.7	13.2	12.9	13.4	14.4	18.2	19.9	13.26	13.21	13.33	13.53	14.31	15.17	1428
PP-autocast	12.7	11.7	13.3	14..3	13.2	13.4	14	14.3	17.7	19.6	13.02	13.11	13.37	13.72	14.36	15.15	1428
ForecastPro	11.5	10.7	11.7	12.9	11.8	12.3	12.6	13.2	16.4	18.3	11.72	11.82	12.06	12.46	13.09	13.86	1428
SmartFcs	11.6	11.2	12.2	13.6	13.1	13.7	13.5	14.9	18	19.4	12.16	12.58	12.9	13.51	14.22	15.03	1428
Theta-sm	12.6	12.9	13.2	13.7	13.4	13.3	13.7	14	16.2	18.3	13.1	13.2	13.44	13.65	14.09	14.66	1428
Theta	11.2	10.7	11.8	12.4	12.2	12.4	12.7	13.2	16.2	18.2	11.54	11.8	12.3	12.5	13.11	13.85	1428
RBF	13.7	12.3	13.7	14.3	12.3	12.8	13.5	14.1	17.3	17.8	13.49	13.18	13.4	13.67	14.21	14.77	1428
ForecastX	11.6	11.2	12.6	14	12.4	12.2	12.8	13.9	17.8	18.7	12.32	12.31	12.46	12.83	13.6	14.45	1428
AAM1	12	12.3	12.7	14.1	14	14	14.3	14.9	18	20.4	12.8	13.2	13.63	14.05	14.78	15.69	1428
AAM2	12.3	12.4	12.9	14.4	14.3	14.2	14.5	15.1	18.4	20.7	13.03	13.45	13.87	14.25	15.01	15.93	1428
SELFNET	11.0	11.2	11.7	12.5	11.6	11.4	10.6	9.6	11	12.7	11.6	11.6	11.4	11.1	11.2	11.4	1428
Propuesta	**10.5**	**10.6**	**10.8**	**11.9**	**10.8**	**10.7**	**9.8**	**8.7**	**10.2**	**10.9**	**10.9**	**10.9**	**10.7**	**10.5**	**10.4**	**10.4**	**10,5**

7.1.3. Observaciones mensuales

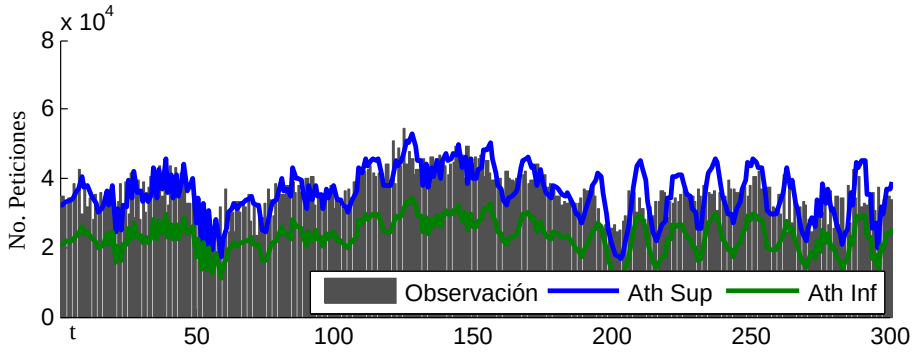
En cuanto a los resultados correspondientes a las series temporales mensuales se han ilustrado en la Tabla 7.3. Como en los anteriores experimentos, los resultados de la propuesta han realizado una mejor precisión para la mayoría de los horizontes de pronósticos en un total de 1428 series temporales, en este caso, evaluadas de t + 1 hasta t+18 donde los valores han oscilado entre 8,7 y 10,9. Gracias a los valores promedios de sMAPE podemos observar nuevamente que el rendimiento obtenido por la propuesta es superior a los otros métodos comentados. Es necesario mencionar que este conjunto de series temporales es el más utilizado en la competencia (con una media de 115 observaciones).

7.1.4. Otras observaciones

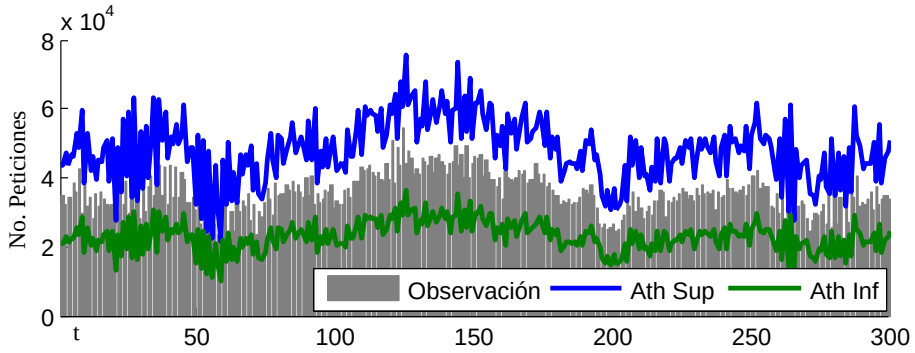
Por último, en la Tabla 7.4 se muestran los resultados obtenidos para un total de 174 series temporales evaluadas en un horizonte de predicción entre 1 y 8. Los resultados han sido significadamente mejores y podemos observar que la precisión se consigue según vaya aumentando el horizonte de predicción, de lo que se puede deducir que la propuesta es capaz de adaptarse según vayan aumentado el número de muestras a analizar. En cuanto a los valores promedio, han sido muy similares a los de la herramienta de SELFNET destacando el limitado uso de recursos que ha necesitado la propuesta.

Tabla 7.4: SMAPE para el otros dataset de M3-Competition

Método	Horizonte de Predicción							Promedio			#Obs
	T+1	T+2	T+3	T+4	T+5	T+6	T+8	1 a 4	1 a 6	1 a 8	
Naive	2.2	3.6	5.4	6.3	7.8	7.6	9.2	4.38	5.49	6.3	174
Single	2.1	3.6	5.4	6.3	7.8	7.6	9.2	4.36	5.48	6.29	174
Holt	1.9	2.9	3.9	4.7	5.7	5.6	7.2	3.32	4.13	4.81	174
Dampen	1.8	2.7	3.9	4.7	5.8	5.4	6.6	3.28	4.06	4.61	174
Winter	1.9	2.9	3.9	4.7	5.8	5.6	7.2	3.32	4.13	4.81	174
Comb S-H-D	1.8	2.8	4.1	4.7	5.8	5.3	6.2	3.36	4.09	4.56	174
B-J automatic	1.8	3	4.5	4.9	6.1	6.1	7.5	3.52	4.38	5.06	174
Autobox 1	2.4	3.3	4.4	4.9	5.8	5.4	6.9	3.76	4.38	4.93	174
Autobox 2	1.6	2.9	4	4.3	5.3	5.1	6.4	3.19	3.86	4.41	174
Autobox 3	1.9	3.2	4.1	4.4	5.5	5.5	7	3.39	4.09	4.71	174
Robust-Trend	1.9	2.8	3.9	4.7	5.7	5.4	6.4	3.32	4.07	4.58	174
ARARMA	1.7	2.7	4	4.4	5.5	5.1	6	3.17	3.87	4.38	174
Automat ANN	1.7	2.9	4	4.5	5.7	5.7	7.4	3.26	4.07	4.8	174
Flores/Pearce 1	2.1	3.2	4.3	5.2	6.2	5.8	7.3	3.71	4.47	5.09	174
Flores/Peace 2	2.3	2.9	4.3	5.1	6.2	5.7	6.5	3.67	7.73	4.89	174
PP-autocast	1.8	2.7	4	4.7	5.8	5.4	6.6	3.29	4.07	4.62	174
ForecastPro	1.9	3	4	4.4	5.4	5.4	6.7	3.31	4	4.6	174
SmartFcs	2.5	3.3	4.3	4.7	5.8	5.5	6.7	3.68	4.33	4.86	174
Theta-sm	2.3	3.2	4.3	4.8	6	5.6	6.9	3.66	4.37	4.93	174
Theta	1.8	2.7	3.8	4.5	5.6	5.2	6.1	3.2	3.93	4.41	174
RBF	2.7	3.8	5.2	5.8	6.9	6.3	7.3	4.38	5.12	5.6	174
ForecastX	2.1	3.1	4.1	4.4	5.6	5.4	6.5	3.42	4.1	4.64	174
Selfnet	1.8	2.3	2.2	2.0	2.3	1.5	2.4	2.1	2.0	2.0	174
Propuesta	**1.8**	**1.9**	**2.4**	**2.1**	**2.3**	**1.5**	**1.6**	**2.1**	**2.0**	**1.9**	**174**



(a) Predicción Convencional



(b) Predicción Adaptativa

Figura 7.1: Ejemplo de adaptación a no estacionariedad.

7.2. Arquitectura Original

Con el fin de facilitar la comprensión del trabajo realizada, en esta fase de experimentación se ha llevado a cabo un sencillo caso de estudio. También se han revisado los resultados obtenidos por DroidSentinel al analizar tráfico de red real.

7.2.1. Caso de Estudio

En la Fig. 7.1, se muestras los resultados obtenidos al calcular la significancia de la estimación del número de peticiones salientes legítimas registradas en un dispositivo Android. Para ilustrar con mayor calidad el ejemplo, se ha considerado una métrica de bajo nivel directamente extraída de los flujos de tráfico monitorizados, ya que estás son más flexibles a cambios en el entorno protegido. En Fig. 7.1a se muestra el intervalo de predicción calculado sobre una configuración estática, definida en una etapa de calibrado que tiene lugar a lo largo de las primeras 20 observaciones. Nótese que, por facilitar la comprensión del ejemplo, la serie temporal analizada

presenta 300 observaciones, siendo mayor que las consideradas en la evaluación de la precisión del sistema. En ella se fija tanto el algoritmo de predicción, como sus parámetros de ajuste. Para K=1.35 el error de predicción medio es del 11.25 Por otro lado, en Fig. 7.1b tanto el algoritmo de predicción como su configuración se han recalibrado en cada nueva observación registrada. En este caso, para K=1.35 el error de predicción medio es del 19.2

7.2.2. Eficacia con tráfico real

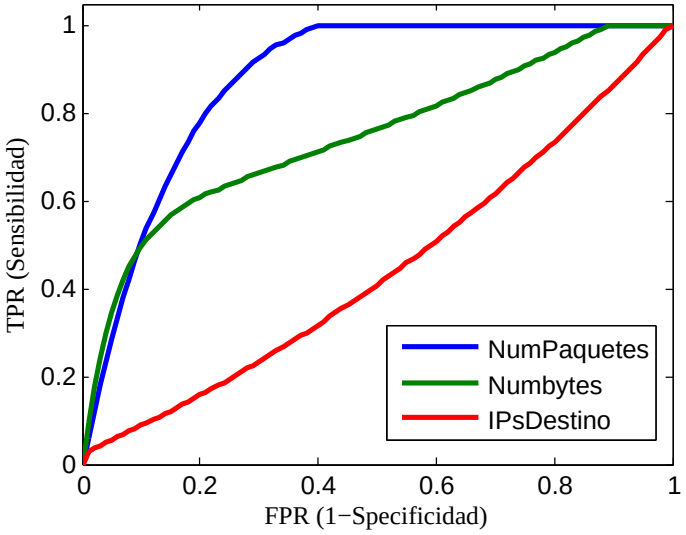
Los resultados obtenidos para las distintas métricas se muestran en Fig. 7.2 sobre el espacio ROC. Este resume la relación entre la variación de la sensibilidad y especificidad registradas al variar el parámetro de ajuste que limita el intervalo de predicción, K en la experimentación realizada. Para ello, se han realizado dos experimentos diferentes: el uso de métricas básicas y el uso métricas agregadas para construcción de las series temporales a analizar.

Métricas Básicas

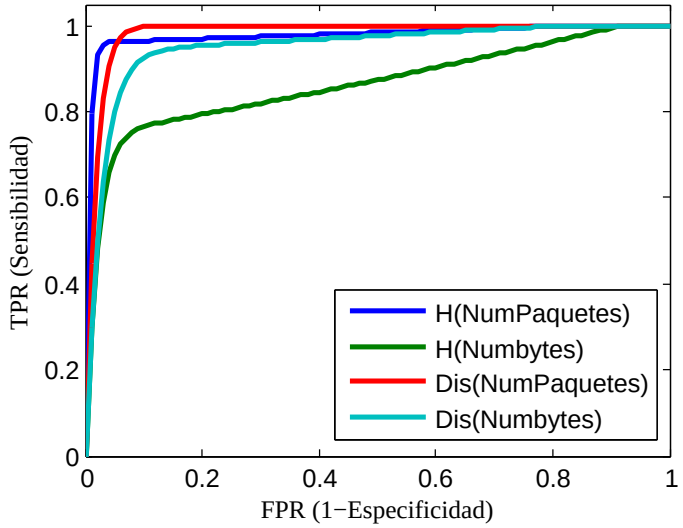
El análisis de tráfico a partir de métricas básicas, en particular número de paquetes (N), total de bytes transmitidos (S) y el número de destinos (D), resultó en la eficacia mostrada en Fig. 7.2a. Las áreas bajo la curva ROC o AUC observados fueron AUC(N)=86.3, AUC(S)=0.729 y AUC(D)=0.45, todos ellos calculados mediante una aproximación trapezoidal con un margen de error máximo de 0.05. Por lo tanto, solo el estudio del total de paquetes enviado por los dispositivos ha sido medianamente viable, debido a que, al participar en un ataque distribuido, a menudo el número de víctimas contra las que se inyecta tráfico es reducido (normalmente un único elemento de red). Además, su carga útil no es muy diferente a la del tráfico legítimo, típicamente amplificada a partir de su paso por elementos de red explotados con fines de amplificación (por ejemplo, servidores DNS).

Métricas Agregadas

En Fig. 7.2b se muestran los resultados en el espacio ROC obtenidos al estudiar dos métricas agregadas: entropía y distancia euclidiana.
La efectividad del análisis basado en entropía ha dependido directamente de la métrica básica a partir de la cual ha sido calculada. Con la entropía del número de paquetes por flujo, los resultados obtenidos han mejorado considerablemente, registrándose AUC=0.985. Este parámetro ha permitido definir una tasa de acierto del 96.1 En Fig. 7.2b se muestra también la eficacia de la propuesta al considerar



(a) Métricas básicas



(b) Entropía

Figura 7.2: Precisión de distintas métricas al variar K

como métrica agregada, la distancia euclidiana entre parámetros del tráfico saliente y entrante. Tal y como puede observarse, el estudio del número de paquetes ha resultado más eficaz, registrándose AUC=0.985. El mejor ajuste ha arrojado una tasa de acierto del 98.5 A raíz de los resultados observados es posible concluir que en la experimentación realizada, y asumiéndose el banco de pruebas adoptado, se ha probado la eficacia de DroidSentinel al detectar ataques de denegación de servicio. Cabe destacar la precisión obtenida mediante el estudio de métricas derivadas del número de paquetes por traza de tráfico y la comparativa de características del tráfico saliente y entrante.

Tabla 7.5: AUC registrado por granularidad al variar la K

Indicador	Granularidad				
	7,5 Seg.	15 Seg.	30 Seg.	1 Min.	2 Min.
nPin	0.75	0.79	0.72	0.75	0.69
nPout	0.84	0.95	0.93	0.93	0.83
nBin	0.61	0.65	0.63	0.67	0.65
nBout	0.75	0.91	0.87	0.84	0.76
H(nPin)	0.63	0.71	0.75	0.67	0.69
H(nPout)	0.65	0.74	0.72	0.69	0.68
H(nBin)	0.64	0.71	0.75	0.65	0.69
H(nBout)	0.63	0.73	0.72	0.71	0.69
nMSE(nP)	0.88	0.96	0.94	0.95	0.85
nMSE(nB)	0.60	0.68	0.68	0.74	0.68

7.3. Arquitectura Adaptada a Redes de 5G

Para este experimento, se ha utilizado la versión de DroidSentinel adaptada a escenarios de 5G. A continuación, se va a detallar los resultados para las diferentes pruebas que se han llevado a cabo para evaluar la herramienta: El impacto de la granularidad, el impacto del perfil de la actividad de los dispositivos e impacto de la intensidad del ataque.

7.3.1. Impacto de la granularidad

En este experimento, se ha medido la precisión del sensor al estudiar los flujos de tráfico capturados en intervalos de tiempo de 7,5 segundos, 15 segundos, 30 segundos, 1 minuto y 2 minutos. Es decir, se ha centrado en el intervalo de monitorización y el ajuste del parámetro K para la calibración de los umbrales adaptativos. Los resultados obtenidos de la evaluación de la exactitud de la herramienta lograda por métrica y configuración se han ilustrado en la Tabla 7.5, donde la efectividad de Droidsentinel se expresa en términos de AUC. Como en la experimentación anterior, estos indicadores de precisión se han calculado con una aproximación trapezoidal cuyo error estimado equivale a 0.05.

La granularidad con mayor precisión fue de 15 segundos por observación, cuyos resultados proporcionados son más precisos que ninguna otra granularidad, en concreto, AUC = 0,96, TPR = 0,93 y FPR = 0,01. Cuando la granularidad es menor, es decir, la duración de la observación es menor, la exactitud de DroidSentinel empeora. Por ejemplo, para 7.5 segundos el mejor AUC evaluado fue de 0.88. De manera análoga, a medida que el nivel de detalle por observación disminuye, la efectividad

de la propuesta disminuye, alcanzando un AUC = 85.2 para la granularidad de 2 minutos por observación. Esto es debido a que cuando se analizan pequeñas observaciones la información recopilada en las mismas tiende a ser menos significativa, por lo tanto, es probable que interfiera el ruido (denominando ruino al tráfico legítimo realizado por un usuario medio). Por ello, cuando el tamaño de la observación es demasiado grande, es posible que las primeras observaciones pertenecientes a ataques pasen desapercibidas entre el tráfico legítimo lo que conlleva a que la herramienta se adapte a dicha no estacionariedad y reajuste los algoritmos analíticos, por lo que no se detecta inicialmente.

Finalmente, es necesario destacar la precisión lograda por métricas directamente relacionadas con la métrica de paquetes totales de entrada (nPin) y de salida (nPout), cuya divergencia (nMSE (nP)) se comportó como el indicador de ataques de DDoS más preciso en esta experimentación. Por otro lado, las soluciones de detección de DDoS basadas en entropía clásica (típicamente desplegadas sobre nodos intermedios y cercanas a las víctimas) demuestran ser menos efectivas que la monitorización en el propio origen de la incidencia.

7.3.2. Impacto del perfil de la actividad de los dispositivos

Con el objetivo de facilitar la compresión de los resultados que se han obtenido durante la experimentación, se ha estudiado el impacto del modo de uso de los dispositivos por los diferentes usuarios analizados en la efectividad de DroidSentinel asumiendo como mejor granularidad aquella obtenida en la experimentación previa. Este experimento se ha ilustrado en la Tabla 7.6. Los tres perfiles según la activad del dispositivo analizado, descritos en la sección anterior, condujeron a los siguientes resultados: Actividades diarias de un usuario: P0 (AUC = 0.96), navegación sintética P1 (AUC = 0.97), Steaming Multimedia P2 (AUC = 0,96). Se ha de tener en cuenta, que, de manera similar a la experimentación anterior, la mejor métrica que se ha obtenido es la diferencia entre paquetes entrantes y paquetes salientes (nMSE (nP)), así como, el número total de paquetes entrantes (nPin) y el número de paquetes salientes (nPout). Nuevamente, las métricas que se basan en la entropía no han sido lo suficientemente efectivas.

Dado que no se han registrado variaciones significativas entre los diferentes perfiles de tráfico analizado, se puede concluir que DroidSentinel ha sido capaz de autocalibrarse de acuerdo con la distribución del tráfico inherente a cada tipo de dispositivo final. De este modo, la propuesta es una solución efectiva independientemente de la naturaleza del dispositivo.

Tabla 7.6: AUC registrada por cada perfil de tráfico con 15 segundos de granularidad

Indicador	Perfil de la actividad		
	P0	P1	P2
nPin	0.86	0.88	0.77
nPout	0.96	0.96	0.95
nBin	0.79	0.72	0.74
nBout	0.92	0.92	0.93
H(nPin)	0.73	0.73	0.62
H(nPout)	0.73	0.76	0.65
H(nBin)	0.70	0.73	0.62
H(nBout)	0.71	0.74	0.65
nMSE(nP)	0.96	0.97	0.96
nMSE(nB)	0.84	0.68	0.79

7.3.3. Impacto de la intensidad del ataque

Por último, este experimento se ha sintetizado en la Tabla 7.7. Esta, resume la precisión que se ha obtenido según el grupo de amenazas, los ataques definidos se basan en inundaciones en los protocolos HTTP (H), TCP (T), UDP (U) con distintos nodos de intensidad. Del mismo modo que la prueba anterior, se ha considerado la mejor granularidad obtenida en la primera experimentación: 15 segundos por observación.

La efectividad fue mejor que en pruebas previas, donde, una vez más, sobresalen las métricas nMSE (nP), (nPin) y (nPout). En este caso, el mejor AUC varió de 0,99 a 1,0 independientemente del subconjunto de intrusión. Esta mejora, se debe a una característica fundamental de la prueba: el factor de ajuste K aplicado por DroidSentinel para configurar su nivel de restricción a la hora de configurar una amenaza específica. En contraste, esta mejora no se ha configurado en los experimentos previos ya que se definió el mismo umbral para todos los métodos de DDoS. En vista de los resultados, se puede deducir que el método propuesto para la detección de ataques de DoS es capaz de adaptarse en función de la intensidad del ataque. Sin embargo, a medida que la especificidad de la amenaza disminuye, DroidSentinel tiende a perder precisión. Esta apreciación debe tenerse en cuenta a la hora de proponer defensas autoorganizadas de propósito general, donde sería recomendable evaluar las diferentes categorías de intrusión por separado.

Tabla 7.7: AUC registrado por tipo de ataque con 15 segundos de granularidad

Indicador	Tipo de Ataque		
	H	T	U
nPin	0.87	0.88	0.82
nPout	1.00	0.95	0.99
nBin	0.79	0.65	0.64
nBout	0.99	0.92	0.96
H(nPin)	0.70	0.78	0.67
H(nPout)	0.68	0.71	0.70
H(nBin)	0.69	0.78	0.65
H(nBout)	0.68	0.69	0.72
nMSE(nP)	1.00	0.96	1.00
nMSE(nB)	0.75	0.64	0.74

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

A lo largo del trabajo realizado se han revisado en profundidad los avances hacia los nuevos escenarios de telefonía móvil (5G) y las tecnologías emergentes que los sustentan. Se ha comprendido que su despliegue integra un complejo y sofisticado ecosistema de soluciones que acarrea importantes desafíos en las diferentes etapas de procesamiento de datos, que afectan desde su almacenamiento hasta a su análisis. El trabajo realizado se ha enmarcado en el proyecto de financiación europea SELFNET - *Framework for Self-Organized Network Management in Virtualized and Software Defined Networks* (Convocatoria: H2020-ICT-2014-2/671672, el cual introduce una arquitectura para la gestión de redes autoorganizadas principalmente sustentada por la virtualización de las funciones de red y las redes definidas por software. En concreto, se ha participado en su cuarto paquete de trabajo (WP4: SDN-*Controlled Self-Monitoring and Detection*), específicamente en la tarea T4.3, donde se lleva a cabo el desarrollo de un conjunto de herramientas analíticas capaces de generar conocimiento a partir de datos capturados por sensores/actuadores previamente agregados. Esto requiere de la incorporación de diversas herramientas analíticas, capaces de elaborar información útil que permita alcanzar un elevado nivel de conciencia situacional acerca del estado del entorno monitorizado, y facilitar su proyección en las observaciones venideras. El trabajo realizado se enmarca precisamente, en las tareas de proyección, habiéndose desarrollado una herramienta analítica que tiene por objetivo la estimación de la evolución de los diferentes indicadores del estado de la red y, por ende, incorpora una potente estrategia de predicción. Con este fin se han tenido en cuenta los desafíos inherentes a los nuevos escenarios emergentes de comunicaciones, haciéndose hincapié en la gran heterogeneidad de la información recibida y su no estacionariedad. Esta solución ha incluido la definición de una es-

trategia de selección de algoritmos predictivos que mejor se adapte a cada instante de tiempo basada en la colección de métricas de series temporales TSFRESH y clasificadores Random Forest, y su auto-calibrado por medio de algoritmos genéticos básicos. Su eficacia ha sido evaluada por medio del estándar funcional propuesto en la M3-Competition, demostrando una importante mejora respecto al componente de inferencia de conocimiento inicialmente implementado por SELFNET.

En una segunda etapa de experimentación, el marco de predicción propuesto ha sido instanciado en un caso de uso concreto y real, donde ha sido adaptado para el análisis de los flujos de tráfico entrantes y salientes en extremos finales en busca de indicios de su participación en ataques de denegación de servicio distribuidos. Esto ha conllevado el considerar métricas propias del estudio de este tipo de incidencias, y a la definición de intervalos de predicción que permitan descubrir cuando una observación es inesperada (es decir, cuando el valor estimado difiere de manera significativa del valor observado), y por lo tanto anómala. La solución propuesta ha sido instanciada en diversos escenarios, entre ellos, en forma de aplicación para sistemas Android, siendo esta versión la que finalmente le ha dado nombre: DroidSentinel. Para su evaluación se ha generado un conjunto de capturas de tráfico de 35 dispositivos diferentes, a partir de las cuales se ha probado su capacidad de distinguir tráfico sospechoso del que no lo es. Dado el interés suscitado, y en colaboración con parte de los investigadores del proyecto SELFNET, DroidSentinel está siendo instanciado como posible solución a este tipo de amenazas en escenarios 5G reales, a día de hoy habiéndose extendido la experimentación a 62 dispositivos diferentes de naturaleza mucho más heterogénea (que incluyen entre otros, relojes y televisores inteligentes).

8.2. Trabajo futuro

A lo largo del documento se han descrito diferentes decisiones de diseño, las cuales han llevado a decidir una aproximación frente a otra con el fin de maximizar ciertos beneficios, pero dejando de lado algunas de las características que brindan las soluciones alternativas. Un ejemplo claro de esto se observa en los algoritmos analíticos implementados (Random Forest, algoritmo genético básico, la batería de algoritmos predictivos seleccionados, etc.), los cuales podrían ser remplazados por estrategias similares en futuras implementaciones. Otras variaciones que serían interesantes de evaluar previo a su despliegue en escenarios diferentes son la colección

de muestras de referencia adoptadas (M3-Competition), las métricas para la decisión del mejor método de predicción (TSFRESH e indicadores de ataques de denegación de servicio) o la función que mide la aptitud de cada calibrado (sMAPE).

En base a los principios de diseño asumidos, cabe destacar que, con el fin de facilitar la comprensión del trabajo realizado, varios aspectos necesarios para su despliegue en escenarios reales han quedado relegados a un segundo plano o no han sido revisados con la profundidad necesaria, sobre los que convendría trabajar de cara a producir futuras versiones de la propuesta. Por ejemplo, este es el caso de la seguridad de los canales de comunicación entre componentes o los diferentes elementos de almacenamiento de la información a procesar, habiéndose asumido su integridad por simplicidad. Tampoco se ha medido el impacto del nuevo Reglamento General de Protección de Datos (RGPD) europeo en las distintas etapas de procesamiento de información, asumiéndose que los sensores y actuadores desplegados tienen plenos permisos para extraer y analizar el encabezado de los paquetes que fluyen a través de los dispositivos protegidos.

Finalmente, cabe destacar el interés que suscita la adaptación de la propuesta a diferentes casos de uso. Si bien el problema de la denegación de servicio actualmente está en el punto de mira de las diferentes organizaciones para la seguridad de la información, existen muchas otras posibles aplicaciones. Por ejemplo, podría adaptarse para la identificación de equipos comprometidos que forman parte de redes de zombis (botnets), la distinción de usuarios por su modo de uso de la red, y la optimización de los servicios de red prestados (mejora de ancho de banda, latencia, estimación de regiones congestionadas, etc.) De hecho, SELFNET actualmente está explorando su implementación para estos últimos casos.

Bibliografía

[1] SELFNET: Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET). Available at: http://www.SELFNET-5g.eu

[2] CCN-CERT, ÏA-16/17 CyberThreats-Trends. 2017 Edition", June 2017, Available at: https://www.ccn-cert.cni.es/en/reports/public/2249-ccn-cert-ia-16-17-cyberthreats-trends-2017-executive-summary-1/file.html

[3] V.A.F. Almeida, D. Doneda, J.S. Abreu, Çyberwarfare and Digital Governance". IEEE Internet Computing, Vol. 21, Issue 2, pp. 68-71, April 2017.

[4] E. Bertino, N. Islam, "Botnets and Internet of Things Security". Computers, Vol. 50 (2), pp. 76-79, February 2017.

[5] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, "DDoS in the IoT: Mirai and Other Botnets". Computer Vol. 50 (7), pp. 80-84, July 2017.

[6] M. Antonakakis, T. April, et al. Ünderstanding the Mirai Botnet". In Proc. of the 26th USENIX Security Symposium, Vancouver, BC, Canada, August 2017.

[7] S. T. Zargar, J. Joshi, D. Tipper, .ª Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks", IEEE Communications Surveys & Tutorials, vol. 15 (4), pp. 2046-2069, 2013.

[8] Q. Yan, F.R. Yu, Q. Gong, J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges". IEEE Communications Surveys & Tutorials, Vol. 18 (1), First quarter 2016.

[9] D. Acarali, M. Rajarajan, N. Kmminos, I. Herwono, "Survey of approaches and features for the identification of HTTP-based botnet traffic". Journal of Network and Computer Applications, Vol. 76, pp. 1-15, December 2016.

[10] DroidSentinel. Available at: https://github.com/borjalor/DroidSentinel

[11] D. Acarali, M. Rajarajan, N. Kmminos, I. Herwono, "Survey of approaches and features for the identification of HTTP-based botnet traffic". Journal of Network and Computer Applications, Vol. 76, pp. 1-15, December 2016.

[12] NGMN Alliance., 5G White Paper, Available online: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[13] Expert Working Group on 5G: Challenges, Research Priorities, and Recommendations. European Technology Platform for Communications Networks and Services (NetWorld2020 ETP), 5G White Paper. Available online: https://networld2020.eu/wp-content/uploads/2014/02/NetWorld2020_Joint-Whitepaper-V8_public-consultation.pdf

[14] Panwar, N., Sharma, S., & Singh, A. K. (2016). A survey on 5G: The next generation of mobile communication. Physical Communication, 18, 64-84.

[15] J. P. Santos, R. Alheiro, L. Andrade, Á. L. Valdivieso Caraguay, L. I. Barona López, M. A. Sotelo Monge, et al: "SELFNET Framework self-healing capabilities for 5G Mobile Networks", Transactions on Emerging Telecommunications Technologies, Vol. 27, No 9, pp. 1225-1232, June 2016.

[16] Bruno López Takeyas, "Introducción a la inteligencia artificial", 2007, Available at: http://www.itnuevolaredo.edu.mx/takeyas/Articulos/Inteligencia%20Artificial/ARTICULO%20Introduccion%20a%20la%20Inteligencia%20Artificial.pdf

[17] Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET)". http://www.selfnet-5g.eu

[18] P. Neves, R. Cale, M.R. Costa, C. Parada, B. Parreira, J. Alcaraz-Calero, Q. Wang, J. Nightingale, E. Chirivella-Perez, W. Jiang, "The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm", International Journal of Distributed Sensor Networks, pp. 1-17, December 2015.

[19] 2016 Cost of Cyber Crime Study & the Risk of Business Innovation, Available at: https://www.ponemon.org/local/upload/file/2016%20HPE%20CCC%20GLOBAL%20REPORT%20FINAL%203.pdf

[20] Symantec Report 2016, Available at: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[21] Symantec Report 2016, Available at: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[22] P. J. Criscuolo, Distributed Denial of Service, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.

[23] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, Detecting DNS Amplification Attacks, in Critical Information Infrastructures Security Lecture Notes in Computer Science, Vol. 5141,pp. 185-196, 2008.

[24] A. Rahul, S. K. Prashanth, B. S. kumarand , and G. Arun, Detection of Intruders and Flooding In Voip Using IDS, Jacobson Fast And Hellinger Distance Algorithms, IOSR Journal of Computer Engineering (IOSRJCE), Vol. 2, no. 2, pp. 30-36, July-Aug. 2012.

[25] Symantec (2014), Internet Security Threat Report 2014. Resource available at `http://www.symantec.com`

[26] Sophos (2014), Security Threat Report 2014. Resource available at `http://www.sophos.com`

[27] L. Ablon, M. C. Libicki, A. A. Golay (RAND Corporation)(2014), Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar. Resource available at `http://www.rand.org`

[28] S. Makridakis, M. Hibon, "The M3-Competition: results, conclusions and implications", International Journal of Forecasting, Vol. 16, Issue 4, pp. 451-176, December 2000.

[29] L. Breiman, Random Forests", Machine Learning, Vol. 45, Issue 1, pp. 5-32, October 2001.

[30] TSFRESH: Time Series Feature extraction based on scalable hypothesis tests. Available at: https://github.com/blue-yonder/tsfresh

[31] P. Mendes, Çombining data naming and context awareness for pervasive networks", Journal of Network and Computer Applications, Vol. 50, pp. 114- 125, April 2015.

[32] M.R. Endsley, "Design and evaluation for situation awareness enhancement", In Proceedings of the 32nd Annual Meeting on Human Factors Society, Santa Monica, CA, US, pp. 97-101, October 1988.

[33] P.G. Mulloy, "Smoothing Data with Faster Moving Averages", Stocks & Commodities, Vol. 12, Issue 1, pp. 11-19, January 1994.

[34] M.B.C. Khoo, V.H. Wong, ."A Double Moving Average Control Chart", Communications in Statistics - Simulation and Computation, Vol. 37, Issue 8, pp. 1696-1708, October 2008.

[35] H. Feng, S. Li, ."Active disturbance rejection control based on weighed-moving-average-state-observer", Journal of Mathematical Analysis and Applications, Vol. 411, Issue 1, pp. 354-361, March 2014.

[36] A.A. Aly, N.A. Salem, M.A. Mahmoud, W.H. Woodall, ."A Reevaluation of the Adaptive Exponentially Weighted Moving Average Control Chart When Parameters are Estimated", Quality and Reliability Engineering International, Vol. 31, Issue 8, pp. 1611-1622, December 2015.

[37] P.G. Mulloy, "Smoothing data with less lag", Technical Analysis of Stocks & Commodities, Vol. 12, Issue 1, January 1994.

[38] R.G. Brown, ."Exponential smoothing for predicting demand", Operations Research, Vol. 5, No. 1, pp. 145-145, 1957.

[39] C.C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages", International Journal of Forecasting, Vol. 20, Issue 1 pp. 5- 10, March 2004.

[40] E.S. Gardner, ."Exponential smoothing: The state of the art-Part II", International Journal of Forecasting, Vol. 22, Issue 4, pp. 637-666, December 2006.

[41] P.R. Winters.,"Forecasting Sales by Exponentially Weighted Moving Averages", Management Science, Vol. 6 (3), pp. 324-342, April 1960.

[42] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, "Forecasting: Methods and Applications". John Wiley & Sons, 1998.

[43] H. Akaike, ."Autoregressive model fitting for control", Annals of the Institute of Statistical Mathematics, Vol. 23, Issue 1, pp. 163-180, December 1971.

[44] S.S. Said, D.A. Dickey, "Testing for unit roots in autoregressive-moving average models of unknown order", Biometrika, Vol. 71, Issue 3, pp. 599-607, December 1984.

[45] S.C. Hillmer, G.C. Tiao, .An ARIMA-Model-Based Approach to Seasonal Adjustment", Journal of the American Statistical Association, Vol. 77, Issue 377, pp. 63-70, Ocober 1980.

[46] S.C. Hillmer, G.C. Tiao, .An ARIMA-Model-Based Approach to Seasonal Adjustment", Journal of the American Statistical Association, Vol. 77, Issue 377, pp. 63-70, Ocober 1980.

[47] B. Hu, X. Li, S. Sun, M. Ratcliffe, .Attention Recognition in EEG-Based Affective Learning Research Using CFS+KNN Algorithm". IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 15 (1), pp. 38-45, February 2018.

[48] R.C. Holte, "Very simple classification rules perform well on most commonly used datasets", Machine Learning Vol. 11, Issue 1, pp. 63-90, April 1993.

[49] M.A. Hall, Çorrelation-Based Feature Selection for Machine Learning". Ph.D dissemination, University of Waikato, April 1999.

[50] L. Rutkowski, M. Jaworski, L. Pietruczuk, P. Duda, "The CART decision tree for mining data streams", Information Sciences, Vol. 266, pp. 1-15, May 2014.

[51] C. O'Reilly, A. Gluhak, M.A. Imran, S. Rajasegarar, .Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment", IEEE Communications Surveys & Tutorials, Vol. 16, No. 3, pp. 1413-1432, third quarter 2014.

[52] D.H. Widyantoro , T.R. Ioerger, J. Yen, "Tracking changes in user interests with a few relevance judgments". In Proc. of the 12th International Conference on Information and Knowledge Management (CIKM), New Orleans, LA, US, pp. 548-551, November 2003.

[53] A. Kamrani, R. Wang, R. Gonzalez, .A Genetic Algorithm Methodology for Data Mining & Intelligent Knowledge Acquisition", Computers & Industrial Engineering Vol. 40, Issue 4, 361-377, September 2001.

[54] S.M.Elsayed, R.A. Sarker, D.L. Essam, .A new genetic algorithm for solving optimization problems". Engineering Applications of Artificial Intelligence, Vol. 27, pp. 57-69, January 2014.

[55] G.R. Ruiz, C.F. Bandera, T.G.A. Temes, A.S.O. Gutierrez, "Genetic algorithm for building envelope calibration". Applied Energy, Vol. 168, pp. 691-705, April 2016.

[56] D.E. Goldberg, K. Deb, .ª Comparative Analysis of Selection Schemes Used in Genetic Algorithms". Foundations of Genetic Algorithms, Vol. 1, pp. 69-93, 1991

[57] J. A. Lima, N. Gracias, H. Pereira, A. Rosa "Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems.ªero 1996.

[58] R. Poli and W.B. Langdon "Genetic Programming with One-Point Crossover"pp. 180-189.

[59] S. T. Zargar, J. Joshi, D. Tipper, .ª Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks", IEEE Communications Surveys & Tutorials, vol. 15 (4), pp. 2046-2069, 2013.

[60] M.H. Bhuyan, D. Bhattacharyya, J. Kalita. "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection", Pattern Recognition Letters, vol. 51, no. 1, pp. 1-7, 2015

[61] G. Vormayr, T. Zseby, J. Fabini, "Botnet Communication Patterns". IEEE Communications Surveys & Tutorials, Vol. 19 (4), pp. 2768-2796, Fist quarter 2017.

[62] S.A. Mehdi, J. Khalid, S.A. Khayam, Revisiting Traffic Anomaly Detection Using Software Defined Networking". In Proc. of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID), Menlo Park, CA, US, pp. 161-180, September 2011.

[63] D.E. Denning, "Framework and principles for active cyber defense". Computers & Security, Vol. 40, pp. 108-113, February 2014.

[64] M.A. Sotelo Monge, J. Maestre Vidal, L.J García Villalba, .ªtropy-Based Economic Denial of Sustainability Detection". Entropy, Vol. 19 (12), No. 649, November 2017.

[65] I. Ozcelik, R.R. Brooks. "Deceiving entropy based DoS detection", Computers & Security, vol. 48, no. 1, pp. 234-245, 2015.

[66] H.Y. Lateef, A. Imran, M.A. Imran, L. Giupponi, M. Dohler, "LTE-advanced self-organizing network conflicts and coordination algorithms". IEEE Wireless Communications, Vol. 22, Issue 3, pp. 108-117, June 2015.

[67] TSFRESH: Time Series Feature extraction based on scalable hypothesis tests. Available at: `https://github.com/blue-yonder/tsfresh`

[68] S. Makridakis, M. Hibon, "The M3-Competition: results, conclusions and implications", International Journal of Forecasting, Vol. 16, Issue 4, pp. 451-176, December 2000.

[69] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Prass, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX". IEEE Communications Surveys & Tutorials, Vol. 16 (4), pp. 2037-2064, May 2014.

[70] M.A. Sotelo Monge, J. Maestre Vidal, L.J. García Villalba, Reasoning and Knowledge Acquisition Framework for 5G Network Analytics". Sensors, Vol. 17(10), No. 2405, October 2017.

[71] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, "Forecasting: Methods and Applications". John Wiley & Sons, 1998.

[72] L. Rutkowski, M. Jaworski, L. Pietruczuk, P. Duda, "The CART decision tree for mining data streams", Information Sciences, Vol. 266, pp. 1-15, May 2014.

[73] Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. IEEE Communications Surveys & Tutorials, 18(3), 1617-1655.

[74] Maimó, L. F., Gómez, Á. L. P., Clemente, F. J. G., Pérez, M. G., & Pérez, G. M. (2018). A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. IEEE Access, 6, 7700-7712.

[75] WarChild DoS test suit. Available at `https://github.com/Souhardya`

[76] Berkeley Packet Filter, Available at: `https://biot.com/capstats/bpf.html`

[77] Pcac4j Available at: `https://www.pcap4j.org/`

[78] Show me Data. Available at: `https://github.com/gjimenezUCM/showMeTheData-2018`

[79] Selenium. Available at: `https://www.seleniumhq.org`

[80] BeautifulSoup Available at: `https://www.crummy.com/software/BeautifulSoup/`

[81] Requests Available at: `http://docs.python-requests.org`

[82] L.E. Bantis, C.T. Nakas, B. Reiser, Çonstruction of confidence regions in the ROC space after the estimation of the optimal Youden index based cut off point". Biometrics, Vol. 70, Issue 1, pp. 212-223, March 2014.

[83] Low Orbit Ion Cannon (LOIC): `https://sourceforge.net/projects/loic/files/`

[84] WarChild DoS test suit. Available at `https://github.com/Souhardya`

[85] Cheung, Y.W., Kon, S.L.: Lag Order and Critical Values of the Augmented Dickey-Fuller Test. Journal of Business & Economic Statistics 13(3) 277-280 (1995)

[86] Internet Noise. Available at: `http://makeinternetnoise.com`

[87] Noiszy. Available at: `https://noiszy.com`

[88] TrackMeNot: Resisting Surveillance in Web Search. Available at: `https://cs.nyu.edu/trackmenot/`

Contribuciones

En el siguiente capitulo se presentan los contribuciones de cada uno de los integrantes del equipo.

Andrés Herranz González

Entre mayo y Junio de 2017 conocimos a través del hermano de un compañero esta propuesta. El grupo de investigación GASS de la Universidad Complutense de Madrid estaba trabajando en un proyecto europeo Horizonte 2020 llamado SELFNET y buscaba incorporar a alumnos de la facultad para hacer una colaboración y a raíz de ella, hacer el TFG.

Mis compañeros Guillermo Rius, Borja Lorenzo y yo, estuvimos muy interesados. La propuesta consistía en trabajar a lo largo del año para el proyecto SELFNET a cambio de recibir una remuneración en concepto de colaboración y la posibilidad de poder hacer el TFG sobre el trabajo realizado, además de divulgar el mismo y participar en congresos y revistas.

Los intereses de los tres estaban relacionados con la seguridad y la inteligencia artificial, y el proyecto que se nos presentaba reunía ambas temáticas, lo cual hacía de ésta una propuesta aún más interesante. Con todo ello, aceptamos la oferta sin dudarlo y empezamos a reunirnos con el codirector del proyecto Jorge Maestre y Marco Sotelo, que también trabajaba en el proyecto y nos ayudaría durante la colaboración.

A lo largo del mes de julio, los tres estuvimos estudiando y aprendiendo acerca de los temas principales del trabajo. Para ello utilizamos la documentación que ellos tenían y algunas fuentes de internet. Aprendimos acerca de 5G, el proyecto SELFNET y algoritmos de predicción. Sobre todo, nos centramos en la última parte, ya que Jorge y Marco nos comentaron que en septiembre había una que presentar una demo, y contaban con nuestra ayuda para desarrollar la herramienta. Es por ello

por lo que distribuimos en categorías los diferentes tipos de algoritmos predictivos, y cada uno nos centramos en un tipo concreto. En mi caso me centré en los algoritmos de medias móviles, y su implementación.

La idea era crear un framework que ejecutase una batería de algoritmos sobre un dataset de series temporales obtenidas del tráfico de red, y que eligiese la mejor predicción y el mejor algoritmo para cada caso. Esta aplicación se presentó en la segunda revisión anual del proyecto que tuvo lugar los días 17-19 septiembre del 2017 en Tel Aviv, Israel, cumpliendo con las expectativas satisfactoriamente.

Una vez terminada la demo, decidimos hacer cambios sobre el sistema para hacerlo adaptativo. La idea era implementar un algoritmo genético capaz de calibrar los parámetros de los algoritmos de predicción de una manera más eficiente y precisa, e implementar un clasificador capaz de seleccionar el mejor algoritmo de predicción para una serie temporal dada. Aunque los tres nos mantuvimos al tanto del avance del resto, Borja se centro más en la implementación del algoritmo genético, y Guillermo y yo en la parte del clasificador.

Para implementar el clasificador tuvimos en primer lugar, que crear una herramienta capaz de extraer las características de una serie temporal. Ante la dificultad para implementarlo por nosotros mismos, decidimos buscar alguna herramienta previamente desarrollada en Github, y así fue como encontramos TSFresh. Posteriormente, creamos el dataset, compuesto por las características de las series temporales y el mejor algoritmo para las mismas (obtenidos haciendo uso de la herramienta previamente creada). Con todo ello, construimos un modelo Random Forest capaz de, dada una serie temporal, extraer sus características, y en base a ellas, seleccionar el mejor algoritmo (sin pasar por la batería de algoritmos).

A mediados de octubre, se nos ocurrió la posibilidad de presentarnos a la hackathon que organizaba el Instituto Nacional de Ciberseguridad (INCIBE) durante el evento CyberCamp, el cual tuvo lugar en Santander los días 1-3 de Diciembre; con idea de hacer un caso de uso para la herramienta que teníamos desarrollada. Unas semanas después tuvimos la respuesta de que el trabajo había sido aceptado y de que nos pagaban el viaje e instancia para asistir al evento y poder competir. El viaje fue una experiencia intensa, en la que aprendimos y trabajamos mucho (y hasta muy tarde). Finalmente, obtuvimos el tercer premio tras conseguir crear una aplicación capaz de detectar, en tiempo real, si el dispositivo estaba participando en

un ataque de denegación de servicio distribuido.

Al finalizar la CyberCamp, estábamos ya a mediados de diciembre, y ya que los 3 trabajábamos en nuestras respectivas empresas y teníamos que estudiar para los exámenes de finales de Enero, decidimos hacer un parón hasta terminar con el primer cuatrimestre.

Una vez terminados los exámenes, Jorge y Marco nos comentaron la posibilidad de presentar el trabajo que estábamos desarrollando en las Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) organizadas por INCIBE, además de en el European Symposium on Research in Computer Security (ESORICS), CORE A. Con este objetivo decidimos ampliar y mejorar el dataset implementando la herramienta Varys para la extracción de los datos del trafico de los dispositivos.

Finalmente, los últimos meses los dedicamos a la redacción del presente documento, en el cual todos trabajamos por igual. El trabajo enviado a las JNIC fue aceptado, por lo que nos disponemos a exponer la semana que viene.

Borja Lorenzo Fernádez

El comienzo de este proyecto tuvo lugar la primera semana de junio tras interesarnos por la propuesta ofrecida por nuestro cotutor Jorge Maestre Vidal, que consistía en la implementación de algoritmos de predicción para la detección de ataques de denegación de servicio en redes de quinta generación. En la primera etapa, se dedicó a la revisión exhaustiva de documentación, cuyo objetivo era adquirir conocimientos sobre las redes emergentes de quinta generación, el proyecto que involucra este SELFNET y, sobre todo, el desarrollo de la parte del framework de predicción.

Una vez adquiridos conocimientos suficientes, así como una idea de lo que iba a implicar el desarrollo de nuestro trabajo de fin de grado, nos pusimos con la parte de implementación de los algoritmos de predicción; como el objetivo principal era dotar al proyecto de una batería de algoritmos nos dividimos los diferentes tipos de algoritmos de predicción que finalmente se han implementado. Particularmente, mi investigación se centró en los modelos autorregresivos, que supuso la implantación de los modelos ARIMA que dan paso a los otros modelos comentados en la memoria como AR, ARMA... Como todo el proyecto, se desarrolló en java, lo que permitió afianzar y extender los conocimientos adquiridos durante la carrera.

La primera versión del proyecto, donde se probaban todos los algoritmos y se quedaba con el mejor de manera secuencial se preparó con el objetivo de formar parte de la demostración de la segunda revisión anual del proyecto SELFNET que aconteció los días 17-19 septiembre del 2017 en Tel Aviv, Israel. Además de esta versión, se implementó el reconocimiento anómalo y la visualización en tiempo real.

Una vez desarrollada esta primera versión se hizo hincapié en la parte de predicción adaptativa, en concreto en la parte de creación del dataset de referencia, la generación del modelo y la implementación del algoritmo genético. Como esta parte se podía modularizar, se me asignó la parte de desarrollo del algoritmo genético; este me permitió profundizar en este ámbito y presentar una solución acorde con los objetivos del proyecto.

A mediados de Octubre se nos presentó la oportunidad de presentar nuestro trabajo como un proyecto en desarrollo para la Hackathon de ciberseguridad organizado por el Instituto Nacional de Ciberseguridad (INCIBE), y enmarcada en el evento Cybercamp 2017, que tuvo lugar en Santander durante los días 1-3. Para ello, era necesario presentar una propuesta, de lo que surgió la idea de presentar un caso de uso de nuestro trabajo centrado en el reconocimiento de ataques de denegación de servicio originados en dispositivos Android, lo que permitía detectar si el dispositivo analizado era parte de una *botnet*.

Una vez aceptados para participar en el Hackathon, se presentó un nuevo reto, ya que ninguno de los integrantes conocíamos el mundo desarrollo en aplicaciones Android. Particularmente, me permitió conocer las tecnologías para desarrollar aplicaciones en dispositivos Android y desarrollar nuestra propuesta en un caso de uso concreto.

Durante el evento, tuvimos la oportunidad de desarrollar la aplicación, algo que supuso un gran avance en el trabajo de fin de grado de lo que nació DroidSentinel. Además, durante el evento tuvimos que preparar diversas presentaciones para definir el proyecto a los diferentes jurados. Concretamente yo hice la presentación que introdujo el trabajo a desarrollar, actualmente disponible en el canal de YouTube del evento. En este Hackathon quedamos terceros, algo que nos motivó especialmente para continuar mejorando nuestro proyecto.

Con la herramienta parcialmente desarrollada, nuestro siguiente objetivo fue el de recolectar un dataset de prueba que nos permitiera evaluar dicho proyecto, para ello desarrollamos una herramienta en Java denominada Varys, la cual permite la recolección de los datos que necesitábamos para crear las métricas para su posterior análisis.

Entre los meses de febrero y abril, nos centramos en la divulgación del proyecto, es decir, el desarrollo de diferentes artículos de investigación contando lo que habíamos hecho para enviarlo a diferentes congresos, como por ejemplo las Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), donde nos aceptaron el trabajo, el cual tendremos que presentar a mediado de junio del 2018. También enviamos un artículo al congreso internacional European Symposium on Research in Computer Security (ESORICS), CORE A, que se celebrará en Barcelona a principios de septiembre.

En el último trimestre del año, se dedicó a la redacción de esta memoria, para la redacción de los puntos que la componen hemos colaborado todos en igual medida.

Guillermo Rius García

Durante el mes de mayo de 2017 mis compañeros y yo definimos las líneas de investigación que más nos interesaban para empezar a buscar un trabajo de fin de grado acorde a ellas. Nos interesaba principalmente la ciberseguridad y la inteligencia artificial por lo que nos informamos acerca de que trabajos ofertaban en el Departamento de Ingeniería de Software e Inteligencia Artificial. Nos informaron de que en ese momento estaban buscando unos alumnos para realizar una herramienta de predicción de anomalías en el tráfico de red y además quisieran alinear este trabajo con una colaboración en el proyecto SELFNET, financiado por la Comisión Europea y parte del programa Horizonte 2020, la cual se detalla en el capítulo 2.2.

Durante el mes de julio empezamos a leer documentación sobre las principales líneas de investigación de nuestro trabajo. Empezamos con conceptos básicos como redes de quinta generación, ataques de denegación de servicio, etc. Con estos conocimientos como base empezamos a leer documentación de SELFNET puesto que este proyecto llevaba dos años en marcha y teníamos que entender bien todo para poder participar en el desarrollo de la herramienta que teníamos que presentar en la segunda revisión anual de SELFNET que tuvo lugar los días 17-19 septiembre del

2017 en Tel Aviv, Israel.

Con los conocimientos adquiridos en la lectura de la documentación, nos costó mucho trabajo coger el ritmo de nuestros compañeros, pero con su ayuda conseguimos entender el desarrollo del código del proyecto que había hasta el momento y ponernos a trabajar.

Empezamos a colaborar en el desarrollo de la herramienta de predicción para la presentación que tenían que hacer nuestros compañeros en Tel Aviv. Esta herramienta constaba de múltiples algoritmos de predicción desarrollados en Java que se pueden agrupar en tres grandes familias: medias móviles, autorregresión y alisamiento. En este momento decidimos paralelizar el trabajo entre los tres por lo que elegimos una familia cada uno. Yo me centré en los algoritmos de alisamiento descritos en el capítulo 4.1.1. Busqué bastante información al respecto puesto que no conocía nada sobre este tipo de algoritmos. Finalmente, con ayuda de nuestros compañeros y toda la información recolectada conseguí desarrollarlos a tiempo para la presentación.

El código presentado ejecutaba secuencialmente todos los algoritmos. Esto estaba bien como prueba de concepto, pero sabíamos que debíamos mejorar su eficiencia. Para ello decidimos desarrollar un modelo de predicción que aprendiendo de los resultados obtenidos en las ejecuciones secuenciales fuera capaz de inferir que algoritmo de predicción se ajustaba mejor a las características de la serie temporal a procesar. Esto nos permitió pasar de ejecutar todos los algoritmos a solo el mejor para cada caso. En esta parte fue en la que más nos centramos mi compañero Andrés y yo.

Además, los resultados obtenidos, aun siendo buenos, podían mejorarse si introducíamos un algoritmo genético que nos permitiese calibrar los parámetros de ajuste de los algoritmos de predicción. En esta parte se centró más nuestro compañero Borja.

Decidimos aplicar esta herramienta a un caso de uso concreto, y lo propusimos para la hackathon celebrada en el marco del evento CyberCamp 2017, evento organizado por el Instituto Nacional de Ciberdefensa (INCIBE) en Santander durante los días 1-3 de Diciembre. En concreto, desarrollamos en Android una App que permitía detectar si tu dispositivo estaba siendo participe de un ataque de denegación de servicio, la cual aprovechaba gran parte de las capacidades predictivas implementadas

hasta entonces. Finalmente, tras un gran esfuerzo y muchas horas de desarrollo, conseguimos el tercer puesto de la categoría absoluta.

Con la finalidad de continuar con la difusión de nuestro trabajo decidimos escribir un artículo para enviarlo a las Jornadas Nacionales de Ciberseguridad (JNIC) 2018 y otro para el European Symposium on Research in Computer Security (ESORICS), siendo el último un congreso CORE A. El desarrollo de estos artículos nos tomó bastante tiempo puesto que debíamos desarrollar una experimentación bastante exhaustiva y se nos solapó con los exámenes de Enero – Febrero. Esta experimentación se basó en un dataset generado con la herramienta Varys, la cual desarrollamos específicamente para ello, compuesto por tráfico de múltiples usuarios. El artículo de las JNIC 2018 ha sido aceptado y será presentado a mediados de junio. El artículo del ESORICS está pendiente de respuesta.

El tiempo restante hasta la entrega se dedicó a la redacción de este documento.