VNiVERSiDAD D SALAMANCA

ESTRATEGIAS AVANZADAS PARA LA IDENTIFICACIÓN PROACTIVA DE TÁCTICAS EMPLEADAS EN CIBERATAQUES MEDIANTE EL USO DE MACHINE LEARNING

AUTOR:

ALBERTO SÁNCHEZ DEL MONTE

DIRECTORES:

ÁNGEL MARÍA MARTÍN DEL REY

ROBERTO CARLOS CASADO VARA

TESIS PARA LA OBTENCIÓN DEL TÍTULO DE DOCTOR POR LA UNIVERSIDAD DE SALAMANCA

OCTUBRE DE 2023

DECLARACIÓN DE AUTORÍA

Ángel María MARTÍN DEL REY, catedrático del Departamento de Matemática Aplicada de la Facultad de Ciencias, Instituto Universitario de Física Fundamental y Matemáticas Aplicadas de la Universidad de Salamanca, y **Roberto Carlos CASADO VARA**, profesor ayudante doctor del Departamento de Matemáticas y Computación de la Universidad de Burgos,

CERTIFICAN

Que el presente documento, titulado ***Estrategias avanzadas para la identificación proactiva de tácticas empleadas en ciberataques mediante el uso de Machine Learning*** ha sido elaborado bajo su supervisión en la Universidad de Salamanca por **Alberto SÁNCHEZ DEL MONTE**, y constituye su tesis doctoral para la obtención del grado de Doctor (PhD) en Ingeniería Informática por la Universidad de Salamanca.

MARTIN DEL REY ANGEL MARIA - 07953200F
Firmado digitalmente por MARTIN DEL REY ANGEL MARIA - 07953200F
Fecha: 2023.10.31 07:20:28 +01'00'

Ángel María Martín del Rey

CASADO VARA ROBERTO CARLOS - 45686237A
Firmado digitalmente por CASADO VARA ROBERTO CARLOS - 45686237A
Fecha: 2023.10.31 07:45:11 +01'00'

Roberto Carlos Casado Vara

SANCHEZ DEL MONTE ALBERTO - DNI 70815812P
Firmado digitalmente por SANCHEZ DEL MONTE ALBERTO - DNI 70815812P
Fecha: 2023.10.30 22:22:38 +01'00'

Alberto Sánchez del Monte

AGRADECIMIENTOS

Quisiera agradecer el apoyo que he recibido de mis mandos y compañeros de la Secretaría de Estado de Seguridad del Ministerio del Interior, y en especial a Fernando por todo lo que ha compartido conmigo a lo largo de este tiempo.

Tampoco quisiera olvidarme de Luis Hernández Encinas, por todo el apoyo que me ha ofrecido en mi estancia en el ITEFI-CSIC, así como a Luis Hernández por su valiosa colaboración.

Gracias a Ángel Martín del Rey por todo el apoyo y confianza que me ha dado durante este tiempo como director de esta investigación, así como a Roberto Carlos Casado Vara por toda su ayuda.

Finalmente quiero dar las gracias a mis hijas, Sofía e Inés, y a mi mujer Irene, por ser mi compañera, amiga y amor en mi paso por la vida.

RESUMEN

En el contexto actual de incremento de la cibercriminalidad se hace patente la necesidad de disponer, por parte de las Fuerzas y Cuerpos de Seguridad del Estado, de herramientas que permitan identificar tácticas y patrones en ataques a sistemas de información u operación. Estas soluciones podrían reducir significativamente los tiempos de análisis y proporcionarían información valiosa para la difícil tarea de atribuir acciones delictivas a organizaciones o individuos.

Así pues, este estudio tiene como objetivo diseñar un modelo robusto de predicción de tácticas o acciones empleadas por ciberatacantes, partiendo de la hipótesis de que la Inteligencia Artificial, específicamente el *Machine Learning*, ofrece herramientas capaces de detectar, trazar y predecir patrones de ciberataques, siempre y cuando estos sistemas estén alimentados con datos diversificados y confiables. En este sentido, la obtención, codificación, ampliación y equilibrio de un conjunto de datos se convierte en un foco principal de esta investigación.

En primer lugar, se ha llevado a cabo un profundo examen del estado actual de la ciberseguridad y la cibercriminalidad de cara a sentar las bases y conceptos necesarios que permitan entender el resto de la investigación. Para ello, se ha expuesto la metodología y pasos que se encuentran detrás de un ciberataque y la importancia que cobran hoy en día las Tácticas, Técnicas y Procedimientos (TTPs) frente a los Indicadores de Compromiso (IOCs), focalizando la investigación en aquellos.

A continuación, se han analizado los principales *frameworks* de ciberinteligencia, enfocando su análisis en la posible ulterior aplicación de Inteligencia Artificial, de modo que un caso práctico acontecido en España ha servido para ilustrar la funcionalidad de estos *frameworks* y constatar que Mitre Att&ck es el más potente y recomendado de ellos para su uso en la investigación.

Ante los desafíos inherentes de la escasez y el desequilibrio de datos se ha optado por diversas estrategias que han codificado, tratado e incrementado artificialmente los datos, logrando así mejorar la calidad y equilibrio del conjunto final.

Finalmente, tras la implementación de técnicas basadas en cadenas de clasificadores y el algoritmo *AdaBoost*, se propone un modelo que alcanza un rendimiento aproximado del 80%, en la predicción de una, dos o tres tácticas desconocidas en un ciberataque.

Esta investigación no solo valida los objetivos e hipótesis planteadas, sino que también proporciona una contribución significativa al ámbito de la ciberseguridad, dotando a los profesionales de herramientas avanzadas para luchar contra los riesgos presentes en el ciberespacio.

ÍNDICE

LISTADO DE FIGURAS

LISTADO DE TABLAS

Capítulo 1. Introducción

1 Motivación

Es un hecho que la delincuencia está migrando de forma progresiva desde un plano físico hacia otro cibernético, con importantes consecuencias estructurales y conceptuales para la sociedad [1]. De forma particular, tal y como puede observarse en la Figura 1, en España la cibercriminalidad ha registrado incrementos de hechos conocidos que van desde cerca de noventa mil infracciones en el año 2016 (92.716) hasta más de trescientas setenta mil en el año 2022 (374.737), representando en la actualidad el 16,1% del total de delitos registrados por las Fuerzas y Cuerpos de Seguridad. Paralelamente el volumen de ataques que no se esclarecen es progresivamente creciente y la sensación de impunidad ante estos delitos por parte de ciudadanos, empresas y administraciones es patente en los últimos años conforme indican los datos de cibercriminalidad publicados por el Ministerio del Interior de España (2022) [2].

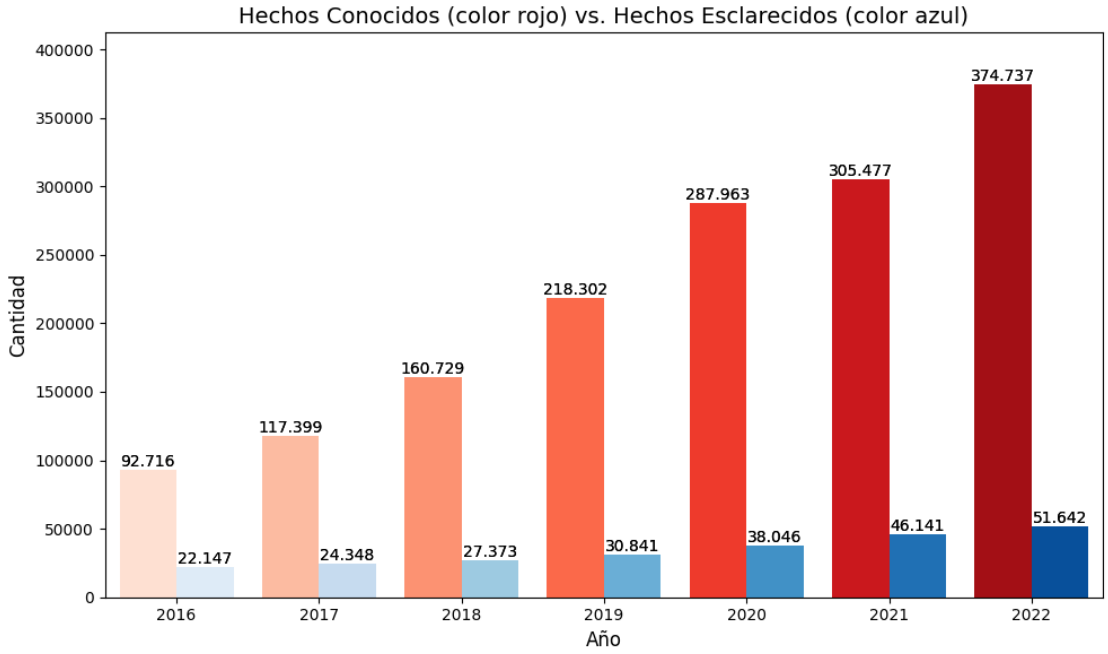

Figura 1. Evolución del número de delitos cometidos a través de las TIC en España a lo largo de los últimos años. Fuente: Elaboración propia con datos del Ministerio del Interior de España

En los últimos años se han detectado varios factores que han motivado estos incrementos tan sustanciales:

1. Tal y como reflejan sucesivamente de forma anual los informes IOCTA -*Internet Organized Crime Threat Assessment*- emanados de la Agencia Europea Europol [3], una proliferación de modelos delincuenciales basados en la subcontratación de uno o varios pasos de un ataque, a través de lo que se ha denominado *Crime as a Service*. Este fenómeno, que, si bien también está presente en la delincuencia tradicional, hace que, por el elevado componente técnico de la criminalidad, introduzca una variable especialmente compleja en la ecuación.

2. Una creciente dependencia de las redes y sistemas de información para el funcionamiento de los servicios esenciales de la sociedad. El comercio electrónico y la prestación de servicios de la Administración pública a los ciudadanos se han desplazado hacia el ciberespacio. Así mismo, la aparición de nuevas tecnologías como el *Internet of Things* (IoT), la Industria 4.0 y las *Smart Cities* han generado nuevos ámbitos de actuación para los cibercriminales.
3. Un aumento de la superficie de exposición en el ciberespacio de activos tecnológicos empresariales y domésticos, principalmente debido al masivo despliegue de soluciones de computación en la nube. Esta circunstancia se ha agudizado con la pandemia mundial ocasionada por el virus SARS-CoV-2 [4].
4. Siguiendo lo indicado por el NIS Investment Report [5], una carencia de inversión en materia de ciberseguridad por parte de empresas, y en especial por parte de Administraciones Públicas.
5. La compleja atribución de infracciones delictivas en el ciberespacio por parte de la Administración de Justicia debido, entre otros factores, a la deslocalización -en especial incrementada por el uso masivo de redes *Darknet* y la generalización del uso de redes privadas virtuales (VPN)- y la diversa legislación penal en los estados objeto y origen de ataques [6].

Las Fuerzas y Cuerpos de Seguridad estatales son los principales responsables de la investigación y persecución de conductas delictivas, incluidas aquellas acontecidas en el ciberespacio, o empleando éste como medio para la comisión. Además de contar con el respaldo legal para la persecución de la cibercriminalidad, el ciudadano y las empresas tienen a estas organizaciones como referencia en la lucha contra el delito en el ciberespacio, acuden a ellos en caso de sufrir un ataque y confían en estos organismos para el mantenimiento de la seguridad ciudadana en el ciberespacio. El volumen de información en poder de estas organizaciones es muy elevado, pero en la actualidad no se dispone de metodología que permita realizar labores de tratamiento e integración de los datos asociados a este tipo de delincuencia tan tecnificada, y por ende no pudiendo elaborar inteligencia de valor en su seno.

La compartición de Indicadores de Compromiso (IOCs) es uno de los principales mecanismos de defensa, prevención y respuesta ante ataques en el ciberespacio. Este concepto ha sido ampliamente desarrollado en la ciberseguridad, y comprende un esquema básico y flexible de compartición en formato XML que incluye *hashes* asociados a una muestra de *malware* (MD5, SHA2, SHA3, etc.) u otros observables de interés (direcciones IP, dominios, etc.). Alrededor de este concepto se han diseñado potentes herramientas de *Cyber Threat Intelligence* (CTI) focalizadas en el tratamiento e intercambio de IOCs de amplio uso en la comunidad internacional, como MISP -*Malware Information Sharing Platform*- [7] u OpenCTI [8], ambas dos auspiciadas de un modo u otro por entidades públicas europeas. Si bien esta información puede ser de gran utilidad por su versatilidad y capacidad de despliegue, en especial a corto plazo para evitar ataques a través del bloqueo de estos indicadores en soluciones antivirus/antimalware/EPP o EDR, y de este modo proteger a la comunidad frente a ataques masivos como el acontecido en 2017 relativo al *ransomware* Wannacry, la determinación de Tácticas, Técnicas y Procedimientos (TTPs) se ha convertido en el mejor camino para afrontar el estudio y respuesta a este fenómeno delincuencial [9]. Esto se fundamenta en el hecho de que el modus operandi empleado por los atacantes, al igual que ocurre en la delincuencia

tradicional, suele mantener trazas de constancia y estabilidad a lo largo del tiempo en la gran mayoría de los casos, en contraposición con el carácter más dinámico del enfoque basado en los IOCs, dado que pueden modificarse y variar con gran facilidad y rapidez, en especial en la actualidad con la proliferación de la computación en la nube.

La Inteligencia Artificial (IA) se ha convertido en una herramienta imprescindible en cualquier campo de la vida real. En este sentido, el *Machine Learning* (ML) o aprendizaje automático es una rama de la inteligencia artificial que busca generar mecanismos de aprendizaje en las máquinas. En la actualidad, la ciberseguridad hace uso de esta rama en dos principales vertientes; la detección de anomalías y la detección de patrones, que principalmente se traducen en la práctica en la detección de intrusiones, en el análisis de *malware* y en la detección de correos electrónicos tipo *spam* y ataques *phishing* [10]. No obstante, el espectro potencial de actuación de la Inteligencia Artificial en el campo de la ciberseguridad es amplio y aún por explorar en muchos casos [9] [10]. La aplicación de la IA en la ciberseguridad no es una opción si se pretende hacer frente adecuadamente al fenómeno delincuencial actual, que aprovecha a su vez los potenciales de la misma en su beneficio [13].

Por ello, en este trabajo se busca la implementación de una metodología que permita contribuir a la identificación de patrones técnicos de actuación de actores en determinados ataques en el ciberespacio basándose en el estudio masivo de TTPs.

2 Hipótesis de trabajo

Las hipótesis que se plantean en esta investigación son las siguientes:

- HIPÓTESIS 1: La Inteligencia Artificial, y de forma particular el *Machine Learning*, proveen herramientas técnicas para la detección, trazado y predicción de patrones de comportamiento en materia de ciberataques basándose en información previa recopilada de este tipo de incidentes. Esta información, si es alimentada por fuentes diversas y confiables, y se estructura y almacena adecuadamente, permitirá disponer de un banco de datos lo suficientemente amplio que garantice su utilidad.

- HIPÓTESIS 2: Una vez se registra un ciberataque, la Inteligencia Artificial permite la atribución, con un determinado grado de precisión y cautela, de la posible existencia de determinadas actuaciones o tácticas ejecutadas por los cibercriminales en un ataque en base a los patrones de comportamiento detectados previamente.

3 Objetivos

En consonancia con las hipótesis planteadas, los objetivos generales y específicos de la tesis son los siguientes:

3.1 Objetivo principal

Obtener un modelo que permita servir de apoyo en la determinación de patrones de comportamiento y en la predicción de tácticas empleadas en determinados ataques a sistemas de información u operación. Este objetivo principal pretende alcanzarse a través de la consecución de los siguientes objetivos específicos.

3.2 Objetivos específicos

- OBJETIVO 1: Analizar la problemática actual existente respecto al incremento de delitos cometidos a través de las Tecnologías de la Información y la Comunicación (TIC) y posibles acciones que puedan ayudar a paliar ese hecho.

- OBJETIVO 2: Determinar aquellos *frameworks* de ciberinteligencia que puedan servir como base de cara a estructurar los datos objeto de tratamiento asociados a ciberataques, con vistas al posterior estudio de estos mediante IA. (Mitre Att&ck, CyberKill Chain, Modelo Diamante…).

- OBJETIVO 3: Identificar un conjunto de datos adecuado y realizar el tratamiento necesario para disponer de una información fiable y suficiente que represente los ciberataques actuales.

- OBJETIVO 4: Definir un modelo estructurado de atribución de patrones y tácticas de ciberataques.

4 Metodología

Para afrontar esta investigación, se plantea el análisis de los diversos *frameworks* de ciberinteligencia que existen en la actualidad para modelizar ataques, incluso en combinación [14] y, en su caso y tras observar su adecuada pertinencia, emplear el *framework* Mitre Att&ck [15]. Esta metodología o enfoque presenta diversos puntos relevantes y de interés; como la versatilidad, el detalle y profundidad técnica y, en especial, la enorme y creciente extensión de su uso en la comunidad de ciberinteligencia a nivel mundial. Mitre Att&ck define las TTPs a través del uso de matrices que recogen las tácticas, técnicas y procedimientos empleados por los delincuentes de forma recurrente. De forma específica, cuenta con catorce grandes grupos de tácticas:

- Reconnaissance o Reconocimiento del adversario (TA0043)
- Resource Development o Desarrollo de recursos (TA0042)
- Initial Access o Acceso Inicial (TA0001)
- Execution o Ejecución (TA0002)
- Persistence o Persistencia (TA0003)
- Privilege Escalation o Escalada de privilegios (TA0004)
- Defense Evasion o Evasión de Defensas (TA0005)

- Credential Access o Acceso mediante credenciales (TA0006)
- Discovery o Descubrimiento (TA0007)
- Lateral Movement o Movimiento lateral (TA0008)
- Collection o Colección (TA0009)
- Command and Control o Mando y control (TA0011)
- Exfiltration o Exfiltración (TA0010)
- Impact o Impacto (TA0040)

Estas tácticas se corresponden con las sucesivas fases teóricas que pueden tener lugar en un ataque determinado, pudiendo no obstante no ser secuenciales y experimentar avances y retrocesos en el proceso de ataque. Si bien este *framework* puede ser un punto de partida, la metodología puede adaptarse, incrementando o reduciendo tanto las TTPs como los actores, el software o cualquier otro parámetro empleado en ataque o relacionado con él. A su vez dispone de un elevado número de técnicas y subtécnicas distribuidas por cada táctica.

En base a esta premisa, la investigación se centrará en determinar una metodología que permita asociar comportamientos o TTPs detectados en un determinado incidente o ataque conforme el modelo Mitre Att&ck a actores prefijados de los que se tiene conocimiento de su modus operandi u otros parámetros con un determinado grado de certidumbre. Para ello se cuenta con la posibilidad de asociar TTPs tanto a determinados actores maliciosos, así como a herramientas software, que puedan ser empleados por aquellos. En este sentido, de cara al tratamiento de todo el volumen de información que se pretende analizar, se emplearán técnicas de *Machine Learning*, de modo que se llevará a cabo un análisis pormenorizado de algoritmos, variables más relevantes y otros parámetros asociados.

Esta investigación se ha llevado a cabo en la franja temporal 2021-2023, empleando para ello los datos disponibles en el *framework* Mitre Att&ck Enterprise, concretamente en el apartado *Working with ATT&CK-Accessing ATT&CK Data* de la página web oficial[1], que ofrece una base de conocimientos amplia con observaciones reales de grupos, software, TTPs, CVE, defensas a aplicar, etc., especificados para tres campos diferentes; “Enterprise” o genérico, “Mobile” o entornos móviles e “ICS” orientado a sistemas de control industrial. Adicionalmente, y de cara a disponer de un mayor espectro de información, se valorará la inclusión de toda aquella información de la que se pueda tener conocimiento en base a informes de ciberinteligencia de consultoras especializadas, trabajo de campo propio, o el uso de herramientas como TTDrill [16] u otras posibilidades basadas en inteligencia artificial [17]–[19].

En la actualidad existen escasas investigaciones focalizadas en el uso de inteligencia artificial en el modelo Mitre Att&ck. La más relevante puede considerarse [20], que aplica algoritmos de clusterización para determinar posibles asociaciones de TTPs y que servirá

[1] https://attack.mitre.org/resources/working-with-attack/

de gran apoyo en la investigación. Así mismo, la investigación se apoyará en los modelos de vectorización de componentes de las matrices Mitre como hacen [21], [22] en entornos móviles para poder realizar predicciones. Por otro lado, y más allá de la inteligencia artificial, existen estudios acerca de la predicción de posibles secuencias de TTPs empleando teoría de juegos [23] y cadenas de Markov [24].

5 Estructura de la tesis doctoral

Esta tesis se organiza alrededor de los objetivos planteados, mediante el desarrollo de los siguientes capítulos:

Capítulo 2; Cibercriminalidad y ciberseguridad. En este capítulo se definirán los conceptos y términos más relevantes para la ciberseguridad que servirán de base para el resto de la tesis doctoral. Junto a ello se analizará brevemente la principal normativa legal de aplicación a nivel español en materia de cibercrimen.

Capítulo 3; Análisis de los marcos de ciberinteligencia para el tratamiento de los datos con IA. En este capítulo se analizan los marcos de inteligencia que pueden ser útiles para la aplicación de algoritmos de *Machine Learning*. Para ello, en primer lugar, se presenta una visión general del concepto de ciberinteligencia y todas sus variantes y posibles aplicaciones en ciberseguridad. A continuación, se detallan tres marcos de ciberinteligencia: Diamond Model o Modelo Diamante, Cyberkill Chain y Mitre Att&ck; se aportan sus fortalezas y debilidades desde una perspectiva que tiene en cuenta la aplicación no sólo desde un punto de vista matemático, sino también desde una perspectiva holística que garantice que el marco utilizado es el más adecuado. Este estudio y análisis pone de relieve la aplicación práctica de los modelos a un caso real de ataque de *ransomware*. Finalmente se concluye que el marco Mitre Att&ck es el más adecuado para la aplicación de técnicas de IA debido a su potencia, su idoneidad para el procesamiento de datos y la existencia de conjuntos de datos disponibles entre otros aspectos derivados del estudio de diecisiete variables.

Capítulo 4; Machine Learning. En este capítulo se detallan los principales algoritmos que pudieran ser de aplicación a la investigación a la par que se plantea la necesidad y utilidad del aprendizaje automático en la ciberseguridad. Se analizan técnicas de aprendizaje supervisado, aprendizaje no supervisado y las métricas más relevantes. Finalmente se plantean las estrategias a seguir para la predicción multiclase focalizadas en determinar varias tácticas desconocidas de forma simultánea mediante Cadenas de Clasificadores y *One Vs Rest*.

Capítulo 5; Tratamiento de datos en el marco de ciberinteligencia Mitre Att&ck. En este apartado, y conforme lo establecido en el capítulo 3, Mitre Att&ck se ha podido estimar como el *framework* de ciberinteligencia más adecuado para poder parametrizar y definir un ciberataque mediante el uso de técnicas de *Machine Learning*. En consecuencia, los datos disponibles en bruto necesitan un profundo tratamiento para la aplicación de un modelo matemático, de modo que se exponen diferentes opciones y se selecciona la codificación *one hot* frente a otras como codificación binaria o mediante *hashing* que presentan menores ventajas. En este capítulo se conforma un primer dataset representativo de muestras de ciberataques.

Capítulo 6; Estudio del incremento y corrección de los datos. Tras analizar el conjunto de datos obtenido tras el tratamiento del capítulo anterior, el objetivo marcado en este apartado es la obtención de datos adecuados, en cantidad, calidad y con suficiente fidelidad a la realidad, para la posterior aplicación de algoritmos de aprendizaje automático a un conjunto de datos binarios que muestran las acciones de ataques en Mitre Att&ck. Para ello se profundiza en estrategias como Neighbourhood Cleaning Rule (NCR), SMOTE, o Borderline SMOTE para incrementar y corregir, en la medida posible, el dataset.

Capítulo 7; Aplicación y análisis de los algoritmos de *Machine Learning*. En este capítulo se define el modelo propuesto, mediante la aplicación de las estrategias para afrontar la ausencia de variables (Classifier Chains, One Vs Rest), los algoritmos escogidos (Support Vector Machines, Regresión Logística, Decision Tree, AdaBoost y Random Forest) y tras su estudio, se determina la solución que ofrece un mejor rendimiento en base a los resultados de una métrica definida *ad hoc* para la investigación.

Capítulo 8; Conclusiones y futuras líneas de investigación. En este capítulo se analizarán las contribuciones que se realizan en los anteriores capítulos, así como las posibles utilizaciones en otras temáticas más allá de la ciberseguridad, y de forma específica del *framework* Mitre Att&ck. Finalmente se incluyen las posibles futuras líneas de investigación que puedan derivarse de esta investigación. En especial atendiendo a las posibles modificaciones que pueda sufrir el modelo de ciberinteligencia empleado con el devenir de la nuevas TTPs que se detecten con el paso del tiempo.

Capítulo 2. Ciberseguridad y Cibercriminalidad

1 Introducción

En este capítulo se pretende ofrecer un enfoque acerca del estado de situación actual en el que se encuentra tanto la ciberseguridad como la cibercriminalidad. En este sentido, se presentan como base los conceptos básicos de ciberseguridad y cibercriminalidad, incluyendo a su vez una reseña al encaje de los ciberataques más relevantes en los tipos penales del código penal español.

2 Conceptos básicos de ciberseguridad y cibercriminalidad

Previamente a la exposición de estos conceptos, parece oportuno explicar en qué consiste y qué es el ciberespacio, entidad en la que se desarrollan ambas fenomenologías, así como la cibernética como disciplina específica. En este sentido, el prefijo "ciber" tiene su origen en el término "kubernan", cuyo significado puede asimilarse a gobernar, de modo que la cibernética se puede definir como la ciencia o el arte de gobernar. Esta premisa nos indica la relevancia del concepto a tratar en este punto. Consecuentemente, el término ciberespacio procede de la contracción de los términos "Cibernética" y "Espacio". Hoy en día este término se emplea ampliamente sin tener un concepto claro del mismo, pero que con una serie de factores puede aproximarse a su concepción siguiendo a [25]:

• Elementos tangibles (ej: hardware)
• Elementos intangibles (ej: software, información)
• Real y virtual
• Actividades e interacciones (su finalidad principal es la comunicación entre personas)
• No solamente internet (ej: SCADA)
• Universalidad o ubicuidad (ámbito global y ausencia de fronteras)
• Transversalidad (implicaciones en el conjunto de la sociedad)
• Basado en la información
• Rapidez y dinamismo
• Anonimato
• Coste reducido
• Capacidad de crecimiento

En base a estos puntos, puede considerarse el ciberespacio como " *un conjunto de sistemas de información interconectados, dependientes del tiempo, junto con los usuarios que interactúan con estos sistemas*" [26]. El ciberespacio por tanto es una figura híbrida, tanto real y virtual como tangible e intangible, que se compone de una serie de capas materiales o físicas (infraestructura), capas semánticas (información), y una serie de capas humanas (individuos).

2.1 Concepto de ciberseguridad

Al igual que sucede con el término ciberespacio, no existe un concepto estandarizado que indique qué se entiende por ciberseguridad, puesto que las definiciones oficiales difieren entre países o entre organizaciones internacionales (ISO, UIT…). No obstante, existe cierta coincidencia en la protección de activos tecnológicos, y muy especialmente

aquellos referentes a la información. De este modo, puede considerarse la ciberseguridad como: "*Conjunto de políticas, herramientas o mecanismos para el resguardo de la confidencialidad, disponibilidad e integridad de los activos tecnológicos de los usuarios del ciberespacio*".

2.2 Objetivos de la ciberseguridad.

En línea con el concepto anterior, los principales objetivos de la ciberseguridad, pivotando sobre la protección integral de activos, siguiendo a [25], pueden concretarse en:

• Alinear la protección de activos con la estrategia de la organización
• Reducir la exposición a los riesgos del ciberespacio
• Asegurar el cumplimiento de leyes, normativas y regulaciones
• Gestionar y mantener la seguridad de la información y sus sistemas
• Mantener un control eficaz de toda la información y sus sistemas
• Asegurar un intercambio de información entre usuarios autorizados
• Crear y mantener la capacidad de detectar, responder y recuperarse de incidentes
• Utilizar sistemas de información fiables
• Fomentar y asegurar una cultura de ciberseguridad y buenas prácticas

2.3 Dimensiones de la ciberseguridad

La confidencialidad, disponibilidad e integridad constituyen el eje alrededor del cual se desarrolla la ciberseguridad en todos sus ámbitos, tanto desde el punto de vista técnico como legal, con medidas enfocadas a cada una de ellas en particular.

La **Confidencialidad** es el atributo de los datos, objetos o recursos que garantiza el acceso y uso oportuno y confiable de los mismos por parte de los usuarios que estén debidamente autorizados. Por ello las medidas asociadas a la confidencialidad se focalizan en la limitación selectiva y robusta de accesos a activos tecnológicos. Así pues, es imprescindible que en la protección de redes y sistemas de información se actúe, tanto en el almacenamiento, en el procesamiento o en el tránsito de los datos. En el extremo opuesto a la confidencialidad puede observarse el concepto de divulgación, que se alcanza una vez las medidas aplicadas para garantizar la confidencialidad son vulneradas.
Los ataques más relevantes en materia de confidencialidad son el compromiso de la información, la captura de tráfico -*sniffing*-, el robo de contraseñas a través de ingeniería social, *shoulder sourfing*, etc. No obstante, ha de tenerse presente que no siempre las vulneraciones de la confidencialidad tienen lugar en el marco de un ataque, los errores humanos siguen siendo muy frecuentes.
Frente a ello, las medidas que pueden aplicarse irán desde una adecuada implementación de protocolos de cifrado de la información, una política de controles de acceso y autenticación, una clasificación de la información en función de su criticidad y por supuesto un profundo entrenamiento del personal de la entidad que reduzca la posibilidad de comisión de errores humanos [27].

La **Integridad** es el atributo de los datos, objetos o recursos que garantiza su fiabilidad y exactitud, de modo que no se alteran sin control los parámetros originales que poseían los

datos. La integridad ha de contemplarse desde una doble perspectiva; impedir que sujetos autorizados hagan modificaciones no autorizadas, así como que sujetos no autorizados hagan modificaciones no autorizadas. En el extremo opuesto a la integridad de los activos se encuentra la alteración de los mismos, que se alcanza una vez las medidas aplicadas para garantizar la confidencialidad son vulneradas.
Los ataques más relevantes que afectan a la integridad son virus, bombas lógicas, *backdoors*, accesos no autorizados, errores en la codificación de aplicaciones, etc.
Las medidas aplicadas para paliar estos ataques pueden variar desde una estricta política de accesos y autenticaciones, implementaciones de criptografía (cotejo de hashes), cifrado de la información y por supuesto un profundo entrenamiento del personal de la entidad que reduzca la posibilidad de comisión de errores humanos [27].

La **Disponibilidad** es la característica, cualidad o condición de la información de encontrarse a disposición, de forma ininterrumpida, de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones. A grandes rasgos, la disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran.
Los ataques más relevantes que afectan a la disponibilidad son las denegaciones de servicio en sus diferentes variantes (*SLOWHTTP, TCP SYN FLOOD, UDP FLOOD, SOCKSTRESS, DNS AMPLIFICATION*, etc.).

Frente a esta tipología de ataques que explotan esta dimensión puede emplearse balanceadores de carga, redes de entrega de contenido (CDNs), firewall adecuadamente configurados, así como técnicas específicas para cada tipología, como puede ser las actuar sobre los tiempos de *SYN-RECEIVED*, habilitar *SYN CACHE* para soportar más peticiones, habilitar *SYN COOKIES*, etc.) [28].

Adicionalmente, hay otras dimensiones no tan relevantes como las anteriores, pero a su vez con afectación a la ciberseguridad de los activos; Autenticación: propiedad que permite identificar el generador de la información, y No repudio: propiedad por la que el emisor no puede negar que envió porque el destinatario tiene pruebas del envío.

2.4 Eventos, incidentes y ataques

De acuerdo con lo recogido en la norma UNE-ISO/IEC 27000:2014, un **evento o suceso de seguridad** de la información hace referencia a una "*Ocurrencia detectada en el estado de un sistema, servicio o red que indica una posible violación de la política de seguridad de la información, un fallo de los controles o una situación desconocida hasta el momento y que puede ser relevante para la seguridad".* Por otro lado, la NIST-800-61 indica que un evento es "*cualquier ocurrencia que pueda observarse en un sistema o red*". Por tanto, un evento puede hacer referencia a una conexión de un dispositivo USB, una petición legítima a un recurso web, o una acción de bloqueo por parte de un firewall a una petición a una IP considerada maliciosa.

De forma más concreta, un **incidente de seguridad** es una violación o amenaza inminente de violación de las políticas de seguridad informática de una organización.

Por otro lado, la CNSSI 4009-2015 lo recoge como "u*n evento que conlleve un peligro real o potencial en la confidencialidad, integridad o disponibilidad de la información que un sistema procesa, almacena o transmite, o que constituya una violación o inminente*

amenaza de violación de las políticas de seguridad, los procedimientos de seguridad o las políticas de uso aceptadas." Por ello, un incidente de ciberseguridad puede hacer referencia a una carencia de disponibilidad de acceso a los sistemas, y por ende a la información, por un corte de suministro de energía eléctrica en un servidor, que haya sido provocado por motivos meteorológicos.

Finalmente, un **ataque**, de acuerdo con la norma UNE-ISO/IEC 27000:2014, es la "*Tentativa de destruir, exponer, alterar, inhabilitar, robar, acceder sin autorización o hacer un uso no autorizado de un activo*". Además, la norma NIST SP 800-82 Rev.2, define ataque como "*toda tentativa de obtener un acceso no autorizado a los servicios, recursos o informaciones de un sistema, o intento de comprometer la integridad, disponibilidad o confidencialidad de un sistema*". De este modo, un ataque puede hacer referencia a la penetración en los sistemas de una entidad financiera por el compromiso de las credenciales de un usuario con privilegios de administrador de sistemas.

En definitiva, puede contemplarse una aproximación a estos tres términos relacionados con la ciberseguridad teniendo en cuenta que se integran de forma progresiva en el subsiguiente, de modo que todo ataque es un incidente, y todo incidente es un evento, englobando este último concepto a los otros dos términos.

2.5 Agentes de ataque

Generalmente los ataques a sistemas de información u operación se llevan a cabo por entidades o individuos, y suelen ser los siguientes:

- ESTADOS: Entidades u organizaciones políticas vinculadas con un territorio que cuentan con fuerte presupuesto y capacidades humanas y materiales en materia ofensiva en el ciberespacio capaces de desarrollar Tácticas Técnicas y Procedimientos (TTPs) avanzados. Estos agentes se encuentran asociados al concepto de ciberguerra o guerra informática, y a acciones de espionaje, sabotaje o desinformación.
- GRUPOS APT: Entidades, en muchos casos asociadas a Estados, que cuentan con alta tecnificación y capacidad para la materialización de ataques, principalmente de penetración y posterior exfiltración de información sensible, en organizaciones estratégicas de todo el planeta. La finalidad de estos agentes es la obtención de lucro económico o la adquisición de ventaja geoestratégica.
- CIBERCRIMINALES: Grupos organizados con capacidad de ejecución de actividades delictivas a través de redes y sistemas de información. Estos agentes conforman la migración natural de la delincuencia tradicional hacia el ciberespacio. Se encuentran fuertemente vinculados con el fenómeno de *Crime As a Service* (CaaS), de modo que no es necesario disponer de elevados conocimientos técnicos de cara a ejecutar acciones.
- CIBERTERRORISMO: Grupos o individuos capaces de ejecutar actividades delictivas con finalidades previstas en el ordenamiento jurídico español[2] asociadas a:
    - Subvertir el orden constitucional, o suprimir o desestabilizar gravemente el funcionamiento de las instituciones políticas o de las estructuras

[2] Artículo 573 Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal.

económicas o sociales del Estado, u obligar a los poderes públicos a realizar un acto o a abstenerse de hacerlo.
    - Alterar gravemente la paz pública.
    - Desestabilizar gravemente el funcionamiento de una organización internacional.
    - Provocar un estado de terror en la población o en una parte de ella.
- GRUPOS SUBVERSIVOS: Grupos o individuos con motivaciones de diversa índole (políticas, económicas, sociales) capaces de alterar el normal funcionamiento de las instituciones mediante acciones a través de las redes y sistemas de información.

2.6 Sistemas de información (IT) y de operación (OT)

La aplicación de las nuevas tecnologías en la sociedad comenzó mediante la implantación de soluciones basadas en las tecnologías de la información que facilitasen la automatización de tareas basadas en datos e informaciones. No obstante, progresivamente, y ante las ventajas competitivas que conllevaba su implantación en entornos industriales, se fueron aplicando soluciones computacionales, denominadas OT (*Operational Tecnologies*) o ICS (*Industrial Control Systems*), complementando las IT (*Information Tecnologies*). A su vez, los ICS se pueden subdividir en dos grandes categorías; por un lado, los sistemas basados en DCS, enfocados hacia industrias basadas en procesos continuos (véase por ejemplo la generación de energía eléctrica o sistemas petroquímicos), y por otro los sistemas SCADA, dirigidos a sistemas industriales con mayor número de procesos discretos (véase por ejemplo la automoción, la alimentación o la electrónica). La integración de tecnologías operacionales en la industria viene representada en la pirámide de automatización industrial, reflejada en la Figura 2.

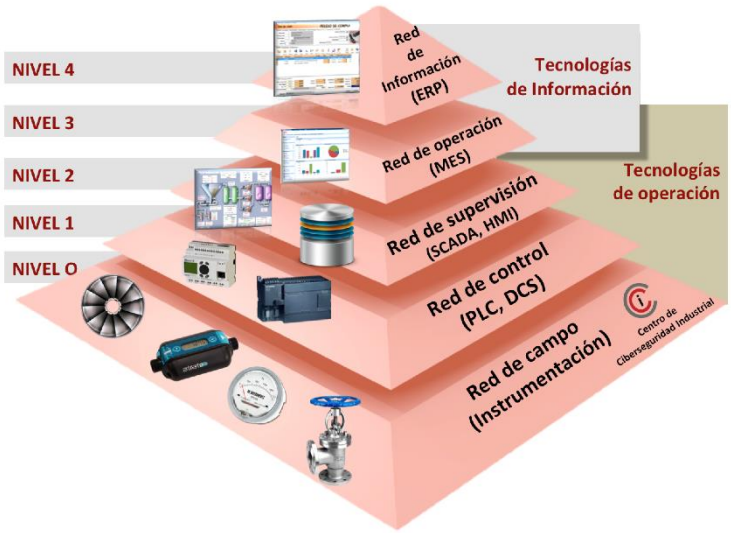

Figura 2. Pirámide de automatización industrial. Fuente: Centro de Ciberseguridad Industrial

En un primer momento, los protocolos y tecnologías empleados en la industria se conformaban en entornos aislados, y estas soluciones, por lo general, eran patentadas. Estas tecnologías lo que permitían era la transformación de señales en datos, y de datos en información comprensible por el usuario. Hoy en día, y ante la necesidad de

interconexión que requiere la operativa diaria de la industria, estos sistemas aislados están paulatinamente migrando hacia tecnologías basadas en protocolos TCP/IP, entornos Windows, navegadores web, etc. A pesar de que esta transición, y la conectividad que permiten, genera beneficios empresariales, también conlleva una serie de riesgos, dado que la seguridad tradicionalmente se ha enfocado hacia la protección de la información, no disponiéndose de esta madurez en entornos industriales [29].

Por otro lado, la seguridad de las tecnologías de información se encuentra enfocada, de forma preponderante, hacia la protección de la confidencialidad de los datos manejados en los sistemas, mientras que las tecnologías de operaciones están dirigidas hacia la disponibilidad que deben ofrecer esta tipología de sistemas, siendo esta disponibilidad un elemento crítico e imprescindible. Este último punto refuerza la necesidad de las autoridades competentes, de disponer de información acerca de posibles incidentes de ciberseguridad que puedan afectar a sistemas OT, ligados íntimamente a la prestación de servicios esenciales por parte de entidades públicas y privadas. Por otro lado, los protocolos de comunicación difieren; en sistemas IT se emplean protocolos de comunicación comunes, como pueden ser el TCP/IP UDP, mientras que los SCI emplean protocolos específicos tipo modbus, profibus etc. Finalmente, el ciclo de vida de los sistemas IT no va más allá de 3 a 5 años en circunstancias normales. Sin embargo, los sistemas Sistemas de Control Industrial tienen un ciclo de más de 15 años, dado que son sistemas que se diseñan en muchos casos de forma específica y no pueden englobarse en un proceso de producción a gran escala [30].

Gran parte de las infraestructuras críticas actuales se sustentan sobre sistemas OT, los ataques a estos sistemas no son tan frecuentes como a los sistemas IT, pero suceden. Prueba de ello es el ataque a una planta depuradora en Estados Unidos a la que se accedió presuntamente tras búsquedas en la herramienta *Shodan*, y mediante el compromiso de la aplicación de escritorio remoto TeamViewer, mediante la que se pudo incrementar en cien veces los valores de sosa cáustica en el agua potable[3] alterando los sistemas de control industrial de la instalación.

2.7 Evolución de los ataques

2.7.1 Tendencias por sector atacado

Si se analizan los ataques a los principales sectores estratégicos, entendiendo éstos como "*cada una de las áreas diferenciadas dentro de la actividad laboral, económica y productiva, que proporciona un servicio esencial o que garantiza el ejercicio de la autoridad del Estado o de la seguridad del país*", tal y como se establece en [31, p. 8]. En base a los datos de la consultora tecnológica FireEye [32], y mostrados en la Figura 3, la evolución que han sufrido a nivel mundial los ataques en función del sector estratégico víctima en los últimos años permite extraer las siguientes conclusiones:

a) El sector **financiero** sigue siendo el ámbito más atacado. Las entidades bancarias son, al igual que en el ámbito físico, el foco de atención de los delincuentes, dado que en sus sistemas se maneja dinero y activos financieros de gran valor.

[3] https://u-gob.com/el-hackeo-que-sufrio-la-planta-de-tratamiento-de-aguas-en-florida/

b) El sector **sanitario**, a raíz de la pandemia, ha pasado a ocupar el segundo lugar en el ranking de sectores atacados. Los hospitales y otros centros médicos son el objetivo favorito de los delincuentes de *ransomware*. En 2020, 560 instalaciones sanitarias se vieron afectadas por ataques de *ransomware* solo en Estados Unidos[4]. Estos incidentes no solo cuestan a las a las víctimas millones de dólares en recuperación, sino que también han provocado retrasos en el tratamiento de los pacientes, y posiblemente la pérdida de vidas. En septiembre de 2020, un ataque de *ransomware* provocó el fallo de los sistemas informáticos en la Clínica Universitaria de Düsseldorf, lo que obligó a trasladar a pacientes en estado crítico a otras instalaciones[5], y En Estados Unidos, un ataque provocó retrasos en el tratamiento de pacientes con cáncer en la University of Vermont Medical Care y otros centros.

c) El sector **energía** está reduciendo progresivamente los ataques. En este sector se incluyen grandes que disponen de gran capacidad de inversión en ciberseguridad que se refleja en una reducción del número de ataques que sufren.

d) El **resto de los sectores** han sufrido un pronunciado descenso por el fuerte ascenso en este último año de los ataques a los sectores financiero y sanitario.

[4] https://thecrimereport.org/2021/08/18/hospitals-cyberattacks/
[5] https://elpais.com/internacional/2020-10-03/ciberataque-a-un-hospital-aleman-en-tiempos-de-pandemia.html

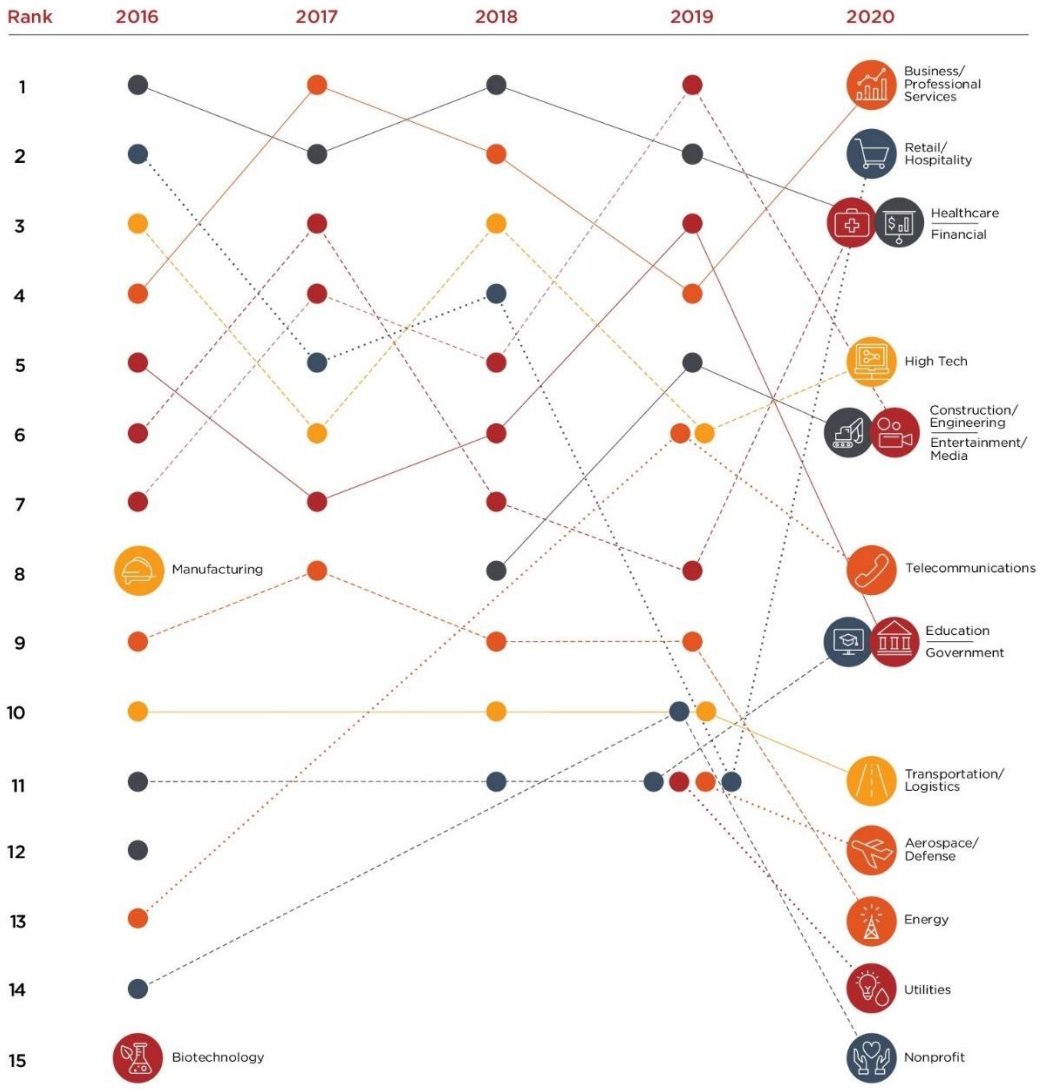

Figura 3. Evolución de los sectores estratégicos a lo largo de los últimos años. Fuente: FireEye

Conforme los datos de la consultora FireEye, publicados en [32], y mostrados en la Figura 4, pueden observarse los precios que alcanzan en foros *underground* la venta de accesos a sistemas de información en base a los sectores estratégicos atacados. Tal y como se muestra en la Figura 3, El sector administración es uno de los sectores más atacados. Una de las principales causas de este hecho puede ser el reducido precio medio los accesos ilegítimos a sus sistemas (2.283 $), en comparación con, por ejemplo, el precio medio de venta de un acceso a un sistema del sector energético con valor medio de más de 40.000 $, o del sector legal 80.000 $. Esto ofrece una idea del porqué de esa oleada de ataques al sector público a nivel internacional y nacional que en España pudo observarse con el ataque al SEPE[6]. Estos ataques no sólo desvían recursos públicos recursos públicos a

[6] El ciberataque al Servicio Público Estatal de Empleo fue un *ransomware* a gran escala que produjo un elevado impacto en la operativa de la Administración española en lo concerniente a los pagos y gestiones de prestaciones por desempleo de sus ciudadanos a lo largo de varias semanas. Puede consultarse más

economías ilícitas, sino que las víctimas incurren en costes que superan con creces los rescates, ya que la administración se ve obligada a pagar por el análisis forense digital, el aumento de personal comunicaciones de crisis y otros costes.

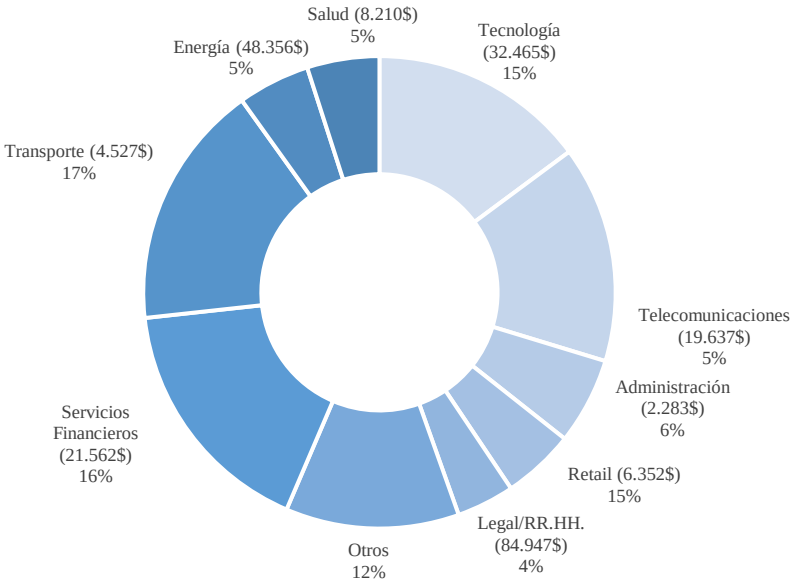

Figura 4. Precio medio de la venta de accesos por sector estratégico y volumen porcentual que representa. Fuente: FireEye

2.7.2 Tendencias por zona geográfica

De acuerdo con la información aportada por la consultora McAfee en su informe de julio de 2020 [33], en el gráfico de la Figura 5 se representa el número de IPs atacantes en eje de ordenadas o eje Y, y el número de organizaciones atacadas en eje de abscisas o eje X. Siendo cada círculo uno de los sectores estratégicos, puede observarse que, de los tres países origen:

A) Las IP origen China, con datos en color rojo, realiza ataques con un muy elevado número de IPs, en consonancia con su elevado volumen de población, y se centra en ataques a servicios financieros, con 290M de ataques.
B) Las IP origen Rusia, con datos en color azul, emplea menor número de direcciones IP, pero el número de ataques a organizaciones, en especial financiera es mucho más elevado. Esto se encuentra relacionado con el supuesto auspicio de organizaciones criminales que se ha argumentado de forma recurrente por organizaciones occidentales [34].
C) Las IP origen Irán, con datos en color verde, realiza ataques mucho más puntuales y de forma más reducida.

información acerca del impacto en la siguiente noticia:
https://www.elmundo.es/economia/2021/03/09/6047578dfc6c83411b8b4795.html

Con respecto a las direcciones IP de destino de ataques, occidente es el blanco de gran parte de los ataques, en especial Estados Unidos y Europa.

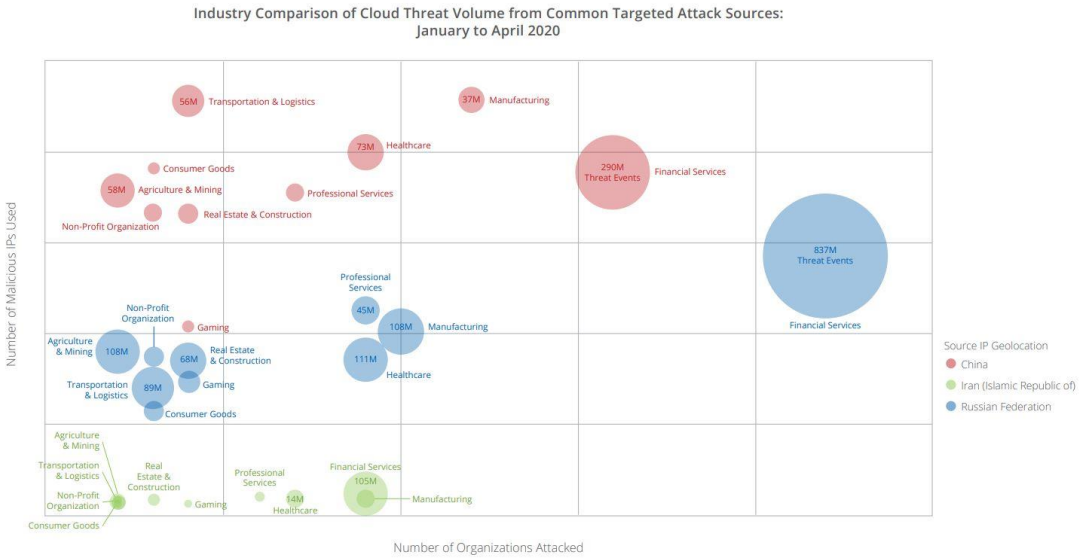

Figura 5. Zona geográfica origen de IP implicadas en ataques. Fuente: McAfee

3 La metodología de un ataque. El caso del *ransomware*.

Los ataques se llevan a cabo siguiendo una serie de pasos que se detallarán más adelante en esta investigación a través de los modelos de ciberinteligencia que estructuran las fases. No obstante, para contextualizar la investigación y profundizar en qué se entiende por un ciberataque, se refleja a continuación un ejemplo de cómo se ejecuta uno de los ataques más relevantes en la actualidad, las infecciones por *ransomware*. Para ello, se agrupan las acciones en tres fases principales:

- Compromiso inicial
- Consolidación y preparación
- Cifrado y explotación

3.1 El compromiso inicial

En este estadio los atacantes consiguen la penetración en los sistemas a través de diversas metodologías. De forma general se lleva a cabo a través del uso de alguno de los tres vectores de ataque que se reflejan a continuación, en el que se toman como referencia los datos estadísticos de la consultora Group-IB [35], reflejados a su vez en el gráfico de la Figura 6:

- Acceso remoto mediante uso de credenciales válidas (52% de las ocasiones)
- *Phishing* o *Spear Phishing* (29% de las ocasiones)
- Explotación de vulnerabilidades de sistemas expuestos (17% de las ocasiones)

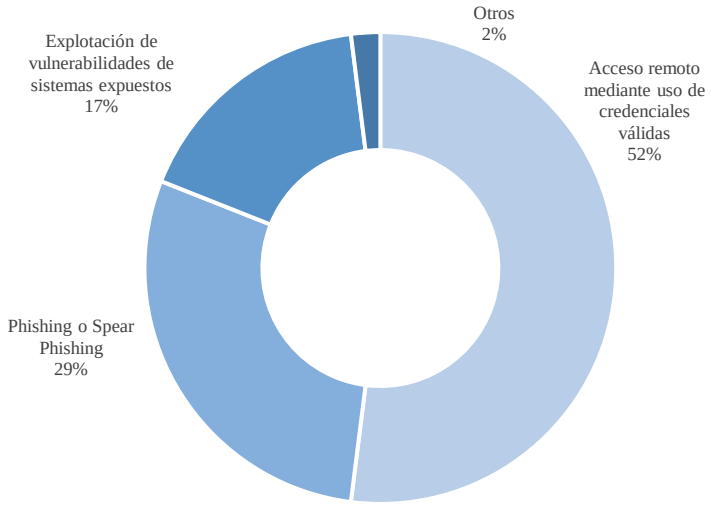

Figura 6. Vectores iniciales de compromiso. Fuente: Group-IB

3.1.1 Acceso remoto mediante uso de credenciales válidas

El comercio y reutilización de credenciales se ha convertido en los últimos años, en el principal vector de entrada para ataques de gran impacto, especialmente *ransomware*. Hay que tener en cuenta que un usuario medio genera numerosas credenciales a lo largo de su vida para múltiples servicios que acaban almacenados en servicios *cloud*, en muchos casos vulnerables. El usuario en muchas ocasiones emplea contraseñas similares para múltiples servicios web, y en muchos casos contraseñas por defecto o triviales como puede extraerse del estudio recogido en [36] y reflejado en la Tabla 1 que refleja los usuarios y contraseñas con mayor presencia.

Tabla 1. Contraseñas y usuarios más empleados. Fuente: ESET

	PASSWORD	USERNAME
1	admin	admin
2	root	root
3	1234	guest
4	12345	1234
5	guest	support
6	password	user
7	support	super
8	Admin	11111
9	super	manager
10	x-admin	tellabs

Existen *leaks* de información de webs y colecciones de gran volumen de contraseñas en foros *underground* (ej: *Collections* #2-#5). Junto a ello, recursos como el portal *Have i been pwned* recopilan leaks de credenciales con gran potencial para el inicio de un ataque.

El modus operandi de los delincuentes para explotar esta vía consiste en que emplean estas colecciones para probar la validez de posibles accesos remotos asociados a esos usuarios y contraseñas, y en caso de que funcionen y permitan accesos a servicios, en especial accesos VPN (*Virtual Private Networks*), venden esas credenciales en foros *underground* o a través de otros canales. Normalmente se trata de foros abiertos para países de la federación rusa, véase XSS o Raidforums (portales que han sufrido consecuencias legales por alojamiento de este tipo de información), así como portales onion de la darknet TOR para el resto de países [34].

3.1.2 Phishing/Spear Phishing

Otro de los vectores de entrada más empleados en ciberataques es el uso de ataques *phishing* o *spear phishing*. La psicología detrás de muchas de estas técnicas basadas en ingeniería social es aprovecharse de las emociones humanas y comportamiento.

El modelo más conocido para explicar este fenómeno es el ciclo de ataque de ingeniería social de Kevin Mitnick, descrito en su libro *The art of deception: controlling the human element of securit*y [37].

Este modelo de ataque tiene cuatro fases: investigación, desarrollo de investigación, desarrollo de la relación y la confianza, explotación de la confianza y utilización de la información.

Desde el inicio de la pandemia del COVID-19 se produjo un notable incremento de distribución de *phishing* a nivel mundial, en especial bajo temática diversa asociada a la enfermedad. Muestra de ello es el elevado número de certificados TLS o SSL que se registraron con temáticas COVID-19, como puede observarse en la Figura 7, en base a los datos de la consultora Sophos [38], donde se refleja el elevado incremento que en el mes de marzo de 2020 se observó del número de certificados asociados a dominios con “COVID” o semejante en su nombre. Esto pone de manifiesto que los atacantes emplean temáticas diversas y adaptables en función de los intereses de las víctimas.

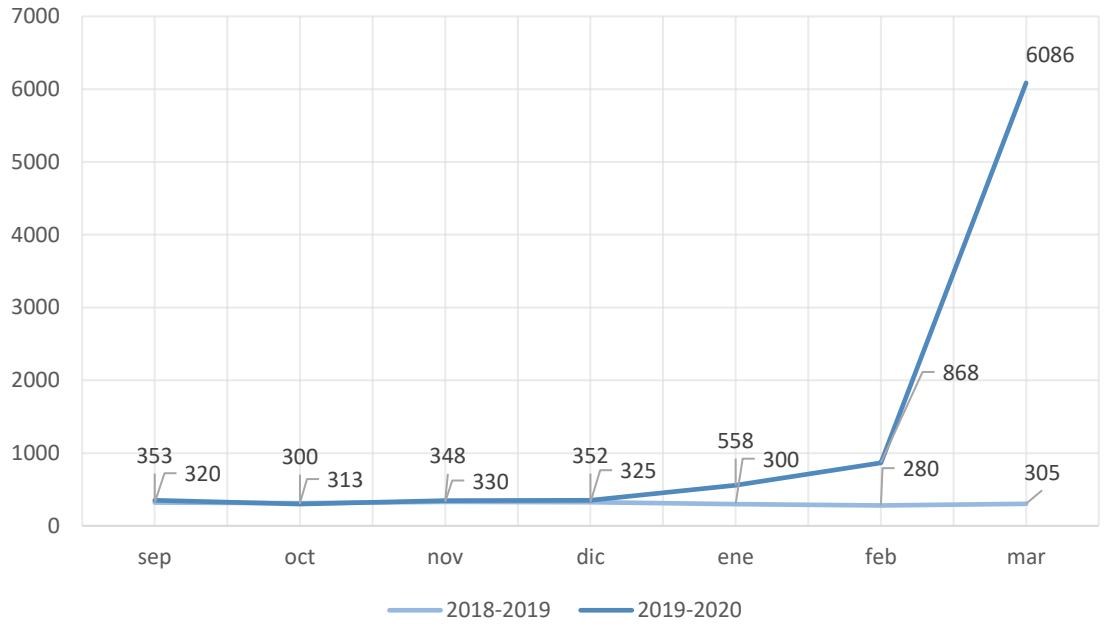

Figura 7. Certificados TLS nuevos por mes asociados a dominios con temáticas "COVID" o "CORONA". Fuente: Sophos

Actualmente, estos ataques presentan las siguientes características:

1. Amenaza mixta o multivectorial. El *Spear phishing* utiliza una mezcla de suplantación de correo electrónico, URLs dinámicas, ficheros maliciosos y drive-by downloads para eludir las defensas tradicionales.
2. Uso de vulnerabilidades *zero day*. Los ataques avanzados de ataques de *spear-phishing* aprovechan las vulnerabilidades de día cero en navegadores, *plugins* y aplicaciones de escritorio para comprometer los sistemas.
3. Uso de links a plataformas de almacenamiento en la nube tipo Drive, OneDrive, Dropbox, etc., sobre todo de cara a ejecutar fraudes.
4. Falsificaciones de correo electrónico bien elaboradas: Las amenazas de correo electrónico de *spear-phishing* suelen estar dirigidas a individuos, por lo que no se parecen mucho parecido con el spam de gran volumen que inunda Internet.

3.1.3 Explotación de vulnerabilidades de sistemas expuestos

Otra de las puertas de entrada de *malware* para los ciberataques es la explotación de vulnerabilidades o fallos en el desarrollo del software, que puede llegar a no ser detectado hasta pasado un tiempo suficiente como para pasar desapercibida la presencia en el sistema (zero days). Generalmente, las aplicaciones y sistemas operativos que se emplean quedan obsoletos con el paso del tiempo y deben actualizarse. Si no se actualizan los sistemas, en especial aquellos críticos, se corre el peligro de que sean comprometidos mediante la explotación de vulnerabilidades. En este sentido, existe una base de datos de vulnerabilidades que las organiza con un código CVE (o Common Vulnerability Exposure), que refleja el año de descubrimiento y un código en su nomenclatura. Normalmente para gran parte de las vulnerabilidades descubiertas se realiza una prueba

de concepto o demostración de la explotación, que lleva asociado un *malware* o *exploit*. Existe un recurso denominado EXPLOIT-DB en la que puede consultarse miles de *exploits*. También en la plataforma Github puede consultarse numerosas pruebas de concepto que demuestren la utilidad de *exploits*. Es decir, los cibercriminales disponen de una base de datos de vulnerabilidades y otra de herramientas de ataque.

En los últimos tiempos, las vulnerabilidades más explotadas han sido aquellas asociadas a accesos remotos a sistemas de información. Las empresas y la administración se han visto obligadas a volcar gran parte de su carga de trabajo a través del uso de acceso remoto a través de protocolos RDP (*Remote Desktop Protocol*), protocolos que habitualmente emplean el puerto 3389 y que permite accesos remotos a equipos de forma ágil. El gran incremento de accesos RDP desde el inicio de la pandemia en marzo del año 2020 provocó un incremento cuantitativo muy notable de estos accesos. En este sentido, de acuerdo con la telemetría de la consultora ESET, se registraron incrementos del +102%, 139,8% y +40% respecto de conexiones de clientes únicos en protocolo RDP a lo largo de los tres últimos trimestres del año 2020. Esto, consecuentemente, fue un considerable foco de penetración en las organizaciones. Así, vulnerabilidades como las relativas a servicios de la empresa Citrix (CVE-2019-19781), Pulse Secure (CVE-2019-11510), Netlogon (CVE-2020-1472) o Log4Shell (CVE-2021-44228) se han consolidado como principal puerta de entrada de los atacantes en los últimos tiempos [36].

3.2 Consolidación y preparación

En esta fase, y una vez el atacante está dentro de los sistemas realiza las conexiones con los servidores de mando y control (C2), que puede asimilarse al campamento base de ataque de los delincuentes, realiza los movimientos laterales necesarios en la red y lleva a cabo la obtención de privilegios *root* para poder desplegar sus capacidades por los sistemas atacados. Una de las cautelas empleadas de cara a conseguir la máxima anonimización posible por parte de estos grupos una vez están dentro de los sistemas es el uso tanto de VPN o *proxies*, como de conexiones torificadas que redirigen el tráfico de salida a TOR desde el panel de C2 o la IP o IPs de ataque. En este sentido, el EC3 (European Cybercrime Centre) de Europol junto con otras policías consiguió retirar el servicio DoubleVPN, que mediante varios saltos permitía un alto grado de anonimización en el lanzamiento de ataques *ransomware*.

De forma específica, se suelen emplear las siguientes herramientas en un ataque ransomware:

- **Backdoor**: Programa cuyo objetivo principal es permitir a un actor de la amenaza emitir comandos de forma interactiva en el sistema en el que está instalado.
- **Credential Stealer**: Una utilidad cuyo propósito principal es acceder, copiar o robar credenciales de autenticación.
- **Downloader**: Un programa cuyo único propósito es descargar (y en su caso ejecutar) un archivo desde una dirección especificada, y que no proporciona ninguna funcionalidad adicional ni admite ningún otro comando interactivo.
- **Dropper**: Un programa cuyo propósito principal es extraer, instalar y potencialmente ejecutar uno o más archivos.
- **Launcher**: Programa cuyo propósito principal es ejecutar uno o más archivos. Se diferencia de un *dropper* o un instalador en que no contiene ni configura el archivo, sino que simplemente lo ejecuta o lo carga.

- **Ransomware**: Programa cuyo propósito principal es realizar alguna acción maliciosa (como cifrar datos), con el objetivo de extraer el pago de la víctima para evitar o deshacer la acción maliciosa.
- **Otros**: Incluye todas las demás categorías de *malware*, como utilidades, keyloggers, tunnelers y mineros de datos.

A lo largo de los últimos dos años se ha podido constatar que se emplea un determinado tipo de software en un elevado número de ataques de relevancia al tener capacidad para aglutinar gran parte de estas TTPs; Cobalt Strike. Han sido innumerables los ataques detectados con presencia de servidores *Command and Control* de Cobalt Strike. Esta herramienta presenta las siguientes características:

- Sobre el papel es una herramienta de auditoria o *red teaming*, pero es empleada por atacantes para fines delictivos.
- *Cobalt Strike* es una herramienta muy empleada en ataques debido a la potencia y versatilidad que tiene. Es un producto de pago con un precio aproximado de 3500$/año. No obstante, hay versiones fácilmente descargables en foros *underground*.
- Tiene su estructura basada en Metasploit y Armitage, un software de red team o explotación muy extendido y presente, por ejemplo, en la suite Kali Linux.
- Basa su funcionamiento en un agente o *payload* denominado Beacon o baliza, que aporta una gran cantidad de funciones para el atacante y permite conexión por HTTP/HTTPS o DNS que incluyen, entre otras, ejecución de comandos, registro de claves, transferencia de archivos, proxy SOCKS, escalada de privilegios, uso de la solución mimikatz, escaneo de puertos y movimiento lateral. En definitiva, se trata de una herramienta que permite manejar todos los pasos del ataque.

Figura 8. Imagen promocional Cobalt Strike. Fuente: HelpSystems.

3.3 Cifrado y explotación

En esta fase final el atacante procede al cifrado de los ficheros, exfiltrando la información al servidor de mando y control y, en muchos casos, destruyendo los back-ups que detecte para evitar la restauración.

Una vez se dispone de acceso al sistema con los privilegios necesarios, el proceso de cifrado generalmente es el mismo; para ello el atacante:

1. Generar de manera totalmente aleatoria claves secretas de algoritmos simétricos robustos para cifrar los archivos. Suele emplearse especialmente AES-256, normalmente por la velocidad de cifrado que ofrece (1Gb/s) y con claves no muy extensas de 128 o 256. Conforme los datos de la consultora Group-IB [35] pueden observarse en la

Tabla 2 las preferencias que tiene cada uno de los grupos atacantes más reconocidos a nivel internacional.

Tabla 2. Algoritmos de cifrado empleado por las principales familias de ransomware. Fuente: Group-IB

Familia de ransomware	Algoritmo de cifrado de archivos	Algoritmo de cifrado de claves
Clop	RC4	RSA-1024
Conti	AES-256	RSA-4096
Darkside	CustomSalsa20	RSA-1024
Dharma	AES-256	RSA-1024
DopplePaymer	AES-256	RSA-2048
Egregor	ChaCha8	RSA-2048
Lockbit	AES-128/256	RSA-2048
Maze	ChaCha8	RSA-2048
Netwalker	ChaCha8	Curve25519
OldGremlin	AES-256	RSA-4096
Prolock	RC6	RSA-1024
Pysa	AES-256	RSA-4096
Ragnar Locker	Custom Salsa20	RSA-2048
RansomEXX	AES-256	RSA-4096
REvil	Salsa20	Curve25519+AES
Ryuk	AES-256	RSA-2048
Sekhmet	ChaCha8	RSA-2048

Estas claves se distribuyen:

a) embebidas dentro del *malware* codificadas para su liberación en el momento oportuno.

b) se solicitan a un servidor C2 de los atacantes, normalmente a través de conexiones torificadas.

c) generación individualizada in situ de claves para cada una de las extensiones a cifrar.

2. Las claves AES que han servido para el cifrado de los archivos se cifran a su vez con criptografía asimétrica, generalmente son RSA 1024-2028 de modo que emplean una clave pública únicamente descifrable con una clave privada en poder de los atacantes.

Por tanto, el mecanismo habitual es cifrar con algoritmos simétricos los ficheros, dado que es mucho más rápido y el volumen suele ser elevado. Posteriormente, la clave secreta empleada se cifra a su vez tras ello con algoritmos asimétricos, mucho más versátiles para manejar la extorsión por parte de los atacantes.

4 Conductas delictivas asociadas

Habitualmente, suele considerarse que todo ataque o incidente dirigido o acontecido en redes o sistemas de información se trata de una conducta delictiva. No obstante, existe, en base a la última ratio que caracteriza el derecho penal, una serie de conductas que no pueden incluirse en las leyes penales pese a tratarse de ataques a la confidencialidad, la integridad o la disponibilidad, bien sea por el reducido impacto que producen en las redes y sistemas o por otras circunstancias que lo desaconsejan, siendo el Derecho Administrativo una posible solución para regular esas conductas más allá del Derecho Penal. Un gran número de los ordenamientos jurídicos a nivel mundial encuentran basados en lo recogido en el convenio de Budapest de 2001 un referente. A continuación, se detalla su contenido al entenderse de utilidad para aproximar el concepto de cibercriminalidad.

4.1 El convenio de Budapest y sus modificaciones

El Convenio de Budapest, oficialmente conocido como el Convenio sobre Ciberdelincuencia, se erige como el primer tratado internacional diseñado específicamente para combatir la ciberdelincuencia. Esta iniciativa se originó bajo el amparo del Consejo de Europa y, a lo largo de los años, ha visto incrementar el número de estados que se han adherido a sus disposiciones.

Este convenio se concibió con dos objetivos esenciales:

- Armonización legislativa: Busca establecer un marco legal común entre los países signatarios. Esta uniformidad se propone tanto en el ámbito penal, definiendo y sancionando conductas delictivas en el ciberespacio, como en el ámbito procesal, estableciendo procedimientos adecuados para la investigación y persecución de estos delitos.
- Cooperación internacional: El convenio promueve una colaboración activa entre los países miembros para enfrentar, de manera efectiva, delitos que, por su naturaleza virtual, no conocen de fronteras.

En la parte sustantiva el convenio define algunos estándares mínimos a la hora de definir los tipos principales de delitos que se consideraban comprendidos en su ámbito. Concretamente:

- Delitos contra los sistemas y datos informáticos
- Delitos informáticos
- Delitos relacionados con el contenido
- Delitos relacionados con la propiedad intelectual

Adicionalmente, aborda aspectos procesales y herramientas de investigación, como la conservación de datos (tanto de contenido como de tráfico), la recopilación y confiscación de evidencia digital y la obtención de datos en tiempo real.

4.2 Tipos penales en España

Con la finalidad de llevar a cabo una asociación de los incidentes notificados, en relación con la presunta infracción penal en la que podrían estar incurriéndose, se empleará como apoyo la siguiente Tabla 3, basada en lo contemplado en la Ley Orgánica 10/1995 de Código Penal y la Circular 3/2017 de Fiscalía General del Estado de España. En esa tabla se recogen de forma esquemática los tipos penales más comunes que pueden presentarse en relación con las tipologías de incidentes que pueden detectarse.

Tabla 3. Principales tipologías delictivas en el ámbito de la seguridad de la información y la operación de las tecnologías en España. Fuente: elaboración propia

TIPO PENAL	TIPOLOGÍA	OBSERVACIONES
Art. 189 Pornografía infantil	Delito público perseguible de oficio	
Art. 197.2 (Intrusión datos personales) **Art. 197 bis1** (Intrusión con acceso ilegítimo) "*CRACKING*" **Art. 197 bis2** (Ej: *Sniffing*/Man In the Middle) "CIBERESPIONAJE" **Art. 197 ter** (Ej: *Malware* / Credenciales) "CIBERMUNICIÓN INTRUSIÓN"	Delito semipúblico*	*En el Art. 197 bis y ter del Código Penal (CP) no se requerirá denuncia siempre que se trate de: -Conductas de funcionario público (art. 198 CP) -Conductas que afecten a los intereses generales (Ej: sea del ámbito competencial del CSIRT de la Administración pública. -Conductas que afecten a una pluralidad de personas (Ej: espionaje informático de organismos e instituciones del Estado)
Art. 264 (Ej: *Deface-ment*/*Ransomware***) "DAÑOS INFORMÁTICOS" **Art. 264 bis** (Ej: DoS/DDoS) "DENEGACIONES DE SERVICIO" **Art. 264 ter** (Ej: Malwa-re/Credenciales) "CIBERMUNICIÓN DAÑOS" **Art. 264 quater** Personas Jurídicas	Delito público perseguible de oficio***	**En incidentes tipo *ransomware* deberá contemplarse posibilidad de presencia del Art. 169 Amenazas. ***El Art. 264 y ss. del CP necesita que la acción y el resultado sean "graves". Si el incidente afecta a Infraestructuras Críticas o Servicios Esenciales, no requerirá la doble gravedad.
Art. 573 Terrorismo "CIBERTERRORISMO"	Delito público perseguible de oficio	
Art. 270 y ss. Propiedad intelectual	Delito semipúblico	
Art. 278 Espionaje industrial	Delito semipúblico	

Capítulo 3. Análisis de marcos de ciberinteligencia para el tratamiento de datos de IA

1 Introducción

Los recientes avances tecnológicos (Internet de las cosas, computación en nube, redes de comunicaciones móviles, etc.) han provocado un cambio de paradigma en lo que respecta a los servicios, los negocios y la gestión y transmisión de datos. Todas estas actividades están migrando desde el mundo físico al mundo cibernético [39], donde son más accesibles y su ejecución resulta más cómoda para el usuario general. Sin embargo, la seguridad de la información en el ciberespacio (ciberseguridad) [40] debe ir acompasada con ese abandono del plano físico, tarea que no siempre es fácil debido a la complejidad y sofisticación de los ciberataques existentes hoy en día [41].
La ciberinteligencia, es decir, las tecnologías y medios para adquirir y analizar información con la finalidad de identificar, rastrear y predecir capacidades, intenciones y actividades en el ciberespacio para mejorar la toma de decisiones [42], se ha convertido en una rama indispensable de la ciberseguridad. La creciente incapacidad para dar una respuesta adecuada a los ataques contra los sistemas de información u operación ha orientado progresivamente los esfuerzos de las organizaciones hacia la aplicación de medidas preventivas eficaces que permitan evitar estos ataques, o al menos reducirlos. Esto se lleva a cabo mediante la aplicación de acciones preparatorias que aumenten la resiliencia en caso de ciberataque, lo que ha dado lugar a un aumento considerable de la importancia de la ciberinteligencia en los últimos años [43]. Los *frameworks* o marcos [44] de ciberinteligencia pueden caracterizarse como esquemas que permiten construir y organizar el trabajo de forma más eficiente proporcionando conceptos, bibliotecas, entidades y patrones de diseño comunes. En este sentido, actualmente existen diferentes marcos que permiten estructurar la información y transformarla en inteligencia a través de un proceso generalmente cíclico. El objetivo de este capítulo es estudiar los marcos de ciberinteligencia más relevantes (en concreto, Diamond Model, Cyberkill Chain y Mitre Att&ck) utilizando un ejemplo real de ciberataque para determinar el más adecuado para su combinación con herramientas de inteligencia artificial (IA). El objetivo final de esta combinación es procesar y analizar con éxito información a gran escala relativa a ciberataques, donde información, en este contexto, se define como en la publicación de la Organización Internacional de Estándares (ISO) ISO/IEC 27000:2018: "*toda comunicación o representación de conocimientos, tales como hechos, datos, acontecimientos, cosas, procesos, ideas, conceptos u opiniones, en cualquier medio o forma, incluida la textual, numérica, gráfica, cartográfica, narrativa o audiovisual que, dentro de un contexto determinado, tenga un significado particular."* [7,8]. Esto permitiría automatizar y explotar la inteligencia generada, permitiendo una mejor protección de las organizaciones y la detección de patrones de comportamiento para inferir las acciones de los atacantes u orientar a las Fuerzas y Cuerpos de Seguridad en su lucha contra la cibercriminalidad. En este sentido, las novedades presentadas en este capítulo son:

- La explicación detallada y el estado actual del arte de los tres marcos de ciberinteligencia más relevantes: Diamond Model, Cyberkill Chain y Mitre Att&ck.

- Un estudio de caso de un ciberataque real para los tres marcos mencionados, haciendo hincapié en los puntos fuertes, los puntos débiles y la mejor perspectiva que ofrece cada uno de ellos.
- La definición de diecisiete variables de interés para comparar, entre otras características, la eficacia, adaptabilidad y sencillez de los marcos.
- Un análisis, basado en estas diecisiete características, del marco más adecuado para combinarlo con métodos de IA y, más concretamente, con modelos de Aprendizaje Automático o *Machine Learning*. Con este análisis, también se concluye la idoneidad de cada marco en función de la situación específica y el conocimiento de interés.

El capítulo se estructura como sigue: el apartado 2 analiza el concepto y la disciplina de la ciberinteligencia, mientras que el apartado 3 incluye el estado actual del arte relativo a los marcos de ciberinteligencia. A continuación, el apartado 4 aplica los modelos a un caso de estudio específico de un ataque real. Los resultados obtenidos se analizan y discuten en el apartado 5. Finalmente, en el apartado 6, se ofrecen las conclusiones y se proponen futuros trabajos relacionados en la materia.

2 Antecedentes de la ciberinteligencia

2.1 Concepto y características

El concepto clásico de inteligencia está relacionado con la cualidad de la mente que permite al ser humano asimilar, comprender, razonar, tomar decisiones y formarse una visión o idea de una realidad concreta [46]. En relación con esto, en el contexto de la seguridad, la inteligencia puede entenderse como "*El producto de la recopilación, evaluación, análisis, integración e interpretación de toda la información disponible que sea inmediata o potencialmente significativa para la planificación y las operaciones."* [29]. La aplicación de esta definición al ámbito del ciberespacio aproxima el concepto de ciberinteligencia, un término con creciente presencia y relevancia en la sociedad actual ante el crecimiento exponencial de la criminalidad, que requiere respuestas reactivas, pero sobre todo preventivas. La ciberinteligencia es, por tanto, una disciplina eminentemente proactiva que utiliza diversas ramas de la seguridad de la información para alcanzar sus objetivos (gestión de vulnerabilidades, gestión de amenazas, respuesta a incidentes, etc.) [47]. Esto se consigue desarrollando herramientas que apoyen la toma de decisiones sobre los riesgos a los que están expuestas las organizaciones. En este sentido, [29] define la ciberinteligencia como "*Actividades de inteligencia en apoyo de la ciberseguridad mediante las que se identifican y detallan las ciberamenazas y se analizan las intenciones y oportunidades de los ciberadversarios con el fin de identificar, localizar y atribuir las fuentes de los ciberataques*". Por tanto, este concepto puede agruparse en tres ramas principales de aplicación en ciberseguridad, que se describirán más adelante: *inteligencia sobre ciberamenazas*, *respuesta a incidentes* y *gestión de vulnerabilidades*.

Los datos utilizados para producir la inteligencia deben basarse en pruebas y cumplir unos criterios mínimos de calidad. Deben por tanto ser útiles para la organización en función de sus características y de sus Necesidades Prioritarias de Información o *Priority Information Requirements* (PIR). [47]. Además, los datos deben ser procesables y manejables, característica básica para procesarlos mediante técnicas de aprendizaje automático [48]. De especial interés en el ciberespacio es el concepto de OSINT (*Open-Source Intelligence Techniques*) [49], que permite la gestión de los datos existentes procedentes de todo tipo de fuentes abiertas para ser procesados con fines de inteligencia.

Con el fin de poder estructurar y procesar esta información y, en particular, de representarla de forma visual y fácil de usar, en los últimos años se ha desarrollado un gran número de plataformas de ciberinteligencia que recopilan inteligencia a partir de diversas *fuentes de* información, generando intercambios principalmente mediante el uso de los estándares STIX [50] o TAXII [51].

2.2 Niveles y tipologías

Las diversas tipologías de destinatarios y públicos objetivos presentes en la gestión de la ciberinteligencia hacen necesario establecer una estratificación o distinción; así, se pueden observar tres niveles [52]:

- El nivel estratégico se centra en apoyar la toma de decisiones sobre las políticas y objetivos de una organización. En este sentido, el conocimiento, estudio y posible evolución de los actores maliciosos involucrados que pueden influir en los riesgos de la organización son esenciales. Un ejemplo es el análisis de la exposición de la organización a los principales grupos que explotan *malware* del tipo *ransomware* o realizan acciones de Amenazas Persistentes Avanzadas (APT).
- El nivel táctico tiene por objeto apoyar la planificación de acciones específicas que permitan alcanzar los objetivos estratégicos de la organización, por ejemplo, el análisis de las Tácticas, Técnicas y Procedimientos (TTP) de ataque de un determinado actor malicioso.
- El nivel operativo se refiere a los conocimientos que permiten tomar decisiones en un corto espacio de tiempo en el marco de las acciones necesarias para prevenir un ataque o incidente determinado, por ejemplo, el análisis de los indicadores de compromiso (IOC) que representan una amenaza para la organización.

El tratamiento de los datos tradicionalmente asociados a la inteligencia operativa, es decir, los indicadores de compromiso (hashes, dominios, IP, etc.) reflejados en los peldaños inferiores de la pirámide del pánico de David Bianco (2013) [17] (véase la Figura 9), es extremadamente complejo debido a la enorme variedad, facilidad de sustitución y escasa correlación entre ellos. Por otro lado, los datos asociados a la inteligencia estratégica son a veces difusos, poco procesables y muy difíciles de obtener de forma estructurada. Por ello, esta investigación se centra en el desarrollo de productos de inteligencia táctica mediante el estudio exhaustivo de las TTP.

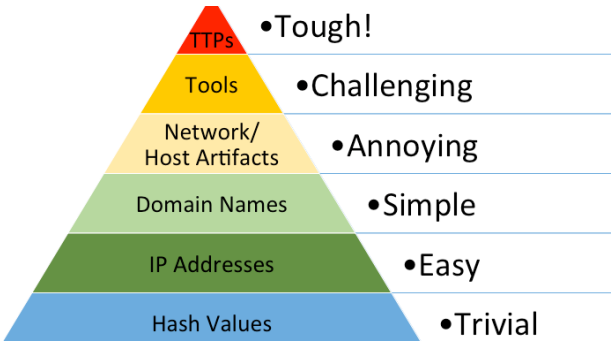

Figura 9. Pirámide del pánico [16]

Por otro lado, se han identificado tres ramas o tipologías de la ciberseguridad como aquellas en las que la ciberinteligencia tiene mayor utilidad y aplicación: inteligencia de amenazas, respuesta a incidentes y gestión de vulnerabilidades.

La inteligencia de amenazas o inteligencia sobre ciberamenazas o *cyberthreat intel* (CTI) está enfocada a la toma de decisiones sobre las políticas y objetivos de una organización en materia de seguridad de la información u operación. Para ello, trata de recopilar y analizar todo tipo de datos para hacer frente a los riesgos de la organización para la protección de sus activos tecnológicos con tres objetivos claros: que la inteligencia sea relevante, que sea precisa y que se ajuste en el tiempo a las necesidades. En este sentido, parece adecuado adoptar un modelo cíclico similar al de la inteligencia militar clásica, ampliamente discutido y estudiado en las últimas décadas, según el cual los productos se elaboran a través de un proceso recurrente y se actualizan en función de la realidad presente en cada periodo de tiempo [46]. Para la correcta implementación de este ciclo, es esencial la cooperación fluida entre los diferentes actores implicados.

- Fase 1. Planificación y gestión. En esta fase se identifican los PIR a partir de la definición previa de los activos críticos y las amenazas relevantes cuya explotación genera un impacto.
- Fase 2. Recogida. La recogida de datos se lleva a cabo siguiendo métodos estructurados basados en los requisitos de la fase anterior.
- Fase 3. Tratamiento. Los datos recogidos se procesan con las técnicas adecuadas.
- Fase 4. Análisis y producción. Se trata de una fase crítica del ciclo en la que la información obtenida en las fases anteriores se transforma en inteligencia. Actualmente, requiere la presencia del factor humano.
- Fase 5. Difusión. Es la fase en la que se proporciona la inteligencia a los responsables de la toma de decisiones para que dispongan de los conocimientos adecuados.
- Fase 6. Utilización. La toma de decisiones basada en el producto entregado se considera otro momento crítico y decisivo del ciclo.

La rama de respuesta a incidentes (IR) se centra en la respuesta integral a los incidentes detectados en la organización. La aplicación de la inteligencia a esta rama se basa en la propuesta de [53].

- Fase 1. Preparación. Esta fase consiste en establecer y formar un equipo de respuesta a incidentes y adquirir las herramientas y recursos necesarios.
- Fase 2. Detección y análisis. El objetivo en esta fase es limitar el número de incidentes que se producirán seleccionando y aplicando una serie de controles basados en los resultados de las evaluaciones de riesgos y, en su caso, tratando los riesgos residuales en primera instancia.
- Fase 3. Contención, erradicación y recuperación. Una vez conocida la afectación de un incidente, se incluyen en esta fase las acciones de respuesta para minimizar los daños y tratar de facilitar la continuidad del negocio en el menor tiempo posible.
- Fase 4. Actividad posterior al incidente. En esta fase se elaboran los informes pertinentes y se analiza el incidente a posteriori con el fin de obtener lecciones aprendidas y generar buenas prácticas.

La rama de gestión de vulnerabilidades (GV) tiene como objetivo la gestión integral de vulnerabilidades en la infraestructura de la organización, donde la ciberinteligencia juega un papel clave. Para ello, se adoptan los enfoques de [19,20] como referencia.

- Fase 1. Descubrimiento. Inventariar los activos que deben protegerse en la organización.
- Fase 2. Priorización. Analizar los riesgos asociados al compromiso de cada activo y su criticidad para la prestación del servicio.
- Fase 3. Evaluación. Determinar un perfil de riesgo de referencia sobre el que actuar.
- Fase 4. Información. Supervisar los activos y describir las vulnerabilidades.
- Fase 5. Corrección. Priorizar y remediar las vulnerabilidades según el perfil de riesgo asociado.
- Fase 6. Verificación. Ejecutar controles que verifiquen que se han eliminado las amenazas.

Vistos estos aspectos, sin duda sería de gran interés poder aplicar técnicas de IA a cada una de las fases de las tres ramas, dado que la IA puede identificar patrones en grandes conjuntos de datos para descubrir amenazas emergentes, mejorando la calidad y la rapidez de la CTI. Además, la predicción precisa de futuras vulnerabilidades permite una mejor gestión y mitigación de las mismas, reforzando la postura de seguridad. En relación con la respuesta a incidentes, la IA puede acelerar el proceso clasificando automáticamente los incidentes por gravedad y sugiriendo acciones de respuesta, liberando a los analistas humanos para que se centren en tareas críticas. Por lo tanto, es esencial encontrar el marco de ciberinteligencia más adecuado para la aplicación de algoritmos de IA.

3 Marcos de ciberinteligencia

Tal y como se ha mencionado anteriormente, en la actualidad existen tres marcos de ciberinteligencia que destacan por encima del resto; utilizan diferentes enfoques, pero todos ellos son de gran utilidad, tal y como se refleja en [55]. En esta Sección, se proporciona una descripción de su funcionamiento y estado del arte.

3.1 Modelo Diamante

El Modelo Diamante es un marco propuesto en [56] que busca un tratamiento integral del análisis de un ataque bajo una premisa sencilla: *el estudio de un adversario que desarrolla capacidades sobre una infraestructura dirigida a una víctima*. Se estructura como una serie de eventos que expresan las cuatro características clave de la definición anterior (adversario, capacidades, infraestructura y víctima) para cada ataque; estas características están estrechamente relacionadas entre sí y se configuran como los vértices de un rombo en forma de diamante (ver Figura 10). Además de las cuatro características clave, se definen metacaracterísticas y valores de confianza, que constituyen la mayor parte del modelo, de la siguiente manera: características clave (adversario, capacidades, infraestructura y víctima); metacaracterísticas (marca temporal, fase, resultado, dirección, metodología y recursos); y valores de confianza (cada característica clave o metacaracterística tendrá asociada una estimación de confianza, que representa la precisión de la fuente de datos o la confianza en las conclusiones extraídas).

Por lo que respecta a los parámetros o características clave de primer nivel, el Adversario es la persona o grupo de personas responsables de la explotación de las capacidades. Este parámetro será generalmente incompleto o incluso desconocido en las primeras etapas del análisis. Con el reciente desarrollo del *Crime as a Service* (CaaS), la distinción entre operador y cliente se hace más evidente.

En segundo lugar, la Capacidad incluye las herramientas utilizadas por el adversario en la explotación de la operación relativas a (1) capacidad/potencial: vulnerabilidades y áreas de exposición que puede emplear en relación con la víctima; (2) Arsenal: el conjunto de herramientas que el atacante puede utilizar, incluidos los paneles de mando y control (C2), entendidos no sólo como la simple infraestructura tecnológica, sino como los canales, estructuras y procedimientos que guían la operación de un ataque.

En tercer lugar, la Infraestructura describe aquellas estructuras físicas o lógicas utilizadas para el despliegue de las capacidades, siendo muy variable en tipo y volumen. Representa, por ejemplo, IP, dominios, direcciones de correo y dispositivos físicos. En concreto, y en línea con lo comentado sobre la proliferación del *CaaS*, se define de la siguiente manera: Tipología 1, totalmente controlada por los atacantes, y Tipología 2, controlada por intermediarios o proveedores de servicios (ISP, registradores de dominios, etc.).

Por último, la característica Víctima representa el objetivo del adversario contra el que se dirigen las capacidades. Puede comprender organizaciones, individuos, direcciones de correo electrónico, direcciones IP, etc. En este sentido, cabe distinguir entre (1) víctima-física, es decir, las organizaciones o sus empleados, y (2) víctima-activo, es decir, la superficie de ataque que se pretende atacar.

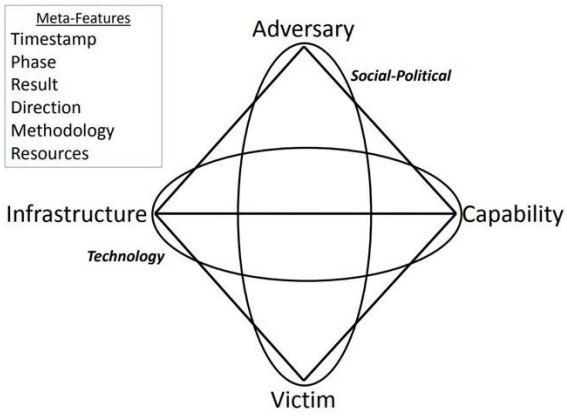

Figura 10. Representación del modelo de diamante [20].

Por otro lado, existe un segundo nivel de parámetros, las metacaracterísticas, que no se detallarán (no son de interés para esta investigación), pero que se mencionarán aquí: marca temporal, fase, resultados, dirección, metodología y recursos.
Este modelo puede aplicarse contemplando diferentes perspectivas en función del vértice que se tome como característica principal entre las cuatro (ver Figura 10) o del eje de dos existentes que se tomen como más relevantes, así se configuran el eje sociopolítico (adversario y víctima) y el eje tecnológico (infraestructura y capacidad). Este marco teórico es muy útil para el análisis integral de los ciberataques, especialmente por la

facilidad con la que es posible prever los movimientos futuros del adversario. Permite estudiar la etiología de la acción y generar hipótesis de autoría con un amplio margen para el analista. Sin embargo, es muy difícil recopilar información de forma estructurada, ya que carece de taxonomía propia. Este marco se aplicó en [57], donde se combinó con técnicas de aprendizaje automático (redes bayesianas) para integrar la detección de correlación de alertas. Una vez generada una alerta, reconstruye automáticamente los escenarios de amenazas pasados y predice las amenazas y vulnerabilidades futuras.
Formalmente, y según los autores, dada una característica *F* y un valor de confianza σ, un evento *E* puede representarse de la siguiente manera:

$$E = ((F_1, \sigma_1), (F_2, \sigma_2), (F_3, \sigma_3), \dots (F_n, \sigma_n))$$

donde *n* es el número de características implicadas. Un ejemplo típico podría ser el siguiente: adversario, capacidad, víctima, infraestructura, marca de tiempo inicial, marca de tiempo final, fase, resultado, dirección, metodología y recursos.

3.2 Cyberkill chain

Centrado principalmente en los ataques APT, este marco ha tenido un fuerte impacto en el campo de la ciberinteligencia desde su publicación en 2010 [58]. Fue desarrollado por la empresa estadounidense *Lockheed Martin* y trata de representar una serie de pasos que un atacante debe ejecutar para alcanzar su objetivo final basándose en el concepto F2T2EA (*Find, Fix, Track, Target, Engage, Assess*), una metodología de referencia en la doctrina militar estadounidense [59] (ver Figura 11). Cyberkill Chain establece siete pasos, consistentes en (1) Reconocimiento: conocer a la víctima mediante técnicas no invasivas; (2) Armamentización: generar la carga maliciosa a entregar; (3) Entrega: entregar el artefacto desarrollado o adquirido en el paso anterior; (4) Explotación: lograr la ejecución de código en el sistema de la víctima mediante la explotación de una vulnerabilidad u otros medios; (5) Instalación: instalar la muestra final de *malware*; (6) Mando y Control (C2): establecer un canal para comunicarse con el *malware* en el sistema de la víctima; y (7) Acciones sobre los Objetivos: lograr el objetivo del ataque, habiendo obtenido pleno acceso y comunicación.
Esta cadena es útil porque proporciona un enfoque estructurado y sistemático para comprender y abordar los ciberataques de principio a fin, lo que facilita la identificación de amenazas y la aplicación de estrategias para defenderse de ellas. Sus ventajas incluyen:

- Alerta temprana o prevención, analizando las debilidades potenciales de la infraestructura tecnológica en cada fase de un posible ataque antes de que pueda producirse.
- Optimización de los recursos, centrando las inversiones y los esfuerzos en las fases más vulnerables de la infraestructura.
- Sensibilizar a los trabajadores que no tienen conocimientos de ciberseguridad, ofreciendo una estructura gráfica y consecutiva que permita entender mejor qué es un ciberataque y los riesgos que genera en la organización.

Aplicando la Cyberkill Chain, es posible analizar las campañas de ataque para observar similitudes y diferencias entre las TTP de los distintos atacantes y, por tanto, la presencia

de patrones que apunten a un actor malicioso concreto. Además, al compartimentar las acciones, es posible observar, desde un nivel superior, los posibles objetivos hacia los que se dirigen las acciones en curso y anticipar el posible éxito de un ataque.

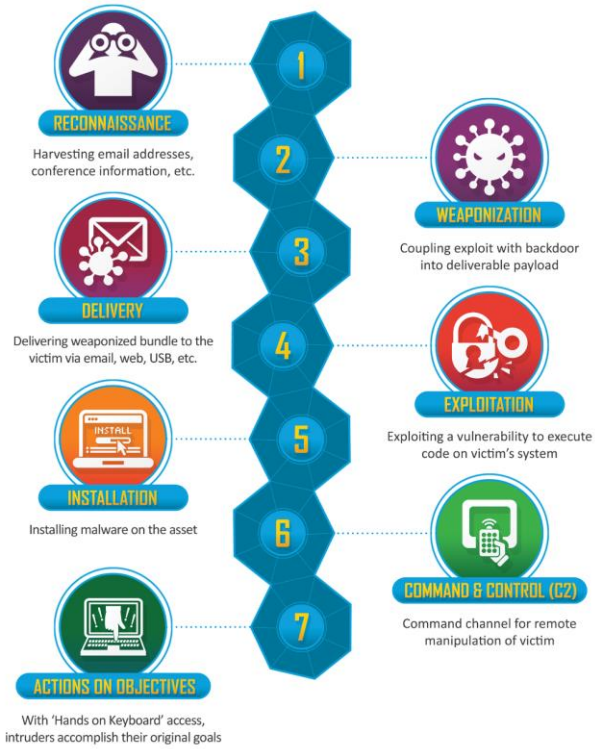

Figura 11. Cyberkill chain [24].

Cyberkill Chain contempla indicadores de tres tipologías:

- Atómicos: indicadores que no pueden descomponerse en partes más pequeñas y conservan su significado en el contexto de un ataque (IP, dominios, etc.);
- Computacionales: derivados de los datos obtenidos en un incidente (hash);
- Comportamentales: colecciones de indicadores atómicos y computacionales (TTP).

Formalmente, siguiendo la representación matemática propuesta en el Modelo del Diamante, dado un paso *S*, un evento *E* puede representarse de la siguiente manera en el modelo de la Cadena:

$$E = ((S_1), (S_2), (S_3), \dots (S_n))$$

donde *n* es el número de pasos implicados.

La gran popularidad de este marco ofrece la posibilidad de encontrar una amplia gama de informes técnicos de diferentes consultorías y CERT/CSIRT que siguen su metodología. A su vez, su estructura inspiró el desarrollo de las tácticas del modelo Mitre Att&ck. Sin

embargo, tiene una estructura muy estática e inflexible, especialmente para ataques que van más allá de las intrusiones, como los ejecutados por grupos APT, lo que ha provocado que se quede progresivamente obsoleto. Basándose en el enfoque APT, [60] implementó una serie de algoritmos tras la extracción, selección y clasificación de características en cada una de las fases de la Cyberkill Chain.

3.3 Mitre Att&ck

MITRE es una organización sin ánimo de lucro fundada en 1958 en Estados Unidos que se dedica a la investigación, el desarrollo y la innovación en tecnologías de la información. Como parte de este trabajo, el modelo ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) [15] fue desarrollado en 2015 como una herramienta de intercambio de conocimientos e información sobre ciberataques centrado en el desarrollo y la aplicación del concepto de TTP, según el cual las tácticas representan por qué un atacante realiza determinadas acciones para llevar a cabo su objetivo, las técnicas representan cómo el atacante realiza las acciones, y los procedimientos representan los pasos detallados para la implementación de las técnicas.
Este marco se utiliza cada vez más en todo el mundo por su versatilidad y, sobre todo, por su representación fiel de un ataque basada en observaciones de ataques reales modelizados con ATT&CK. Los datos implementados en la base de datos de TTPs proceden de informes de inteligencia elaborados por entidades públicas y privadas de todo el mundo y se han convertido en un repositorio de gran interés y utilidad para la ciberseguridad en todo el mundo (véase la Figura 12).

Figura 12. Interés por Mitre Att&ck según las búsquedas mundiales en Google. Fuente: Tendencias de Google.

La columna vertebral del modelo consiste en la representación de las TTP mediante matrices en las que las columnas representan cada una de las tácticas definidas. Los valores de la matriz representan las técnicas que pueden aplicarse en el análisis de un ataque.

Formalmente, considerando las tácticas como columnas y las técnicas como filas, la matriz M puede definirse como M_{ij}, donde "i" es el índice de la técnica y "j" es el índice de la táctica. Así, para un ciberataque concreto que utilice determinadas técnicas y, en consecuencia, determinadas tácticas, se puede asignar el valor 1 a las celdas correspondientes de la matriz, y 0 a las demás. Por ejemplo, si en la táctica 2 se utiliza la técnica 3, entonces $M_{32} = 1$.
Así, considerando los elementos de la matriz M, un suceso E puede representarse de la siguiente manera:

$$E = ((M_{11}), (M_{12}), (M_{13}), \dots (M_{mn}))$$

donde m es el número de tácticas implicadas, y n es el número de técnicas implicadas en cada táctica. Existen diferentes dominios técnicos en función del objetivo del análisis: Enterprise, centrado en ataques a entidades; Mobile, referido a ataques en entornos móviles; e ICS, referido a posibles incidentes en Sistemas de Control Industrial. Asimismo, cada dominio técnico es diferente según se aplique a un sistema operativo u otro. Las principales aplicaciones de este marco son la detección y análisis de ataques, simulación y *Red Team* en infraestructuras y, en particular, CTI o inteligencia de amenazas.
Por último, la presencia de una taxonomía de TTPs ha permitido con el tiempo reunir información homogeneizada sobre los parámetros de los diferentes grupos de ataque, el software implicado, las vulnerabilidades según su *Common Vulnerability Exposed* (CVE), etc. En la actualidad, son escasas las investigaciones centradas en el uso de la IA en el modelo Mitre Att&ck. El más relevante es probablemente el estudio realizado en [20], que aplicó algoritmos de clustering para determinar posibles asociaciones de TTPs. Por otro lado, y más allá de la IA, existen estudios sobre la predicción de posibles secuencias de TTPs utilizando la teoría de juegos [23] y cadenas de Markov [61].

3.4 Comparación de marcos

Cada uno de los tres marcos analizados tiene puntos de interés que pueden ser de gran valor en un análisis de inteligencia y enriquecerlo. En concreto, el Modelo Diamante ofrece una visión holística y estratégica de los ataques con una aproximación al concepto clásico de inteligencia; Cyberkill Chain se centra en los aspectos tácticos que ejecuta un atacante; y Mitre Att&ck desarrolla un amplio abanico de conceptos desde el punto de vista técnico que permiten un análisis detallado de un ataque.
Por otro lado, Cyberkill Chain y Mitre Att&ck se estructuran y organizan en diferentes fases o etapas, que pueden ayudar a comprender y visualizar las acciones del atacante. En definitiva, la combinación de detalle, adaptabilidad, adopción generalizada, utilidad práctica y versatilidad han hecho que los tres modelos analizados destaquen y sean ampliamente reconocidos en el ámbito de la ciberseguridad y la inteligencia de amenazas. Su popularidad también se ve reforzada por su continuo respaldo y uso por parte de la comunidad mundial de ciberseguridad.
Si bien es cierto que los tres modelos reflejados anteriormente son ampliamente conocidos y utilizados en la comunidad de ciberseguridad, existen otras alternativas que no serán analizadas en este capítulo por ser menos utilizadas, pero considerando de interés su mención:

- Marco de Ciberseguridad del NIST [62] es un marco que se centra en la gestión de riesgos de ciberseguridad y proporciona directrices en áreas como la identificación, protección, detección, respuesta y recuperación. Aunque su uso está muy extendido, no se centra en el análisis de la ciberinteligencia, sino en el análisis de riesgos de las entidades.

- DREAD (*Damage, Reproducibility, Exploitability, Affected users, Discoverability*) [63] es otro método desarrollado por Microsoft para evaluar y clasificar los riesgos asociados a las amenazas de seguridad en software y sistemas informáticos.

Paralelamente a Att&ck, Mitre ha desarrollado proyectos marco para diversos fines en el ámbito de la ciberseguridad no tan centrados en la inteligencia, como TARA [64] (*Threat Assessment and Remediation Analysis)* para la gestión de riesgos, CAPEC (*Common Attack Pattern Enumeration and Classification*) para el análisis de vulnerabilidades y patrones de ataque [65] y D3FEND [66] para la estrategia de respuesta a los ataques.

4 Materiales y métodos

Con el fin de observar el comportamiento de cada uno de los modelos ante un ciberataque real, se realizó una aplicación práctica que permitiera mostrar las debilidades y fortalezas de cada modelo y obtener elementos de juicio que apoyaran la toma de decisiones sobre qué marco de ciberinteligencia utilizar en la investigación. Para ello, se desarrollaron una serie de ítems para aplicar a posteriori a cada marco de forma que se pudieran construir las fases o secciones de cada modelo siguiendo la propuesta de [67].
El ciberataque analizado corresponde a un ataque de *ransomware* registrado en una entidad pública española en 2021. Esta entidad pública emplea a más de 7.000 trabajadores, repartidos por toda la geografía española, y gestiona un presupuesto elevado, dependiendo, en todo caso, de las transferencias asociadas a los presupuestos generales del Estado gestionados por otra entidad gubernamental. Un ataque tipo *ransomware* es una actividad delictiva que basa su éxito en la ejecución de amenazas a la víctima, ya sea mediante el posible borrado de la información robada o mediante su posible publicación en abierto. Este tipo de ataque se ha generalizado en los últimos tiempos debido al beneficio económico que ofrece.
Por razones de seguridad, las referencias temporales y nominales se harán bajo el seudónimo RASK; del mismo modo, los dominios y otros IOCs de la identidad de la víctima se modificarán en consecuencia para evitar su posible identificación.

> *El 9 de julio de 2021, se detectó un incidente de ciberseguridad de tipo ransomware en los sistemas de la entidad pública española (RASK), con una infección masiva por el malware "RYUK". Este ataque habría afectado a varios sistemas de información y comunicaciones de esta Administración, así como a servicios de correo electrónico y web, y a puestos de trabajo de funcionarios de la entidad. Las primeras evidencias del ataque se detectaron en la madrugada del 9 de julio de 2021, y se determinó que se trataba del conocido malware tipo ransomware de la familia RYUK. Las pruebas analizadas durante la investigación sugirieron que la intrusión en la red podría haberse llevado a cabo utilizando credenciales comprometidas que permitían el acceso a la red a través de Citrix. Así, se detectó la venta previa de dos credenciales de usuario en foros clandestinos (08s-in08 y admdp08), siendo esta última una cuenta de administrador que daba acceso a los siguientes servicios:*
>
> - *vtagex.rask[.]es (usuarios 08s-in08 y admdp08)*
> - *mytime.rask[.]es:1124 (usuario 08s-in08)*
> - *intraprod.rask[.]es (usuario PROD08)*

- *e-mail.rask[.]es (usuarios admdp08, 08s-in08 y 08dpucr)*
- *supportcx.rask[.]es (usuario admdp08)*
- *RASK Office 365 (usuario 08s-in08@rask.es)*

El vector de ataque utilizado por los atacantes consistió en acceder a la infraestructura de RASK utilizando credenciales legítimas robadas en otra operación y vendidas en foros clandestinos siguiendo la práctica habitual de este tipo de grupos, que externalizan la obtención del acceso y se centran en la propia intrusión y cifrado. El hecho de que el atacante dispusiera de credenciales de acceso "admin" indica que la víctima podría no haber sido seleccionada específica y expresamente por el atacante, lo que favorecería la tesis de un ataque puramente cibercriminal y no de ciberespionaje o búsqueda específica de información sensible. Así, la motivación de este ataque sería la obtención de un beneficio económico suficiente a través de actividades delictivas que implican amenazas a RASK, entidad que, por su volumen e importancia, podría reportar suficientes beneficios como para que el ataque mereciera la pena.

Las primeras conexiones maliciosas a la infraestructura se registraron el 1 de marzo de 2021. El filtrado de las conexiones realizadas por uno de los usuarios reveló inicios de sesión en máquinas con caracteres del alfabeto ruso. Se analizaron los registros de DNS, cortafuegos y proxy de la organización, y se detectó la presencia de varios archivos dll, cuyo análisis demostró que se trataba de balizas del software Cobalt Strike y/o SystemBC (por ejemplo, qws.dll). Cuando el atacante consiguió acceder a la red, desplegó la herramienta Cobalt Strike Beacon, que estableció conexiones con los siguientes servidores de mando y control (C2) a lo largo de marzo de 2021:

Timestamp (UTC) / IP del servidor C2 / Dominio del servidor C2:

01/07/2021 17:31 ---.26.29[.]242 culunk[.]com

01/07/2021 22:40 ---.141.84[.]190 smadst.com

01/07/2021 22:48 ---.141.84[.]190 smadst[.]com

06/07/2021 16:53 ---.26.29[.]245 eochea[.]com

06/07/2021 16:54 ---.26.29[.]245 eochea[.]com

06/07/2021 17:31 ---.141.87[.]76 dorkedit[.]com

06/07/2021 17:54 ---.26.29[.]245 eochea[.]com

06/07/2021 18:06 ---.141.87[.]76 dorkedit[.]com

06/07/2021 18:53 ---.26.29[.]245 eochea[.]com

06/07/2021 23:01 ---.26.29[.]245 eochea[.]com

08/07/2021 15:27 ---.26.29[.]245 eochea[.]com

08/07/2021 19:31 ---.26.29[.]245 eochea[.]com

08/07/2021 21:35 ---.26.29[.]245 eochea[.]com

08/07/2021 23:20 ---.26.29[.]245 eochea[.]com

08/07/2021 23:24 ---.26.29[.]245 eochea[.]com

Esta herramienta permitía al atacante realizar un reconocimiento de la red, obtener más credenciales de diferentes usuarios (incluidos administradores) tras acceder al Controlador de Dominio, e identificar nombres de ordenadores y servidores. El 8 de julio, a las 21:27, se detectó el archivo "desktop.dll", malware

de la familia BazarLoader. El 9 de julio, a las 04:29:35, se observó un archivo "82.exe" ejecutándose en varios ordenadores desde la ruta C:\Windows\Temp\, perteneciente a la familia de malware Ryuk.
Una vez que el atacante hubo completado el reconocimiento de la red, desplegó y ejecutó el malware Ryuk en tantos ordenadores como le fue posible mediante movimientos laterales. Para ello utilizó credenciales de administrador y recurrió a la utilidad PsExec, que permite la ejecución remota de comandos. Además, el malware incluye sus propias capacidades de replicación que le permitieron propagarse por la red con gran rapidez. Estas acciones tuvieron lugar el 9 de julio de 2021, día en que se detectó el compromiso y el cifrado de la información.
Un análisis de la información en fuentes abiertas muestra que las IP utilizadas en el ataque corresponden a rangos de IP asociados a paneles de mando y control del grupo UNC1878/Wizard Spider.
Las acciones de investigación se centraron en la detección del vector de ataque (acción clave para la investigación posterior), la caracterización de las herramientas utilizadas en el ataque (Cobalt Strike, SystemBC, BazarLoader, Ryuk, PsExec), y la búsqueda de IOCs (principalmente direcciones IP). Estas acciones se llevaron a cabo sin la aplicación de ninguno de los marcos de inteligencia analizados en este documento. Como se ha mencionado anteriormente, la atribución al grupo UNC1878 se realizó únicamente asociando IPs a los rangos de IP del propio grupo, sin tener en cuenta información valiosa como las TTPs, así como el contexto y los antecedentes de este ataque.

Previo al tratamiento por cada uno de los tres marcos o modelos, se procede a estructurar toda la información del incidente de forma que se disponga de una visión global del mismo.

4.1 Aplicación del Modelo Diamante

Los datos de los incidentes están estructurados según el Modelo del Diamante siguiendo los cuatro vértices. Para simplificar el análisis y evitar sesgos, se han omitido los valores de confianza.

- Víctima: La entidad pública analizada (RASK) es una víctima del sector de la Administración Pública en España, con una ubicación geográfica distribuida por todo el territorio español. Su popularidad es alta, y su posicionamiento político neutral. La capacidad económica es baja, ya que no dispone de recursos propios. La madurez de la organización es baja, como demuestran las auditorías posteriores. La organización es altamente crítica, ya que es una entidad esencial para el ejercicio de las funciones del Estado español. Los tipos de sistemas afectados son ordenadores de sobremesa y servidores, incluidos los cinco controladores de dominio. La información no es clasificada, pero es información personal sujeta a legislación. No se tiene constancia de que la organización haya expuesto CVEs. La víctima pertenece al sector de la Administración española, que presenta graves carencias en materia de ciberseguridad, como demuestran los sucesivos informes nacionales sobre el estado de la seguridad de los sistemas TIC publicados por el CSIRT nacional [68].
- Capacidad: El *malware* utilizado fue, como mínimo, Cobalt Strike, Ryuk, Bazar Loader y SystemBC. No hay evidencias de explotación de vulnerabilidades porque no era necesario, ya que el atacante disponía de credenciales de administrador. El

malware utilizado no refleja una gran sofisticación, ya que son evoluciones de malware conocido y ampliamente utilizado por los ciberdelincuentes en sus ataques. Requiere la intervención del adversario, pero no víctimas para el ataque. Aunque se desconoce el método inicial de obtención de credenciales, no hay evidencias de la aplicación de técnicas de ingeniería social. No forma parte de ninguna campaña de explotación o distribución de *malware* a través de phishing u otro tipo de ataques.

- Infraestructura: Una vez analizada la información desde el punto de vista técnico, se facilitaron las siguientes direcciones IP y dominios que conformaban los principales C2:
  ---.26.29[.]242 (Media Land LLC- San Petersburgo 09-09-2020) resoluciones culunk[.]com (registro 22/02/2021)
  --.141.84[.]190 (Media Land LLC- San Petersburgo 14-11-2019) resoluciones de smadst[.]com (registro 22/02/2021)
  ---.26.29[.]245 (Media Land LLC- San Petersburgo 09-09-2020) resoluciones eochea[.]com (registro 03/02/2021)
  --.141.87[.]76 (Media Land LLC- San Petersburgo 25-12-2020) dorkedit[.]com resoluciones (registro 03/03/2021)

  Los dominios culunk[.]com, smadst[.]com, eochea[.]com y dorkedit[.] se registraron en estrecha proximidad temporal y tienen coincidencias de registro, alojamiento y certificado SSL con dominios UNC1878 previamente identificados. [69]. Los conjuntos de dominios UNC1878 se registraron a través de *NameCheap* u *OpenProvider*, utilizaron sus propios servidores de nombres, están alojados en servidores dedicados en los sistemas de Russian Federation Media Land LLC y utilizan varias cadenas de certificados SSL.
- Adversario: La probable atribución mediante la identificación de las TTP, así como mediante la coincidencia de rangos de IP y dominios con los servidores de Cobalt Strike, puede asociarse presumiblemente con UNC1878, un grupo cibercriminal de Europa del Este con conexiones con Ucrania y Rusia, cuya finalidad es económica. Los grupos UNC pueden evolucionar, fusionándose con el tiempo con otros grupos y derivando potencialmente en actores con nombres de amenazas reconocidos, como "Amenazas Persistentes Avanzadas" (APT) o "Grupos de Hacking con Motivación Financiera" (FIN). [70]. Algunas investigaciones de expertos han atribuido el nombre de "Grupo Uno" al actor UNC1878, afirmando que sus objetivos son "indiscriminados" y sus infecciones "oportunistas" y asignándole las siguientes características [71]:
  1. Vector de infección: correos electrónicos tipo *phishing*, que suelen contener enlaces.
  2. Velocidad de ejecución: el tiempo entre la infección inicial y el cifrado se ha reducido recientemente de unos 2-5 días a entre 3 y 6 horas.
  3. Uso coherente de muestras de Cobalt Strike autofirmadas.
  4. Uso de herramientas legítimas a lo largo de la cadena de infección posterior al compromiso: Cobalt Strike, Empire, Meterpreter, Mimikatz, Kerbrute, Kerberoast, BloodHound, AdFind.
  5. Ausencia de exfiltración o publicación de información sobre sus víctimas.

  Según varios proveedores de ciberseguridad, una quinta parte de las intrusiones relacionadas con *ransomware* en 2020 se debieron a Ryuk, el 83% de las cuales

se atribuyeron al grupo UNC1878. En cuanto a los orígenes de este grupo, actualmente se desconocen. A pesar de dirigir sus ataques a objetivos sin seguir ningún patrón detectado, algunos investigadores habrían señalado que el grupo UNC1878 está seleccionando en la franja temporal objeto de estudio objetivos vinculados preferentemente a la prestación de servicios sanitarios en Estados Unidos (EEUU) [72]. Aunque el grupo UNC1878 ha sido descrito hasta ahora como una única entidad no categorizada, otros especialistas en ciberseguridad habrían señalado que detrás de este actor de amenaza se encuentra el mismo grupo que el alojado bajo el apelativo de Wizard Spider, también conocido como Gold Blackbourn [73]. La información pública sobre Wizard Spider afirma que se trata de un actor de amenazas al que se atribuyen orígenes rusos y el desarrollo del troyano bancario Trickbot. Los principales objetivos de este actor son organizaciones de los ámbitos de la defensa, las finanzas, la administración pública, la sanidad y las telecomunicaciones a escala mundial. [74].

Se adjunta la siguiente tabla para facilitar la estructura de los datos en cada uno de los vértices del modelo.

Tabla 4. Aplicación del modelo Diamante al caso analizado.

VÉRTICE	**INDICADORES**
1. VÍCTIMA	Víctima física: **Sector Público **Distribución geográfica a lo largo de toda España **Popularidad alta **Posicionamiento político neutral **Capacidad económica baja **Madurez baja **Organización esencial para el Estado **Información no clasificada Víctima Activos: *workstations*, servidores y controladores de dominio
2. CAPACIDAD	Uso de C2 Uso de *malware*: **Cobalt Strike **Ryuk **Bazar Loader **SystemBC Uso de Credenciales legítimas para acceso y movimiento lateral [...]
3. INFRAESTRUCTURA	C2: **Dirección IP-Dominio ---.26.29.242-culunk[.]com **Dirección IP-Dominio ---.26.29[.]245-eochea[.]com **Dirección IP-Dominio --.141.84[.]190-smadst[.]com **Dirección IP-Dominio --.141.210[.]78-choopa[.]com **Dirección IP-Dominio --.141.87[.]60-vultr[.]com **Dirección IP-Dominio --.141.87[.]76-dorkedit[.]com [...]
4. ADVERSARIO	Grupo de Europa del Este Finalidad económica Sin patrón de objetivo definido claramente

4.2 Aplicación del modelo de cyberkill chain

Los datos asociados al incidente se estructuran según los siete pasos del modelo de *Lockheed Martin* (ver Tabla 5), de forma que cada paso incluye los indicadores de compromiso asociados y cualquier otra información que pueda ser de interés para analizar este ataque.

Tabla 5. Aplicación del modelo Cyberkill Chain al caso analizado.

FASE DE ATAQUE	INDICADORES
1. RECONOCIMIENTO	Reconocimiento de objetivos y acceso Compra de credenciales de acceso en foros clandestinos: **vtagex.rask[.].co.uk (usuarios 08s-in08 y admdp08) **myhours.rask[.].co.uk:1124 (usuario 08s-in08) **intraprod.rask.co.uk (usuario PROD08) **e-mail.rask[.]es (usuarios admdp08, 08s-in08 y 08dpucr) **supportcx.rask[.]es (usuario admdp08) **Office 365 por RASK (usuario 08s-in08@rask[.]es)
2. ARMAMENTIZACIÓN	DLL_Bazar, archivo de carga: "desktop.dll" DLL_CobaltStrike, fichero "qws.dll" (08/07/2021 21:31) DLL_CobaltStrike, fichero "test.dll" (06/07/2021 17:31) Ryuk, archivo "82.exe" Ryuk archivo "6hr.exe" Ryuk, archivo "6hr.exe" (06/07/2021 17:31) Cobalt Strike: Beacon Cobalt Strike 87fb204.exe Beacon Cobalt Stike f59173f.exe [...]
3. DESPLIEGUE	Acceso con credenciales de administrador “admdp08” Movimientos laterales con PsExec (desde 01/07/2021 12:05) Despliegue de balizas Cobalt Strike (desde 06/07/2021 17:21) Acceso a listados de Domain Controller (DC) (06/07/2021 18:14) [...]
4. EXPLOTACIÓN	Muestra BazarLoader "desktop.dll" en DC y ejecución del *malware* SystemBC y Cobalt Strike (08/07/2021 21:27 21:31)
5. INSTALACIÓN	Malware Ryuk "82.exe" (09/07/2021 19:24) y "6hr.exe" (09/07/2021 4:17) en C:\Windows.
6. C2	Cobalt Strike y SystemBC conexiones a los siguientes C2: Dirección IP-Dominio ---.26.29.242-culunk[.]com Dirección IP-Dominio ---.26.29[.]245-eochea[.]com Dirección IP-Dominio --.141.84[.]190-smadst[.]com Dirección IP-Dominio --.141.210[.]78-choopa[.]com Dirección IP-Dominio --.141.87[.]60-vultr[.]com Dirección IP-Dominio --.141.87[.]76-dorkedit[.]com [...].
7. ACCIONES A OBJETIVOS	Cifrado masivo de información en ordenadores con el algoritmo simétrico AES256, y posterior cifrado de la clave AES con el algoritmo asimétrico RSA.

4.3 Aplicación del modelo Mitre Att&ck

Los datos asociados al incidente se estructuran según la versión de la matriz v10.1 en el navegador web e incluyen las técnicas indicadas con sus correspondientes identificadores numéricos.

Tabla 6. Aplicación del modelo Mitre Att&ck al caso analizado.

TÁCTICAS	TÉCNICAS
RECONOCIMIENTO (TA0043)	T1589: Recopilación de información sobre la identidad de la víctima; T1591: Recopilación de información sobre la organización de la víctima.
DESARROLLO DE RECURSOS (TA0042)	T1584: Compromiso de Infraestructura; T1588: Obtención de Capacidades.
ACCESO INICIAL (TA0001)	T1078: Cuentas válidas
EJECUCIÓN (TA0002)	-
PERSISTENCIA (TA0003)	T1078: Cuentas válidas
ESCALADA DE PRIVILEGIOS (TA0004)	T1078: Cuentas válidas
EVASIÓN DE DEFENSAS (TA0005)	T1078: Cuentas válidas
CREDENCIALES DE ACCESO (TA0006)	-
DESCUBRIMIENTO (TA0007)	T1069: Descubrimiento de Grupos de Permisos
MOVIMIENTO LATERAL (TA0008)	T1210: Explotación de Servicios Remotos; T1021: Servicios Remotos
COLECCIÓN (TA0009)	T1005: Datos del sistema local
MANDO Y CONTROL (TA0011)	T1071: Protocolo de la capa de aplicación
EXFILTRACIÓN (TA0010)	T1041: Exfiltración a través del Canal C2
IMPACTO (TA0040)	T1486: Datos cifrados por impacto

Como se refleja en la tabla anterior, se puede observar que la posesión de credenciales de administrador (admdp08) redujo la necesidad de desplegar técnicas en varias tácticas, allanando el camino a los atacantes, especialmente en escalada de privilegios y adquisición de persistencia. Aunque los indicadores de compromiso no se muestran explícitamente, se obtiene una matriz estructurada con la secuencia utilizada en el ataque que representa los TTP, indicadores mucho más fiables para el análisis de un ataque dada la volatilidad de los IOCs. La representación de la secuencia a través de una matriz permite realizar comparaciones con otros ataques que, a su vez, pueden plasmarse en una matriz Mitre. En este sentido, este modelo permite la comparación del ataque RASK con otros ataques y un tratamiento matemático para ayudar en esta tarea.

5 Resultados

Siguiendo la metodología propuesta en [75], se han implementado diecisiete variables de interés para evaluar las posibles aplicaciones de la IA a cada marco. Por ello, el enfoque aborda no sólo perspectivas matemáticas, como la facilidad de parametrización o la creación de variables en la información presentada o incluso la existencia de conjuntos de datos, sino también consideraciones relevantes para analizar la potencia o utilidad del marco y la capacidad de adaptación (entre otros aspectos). De este modo, se garantiza la

realización de un análisis exhaustivo. Así pues, para estudiar cada uno de los marcos presentados y su idoneidad para la investigación en IA basada en datos a gran escala, se evalúa cada uno de ellos con las siguientes variables de interés:

- Madurez: refleja que el desarrollo del marco ha alcanzado un grado suficientemente elevado.
- Flexibilidad: se refiere a la capacidad de adaptarse a otros entornos fuera del marco.
- Popularidad: relativa al alcance y uso en la comunidad mundial de ciberseguridad.
- Taxonomía propia: existencia de una categorización propia de sus características.
- Conjuntos de datos: existencia de fuentes abiertas de repositorios de datos referidos a ese marco.
- Software propietario: desarrollo de herramientas nativas que permitan su aplicación.
- Adaptación a diferentes ataques: indica si el marco puede utilizarse para diferentes tipos de ataques (*ransomware*, ataques APT, DDoS, etc.).
- Actualización: si hay una actualización constante por parte de la comunidad o del desarrollador.
- Facilidad de uso: usabilidad del marco por usuarios inexpertos.
- Parametrización: indica la creación de variables o argumentos para definir los puntos de interés más relevantes de un ataque.
- Granularidad: indica si el nivel de detalle del modelo es alto.
- Visualización: indica si la información puede visualizarse de forma gráfica.
- Fácil integración con otros sistemas: refleja la posible integración con sistemas de identificación o protección (por ejemplo, IDS/IPS o antivirus).
- Orientación: indica en qué parámetro se centra principalmente la atención del modelo: los atacantes, los activos de la organización o el software utilizado en el ataque.
- Escalabilidad: indica si un marco tiene la capacidad de seguir funcionando correctamente cuando la cantidad de datos cambia de tamaño o volumen.
- Interoperabilidad: capacidad de un marco para compartir datos y facilitar el intercambio de información y conocimientos con otras herramientas.
- Rendimiento: indica si un marco es eficiente a la hora de trabajar con datos a gran escala.

La Tabla 7 indica si cada uno de los marcos cumple los requisitos descritos según su ejecución en el ejemplo de ciberataque tratado en este capítulo. El análisis de estos criterios se considera relevante para obtener resultados satisfactorios en cualquier investigación basada en IA. Ciertamente, algunas de las características mencionadas juegan un papel importante en este sentido, como la existencia de conjuntos de datos validados, el nivel de granularidad, la escalabilidad, el rendimiento o la flexibilidad. Podría parecer que otras, por ejemplo, la facilidad de uso, la popularidad o la madurez, deberían pasar a un segundo plano en este contexto, pero completan todas las implicaciones que supone implementar y construir un modelo de IA. Por lo tanto, puede afirmarse que el cumplimiento de todas las variables mencionadas es importante para

categorizar un marco como apto y adecuado para fusionarlo con algoritmos de IA para el análisis de datos a gran escala.

Tabla 7. Análisis de las variables propuestas en cada uno de los modelos.

	MODELO DIAMANTE	**CYBERKILL CHAIN**	**MITRE ATT&CK**
MADUREZ	SÍ	SÍ	SÍ
FLEXIBILIDAD	NO	NO	SÍ
POPULARIDAD	NO	SÍ	SÍ
TAXONOMÍA PROPIA	NO	NO	SÍ
DATOS	NO	NO	SÍ
SOFTWARE PROPIETARIO	NO	NO	SÍ
ADAPTACIÓN A DIFERENTES ATAQUES	SÍ	NO	SÍ
ACTUALIZACIÓN	NO	NO	SÍ
FACILIDAD DE USO	SÍ	NO	NO
PARAMETRIZACIÓN	NO	NO	SÍ
GRANULARIDAD	SÍ	NO	SÍ
VISUALIZACIÓN	SÍ	SÍ	SÍ
FÁCIL INTEGRACIÓN CON OTROS SISTEMAS	NO	SÍ	SÍ
ORIENTACIÓN	ATACANTES ACTIVOS	ATACANTES	SOFTWARE PARA ATACANTES ACTIVOS
ESCALABILIDAD	NO	NO	SÍ
INTEROPERABILIDAD	SÍ	SÍ	SÍ
RENDIMIENTO	NO	NO	SÍ

Según el análisis mostrado en la Tabla 7, se puede concluir que el marco Mitre Att&ck es el más adecuado para procesar datos a gran escala con fines de ciberinteligencia basándose en los siguientes puntos de interés detallados a continuación.

Si bien es cierto que el Modelo Diamante presenta buenos resultados en cuanto a madurez, facilidad de uso, visualización -con su característico diamante que ayuda a comprender la información de forma sencilla- e incluso granularidad, con la presencia de rasgos detallados y meta-características, este modelo presenta carencias significativas en variables especialmente relevantes para la aplicación de la IA, como son la falta de capacidad de parametrización, la baja escalabilidad, la reducida flexibilidad y la ausencia de conjuntos de datos.

Cyberkill Chain destaca por su madurez, popularidad -es ampliamente conocida en la comunidad- y capacidad para integrarse fácilmente con otros sistemas, pero, al igual que el Modelo Diamante, no posee las características necesarias para una implementación óptima de la IA, como la flexibilidad, la parametrización, la escalabilidad y la existencia de conjuntos de datos.

Mitre Att&ck es flexible, tiene su propia taxonomía, está referenciado en conjuntos de datos, dispone de software propio, se actualiza periódicamente y permite la parametrización y la granularidad. Estos factores son esenciales para la implantación de

la IA, ya que la existencia de datos estructurados y la capacidad de adaptar y personalizar el modelo y disponer de un alto nivel de detalle pueden facilitar el entrenamiento de los algoritmos de aprendizaje automático. Además, su capacidad para trabajar con éxito con conjuntos de datos de volumen variable y para compartir sus conocimientos con otras herramientas también son apropiadas para su combinación con algoritmos de IA. Además, otras variables que pueden ser de interés para un análisis global de la potencia del marco, como la popularidad, la madurez y la orientación, muestran que supera al resto de modelos por un amplio margen.
Esta idea se ve reforzada por el potente conjunto de datos disponible en el sitio web de Mitre Att&ck, que permite un tratamiento exhaustivo de los datos para las técnicas de aprendizaje automático, y por la amplia comunidad que desarrolla nuevas implementaciones con regularidad. El uso de los otros dos modelos analizados requeriría la creación de un conjunto de datos *ad hoc* sin una taxonomía homogénea como punto de partida, lo que dificultaría enormemente el tratamiento de los datos para alcanzar los objetivos de esta investigación.

Una vez planteada la utilización del marco Mitre Att&ck, en la Figura 13 se presenta el tratamiento que se le puede dar para alcanzar los objetivos de aplicación de las técnicas de IA. Se puede observar que el tratamiento de los datos recogidos de diversas fuentes (incluido el repositorio oficial de Mitre Att&ck) permitirá enriquecer cada una de las fases de las ramas de ciberinteligencia (CTI, IR y VM) mediante el desarrollo de modelos de predicción de patrones de comportamiento. Toda la información obtenida deberá pasar por el tamiz del *framework* elegido, es decir, Mitre Att&ck, por lo que toda la información deberá convertirse en matrices o, en su caso, vectores, a los que se aplicarán las técnicas de *Machine Learning* adecuadas para obtener secuencias o autorías. Aunque esta aplicación permitirá profundizar en aspectos puramente técnicos, hay que tener en cuenta que el Modelo Diamante y la Cyberkill Chain ofrecen una mejor comprensión de la esfera sociopolítica y su papel en el contexto de un ataque, lo que puede ayudar y, en muchos casos, ser decisivo a la hora de afrontar un ciberataque.

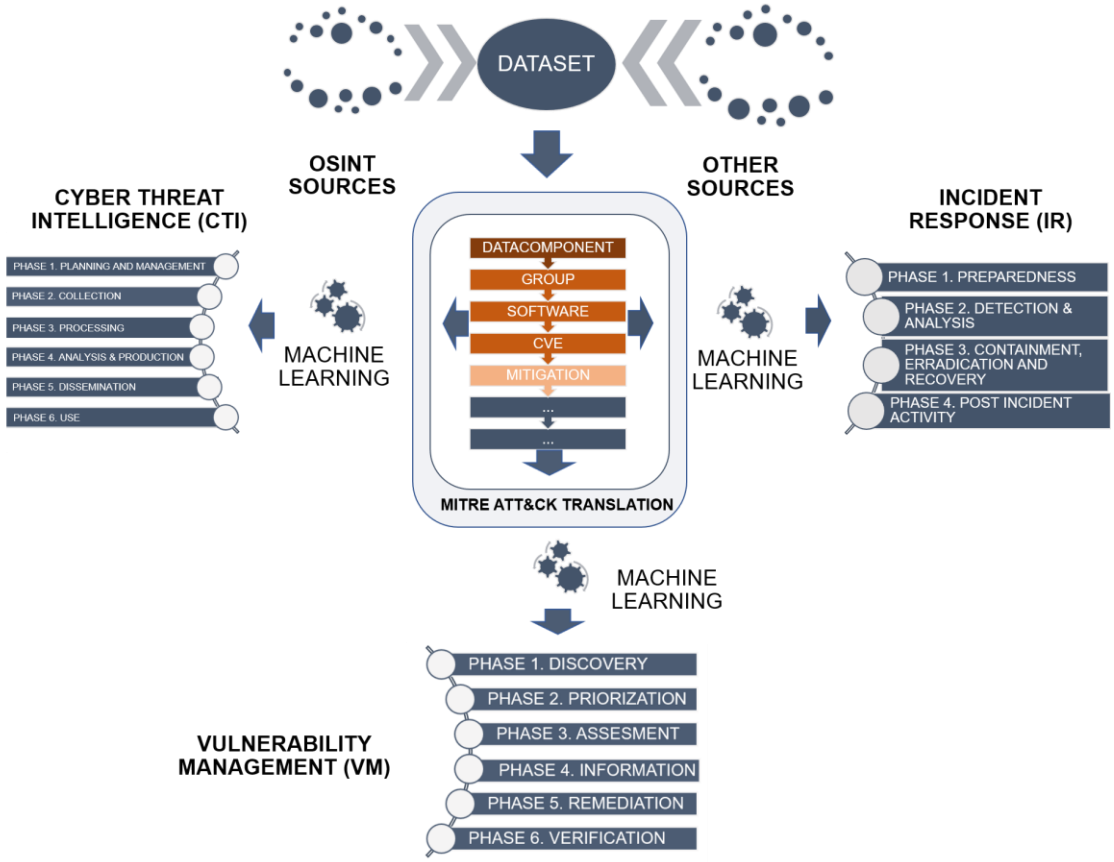

Figura 13. Aplicación del marco Mitre Att&ck a la investigación de aprendizaje automático en ciberinteligencia. Fuente: Elaboración propia.

Los trabajos alternativos en la literatura actual sobre IA y ciberseguridad y ciberinteligencia suelen centrarse en la elaboración de protocolos de autenticación de usuarios [46-48]; el conocimiento de la situación de la red [49,50]; la supervisión de comportamientos peligrosos [51,52]; e identificación de tráfico anómalo [53,54]. En todos los casos, el objetivo principal de los esquemas propuestos es predecir o identificar una situación anómala. En este capítulo, se analizan los tres marcos de ciberinteligencia más relevantes para evaluar su potencial de fusión con tecnologías de IA no sólo con este propósito, sino también con dos objetivos adicionales: la identificación precisa de futuras vulnerabilidades en el sistema y la creación de un protocolo que pueda clasificar rápidamente un ciberincidente, proponiendo automáticamente acciones de respuesta para mitigar su efecto.

6 Conclusiones

Hoy en día, es incuestionable que la ciberinteligencia es un área esencial de la ciberseguridad. Las dificultades para ofrecer respuestas adecuadas a las actividades maliciosas han puesto en valor las medidas preventivas, cobrando especial importancia los marcos de ciberinteligencia. En este capítulo se analizan los marcos de inteligencia que pueden ser útiles para la aplicación de algoritmos de aprendizaje automático. Para ello, en primer lugar, se presenta una visión general del concepto de ciberinteligencia y

todas sus variantes y posibles aplicaciones en ciberseguridad. A continuación, se detallan tres marcos de ciberinteligencia: Diamond Model, Cyberkill Chain y Mitre Att&ck; se aportan sus fortalezas y debilidades desde una perspectiva que tiene en cuenta la aplicación no sólo desde un punto de vista matemático, sino también desde una perspectiva holística que garantice que el marco utilizado es el más adecuado. Este estudio y análisis pone de relieve la aplicación práctica de los modelos a un caso real de ataque de *ransomware*. Aunque los tres marcos ofrecen diferentes ventajas, se concluye que el marco Mitre Att&ck es el más adecuado para combinar con técnicas de IA debido a su potencia, su idoneidad para el procesamiento de datos y la existencia de conjuntos de datos disponibles.

Capítulo 4. Machine Learning

1 Introducción

La Inteligencia Artificial se ha convertido en una herramienta de utilidad en cualquier campo de la vida real. En este sentido, el *Machine Learning* (ML) o aprendizaje automático es una rama de la inteligencia artificial que busca generar mecanismos de aprendizaje en las máquinas. En la actualidad, la ciberseguridad hace uso de esta rama en dos principales vertientes; la detección de anomalías y la detección de patrones, que principalmente se traducen en la práctica en la detección de intrusiones, en el análisis de *malware* y en la detección de correos electrónicos tipo spam y ataques *phishing*. No obstante, el espectro potencial de actuación de la Inteligencia Artificial en el campo de la ciberseguridad es amplio y aún por explorar en muchos casos. La aplicación de la IA en la ciberseguridad no es una opción si se pretende hacer frente adecuadamente al fenómeno delincuencial actual, que aprovecha a su vez los potenciales de esta en su beneficio.

El problema del reconocimiento de patrones en conjuntos de datos ha estado presente a lo largo de la historia. A modo de ejemplo, las extensas observaciones astronómicas de Tycho Brahe en el siglo XVI permitieron a posteriori a Johannes Kepler descubrir las leyes empíricas del movimiento planetario. Del mismo modo, el descubrimiento de regularidades en los espectros atómicos de la física cuántica a principios del siglo XX se nutrió de la observación de grandes volúmenes de datos [48].

Grosso modo, el término inteligencia artificial hace referencia a cualquier técnica que permita a las máquinas imitar, reproducir o en su caso superar, el comportamiento humano ante la toma de decisiones de forma independiente o con una mínima intervención humana [80].

De forma concreta, el *Machine Learning* es un campo de la inteligencia artificial que ha cobrado gran importancia desde la década de 1990 debido a su aplicación práctica en la vida cotidiana. El ML se ha focalizado en investigar, entender y construir métodos que permitan aprovechar la información contenida en un conjunto de datos para la mejora de una serie de tareas. Por ello, los algoritmos de *Machine Learning* buscan la construcción de un modelo basado en un dataset o conjunto de datos iniciales que permita la toma de decisiones ante problemas complejos, como puede ser la predicción de un cierto parámetro, las dependencias o correlaciones entre los datos presentes.

Así pues, el *Machine Learning* hace referencia a la mejora del rendimiento de un programa informático, sobre la base de la experiencia, con respecto a determinadas tareas y medidas de rendimiento [81]. Esto se consigue aplicando algoritmos que aprenden de forma iterativa a partir de datos de entrenamiento específicos del problema, especialmente en tareas relacionadas con datos de alta dimensión, como la clasificación, la regresión y la agrupación, el ML muestra una buena aplicabilidad. Al aprender de los cálculos anteriores y de bases de datos masivas, puede ayudar a producir decisiones fiables y repetibles.

Por esta razón, los algoritmos de ML se han aplicado con éxito en muchos ámbitos, como la como la detección de fraudes, la puntuación de créditos, el análisis de la mejor oferta [82].

En términos generales, las entradas del proceso de modelado son las instancias que contienen un conjunto de m parámetros, $X = \{ X_1, ..., X_m)\}$ donde cada parámetro Xi puede tomar un valor de su propio conjunto de valores posibles χi, y n vectores de características o instancias, $x_i = (x_1, ..., x_m) \in \chi = (\chi_1, ..., \chi_m)$. Más concretamente, el modelo a aprender procesará esta información para producir una representación de conocimiento del conjunto de datos.

2 Esquema de un modelo matemático

La modelización a través de las matemáticas es una de las herramientas más potentes que dispone el ser humano para explicar y predecir el comportamiento de un determinado sistema complejo. El avance de la computación en los últimos años ha motivado que la modelización y posterior simulación de fenómenos empleando recursos software sea la alternativa más viable y económica para ello. No obstante, el proceso empleado para el desarrollo del modelo matemático (teórico) no varía en lo sustancial pese a la preeminencia de la computación, de modo que el eje principal alrededor del que se vertebra el modelo es el planteamiento y la comprobación de una serie de hipótesis. Así pues, el desarrollo de un modelo matemático puede estructurarse en torno a un proceso cíclico (véase Figura 14), que se compone de las siguientes fases:

1. Descripción del fenómeno objeto de estudio.
2. Elección de las variables asociadas al sistema.
3. Estudio de las relaciones entre variables.
4. Elaboración de un *dataset* adecuado y que represente el fenómeno.
5. Construcción del modelo expresando la idealización del mismo en términos matemáticos.
6. Aplicación práctica de la modelización al sistema complejo mediante la computación del mismo.
7. Validación y comprobación de la utilidad del modelo. En caso de detectar carencias se procede a iniciar de nuevo el proceso mediante la ejecución de la fase inicial.

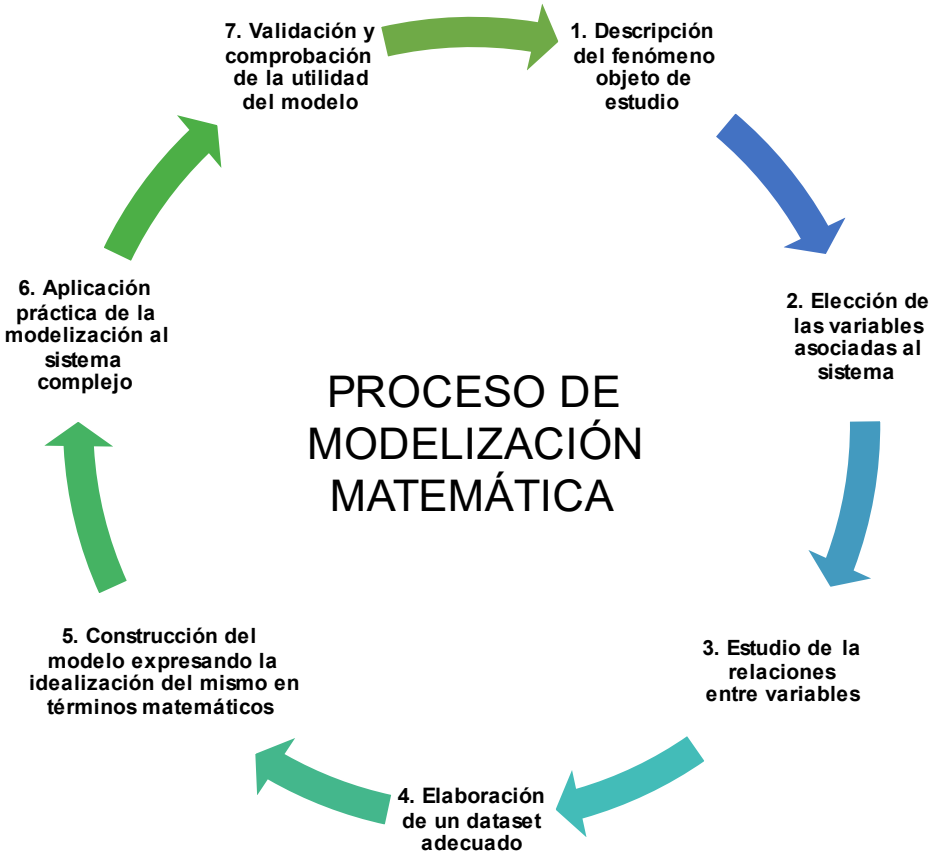

Figura 14. Proceso de modelización matemática.

3 Tipos de aprendizaje

Tradicionalmente, el *Machine Learning* se ha dividido en tres grandes grupos, a saber; Aprendizaje supervisado, Aprendizaje no supervisado y Aprendizaje por refuerzo. Se detallan a continuación las características de cada una de estas ramas.

3.1 Aprendizaje supervisado

En el proceso de modelado, los datos iniciales empleados en la fase de entrenamiento se encuentran asociados a determinadas etiquetas. El objetivo es aprender una regla que permita transformar inputs en outputs sobre la base de esas etiquetas, pudiendo ser estos últimos continuos o discretos. Ejemplos de esta tipología son la clasificación y la regresión, siendo de gran utilidad en muy útil en problemas de investigación biológica, biología computacional y bioinformática [83], [84].

3.2 Aprendizaje no supervisado

En esta rama del *Machine Learning* los datos iniciales empleados en la fase de entrenamiento carecen de etiquetado y se no se tiene conocimiento previo. El modelo se focaliza en el objetivo de analizar y deducir peculiaridades o rasgos comunes de las instancias presentadas para descubrir similitudes o asociaciones entre las muestras; es

decir, reconocer patrones para poder etiquetar nuevas entradas. Ejemplos de esta tipología son los modelos de *clustering* [85].

3.3 Reinforcement learning

Se trata de un tipo de aprendizaje basado en ensayo-error que busca maximizar la noción de recompensa acumulada. El algoritmo aprende observando el mundo que le rodea tras recibir *feedback* de las acciones que realiza [86].

4 Algoritmos empleados en la investigación

Los algoritmos que se emplearán en la investigación son tanto lineales (Support Vector Machines y Regresión Logística), algoritmos no lineales (Árboles de decisión) y algoritmos de ensamblado, tanto *boosting* (Adaboost) como *bagging* (Bosques aleatorios), de modo que se va a tener una amplia variedad de tipologías para contrastar su rendimiento.

4.1 Algoritmo Support Vector Machines (SVM)

Los *Support Vector Machines* (SVM) o máquinas de soporte de vectores son modelos que permiten analizar datos para la clasificación y la regresión de valores. Fueron desarrollados en 1964 por Vladimir Vapnik y su equipo en los laboratorios de AT&T, y se han convertido en la actualidad en uno de los métodos más empleados por la madurez y la robustez que ofrecen sus resultados para la predicción [48] [87]. En especial, se han empleado en bioinformática y reconocimiento de textos e imágenes.

Los SVM buscan encontrar un hiperplano en el caso de problemas binarios, o varios hiperplanos en problemas multiclase que permitan, entre clases de datos etiquetados en un conjunto de entrenamiento, separar de forma óptima los puntos con el mínimo error, primando la reducción del error sobre la clasificación [88].

Para ello, los puntos más cercanos al hiperplano se conocen como *support vectors*, de modo que este algoritmo tratará de determinar la máxima separación entre esos puntos. Se asume que cuanto mayor sea la distancia perpendicular entre los dos puntos opuestos más cercanos relacionados con su etiqueta, mejor será el proceso de clasificación. Los nuevos valores se asignan a ese mismo espacio y se predice su pertenencia a una categoría en función del lado en el que se encuentren.

Formalmente se tiene que, definida la función:

$$y(x) = w^T \phi(x) + b$$

donde $\phi(x)$ denota una transformación fija en el espacio definido, y se explicita el parámetro de sesgo o bias b (véase Figura 15). El conjunto de datos de entrenamiento comprende N vectores de entrada $x_1,\ldots, x_N$, con correspondientes valores objetivo $t_1,\ldots, t_N$ donde $t_n \in \{-1, +1\}$, y los nuevos puntos de datos x se clasifican según el signo

de $y(x)$. En consecuencia, la distancia de un punto x_n a la superficie definida viene dada por:

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T\phi(x) + b)}{\|w\|}$$

El margen vendrá dado por la distancia perpendicular al punto más cercano x_n del dataset, y se buscará optimizar el parámetro *w* y *b* de modo que pueda maximizarse esa distancia. Por ello, el margen máximo vendrá definido por:

$$\arg\max_{w,b}\left\{\frac{1}{\|w\|}\min_n[t_n(w^T\phi(x) + b)]\right\}$$

De modo que al no existir dependencia de n y w puede sacarse fuera de la ecuación $\|w\|^{-1}$. Este complejo problema de optimización puede simplificarse mediante la búsqueda de la maximización de $\|w\|^{-1}$, lo que equivale a minimizar $\|w\|^2$, concluyendo en:

$$\arg\min_{w,b}\left\{\frac{1}{2}\|w\|^2\right\}$$

Parece que el parámetro de sesgo b ha desaparecido de la optimización. Sin embargo, se determina implícitamente a través de las restricciones, ya que éstas requieren que los cambios en w se compensen con cambios en b.

Hasta ahora se ha asumido que los puntos del *dataset* de entrenamiento son linealmente separables en el espacio. Por ello, los SVM ofrecerán una separación exacta de datos. No obstante, en la práctica esta circunstancia no suele producirse, de modo que deberá permitirse cierto grado de clasificación errónea.

Para ello se introducirán las variables de holgura para cada punto $\xi_n \geq 0$, donde n=1, 2, 3, ..., N, de modo que $\xi_n \geq 0$ para los datos dentro de la frontera de delimitación, y $\xi_n = |t_n - y(x_n)|$ para otros puntos. Por ello, los datos que se encuentren en la frontera de decisión tendrán un valor $\xi_n = 1$ (véase Figura 15 y Figura 16).

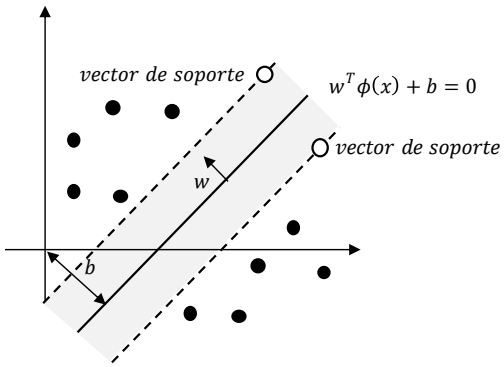

Figura 15. Support Vector Machines (I)

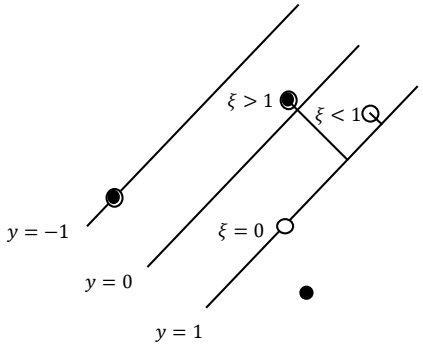

Figura 16. Support Vector Machines (II)

El objetivo por lo tanto será maximizar el margen mientras se penalizan los puntos que se encuentren en el lado incorrecto de la frontera de clasificación, de modo que se buscará minimizar la siguiente expresión, donde el parámetro C controla la compensación entre la penalización y la variable de holgura.

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|w\|^2$$

En caso de que los datos no permitan una separación lineal, se observa la circunstancia de que determinados elementos no serán clasificados en el lado correcto. Estos puntos se conocerán como el coste (C), de modo que el modelo tratará de reducir al máximo ese valor. Esta problemática de la linealidad se resuelve mediante el uso de *kernel*, que transforma los datos a una dimensión superior donde sí podrá realizarse la separación (véase Figura 17). Los *kernel* más empleados son funciones polinómicas, funciones gaussianas y funciones sigmoides [89].

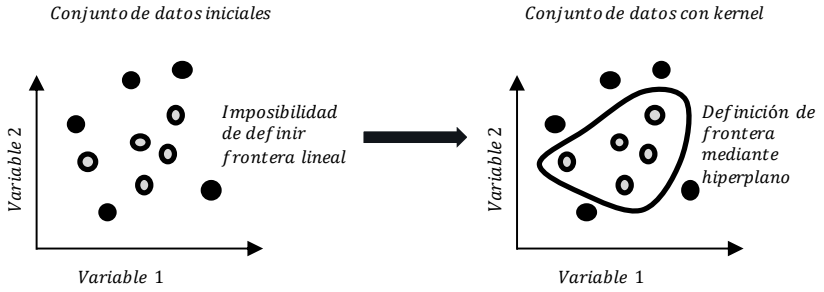

Figura 17. Aplicación de kernel en SVM

Ahora bien, el tipo de función *kernel* para un determinado problema no se aprende de los datos, sino que habrá que especificarlo. Por ello, la elección de la función *kernel* es un hiperparámetro categórico (un hiperparámetro que toma valores discretos, no continuos). Por lo tanto, el mejor enfoque para elegir el kernel de mejor rendimiento es el ajuste de hiperparámetros o *hyperparameter tunning*.

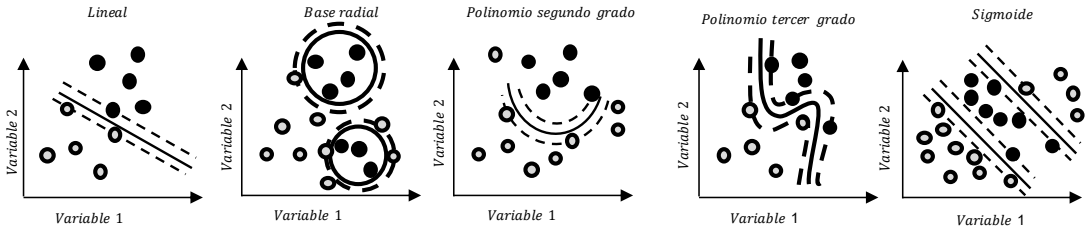

Figura 18. Hiperparámetros de núcleo y grado SVM.

La capacidad del algoritmo para definir correctamente los límites estará delimitada por la selección del kernel y los parámetros que se establezcan. A menudo se adopta una estrategia de *grid search* que pretende examinar exhaustivamente todas las combinaciones posibles de los parámetros que se comprobarán mediante una metodología de *cross validation* [90].

El algoritmo SVM tiene bastantes hiperparámetros que ajustar, siendo crucial esta acción [91]. Los hiperparámetros más importantes a tener en cuenta son los siguientes:

- El hiperparámetro del núcleo (véase Figura 17)
- El hiperparámetro de grado, que controla la flexibilidad del límite de decisión para el núcleo polinómico dentro de la elección polinómica (véase Figura 18)
- El hiperparámetro de coste o C, que controla lo "duro" o "blando" que es el margen. (véase Figura 19)

  Este hiperparámetro asigna un coste o penalización a tener casos dentro del margen o, dicho de otro modo, indica al algoritmo lo malo que es tener casos dentro del margen. Un coste bajo indica al algoritmo que es aceptable tener más casos dentro del margen y dará lugar a márgenes más amplios menos influenciados

por las diferencias locales cerca del límite de la clase. Un coste alto impone una penalización más dura por tener casos dentro del margen y dará como resultado márgenes más estrechos e influenciados por las diferencias locales cerca del límite de la clase.

- El hiperparámetro gamma, que controla la influencia de los casos individuales en la posición del límite de decisión (véase Figura 19)
  Este hiperparámetro controla la influencia que tiene cada caso en la posición del hiperplano, y es utilizado por todas las funciones del *kernel* excepto el *kernel* lineal. Cuanto más grande sea gamma, más atención tendrá cada caso y más granular será el límite de decisión (lo que puede llevar a un exceso de ajuste). Cuanto más pequeña sea la gamma, menos atención se prestará a cada caso y menos granular será el límite de decisión (lo que puede llevar a un ajuste insuficiente).

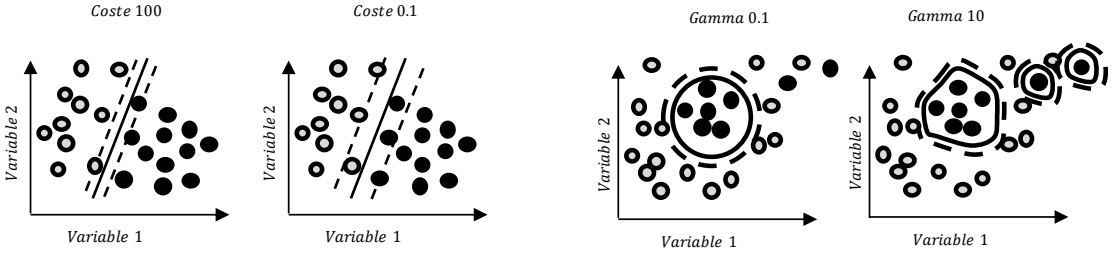

Figura 19. Hiperparámetros de Coste (izquierda) y Gamma (derecha) para SVM.

Hasta ahora se ha tenido en cuenta los SVM para la clasificación en dos clases (clasificación binaria), pero, tal y como se pretende aplicar en esta investigación, existe la posibilidad de tener que clasificar en más de dos clases. Por ello se empleará múltiples modelos simultáneos que permitan la clasificación de los datos, que funcionarán:

- Uno contra todos (OVA): se crean tantos modelos como clases existan. Cada modelo separa los datos de esa clase respecto del resto de la mejor forma posible. La clase se determina mediante la validación del modelo que separe correctamente (véase Figura 20).
- Uno contra uno (OVO): se crea un modelo por cada par de clases que existan, de modo que se construyen $k(k-1)/2$ modelos, siendo k el número de clases. Cada modelo separa los datos de esas dos clases obviando la existencia de las otras. La clase se determina mediante la elección de aquella que tenga mayor frecuencia de aparición en los modelos (véase Figura 20)

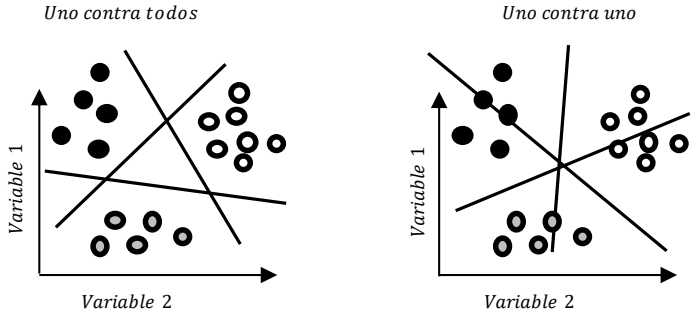

Figura 20. Modelos para clasificación multiclase SVM.

Tabla 8. Fortalezas y debilidades de SVM

Fortalezas	**Debilidades**
Muy buen rendimiento con un claro margen de separación	Combinaciones de kernels y modelos con un *cross validation* costoso computacionalmente
Buen rendimiento en espacios de alta dimensión	Tiempo de entrenamiento elevado cuando el conjunto de datos es grande
Buena eficiencia de memoria	Los datos han de ser separables con un kernel adecuado

4.2 Algoritmo Regresión Logística

La regresión logística es un algoritmo de aprendizaje supervisado que clasifica nuevos datos calculando las probabilidades de que los datos pertenezcan a cada una de las clases, de forma que mide la relación entre variables dependientes y una o más variables independientes mediante la estimación de las probabilidades usando la función logística siguiendo un flujo recogido en la Figura 21. Este algoritmo puede manejar tanto predictores continuos como categóricos, y modela una relación lineal entre los predictores y las probabilidades logarítmicas de pertenecer a la clase positiva [89].

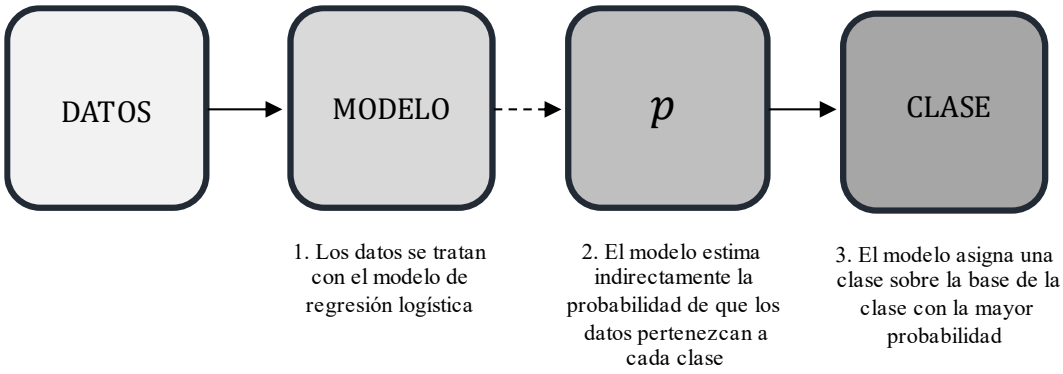

Figura 21. Flujo del algoritmo de regresión logística para clasificación.

Actualmente la regresión logística se usa en varios campos, incluido el aprendizaje automático, la mayoría de los campos médicos y las ciencias sociales.

Formalmente, puede expresarse del siguiente modo:

$$posibilidad = odds = \frac{probabilidad\ de\ acierto}{probabilidad\ de\ error}$$

Expresado en función de la probabilidad de acierto (p), se tiene que:

$$odds = \frac{p}{(1-p)}$$

Tomando logaritmos a ambos lados de la ecuación:

$$\log(odds) = \log\ (\frac{p}{1-p})$$

Despejando p se obtiene la función sigmoide (véase Figura 22):

$$p = \frac{e^{log(odds)}}{1+e^{log(odds)}} = \frac{1}{1+e^{-log(odds)}}$$

Siendo k el número de variables a predecir en el *dataset* puede definirse que:

$$\log(\frac{p}{1-p}) = \log(odds) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \dots \beta_k x_k$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \ldots \beta_k x_k)}}$$

Figura 22. Función sigmoide

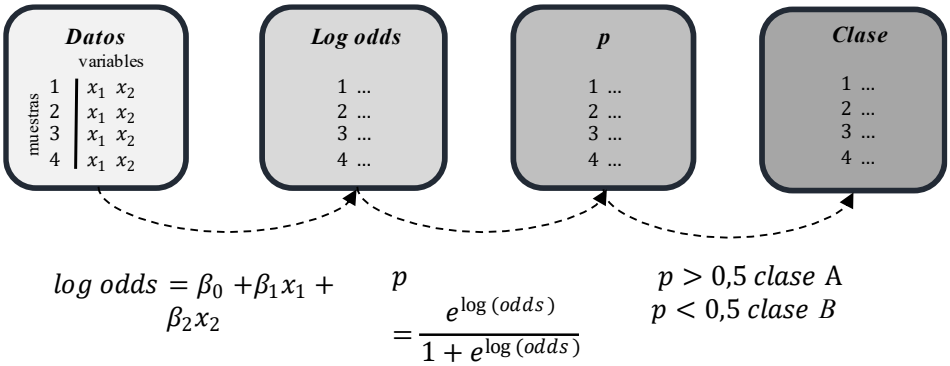

Figura 23. Estimación de clases en la regresión logística tipo bivariable

Puede observarse de forma esquemática el proceso que sigue el algoritmo (véase Figura 23). En primer lugar, se determinan las probabilidades logarítmicas o *logit* en base a los valores de las variables en cada muestra. Posteriormente esas probabilidades logarítmicas se convierten a probabilidades empleando la función logística. Finalmente, en base al valor obtenido se clasifica como una determinada clase en función de si es superior o inferior a 0,5 el valor obtenido.

El proceso anterior es válido para muestras bivariables, pero en la regresión logística multinomial, en lugar de estimar un solo *logit* para cada caso, el modelo estima un logit para cada caso para cada una de las clases de salida. Estos *logits* luego se pasan a una ecuación llamada función *softmax* (véase Figura 24) que convierte estos *logits* en probabilidades para cada clase, que suman 1. Luego, la clase que tenga la mayor probabilidad se selecciona como clase de salida [89].

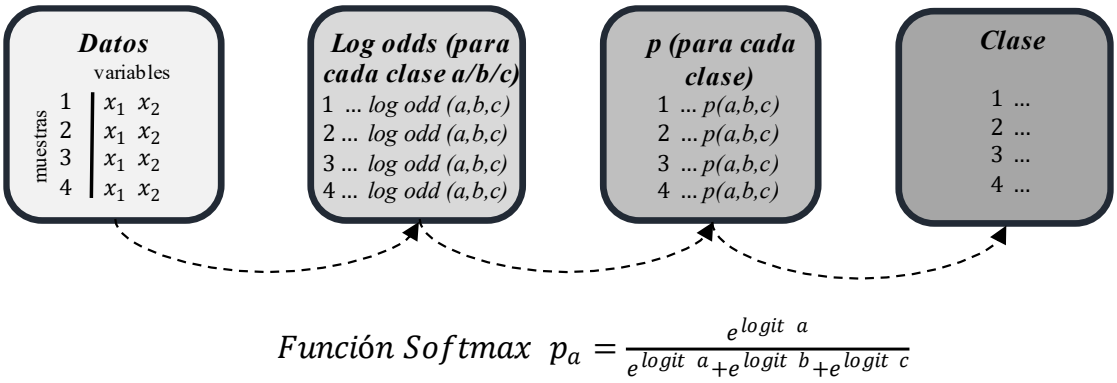

Figura 24. Estimación de clases en la regresión logística tipo multivariable

Los principales hiperparámetros que pueden ajustarse en una regresión logística conforme [92] son:

- La penalización o penalty, que se refiere al tipo de término de regularización que se va a emplear, con dos opciones principales:
  - 'l1': Regularización Lasso. Puede llevar a que algunos coeficientes sean exactamente cero, actuando como una especie de selección de características.
  - 'l2': Regularización Ridge (la que estás usando). Penaliza los coeficientes grandes, pero no necesariamente los hace cero.
- El parámetro de regularización C, aspecto ya reflejado en Support Vector Machines y, finalmente,
- El tipo de resolutor, o el algoritmo utilizado para optimizar y encontrar los mejores coeficientes para el modelo, siendo los posibles valores ’newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, o ‘saga’.

Es importante destacar, de cara al ajuste, que el resolutor tiene correlaciones con la penalización y la regularización.

Tabla 9. Fortalezas y debilidades de la regresión logística

Fortalezas	Debilidades
Puede predecir variables categóricas y continuas	Se asume que existe linealidad entre las variables independientes y los *logs odds*
Algoritmo sencillo y eficaz	Necesita una muestra mucho mayor en comparación con otros algoritmos
Requiere menos supuestos que, por ejemplo, la regresión lineal	Presenta problemas con clases desbalanceadas
Fácilmente interpretable	No es adecuada para relaciones no lineales

4.3 Algoritmo Decision Tree

El Árbol de Decisión o Decision Tree es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión y que destaca por su versatilidad y fácil comprensión. Usa una estructura de árboles en la que cada uno de los nodos representa una característica o atributo, los enlaces o ramas representan una decisión o regla y cada una de las hojas representa un resultado (véase Figura 25) [93].

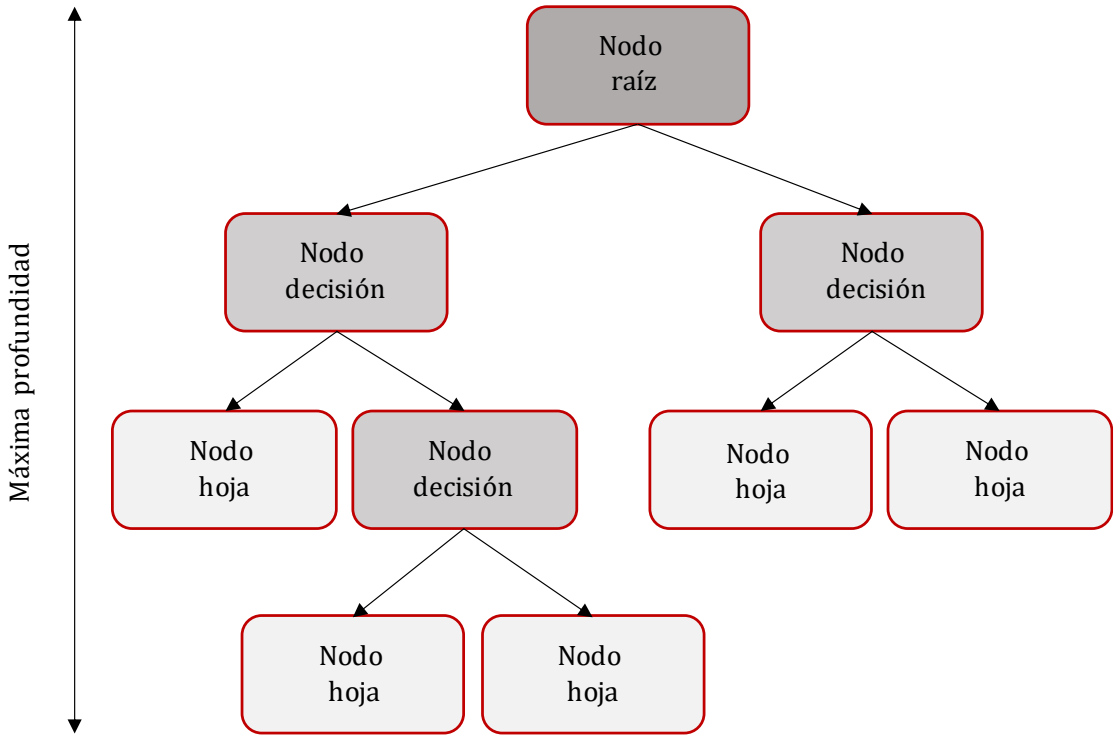

Figura 25. Esquema del algoritmo Decision Tree

Este algoritmo funciona siguiendo una serie de pasos para llevar a cabo tareas de clasificación y regresión. El proceso comienza con la selección de atributos o características. En cada nodo del árbol, el algoritmo elige el atributo que proporciona la mejor división del conjunto de datos basándose en:

a) La Ganancia de información -o Information Gain- que va a informar acerca de cuánta información se obtiene al dividir un conjunto de datos según una característica específica. Se basa en el concepto de entropía, que mide el desorden o la incertidumbre en un conjunto de datos. Cuanto mayor sea la ganancia de información, mejor será la división, porque significa que la característica seleccionada está reduciendo la incertidumbre (o el desorden) en el conjunto de datos. Formalmente queda definido como:

$$IG(S,F) = Entropía\ (S) - \left(\frac{|S_1|}{|S|} \times Entropía\ (S_1) + \frac{|S_2|}{|S|} \times Entropía\ (S_2)\right)$$

Donde:

- $Entropía\ (S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$
- p_+ y p_- son las proporciones de ejemplos positivos y negativos en S respectivamente
- S_1 y S_2 son los subconjuntos resultantes de la división de S en base a la característica F

b) El índice de Gini -o Gini Index-, que va a medir la impureza de un dataset. Un valor de Gini de 0 indica que todos los elementos del conjunto pertenecen a una sola clase, mientras que un valor de 0.5 (para un problema de clasificación binaria) indica que las clases están perfectamente mezcladas. Al dividir según una característica, el objetivo es minimizar el índice de Gini (ponderado), buscando la menor impureza posible. Formalmente queda definido como:

$$Gini\ (S) = 1 - ({p_+}^2 + {p_-}^2)$$

$$Gini\ Ponderado\ (S,F) = \frac{|S_1|}{|S|} \times Gini\ (S_1) + \frac{|S_2|}{|S|} \times Gini\ (S_2)$$

c) Otros; Ratio de Ganancia (Gain Ratio), Reducción de la Varianza o Índice de Chi-cuadrado.

Una vez tomada la decisión referente al atributo, el conjunto de datos se divide en subconjuntos de acuerdo con ese atributo. Cada nodo de decisión en el árbol tiene dos o más ramas, y cada nodo hoja del árbol va a representar una decisión final o una predicción [94], [95].

Los principales hiperparámetros que pueden ajustarse en un árbol de decisión conforme [92] son:

- *min_samples_split*: Se trata del número mínimo de muestras que debe haber en un nodo del arbol para que se lleve a cabo una división. Si un nodo tiene menos muestras que *min_samples_split*, no se dividirá y, por lo tanto, será un nodo final u hoja.
- *min_samples_leaf*: Número mínimo de muestras que debe tener un nodo que tenga la consideración de hoja. Si una división propuesta resulta en nodos hoja con menos muestras que *min_samples_leaf*, la división no se llevará a cabo. Esto puede ser útil para garantizar que las hojas tengan un número mínimo de muestras y evitar hojas con muy pocas muestras que podrían ser producto de ruido en los datos.
- *max_depth*: Se trata de la profundidad máxima que tendrá el árbol de decisión. Este hiperparámetro informa acerca de la longitud del camino más largo desde la

raíz hasta una hoja determinada, por el que *max_depth* va a limitar este valor. Es un modo de controlar el sobreajuste (*overfitting*), puesto que un árbol excesivamente profundo puede aprender demasiado sobre el conjunto de entrenamiento y no generalizar bien ante la presencia de nuevos datos.

Existen varias técnicas de poda del árbol para modificar características del árbol, y así evitar el sobreajuste, entre ellas; poda reducida de error, poda basada en complejidad, poda basada en profundidad, así como poda basada en el mínimo número de ejemplos existentes en un nodo. Varios de los anteriores hiperparámetros realizan funciones de poda, tanto en la profundidad del árbol (max_depth), como en el número mínimo de ejemplos (min_samples_split/ min_samples_leaf).

Tabla 10. Fortalezas y debilidades de Decision Tree

Fortalezas	Debilidades
Fácil de entender e interpretar.	Propenso al sobreajuste, especialmente con datos ruidosos o cuando el árbol es muy profundo.
Requiere poco preprocesamiento de datos (no necesita normalización).	Puede ser inestable debido a pequeñas variaciones en los datos.
Puede manejar tanto variables categóricas como numéricas.	Las decisiones son basadas en una estructura de árbol, que no siempre captura bien las relaciones lineales.
Capaz de manejar problemas de múltiples salidas.	Presenta limitaciones al usar variables continuas y categóricas con muchos niveles, generando árboles innecesariamente complejos.

4.4 Algoritmo AdaBoost

El AdaBoost (Adaptive Boosting) [96] es un algoritmo de ensamblado del tipo *boosting* que se basa en la combinación de varios submodelos denominados débiles (generalmente se trata de árboles de decisión de poca profundidad) para formar un modelo más fuerte y preciso (véase Figura 26). A diferencia de los métodos de *bagging* como Random Forest, el *boosting* trabaja en serie y ajusta el peso de las observaciones en función de los errores de los modelos anteriores.

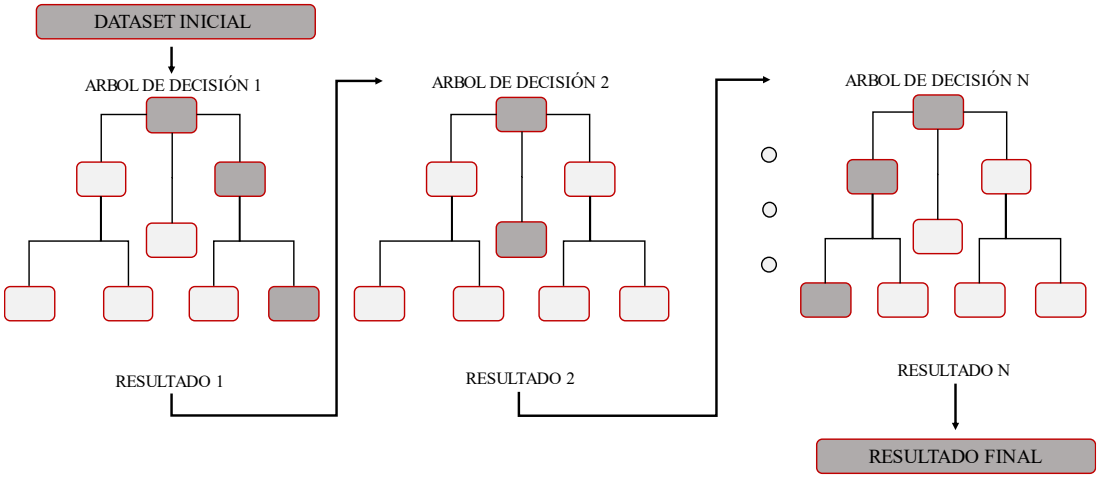

Figura 26. Esquema del algoritmo AdaBoost con árboles de decisión

A lo largo de cada iteración en el proceso de entrenamiento, AdaBoost se focaliza en aquellas observaciones clasificadas incorrectamente en la iteración anterior, y trata de ajustar sus pesos para un mejor rendimiento del siguiente modelo débil. Finalmente, todos los modelos se combinan para hacer una predicción final, donde cada modelo tiene un peso basado en su precisión.

Los hiperparámetros más comunes en AdaBoost son los mismos que los reseñados en Decision Tree [92] para la investigación, pero teniendo en cuenta además el hiperparámetro n_estimators, que no es otro aspecto que el número de árboles presentes, y la tasa de aprendizaje.

Tabla 11. Fortalezas y debilidades de AdaBoost

Fortalezas	Debilidades
Capacidad de combinar múltiples modelos débiles para formar uno fuerte, lo que puede resultar en una mayor precisión.	Sensible al ruido en los datos.
Menos propenso al sobreajuste que otros modelos, especialmente cuando el conjunto de datos no es muy grande.	El proceso iterativo puede ser computacionalmente costoso para conjuntos de datos grandes.
Puede emplearse con varios algoritmos de aprendizaje base o débiles, no sólo árboles de decisión.	La elección de un modelo débil inapropiado puede conducir a un bajo rendimiento.
Algoritmo adaptativo, ya que se focaliza en los datos mal clasificados.	Carece de la facilidad de interpretación de un árbol de decisión simple.

4.5 Algoritmo Random Forest

El Random Forest es un algoritmo que combina múltiples árboles de decisión para obtener una predicción más precisa y robusta que un solo árbol (véase Figura 27). Se utiliza para clasificación y regresión, y es especialmente útil para lidiar con grandes conjuntos de datos y conjuntos con un gran número de características [97].

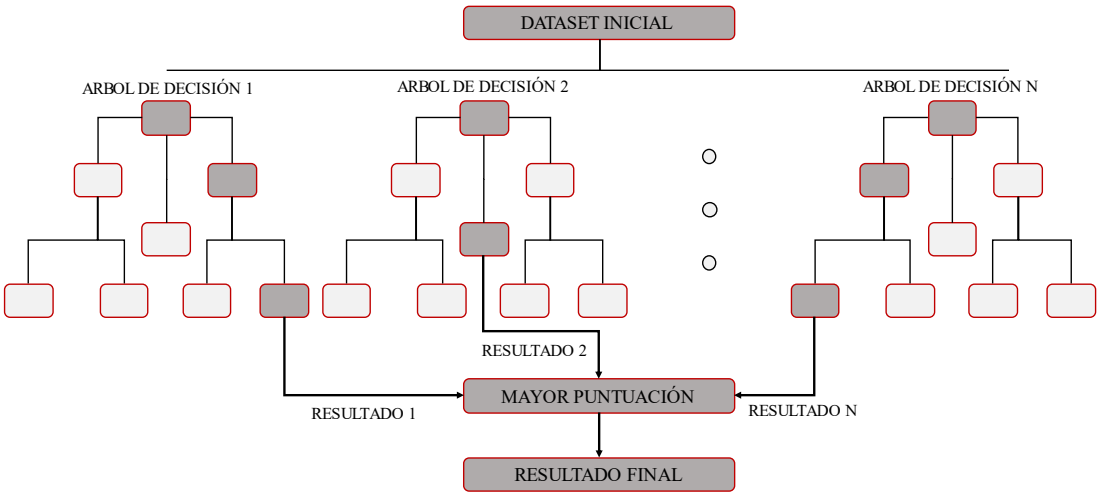

Figura 27. Esquema del algoritmo Random Forest o Bosques aleatorios

Los bosques aleatorios son algoritmos *ensemble* al emplear gran número de modelos simples, del tipo *bagging*, al emplear la técnica de *bootstraping* [97]. Funcionan construyendo múltiples árboles de decisión durante el entrenamiento y produciendo el promedio de las predicciones (en caso de regresión) o la clase que tenga la mayoría de los votos por parte de los árboles (en caso de clasificación).

Cada árbol se construye seleccionando aleatoriamente subconjuntos del conjunto de datos y características. Este proceso de selección aleatoria asegura que los árboles sean diversos y, por lo tanto, menos susceptibles al sobreajuste.

Los hiperparámetros más comunes en Random Forest son los mismos que los reseñados en Decision Tree [92] para la investigación, pero teniendo en cuenta además el hiperparámetro *n_estimators* que no es otro aspecto que el número de árboles presentes en el bosque.

Tabla 12. Fortalezas y debilidades de Random Forest

Fortalezas	Debilidades
Puede manejar conjuntos de datos grandes y tiene una alta precisión.	Menos interpretable que un árbol de decisión simple.
Puede manejar tanto variables categóricas como numéricas.	El rendimiento disminuye en datasets con gran número de variables.
Proporciona una estimación de qué variables son importantes en la clasificación.	No es muy recomendable para datos con relaciones lineales; en esos casos, los modelos lineales (Ej: SVM, Regresión Logística) podrían funcionar mejor.
Menor sobreajuste que los árboles de decisión simples. La poda no es imprescindible como en los árboles simples	No es muy recomendable para la realización de extrapolaciones

5 Estrategias de clasificación de Machine Learning empleadas en la investigación

De cara a poder implementar los anteriores algoritmos, es necesario estructurar las posibles variables que puedan desconocerse, es por ello por lo que en la investigación se van a evaluar las siguientes metodologías de tratamiento de variables:

5.1 Classifier Chains

Las cadenas de clasificadores representan un método avanzado en aprendizaje automático diseñado para la transformación del problema en la clasificación multietiqueta. Esta técnica logra integrar la eficiencia computacional del método de relevancia binaria, al tiempo que tiene en cuenta las dependencias entre las etiquetas para la clasificación.

El punto fuerte de este método es que cada clasificador, después del primero, se entrena no solo en las características de entrada, sino también en las etiquetas predichas por todos los clasificadores anteriores en la cadena enriqueciendo la información. Este enfoque permite que las relaciones entre las etiquetas se tengan en cuenta durante el entrenamiento, lo que puede mejorar el rendimiento en tareas donde las etiquetas están correlacionadas de manera compleja [98].

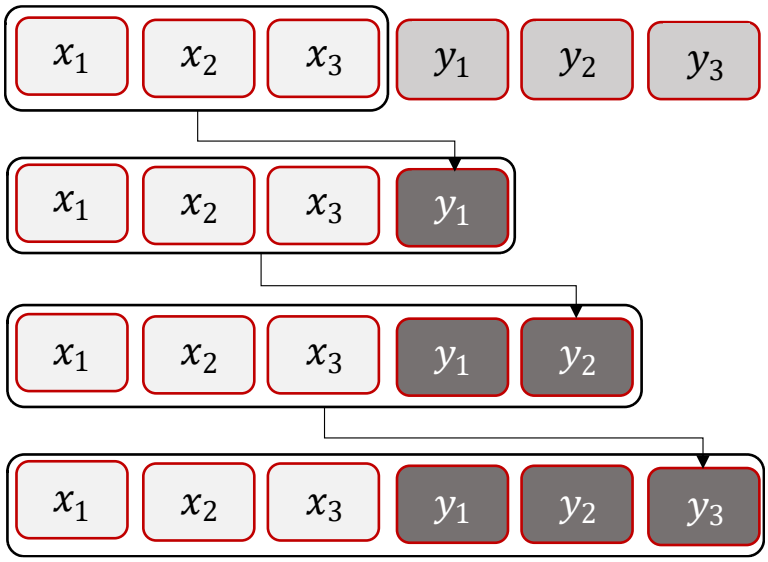

Figura 28. Ejemplo de una cadena de clasificadores

Tabla 13. Fortalezas y debilidades de las cadenas de clasificadores

Fortalezas	Debilidades
Consideración de dependencias de etiquetas	Sensibilidad frente al orden de etiquetas
Eficiencia computacional	Complejidad ante elevado número de etiquetas
Flexibilidad	Generalización limitada si no identifica relaciones

5.2 One Vs Rest (OvR)

El método Uno Contra el Resto (One Vs Rest, OvR) es una técnica de clasificación multiclase que implica entrenar un clasificador separado para cada clase contra todas las otras clases. Es decir, para N clases, se entrenan N clasificadores distintos, cada uno diseñado para reconocer una clase específica frente a todas las demás. De cara al entrenamiento, y respecto a las variables desconocidas, aquella asociada al clasificador tendrá una etiqueta positiva "1" y el resto de las clases se etiquetarán como "0". En el momento de realizar una predicción, todos los clasificadores emiten su predicción y se selecciona la clase que haya sido predicha por su clasificador correspondiente. Aunque es un método simple y directo, OvR es muy eficaz en una amplia gama de tareas de clasificación multiclase [48].

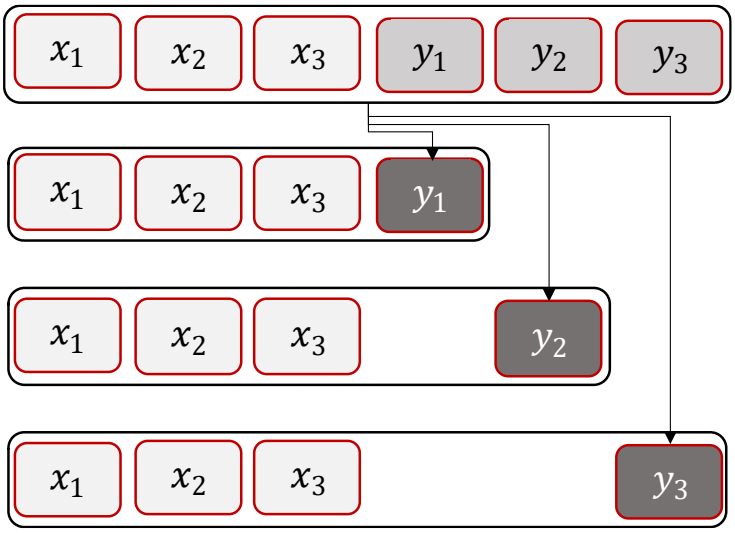

Figura 29. Ejemplo de la estrategia One Vs Rest

Tabla 14. Fortalezas y debilidades de OvR

Fortalezas	Debilidades
Simplicidad	Necesidad de entrenar múltiples clasificadores para conjuntos con muchas clases
Independencia de los clasificadores	Rendimiento comprometido si el clasificador binario base no es robusto o sufre de sobreajuste (*overfitting*)
Eficiencia computacional, especialmente con un número moderado de clases	No captura las relaciones o dependencias entre diferentes clases, ya que trata a cada clase independientemente del resto.

6 Las métricas de rendimiento

Existen gran variedad de indicadores para evaluar el rendimiento de un modelo de *Machine Learning*. La matriz de confusión (véase Figura 30) se considera una de las formas más estandarizadas de presentar los detalles de la evaluación. Esta herramienta muestra los valores de los Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos que servirán para la elaboración ulterior de otra serie de indicadores. La validez de las métricas varía dependiendo de cada caso. Supóngase que se analiza si una transacción financiera pudiera ser fraudulenta o autorizada, es imprescindible conocer los falsos negativos, dado que un valor significativo puede devenir en una sustancial pérdida financiera [99].

- Verdaderos Positivos (True Positives -TP-): Recuento del número de muestras que son correctamente clasificadas como positivas en el modelo de ML.
- Verdaderos Negativos (True Negatives -TN-): Recuento del número de muestras que son correctamente clasificadas como negativas en el modelo de ML.
- Falsos Positivos (False Positives -FP-): Recuento del número de muestras que son incorrectamente clasificadas como positivas en el modelo de ML.
- Falsos Negativos (False Negatives -FN-): Recuento del número de muestras que son incorrectamente clasificadas como negativas en el modelo de ML.

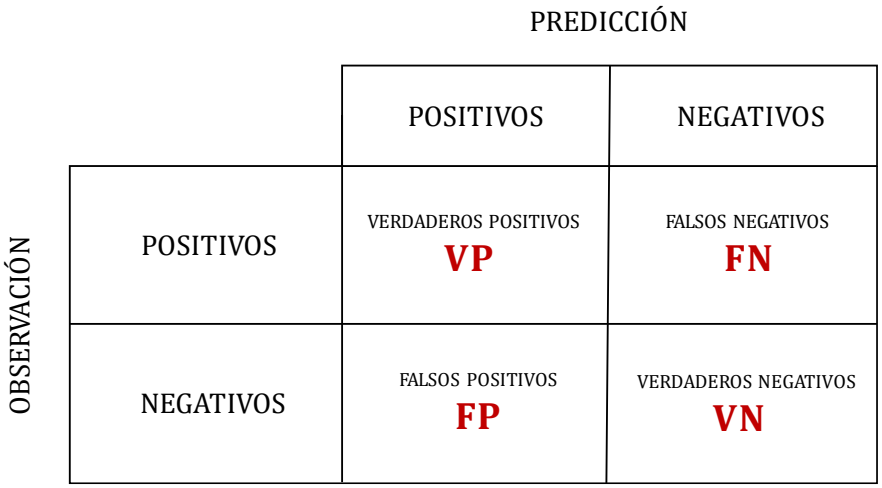

Figura 30. Matriz de confusión

En consonancia con lo indicado más arriba, derivan una serie de métricas de interés que se emplearán en la investigación para analizar los resultados de los modelos que se definan:

Precisión (Precision): Es la proporción de Verdaderos Positivos (TP) entre la suma de todas aquellas instancias clasificadas como positivas por el modelo. Este indicador informa sobre la capacidad del modelo para detectar correctamente instancias positivas.

$$Precisión = \frac{TP}{(TP + FP)}$$

Sensibilidad (Sensitivity o Recall): Es la proporción de Verdaderos Positivos (TP) entre la suma de ese mismo valor y el número de instancias incorrectamente clasificadas como negativas, o Falsos Negativos (FN). Este indicador informa sobre la capacidad del modelo para detectar correctamente aquellas instancias realmente positivas.

$$Sensibilidad = \frac{TP}{(TP + FN)}$$

Especificidad (Specificity): Es la proporción entre aquellas instancias correctamente clasificadas como negativas o Verdaderos Negativos (TN) y ese mismo valor sumado al número de instancias incorrectamente clasificadas como positivas o Falsos Positivos (FP). Este indicador informa sobre la capacidad del modelo para detectar correctamente aquellas instancias realmente negativas.

$$Especificidad = \frac{TN}{(TN + FP)}$$

Exactitud (Accuracy): Es la proporción entre la suma de aquellas instancias correctamente clasificadas (TP+TN) y todas las instancias sometidas al modelo (TN+TP+FP+FN). Este indicador informa sobre la capacidad del modelo para clasificar correctamente las instancias.

$$Exactitud = \frac{(TP + TN)}{(TN + TP + FP + FN)}$$

Valor-F (F1-score): Es un indicador que combina la Sensibilidad (Recall) y la Precisión en un solo valor. Su cálculo se realiza mediante la media armónica entre la Sensibilidad y la Precisión del modelo.

$$F1 - score = \frac{(2 * Precisión * Sensibilidad)}{(Precisión + Sensibilidad)}$$

Curva ROC (Received Operating Characteristic Curve): Es un gráfico ampliamente usado para el análisis de modelos de *Machine Learning* que permite representar la tasa de Verdaderos Positivos (eje y) en función de la tasa de Falsos Positivos (eje x), mostrando el rendimiento de clasificación a través de diferentes umbrales.

Área bajo curva ROC -AUC-: Representa el rendimiento del modelo, de modo que cuanto más cercano a 1 sea su valor, mejor será el rendimiento de este.

Capítulo 5. Tratamiento de datos del marco de ciberinteligencia Mitre Att&ck para la aplicación de algoritmos de Machine Learning

1 Introducción

En base al análisis realizado en el capítulo 3, Mitre Att&ck [15] se ha podido estimar como el *framework* de ciberinteligencia más adecuado para poder parametrizar y definir un ataque en la actualidad. A raíz de profundizar en el estudio de los datos disponibles, se constata que los datos no pueden manejarse tal y como se encuentran en bruto, y es imposible la aplicación del modelo matemático sin llevar a cabo un tratamiento de los mismos. Debe tenerse presente que un buen conjunto de datos tiene una influencia crucial en el diseño de algoritmos de *Machine Learning* si se quiere garantizar la calidad, representatividad, generalización, eficiencia computacional e interpretabilidad de los modelos. Ante este panorama, se observan los siguientes retos y limitaciones con los datos de Mitre.

1.1 Retos

El marco Mitre Att&ck está en constante actualización, habiendo variado en numerosas ocasiones su estructura y contenidos, si bien es cierto que la nomenclatura y gran número de las tácticas se mantienen a lo largo del tiempo. Esta circunstancia genera la necesidad de disponer de un programa de actualización del modelo a desarrollar que permita adaptarse a los cambios del *framework*.

Otro de los retos que se observa es el reducido número de observaciones que pueden tratarse en la investigación. Si bien es cierto que el paso del tiempo permitirá disponer de una mayor base de datos, los datos actuales son reducidos y ha de contemplarse un mecanismo que permita ampliar las observaciones.

Los grupos suelen representar entidades que disponen de un gran número de tácticas, técnicas y procedimientos (TTPs), mientras que el software representa acciones con muy pocas técnicas. Además, la presencia de subtécnicas genera distorsiones, por lo que no se ha contemplado en la investigación esta modificación y se ha optado por incluir únicamente referencias a las técnicas, reduciendo considerablemente la necesidad de computación.

Finalmente, debe tenerse presente que la adopción y aplicación efectiva de *Mitre Att&ck* requerirá una inversión significativa en tiempo y recursos dentro de las organizaciones de cara a que los profesionales puedan familiarizarse con este marco y adquieran el profundo conocimiento necesario para la explotación de los algoritmos que se van a desarrollar.

1.2 Limitaciones.

Los datos contenidos en el repositorio de Mitre Att&ck no son representativos de todas las técnicas posibles utilizadas por los actores asociados con los datos observados, sino

un subconjunto de lo que de lo que ha estado disponible a través de informes públicos y de código abierto.

La segunda limitación que se observa es el posible sesgo que tendrán los datos, bien por el propio algoritmo que extraiga la información de forma automática de informes o bien mediante el propio sesgo cognitivo del analista de ciberinteligencia.

El presente capítulo se organiza de modo que en el segundo apartado se presenta la metodología propuesta, consistente en la exploración y gestión de los datos origen, así como el tratamiento de las variables identificadas. En el tercer apartado se muestran los resultados obtenidos y en el cuarto y último apartado se muestran las conclusiones derivadas del trabajo realizado.

2 Metodología

2.1 Exploración de los datos.

La exploración de datos es un paso esencial en el desarrollo del modelo de inteligencia artificial que se va a plantear en la investigación. Se va a buscar información, descubrir patrones y relaciones e identificar posibles problemas futuros para tomar decisiones informadas acerca de los datos. Para ello se realizarán las fases que se recogen en los siguientes apartados.

2.1.1 Identificación del dataset

El *dataset* inicial se conformará mediante la aglutinación de cinco conjuntos de datos reflejados en el apartado "*Working with Mitre Att&ck*" [100] de la web de Mitre y de acceso público, concretamente aquellos datos referidos al tipo *Enterprise* o modelizado estándar que no haga referencia a sistemas de control industrial o tecnologías móviles. En este sentido se emplearán los datos publicados en la web a fecha 14/01/2023. Así pues, se identifican; Fuente 1: Dataset "dfg" correspondiente con los grupos identificados. Fuente 2: Dataset "dfs" correspondiente con las aplicaciones software identificadas. Fuente 3: Dataset "dfta" correspondiente con las tácticas del modelo. Fuente 4: Dataset "dfte" correspondiente con las técnicas del modelo. Fuente 5: Dataset "dftr" correspondiente con las relaciones existentes entre los anteriores datos.

2.1.2 Identificación de variables

Las variables se asociarán con las técnicas o, en su caso, las tácticas Mitre Att&ck, representadas con nomenclatura "*TXXXX*" o "*TAXXXX*". Estos valores son de tipo categórico, mutuamente excluibles entre sí y sin ningún tipo de orden. El número total de tácticas es de 14, y el número de técnicas es muy elevado, ascendiendo a 193, de modo que se deberá especificar a lo largo del proceso de tratamiento de datos diferentes alternativas para poder trabajar con un número que represente adecuadamente la información sin incluir información redundante y que dificulte el trabajo computacional. Las variables que se consideran de interés se denominarán variables respuesta o dependientes mientras que aquellas que se van a medir se denominarán variables predictoras, explicativas o independientes.

2.1.3 Limpieza de datos

No se observa la presencia de valores nulos o erróneos, de modo que pueden obviarse las tareas de limpieza y sustitución de datos. De cara a obtener un marco óptimo de trabajo se llevan a cabo las siguientes acciones de limpieza en cada conjunto de datos:

- Dfg: se reduce el número de columnas únicamente a las siguientes: "*ID*", "*name*", "*associated groups*" y se eliminan las restantes. Se conforma un dataset con 3.604 técnicas observadas en 133 actores, que tienen ligado un nombre y una serie de grupos asociados.
- Dfs: se reduce el número de columnas únicamente a las siguientes: "*ID*" y "*source_name*", y se eliminan las restantes. Se conforma un *dataset* con 9.420 técnicas observadas en 544 aplicaciones.
- Dfta: se reduce el número de columnas únicamente a las siguientes: "*IDF*", "*name*" y se eliminan las restantes para que pueda observarse con claridad la identificación y nombre de cada táctica.
- Dfte: Se identifica si cada una de las técnicas tiene una o varias subtécnicas (en cuyo caso se simplificarán a técnicas) y por otro lado se asociará cada técnica con las tácticas en las que se encuentre presente, que pueden ser una, dos, tres o hasta cuatro posibles técnicas. Posteriormente, se asocia el nombre de la táctica con el identificador de táctica en formato *TAXXXX*.

A continuación, se tratará el dataset "Dftr", que ejercerá como un diccionario de consulta y que va a permitir, para cada técnica identificar las tácticas que tiene asociadas.

- Dftr: se seleccionan las columnas "*source type*"," *target ID*"," *target type*" y" *source ID*". Hay que tener presente que se eliminarán las posibles subtécnicas que aparezcan al asociar las mismas a la técnica raíz.

Finalmente se llevará a cabo un cotejo para que se pueda crear un nuevo *dataset* con los siguientes campos: "*source ID*", "*ID*" y "*táctica*" que relaciona la entidad (software o grupo) con su identificación con una determinada táctica. Se identifican 13.023 observaciones en ese nuevo conjunto de datos. Estas observaciones se intentarán agrupar en base a la pertenencia a una determinada entidad, bien sea un grupo, bien sea una aplicación de software determinada. Para cada una de estas entidades se dispondrá del conjunto de técnicas o tácticas con las que suelen operar. Es reseñable que se observan únicamente 180 técnicas en los datos de entidades, lo que indica que hay 13 técnicas no reflejadas en los datos que, en todo caso se tendrán en cuenta ante posibles futuras inclusiones.

2.2 Gestión de los datos

2.2.1 Codificación de la información

Para poder aplicar los algoritmos de inteligencia artificial es imprescindible la codificación de la información de las variables categóricas identificadas. Mediante ello, se conseguirá mayor precisión, consistencia, facilidad de uso e interoperabilidad con otros sistemas y aplicaciones. Se van a analizar a continuación tres opciones interesantes para la codificación de los datos (*one-hot*, binaria y *hashing*). En todos los casos se llevará a cabo, de una forma u otra, un tratamiento binario de los datos mediante vectorización de las tácticas o técnicas Mitre Att&ck como reflejan [4][5] para las tácticas. La binarización es una metodología de representación de datos mediante el empleo de únicamente los dos dígitos binarios (0 y 1). Cada dato se representa como una secuencia de unos y ceros, donde cada dígito binario se denomina "bit". Esta codificación de emplea ampliamente en la electrónica y la informática para obtener una información digital. La inmensa mayoría de los sistemas informáticos de la actualidad emplean la codificación binaria para la representación de datos, facilitando su integración en los ecosistemas electrónicos en base a la presencia o ausencia de señales eléctricas para almacenar y procesar información.

2.2.1.1 One-Hot Encoding

Es una técnica empleada para la conversión en valores numéricos de variables categóricas. La técnica basa su funcionamiento en la creación de una columna separada para cada valor único en la variable categórica. Existen diversas aplicaciones de este método; *Effects Coding*, *Contrast Coding*, etc. pero el método que se va a aplicar es la codificación *Dummy*, descrita por [102]. Este método va a convertir variables categóricas en indicadores o variables "*dummy*", de modo que, para cada categoría única en la columna dada, se crea una nueva columna en el conjunto de datos. Cada una de estas nuevas columnas tendrá un valor de 1 cuando la categoría original coincida con la columna y un 0 en caso contrario.

De este modo, siendo V un vector de longitud M con N únicas categorías:

$$V = [v_1, v_2, \dots, v_M]$$

Mediante la aplicación de *one-hot encoding* lo que se va a obtener es una matriz B en la que cada fila corresponde a un elemento de V, teniendo un 1 en la columna asociada a esa categoría y un cero en las restantes columnas.

$$B(i,j) = I(v_i == c_j)$$

Donde:

- $B(i,j)$ va a ser el elemento de la fila i-ésima y columna j-ésima de la matriz B
- $v_i$va a ser el elemento i-ésimo del vector de variables categóricas V
- c_j va a ser la j-ésima categoría única del vector V
- $I(x)$ es una función indicatriz que va a devolver 1 si la condición x es cierta y 0 en otro caso

Dado que las entidades (grupos o software) tienen asociadas varias técnicas, esta variedad de codificación permite de forma idónea esa representación ya que la suma de dos vectores de técnicas refleja la posible coexistencia de dos técnicas en una entidad.

En el caso concreto de la investigación, se dispone de un conjunto de datos que contiene información sobre el tipo de técnica empleada, bien por un determinado grupo de ataque o bien por un software concreto. La variable categórica puede tomar, teóricamente, 193 valores posibles correspondientes con cada una de las técnicas disponibles. Por tanto, la representación será del tipo que se refleja en la siguiente tabla:

Tabla 15. Representación de los datos mediante One-hot encoding

	T0001	T0002	T0003	T0004	…	T0193
T0001	1	0	0	0		0
T0002	0	1	0	0	.	0
T0003	0	0	1	0	.	0
T0004	0	0	0	1	.	0
…		.	.	.	.	0
T0193	0	0	0	0	.	1

2.2.1.2 Codificación binaria

Esta técnica convierte cada valor categórico en una secuencia binaria de longitud fija en base al empleo de un número de bits determinados. Así pues, sea V un vector de longitud M con N únicas categorías:

$$V = [v_1, v_2, \dots, v_M]$$

Para llevar a cabo esta técnica se puede identificar cada variable categórica con un número (generalmente correlativo) al que se le va a transformar a posteriori en una representación binaria. De este modo se va a disponer de un mapeo binario.

$$B(i,j) = I(v_i, c_j, k)$$

Donde:

- $B(i,j)$ va a ser el elemento de la fila i-ésima y columna j-ésima de la matriz B
- $v_1$va a ser el elemento i-ésimo del vector de variables categóricas V
- c_j va a ser la j-ésima categoría única del vector V
- k va a ser la k-ésima posición bit en la representación binaria de la categoría
- $I(x,y,z)$ es una función indicatriz que va a devolver el valor del bit z-ésimo en la representación binaria de x si x es igual a y, e indefinido en caso contrario

Dado que las entidades (grupos o software) tienen asociadas varias técnicas, esta variedad de codificación no permite esa representación, ya que la suma de dos vectores de técnicas representaría una nueva técnica, no la posibilidad de que coexistan esas dos técnicas.

Adaptándolo a la particularidad de la investigación, la variable categórica que representa el tipo de técnica puede tomar 193 valores posibles correspondientes con cada una de las técnicas disponibles, por lo que el logaritmo en base 2 de 193 es aproximadamente 7,6. Por tanto se hace necesaria una secuencia binaria de al menos 8 bits para representar cada valor. Si el valor que se asigna a la primera variable es 0, se representaría como 00000000. Si el valor asignado a la segunda variable es 1, se representa como 00000001. Y así sucesivamente, para cada una de las 193 categorías. Por tanto, la representación será del tipo que se refleja en la siguiente tabla:

Tabla 16. Representación de los datos mediante codificación binaria

	VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8
T0001	0	0	0	0	0	0	0	1
T0002	0	0	0	0	0	0	1	0
T0003	0	0	0	0	0	0	1	1
T0004	0	0	0	0	0	1	0	0
…		.	.	.	.	.	.	.
T0193	1	1	0	0	0	0	0	1

2.2.1.3 Codificación mediante hashing

El *hashing* es una técnica que utiliza una función matemática para asignar una cadena de texto de longitud variable a un valor numérico de longitud fija [103]. Esta técnica se utiliza comúnmente para reducir la dimensionalidad de los datos de entrada, pero ha de tenerse presente las posibles colisiones. Desde el punto de vista de creación de nuevas categorías es una opción muy útil ya que no obliga a recalcular al incluir nuevas categorías.

Sea V un vector de longitud M con N únicas categorías:

$$V = [v_1, v_2, \dots, v_M]$$

Para aplicar esta codificación se aplicará una función *hashing* h a cada una de las categorías. A continuación, se aplicará una función módulo para limitar el número de columnas en la representación codificada:

$$B(i,j) = I(h(v_i)\ mod\ k == j)$$

Donde:

- $B(i,j)$ va a ser el elemento de la fila i-ésima y columna j-ésima de la matriz B
- $v_1$va a ser el elemento i-ésimo del vector de variables categóricas V

- h es la función de hashing
- c_j va a ser la j-ésima categoría única del vector V
- k es el número de columnas en la representación codificada
- $I(x == y)$ es una función indicatriz que va a devolver 1 si x es igual a y, y 0 en caso contrario.

Para aplicar *hashing*, al caso concreto de la investigación, en primera instancia se empleará una función *hash* para cada cadena de texto que representa la técnica (aunque podría haberse realizado mediante un número correlativo asignado) y se aplicará, por ejemplo, una *k*=20 que se refleja mediante la creación de 20 nuevas variables o columnas. Por tanto, la representación será del tipo que se representa en la siguiente tabla:

Tabla 17. Representación de los datos mediante codificación hashing

	VAR1	VAR2	VAR3	VAR4	…	VAR20
T0001	0	0	0	0		1
T0002	0	1	0	0	.	0
T0003	0	0	1	0	.	0
T0004	0	1	0	0	.	0
…		.	.	.		0
T0193	0	0	0	0		0

2.2.1.4 Análisis de las codificaciones desarrolladas

De cara a reflejar de manera sencilla las fortalezas y debilidades que se han observado tras el estudio de estos tres métodos se ha elaborado la Tabla 18. Si bien es cierto que la inclusión de nuevas categorías implica una nueva distribución de codificación del *dataset*, la elección del método *one-hot* parece el más idóneo, en especial ante la capacidad que tiene de cara a agrupar las técnicas de las diferentes entidades en un único vector, algo crucial para la investigación.

Tabla 18. Análisis de las características de cada método de codificación de los datos

	ONE HOT	BINARIO	HASHING
SIMPLICIDAD	+++	++	+
INTERPRETABILIDAD	+++	++	+
DIMENSIONALIDAD	+++	++	++
ROBUSTEZ ANTE NUEVAS CATEGORÍAS	+	++	+++
EFICIENCIA ALMACENAMIENTO	+	++	+++
COLISIONES	NO	NO	SÍ
COMPATIBILIDAD ALGORITMOS	+++	++	++
PERMITE VARIAS TÉCNICA EN UN VECTOR	SÍ	NO	NO

Así pues, se van a agrupar en la variante *one-hot* los datos por entidades (grupos o aplicaciones de software), en las que cada una de ellas va a representar una observación, teniendo un conjunto de técnicas representadas. Dado que es posible que un grupo registre más de una observación relativa a una técnica, y dado que se van a conformar variables binarias, se eliminará el valor por el de la unidad en caso de que sea superior a uno. Por tanto, la representación será del tipo representado en la siguiente tabla:

Tabla 19. Representación de los datos mediante codificación one-hot y agrupación de observaciones por entidades

	T0001	T0002	T0003	T0004	…	T0193
Entidad 1	1	0	0	1		0
Entidad 2	0	1	0	0	.	1
Entidad 3	0	0	1	0	.	0
Entidad 4	1	0	0	1	.	0
…		.	.	.	.	0
Entidad 750	0	0	0	0	.	1

2.2.2 Análisis de las variables

El análisis de variables es un paso fundamental al permitir comprender los datos de forma profunda, identificar patrones y relaciones y en base a ello poder tomar decisiones fundamentadas, por ejemplo, en materia de supresión o agrupación de variables. De forma concreta, esto va a permitir seleccionar las variables más relevantes e identificando aquellas que tengan mayor relación con el resultado, reduciendo la complejidad del modelo y su calidad.

Además, con la codificación *one-hot* se va a disponer de una elevada dimensionalidad que es necesaria reducir para evitar problemas computacionales y de adaptación futura del *framework*. En este apartado se lleva a cabo un estudio de una aproximación en base a PCA, LDA o la agrupación de variables por tácticas, y el análisis de correlaciones y varianzas de las variables, descartando la selección de categorías más frecuentes ante el riesgo de alterar el modelo al no representar las variables con menor frecuencia.

2.2.2.1 Estudio del uso de PCA para la reducción dimensional

El Análisis de Componentes Principales (PCA) [104] es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos de modo que se pueda conservar la mayor cantidad posible de variabilidad posible y por tanto de información. Esta técnica es de especial utilidad en aquellos conjuntos de datos con un número de variables elevado, de forma que se va a ver reducido significativamente el número de variables, siendo sustituidas por los componentes principales.

Tabla 20. Proporción de varianza y varianza acumulada tras aplicar PCA

	PROPORCIÓN DE VARIANZA	VARIANZA ACUMULADA
PC1	13,9	13,92
PC2	5,06	18,97
PC3	3,76	22,73
PC4	3,09	25,83
PC5	2,76	28,59
…	…	…
PC10	1,93	39,78
…	…	…
PC20	1,29	55,48
…	…	…
PC30	0,97	66,55
…	…	…
PC40	0,73	74,93
…	…	…
PC50	0,54	81,16

Tras llevar a cabo el análisis en el *dataset* que contiene las 180 variables (recordar que, pese a que el número de variables que indica Mitre Att&ck es de 193, en el dataset únicamente se reflejan 180), se reflejan los resultados más relevantes en la Tabla 20, de la cual se pueden extraer las siguientes conclusiones:

- La primera componente principal (PC1) explica un 13.9% de la varianza total del conjunto de datos.
- La suma de las proporciones de varianza de las primeras 10 componentes principales refleja un total de aproximadamente el 39.8% de la varianza explicada.
- Considerando las primeras 50 componentes, solo se logra explicar alrededor del 81.2% de la varianza.

Para ayudar a la comprensión se ofrece la Figura 31 en la que se observa la varianza acumulada e individual de las componentes principales.

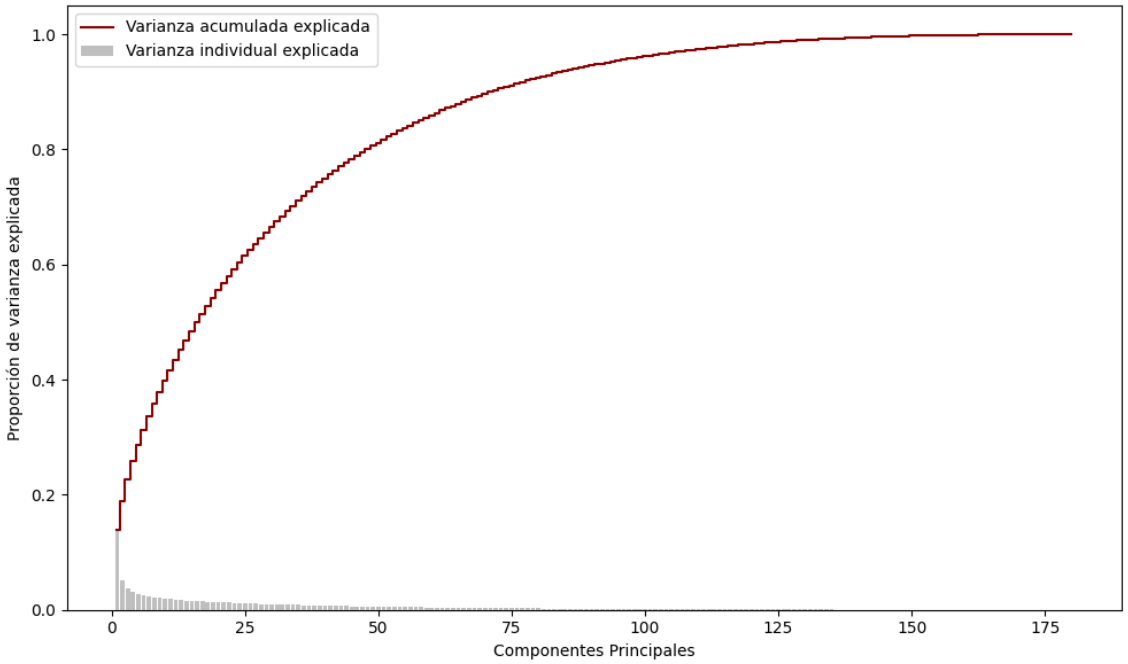

Figura 31. Varianza acumulada e individual de las variables PCA

Para que el PCA sea realmente beneficioso, se esperaría que unas pocas componentes principales expliquen una gran proporción de la varianza total. Para el caso concreto que se está analizando, se requeriría conservar un número significativamente alto de componentes para retener la mayoría de la información del *dataset* original. Esto minimiza la ventaja de la reducción de dimensionalidad, ya que no se logra una simplificación significativa del conjunto de datos.

Por tanto, en este caso particular, no se recomienda emplear PCA como técnica de reducción de dimensionalidad. No proporciona una reducción suficientemente significativa para justificar su implementación, y podría dar lugar a la pérdida de información relevante.

2.2.2.2 Estudio del uso de LDA para la reducción dimensional

El Análisis Discriminante Lineal (LDA) [105] es una técnica empleada con la finalidad de reducir la dimensionalidad de un *dataset* basándose en las diferencias entre categorías o clases, con el objetivo principal de maximizar la separación que exista entre ellas. Esta técnica es especialmente valiosa en conjuntos de datos con múltiples variables, permitiendo la generación de componentes discriminantes que reflejen la máxima variabilidad entre las clases. LDA, a diferencia de técnicas no supervisadas, como es el caso de PCA, se centra en localizar los ejes que proporcionen la mejor distinción entre grupos predefinidos, haciéndolo especialmente de utilidad para problemas de clasificación.

Tabla 21. Proporción de varianza y varianza acumulada tras aplicar LDA

	PROPORCIÓN DE VARIANZA	VARIANZA ACUMULADA
LD1	46,11	46,11
LD2	11,69	57,8
LD3	3,49	61,29
LD4	2,93	64,22
LD5	2,52	66,74
…	…	…
LD10	1,63	72,72
…	…	…
LD20	0,83	88,44
…	…	…
LD30	0,41	94,78
…	…	…
LD40	0,23	97,83
…	…	…
LD50	0,10	99,38

Tras reflejar los resultados más relevantes en la Tabla 21, se pueden extraer las siguientes conclusiones:

- La primera componente discriminante (PC1) explica un 46,11% de la varianza total del conjunto de datos.
- La suma de las proporciones de varianza de las primeras 10 componentes discriminantes refleja un total de aproximadamente el 72,72% de la varianza explicada.
- Considerando las primeras 50 componentes, se logra explicar alrededor del 99,38% de la varianza.

Para ayudar en la interpretación de estos datos se ofrece la Figura 32, que recoge todos los datos de la reducción:

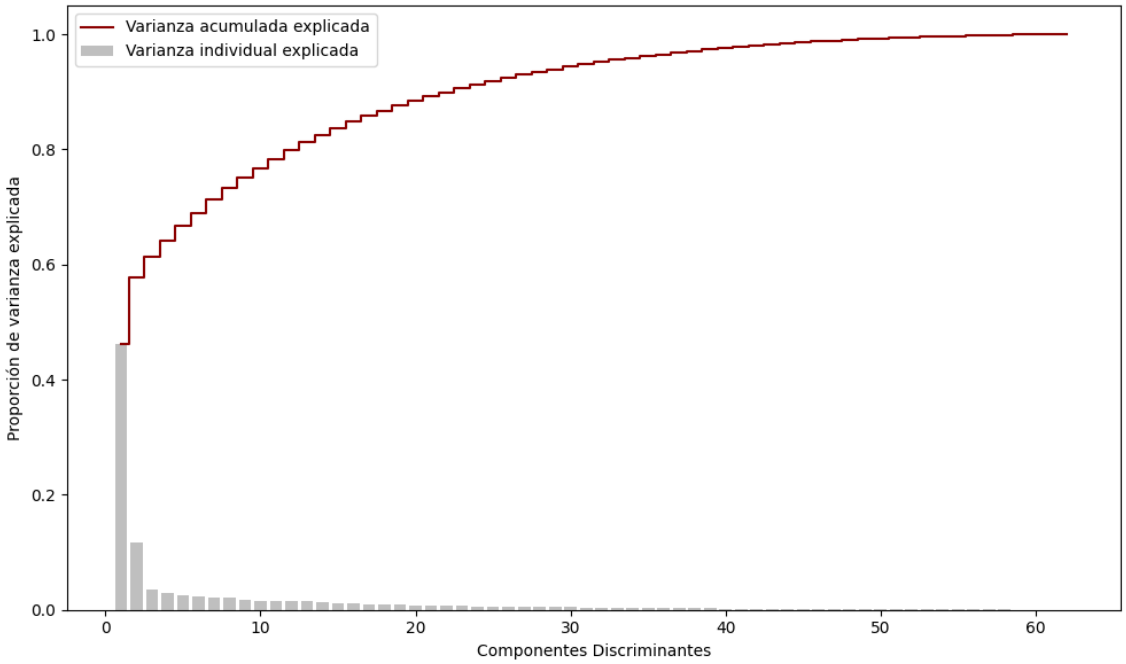

Figura 32. Varianza acumulada e individual de las variables LDA

Se observa que el primer componente es muy dominante, está capturando gran varianza. Este hecho conduce a afirmar que hay una dirección en el espacio de características que discrimina bien entre las clases. A partir del segundo componente en adelante, la contribución de cada uno de ellos a la varianza total disminuye muy significativamente, lo que sugiere que estos componentes adicionales proporcionan menor información que pueda servir de ayuda.

Conforme los datos mostrados, esta técnica de reducción dimensional es de mayor utilidad que la técnica PCA mostrada en el apartado anterior, consiguiendo una aceptable representación con un número de variables contenido. No obstante, la compleja interpretabilidad de esta reducción dimensional motiva que se continúe explorando posibilidades de reducción del número de variables.

2.2.2.3 Agrupación por tácticas del framework

La aglutinación de las 193 técnicas de Mitre Att&ck en las 14 tácticas puede ser una solución factible de cara a reducir la elevada dimensionalidad y por ende reducir y simplificar el modelo. Además, las 14 tácticas del marco proporcionan una estructura más simple, interpretable, lógica y coherente para representar las actividades y tácticas empleadas en un ataque, lo que puede ayudar a mejorar la interpretación de los resultados del modelo. Así pues, siendo T el conjunto de técnicas, y TA el de tácticas se definirá una función tal que:

$$f: T \rightarrow TA$$

De modo que para cualquier t_i en T, $f(t_i) = TA_j$, donde TA_j es la táctica correspondiente a la técnica t_i.

Así pues, las variables se reducirán obteniendo las siguientes:

TA0043: Reconnaissance
TA0042: Resource Development
TA0001: Initial Access
TA0002: Execution
TA0003: Persistence
TA0004: Privilege Escalation
TA0005: Defense Evasion
TA0006: Credential Access
TA0007: Discovery
TA0008: Lateral Movement
TA0009: Collection
TA0011: Command and Control
TA0010: Exfiltration
TA0040: Impact

Esta aglutinación se va a llevar a cabo de modo que se considerará positiva una variable cuando se detecte en la observación la presencia de una técnica asociada. Si bien es cierto que puede darse el caso en el que una observación cuente con gran número de técnicas y el valor sea el mismo (1) que en el caso de que únicamente se hubiera detectado una técnica, es destacable que la investigación va focalizada a detectar la presencia o ausencia de las tácticas en un determinado ataque, siendo indiferente se esa táctica se ha logrado mediante la ejecución de una o varias técnicas.

Se va a obtener un *dataframe* con 750 observaciones de cada una de las 14 variables que representan las columnas.

A continuación, se analiza la correlación y varianza de esas 14 variables seleccionadas.

2.2.2.3.1 **Análisis de la correlación de las variables**

El análisis de la correlación de las variables es un paso fundamental que va a permitir determinar cómo se relacionan entre sí las diferentes variables. De forma ideal, las variables debieran ser relevantes o relacionadas con las variables objetivo, e independientes entre sí. Para el caso concreto de la investigación ha de tenerse presente que la variable (o táctica del ciberataque) objetivo a determinar puede cambiar en función del interés del investigador que emplee el modelo. Así pues, cuando dos tácticas están altamente correlacionadas, pueden ser redundantes y la inclusión de ambas en el modelo puede llevar a un sobreajuste, de modo que se seleccionará y mantendrá de entre los pares identificados, aquella que tenga una mayor frecuencia de unos. Esto se realizará para aportar más información, reducir el ruido que provocan las entradas nulas y mejorar el posible desbalanceo.

Para este análisis se va a emplear el coeficiente ϕ (coincidente con el coeficiente de correlación de Pearson al tratarse de variables binarias y con valores prácticamente idénticos al uso de ϕ_c) [106]. Se analizan los pares de variables con una mayor correlación marcando un umbral de 0,7 para determinar una correlación fuerte.

Se observa que las variables TA003 y TA004 (véase Figura 33) presentan una fuerte correlación de 0,89, por lo que se procederá a eliminar la que presente menor frecuencia, en este caso TA004, que tiene presencia en el 67,69% de las observaciones, mientras que

la TA003 está presente en el 70,76% de las observaciones. Así pues, el dataset obtenido estará conformado por 13 variables y 750 observaciones.

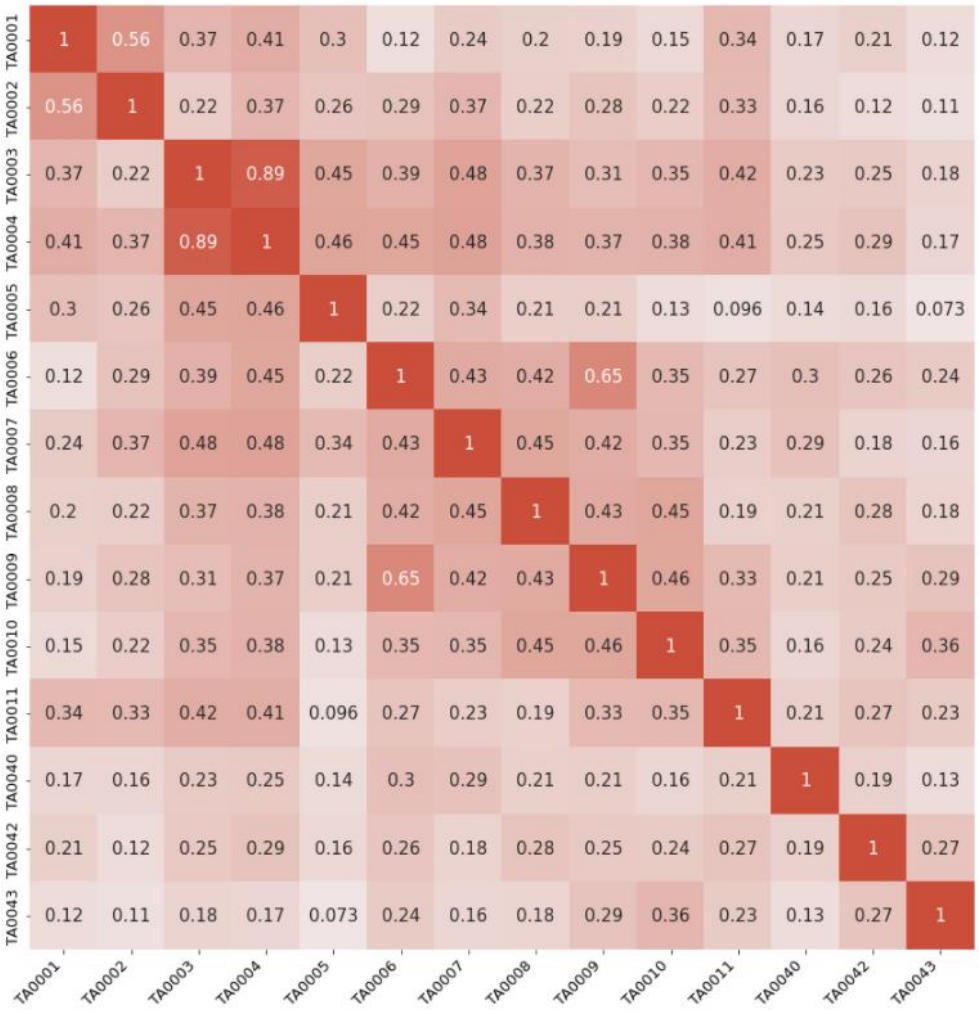

Figura 33. Matriz de correlación de las variables analizadas

2.2.2.3.2 **Análisis de la varianza de las variables**

La identificación de variables con poca varianza se llevará a cabo para identificar aquellas que no están contribuyendo significativamente a la variabilidad en los datos. Para estos datos, teniendo presente el carácter binario de las variables, la varianza se va a calcular como $p * (1 - p)$, donde p es la proporción de unos en esa variable.

Para el caso concreto de la investigación, se va a buscar identificar variables cuya varianza sea al menos igual al 82% de la varianza máxima de una variable binaria en ese *dataset* (con p = 0.82), por lo que, en este caso, el umbral de varianza es 0.1476.

Se observa que las variables TA005, TA008, TA042, y TA043 (véase Figura 34) presentan una varianza menor al umbral marcado, por lo que se procederá a eliminarlas.

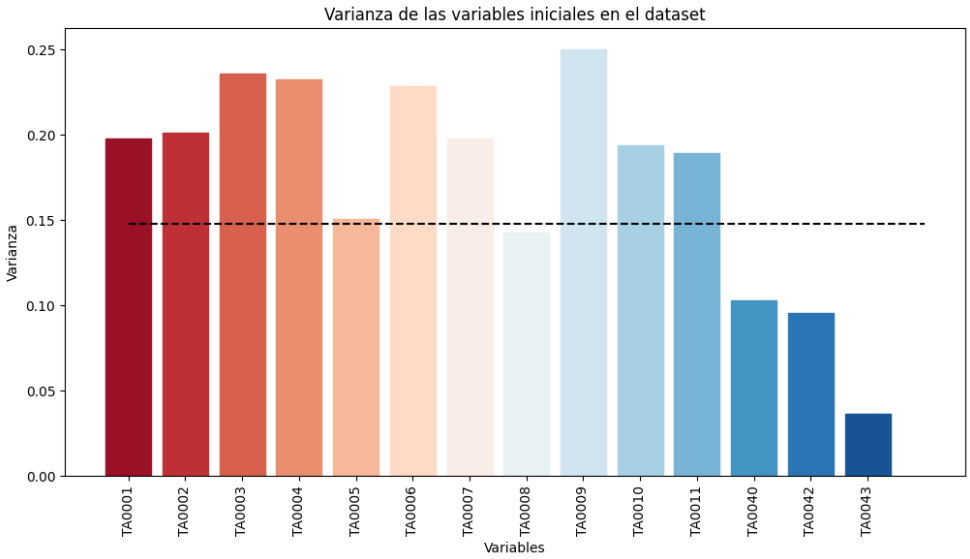

Figura 34. Varianza de las variables iniciales

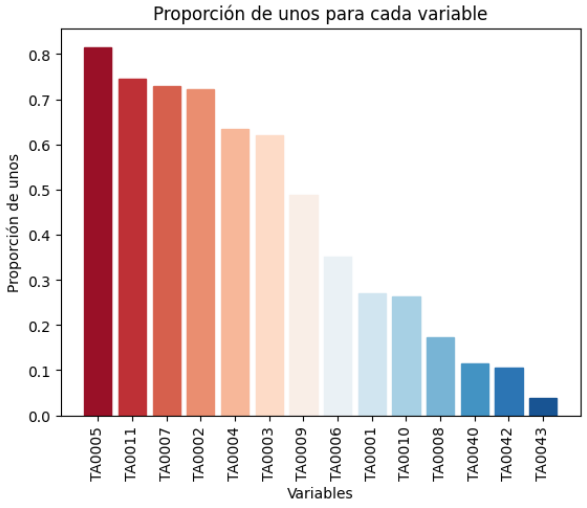

Figura 35. Frecuencia de las variables en el dataset final

3 Resultados

Mediante la aplicación del método propuesto se obtiene un *dataset* (véase Tabla 22) de carácter binario (unos y ceros) que permite representar observaciones asociadas a entidades detectadas (grupo o software). El número de observaciones registradas es de 750, el número de variables es de 9 ('TA0001', 'TA0002', 'TA0003', 'TA0005', 'TA0006', 'TA0007', 'TA0009', 'TA0010', 'TA0011'), y la distribución de frecuencias de positividad es la que se representa en la Figura 35.

Tabla 22. Dataset obtenido tras la aplicación de la metodología

	TA001	TA002	TA003	TA005	…	TA011
Observación 1	1	0	0	0		0
Observación 2	0	1	0	0	.	0
Observación 3	0	0	1	0	.	0
Observación 4	0	1	0	1	.	0
…		.	.	.		0
Observación 750	0	0	0	0		1

4 Conclusiones

La ingeniería de datos y el posterior tratamiento de datos ha permitido obtener un dataset idóneo, muy novedoso, y de carácter binario, que va a permitir aplicar modelos de predicción o clasificación de sus variables mediante técnicas de inteligencia artificial empleando la vectorización de los datos Mitre Att&ck. El análisis de los métodos de codificación ha puesto de relieve que *one-hot* es el método más idóneo, por su simplicidad, interpretabilidad y, en especial por la capacidad que tiene de agrupar varias técnicas en un único vector de forma coherente. Así mismo, el análisis de la correlación y varianza de las variables se ha demostrado de utilidad para reducir dimensiones, pero representando el fenómeno con suficiente fidelidad.

Pese a ello, el análisis en detalle de los datos obtenidos pone de relieve la necesidad de intervención para poder corregir los desbalanceos y, en especial, el reducido número de observaciones para la ulterior aplicación del algoritmo seleccionado en el modelo o modelos a desarrollar.

Capítulo 6. Estudio del incremento y corrección de los datos

1 Introducción

Los conjuntos de datos binarios presentan ciertas ventajas para la posterior aplicación de algoritmos de aprendizaje automático, como la fácil interpretación de la información, la eficiencia computacional al trabajar con unos y ceros, y la fácil traslación a problemas prácticos de la vida real. Sin embargo, ha de tenerse en cuenta que estos datos suelen tener una alta dimensionalidad y a menudo están desequilibrados o desbalanceados. Así pues el desabalanceo de datos puede generar sesgos que conduzcan a una ejecución errónea del algoritmo que se aplique posteriormente [107].

Mitre Att&ck [15] es una base de conocimientos de ciberinteligencia basada en una matriz que muestra las técnicas de ataque utilizadas para violar una red o un sistema y las tácticas o pasos utilizados para conseguirlo, así como las técnicas específicas. Tiene muchas ventajas, pero una de sus limitaciones es la escasez de datos sobre algunos de los vectores de ataque. Esto se debe principalmente a que el marco lleva aplicándose relativamente poco tiempo al haberse implantado en el año 2015 y, en especial, a que la recopilación de datos sobre ciberataques a veces puede resultar compleja, fundamentalmente si se trata de datos de ataques reales en los que las organizaciones afectadas son reacias a mostrar las deficiencias que llevaron al éxito de un determinado ataque. En consecuencia, la escasez de datos puede hacer que un modelo de aprendizaje automático basado en la matriz Mitre Att&ck sea menos eficaz para predecir y defenderse de futuros ataques, ya que, si los patrones de ataque no se observan con suficiente frecuencia, es menos probable que se desarrollen herramientas y medidas de defensa eficaces en respuesta a los resultados mostrados por el algoritmo. Así, las técnicas de ataque menos conocidas pueden volverse más eficaces, ya que los equipos de defensa (*Blue Teams*) no están preparados para ellas, creando una especie de invisibilidad en beneficio de los ciberdelincuentes.

El objetivo marcado en este capítulo es la obtención de datos adecuados, en cantidad, calidad y con suficiente fidelidad a la realidad, para la posterior aplicación de algoritmos de aprendizaje automático a un conjunto de datos binarios que muestran las acciones de ataques en Mitre Att&ck.

2 Los datos desbalanceados

Cuando un conjunto de datos presenta una distribución no equilibrada de los datos entre las diferentes clases se considera que existe desbalanceo en ese conjunto. Este fenómeno presenta especial incidencia en los problemas de clasificación de muestras, dado que la subrepresentación de una determinada clase puede generar sesgos que devengan en una errónea clasificación [107]. Por ejemplo, en un *dataset* de muestras de correos electrónicos para identificar aquellos que sean tipo *phishing*, puede haber muchas más comunicaciones deseadas que no deseadas (es lo habitual). Si el modelo se entrena en un *dataset* desbalanceado, puede tender a clasificar erróneamente la mayoría de los correos como deseados, por ejemplo, con unos valores de 1.000.000 de VN (o verdaderos negativos) y 20 VP (o verdaderos positivos) en la diagonal de la matriz de confusión, lo

que haría que el modelo no fuera útil en la detección de *phishing* al presentar un rendimiento muy elevado e irreal. Por otro lado, en problemas de regresión, el impacto del desbalanceo no es tan marcado, lo que no impide que se realicen predicciones erróneas.

No obstante, ha de tenerse presente que la corrección del equilibrio en los datos no siempre devendrá en un mejor rendimiento del modelo [108], por lo que es imprescindible analizar los resultados obtenidos con la aplicación de técnicas de balanceo y sin ellas. Así mismo, el incremento de muestras puede generar un incremento del tiempo de entrenamiento y causar sobreajuste del modelo [109].

Para corregir el problema, existen varias técnicas de remuestreo, tanto desde el punto de vista de acciones sobre los datos como sobre los algoritmos, como son:

Sobremuestreo de la clase minoritaria. Esta técnica se puede ejecutar, por ejemplo, duplicando de forma aleatoria muestras existentes de la clase minoritaria o también mediante la generación de datos sintéticos mediante técnicas como SMOTE (*Synthetic Minority Over-sampling Technique*) [110], que crea nuevos ejemplos sintéticos para la clase minoritaria del *dataset* mediante interpolaciones en lugar de duplicar los datos (véase Figura 36), o ADASYN (*Adaptive Synthetic Sampling Approach for Imbalanced Learning*) [111], que crea nuevos ejemplos sintéticos para la clase minoritaria, pero que a diferencia de otros métodos de sobremuestreo, adapta el nivel de sobremuestreo para cada ejemplo concreto de la clase minoritaria.

Así mismo se han desarrollado variantes de SMOTE, como Borderline SMOTE [112], que se utiliza específicamente en conjuntos de datos en los que la clase minoritaria está en la frontera (*borderline*) entre la clase mayoritaria y la clase minoritaria, tal y como se observa en la Figura 37. Esta técnica genera muestras sintéticas cerca de la frontera de la clase minoritaria, lo que ayuda a mejorar la precisión del modelo. Basa su funcionamiento en diferenciar tres tipos de instancias: *Danger*: o instancias de frontera entre clases. Es aquí donde se generan nuevos valores; *Noise*: o instancias muy alejadas de la frontera y que no contribuyen, y que son eliminadas; y *Safe*: instancias moderadamente lejos de la frontera y que no requieren instancias adicionales.

El objetivo por tanto es aumentar el volumen de datos de la clase con menor presencia y equilibrar así el *dataset*. No obstante, ha de tenerse presente al aplicar estas técnicas el aumento del coste computacional, y, en especial, el riesgo de sobreajuste del modelo si los datos no tienen en cuenta la distribución original incrementar.

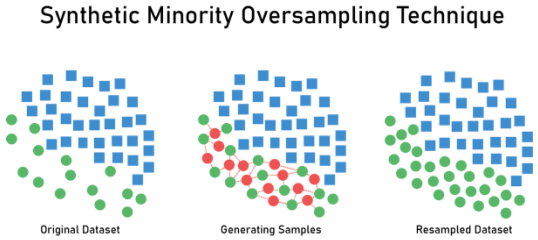

Figura 36.Generación de instancias mediante SMOTE. Fuente: Emilia Orellana

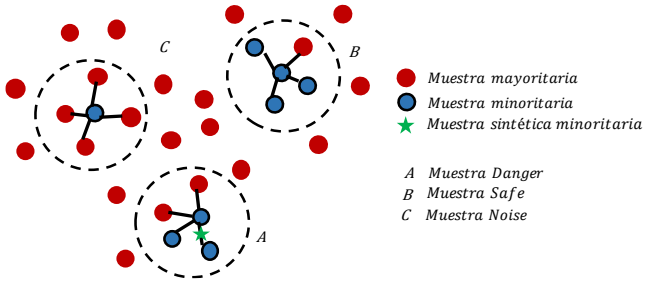

Figura 37.Tipologías de instancias que se generan mediante Borderline SMOTE

Submuestreo de la clase mayoritaria: Esta técnica implica la eliminación de algunas de las muestras de la clase mayoritaria para forzar el equilibrio. Para ello se puede emplear la eliminación aleatoria, basada en *Near Miss* [113] o *Tomek links* [114], [115], que localiza pares de diferentes clases que se encuentren muy cerca para eliminar el ejemplo de la clase mayoritaria en cada par. Por último, *Neighborhood Cleaning Rule* (NCR) [116], es un algoritmo de aprendizaje basado en instancias que reduce el tamaño del conjunto de entrenamiento eliminando instancias ruidosas e instancias de clase mayoritaria que se solapan con instancias de clase minoritaria, mejorando el rendimiento de la clasificación.

Es importante tener en cuenta que, si una técnica funciona bien en un conjunto de datos desequilibrado, no significa necesariamente que vaya a funcionar en un conjunto de datos diferente [117], así como que eliminar ejemplos de la clase mayoritaria puede derivar en la pérdida de información relevante que reduzca la capacidad del modelo para generalizar resultados a nuevos datos.

Combinación de sobremuestreo y submuestreo: Esta técnica implica aplicar de forma conjunta las dos técnicas anteriores a los datos. Para ello se sobremuestrea la clase minoritaria y se submuestrea la clase mayoritaria al mismo tiempo. Esta aplicación conjunta permite combinar los beneficios y mitigar las desventajas. Entre otras opciones, existe la combinación de SMOTE y *Tomek links*, denominada SMOTE-Tomek [118], que combina estas dos técnicas para abordar el desequilibrio de clases en un conjunto de datos. Primero, se aplica TOMEK para eliminar las muestras redundantes entre las clases y posteriormente se aplica SMOTE para la generación de nuevas muestras sintéticas para la clase minoritaria.

Selección de características: La selección de características o la agrupación de estas puede realizarse mediante la elección de un subconjunto relevante de características del *dataset*. Esto puede ayudar a reducir la dimensionalidad del problema y mejorar la precisión del modelo.

Ajuste de pesos: En determinados algoritmos de regresión, es posible ajustar los pesos de las muestras para dar más importancia a las muestras de la clase minoritaria. Por ejemplo, en la regresión logística, se pueden utilizar los pesos inversamente proporcionales a la frecuencia de las clases para corregir el desbalanceo.

Es importante tener en cuenta que estas técnicas no siempre mejoran el rendimiento del modelo y pueden depender del problema específico y del *dataset* en cuestión. Por lo tanto, es recomendable experimentar con varias técnicas y evaluar el rendimiento del modelo

en un conjunto de validación antes de seleccionar la técnica adecuada para corregir el desbalanceo. Existen muchos métodos para el tratamiento de los datos desequilibrados, pero el estudio habrá de centrarse en determinar la técnica concreta que será de utilidad -o generar una nueva- de acuerdo con la necesidad existente en cada caso.

3 La separabilidad de los datos

La separabilidad de los datos hace referencia a la capacidad de diferenciar o poder separar las distintas clases en un espacio de características, de modo que si dos o más clases no son fácilmente separables puede llegar a ser extremadamente complejo para un modelo de *Machine Learning* distinguir entre una y otra con precisión.

En muchos casos se plantean problemas que no ofrecen una clara separación lineal entre las clases del *dataset*. En estas ocasiones, el uso de modelos lineales simples, como por ejemplo SVM o la regresión logística, pueden no ser adecuados, ya que su frontera de decisión entre clases es una línea (o un plano en su caso) y no van a poder capturar relaciones complejas entre las clases de forma correcta [119].

Generalmente el origen de este problema de no separabilidad suele estar asociado con el ruido, la presencia de características insuficientes o irrelevantes, o la complejidad y superposición de las clases.

Para abordar la no separabilidad de las clases se pueden utilizar varias estrategias ya empleadas en la investigación:

Transformación de características: Pueden emplearse técnicas como el Análisis de Componentes Principales (PCA) o el Análisis Discriminante Lineal (LDA).

Uso de modelos no lineales: Algoritmos como máquinas de soporte vectorial con *kernels* no lineales, redes neuronales o árboles de decisión pueden capturar relaciones no lineales y separar clases que no son linealmente separables.

Incorporación de características adicionales: La incorporación de características adicionales e información de contexto puede ayudar a la separabilidad de las clases.

4 Los datos escasos

Un conjunto de datos escaso puede representar un desafío a la hora de afrontar un problema de *Machine Learning*. Es recomendable disponer de muestras suficientes, por lo que caso de no tener suficiente volumen es recomendable recopilar o crear nuevos datos que representen la variabilidad. En estas acciones se ha de tener presente la distribución inicial de los datos para evitar sobreajustes, sesgos, reducir la incertidumbre en las estimaciones, así como la dificultad de detección de patrones.

No existe una única herramienta para abordar el problema de la escasez de datos, pero se muestran a continuación una serie de posibles soluciones:

Recopilación adicional de datos: caso de ser posible, la mejor opción es incrementar el volumen de datos, bien mediante la recogida manual (ej: encuestas, cuestionarios, etc.) o automatizada (ej: *scraping*) de diversas fuentes fiables y de calidad. Así mismo puede contemplarse la reutilización de datos mediante transferencia de aprendizaje, de modo

que los conocimientos adquiridos en una tarea se reutilizan para mejorar el rendimiento en otra tarea relacionada [120].

Generación de datos sintéticos: Esta técnica conlleva la generación de nuevos datos sintéticos o artificiales a partir de los datos iniciales. Puede ser de utilidad cuando no es posible recopilar más datos de forma manual o automatizada. Actualmente se emplean los GANs (*Generative Adversarial Networks*) [121], que son redes compuestas por un generador y un discriminador. El discriminador intenta clasificar correctamente los datos como reales o sintéticos, mientras que el generador intenta engañar al discriminador para que clasifique sus salidas sintéticas como reales. Por otro lado, pueden emplearse métodos como el enmascaramiento o *masking*, basado en la ocultación o eliminación de ciertos valores o características para mejorar la calidad y la precisión del análisis de datos o del modelo de aprendizaje automático. Este método, en principio empleado para ocultar determinados valores principalmente por razones de protección de datos puede tener su utilidad al crear una versión de los datos que se ve estructuralmente similar al original, pero oculta (enmascara) información desconocida por el investigador y en caso de tener pocos datos puede provocar *overfitting*.

Por otro lado, las técnicas *one shot learning* [122] ofrecen un rendimiento muy interesante con un volumen de muestras muy reducido.

Aunque para el caso particular de un conjunto de datos relacionado con Mitre Att&ck no se han encontrado soluciones específicas, para abordar el problema genérico de un conjunto de datos disperso o desequilibrado se han desarrollado varias soluciones en otros campos. Entre ellas se encuentra ROSE (*Random Over-Sampling Examples*) [123] que se basa en *Bootstrap* y se centra en el sobremuestreo de datos aleatorios para igualar el número de instancias de las dos clases en el conjunto de datos de entrenamiento, y también incluye otras características, como la capacidad de realizar validación cruzada y selección de características. Por otro lado, [124] aborda el problema desde un punto de vista holístico, actuando sobre el preprocesamiento de datos y el ensamblaje de algoritmos en el contexto de intrusiones en redes y sistemas de información. En [125] se adaptan dos de los métodos más conocidos (SMOTE y submuestreo) a tareas de regresión y en [126] se discute en detalle la posible aplicación de técnicas de submuestreo.

5 La casuística del *dataset* objeto de estudio

Los datos analizados se han obtenido tras la manipulación de la información contenida en el repositorio oficial "*Working with Mitre Att&ck"* [100], de forma que se han creado variables en función de las Tácticas detectadas en un ataque, indicando con "1" su presencia y con "0" su ausencia, tal y como se ha explicado en el capítulo previo. Estos datos, una vez manipulados para su procesamiento, muestran una fuerte disparidad en cuanto a la frecuencia de positividad en las variables seleccionadas para el estudio, con una variabilidad entre el 20% y el 80% aproximadamente, como puede observarse en la Figura 38. Además, el número de observaciones del conjunto de datos inicial es de sólo 750, número que pese a poder ser suficiente no deja de ser escaso.

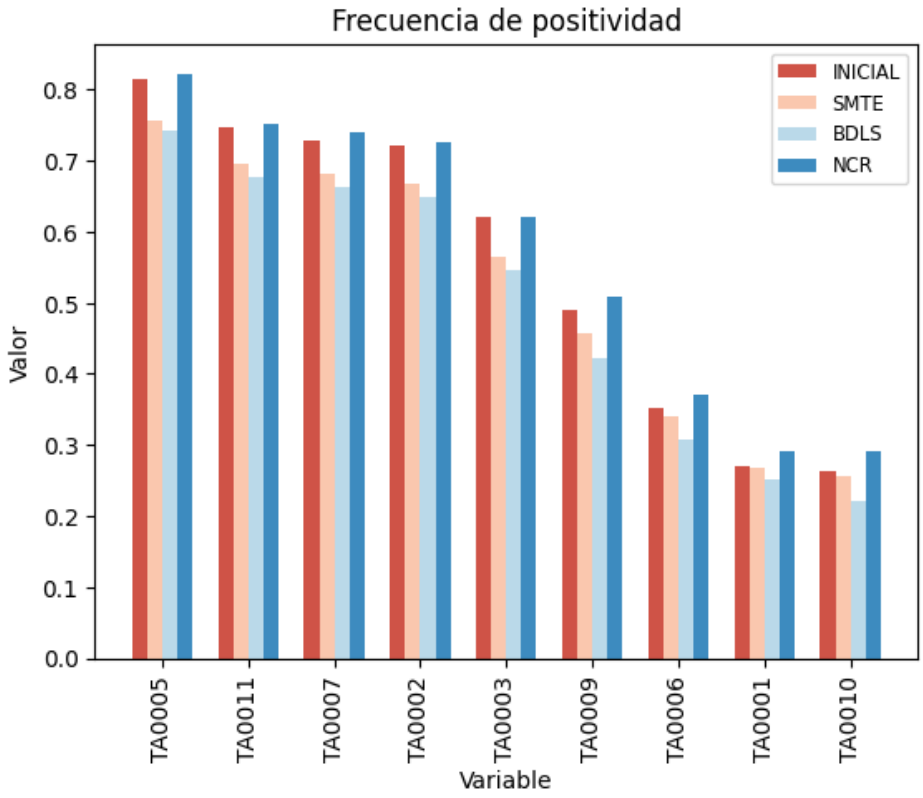

Figura 38. Positividad de cada variable tras el tratamiento

6 Hipótesis de trabajo

Para corregir el desequilibrio y el reducido número de observaciones en el conjunto de datos binarios extraídos de "*Working with Mitre Att&ck*", en este capítulo se considera apropiado explorar la aplicación de varias técnicas de remuestreo y generación sintética que pueden abordar los posibles desequilibrios entre clases, así como aumentar el volumen de observaciones. Basándose en esto, se plantea la hipótesis de que existe una diferencia significativa en los resultados al corregir y aumentar un conjunto de datos binarios utilizando diferentes técnicas, y que una de las técnicas (SMOTE, Borderline SMOTE o NCR) produce un mejor rendimiento en términos de precisión y de cantidad de nuevas observaciones generadas, en comparación con las demás.

Derivado de esta hipótesis, el objetivo de la investigación es comparar la eficacia de las técnicas de sobremuestreo SMOTE, Borderline SMOTE y la estrategia de limpieza NCR para mejorar el rendimiento de los modelos de clasificación en conjuntos de datos binarios no equilibrados. Para ello, se comparará la precisión de los modelos de clasificación, tras la aplicación de cada método, con la precisión y el AUC de los modelos entrenados sobre los datos originales no corregidos, con el objetivo de aumentar el número de observaciones y corregir el desequilibrio de clases.

Para probar la hipótesis, se propone una arquitectura basada en los siguientes puntos:

1. Preprocesamiento de datos
2. Aplicar el método propuesto
3. Selección del modelo

4. Formación y evaluación de modelos
5. Comparación de técnicas
6. Interpretación de los resultados y conclusiones

7 Método propuesto

El método propuesto consiste en la ejecución progresiva de dos fases que permitirían corregir y aumentar los datos binarios. Para el desarrollo y análisis del método propuesto se ha utilizado el lenguaje de programación *Python* y la librería *Scikit learn* [127], que se utilizarán sobre un conjunto de datos (previamente manipulado y adaptado) obtenido del repositorio oficial "*Working with Mitre Att&ck*" [100].

- **FASE I. Selección de características (Fase preliminar):** Para predecir la clase minoritaria, ciertas características pueden tener un mayor impacto en el algoritmo que otras. Por lo tanto, mediante este método se pueden seleccionar ciertas características relevantes del conjunto de datos para mejorar el rendimiento. Se utilizará un primer filtrado para agregar las variables y, a continuación, se utilizarán medidas estadísticas, como la correlación, tal y como hace [128] o la varianza.
- **FASE II. Aplicación de técnicas:** Se tratará de realizar un submuestreo de las clases mayoritarias y un sobremuestreo de las clases minoritarias aplicando tres posibles opciones: variante a) SMOTE; variante b) Borderline SMOTE; y variante c) Neighborhood Cleaning Rule.

En el primero de los casos se seleccionan los k-NN (*K-Nearest Neighbours*) conforme distancia euclídea y se escoge uno de ellos. A continuación, se opera con la diferencia entre el vector original y el seleccionado, multiplicando por un valor entre 0 y 1 y ajustando para que continúe siendo un valor binario el vector resultante conforme el *dataset* original.

En la segunda técnica se aplica lo anterior diferenciando las tres tipologías de instancias propias de la variante.

Finalmente, en la tercera técnica (NCR), se eliminarán aquellas instancias de la clase mayoritaria que están más cerca de las instancias de la clase minoritaria o de la frontera entre las clases.

Para la aplicación de estas técnicas se ha variado de forma secuencial la variable objetivo entre las nueve existentes, de modo que se ha ido analizando la ampliación o reducción del *dataset* para cada una de las variables y concatenado los datos generados, lo que explica que, en cada caso, independientemente de ser sobremuestreo o submuestreo, el número de valores va a incrementar.

8 Resultados

Mediante la aplicación de la propuesta en sus diferentes variantes, se observa un ajuste progresivo en los valores de clase. Así, en todas las variantes de la Fase II se observa que las técnicas de balanceo de datos (SMOTE, Borderline SMOTE y Neighborhood Cleaning Rule) mejoran la precisión media respecto a los datos originales, y en particular, la aplicación de la variante c) Neighborhood Cleaning Rule.

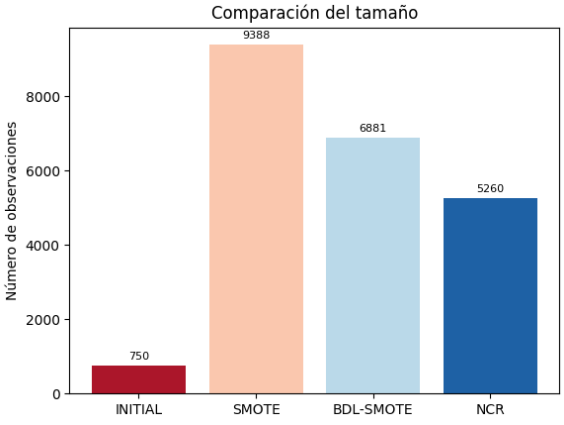

Figura 40. Número de observaciones de cada conjunto de datos después del tratamiento

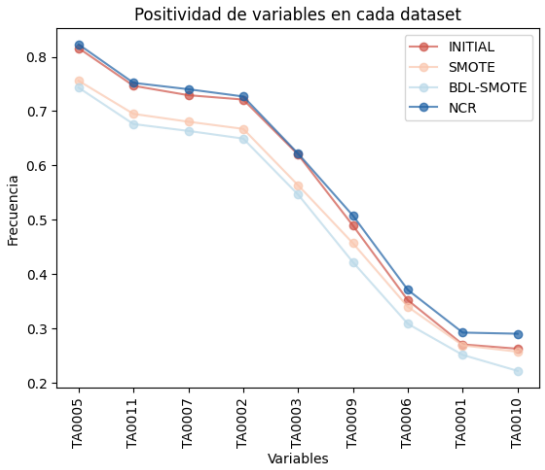

Figura 39. Positividad de cada conjunto de datos

En cuanto al incremento del número de observaciones (véase Figura 40) se observa que en las tres variantes es muy significativo, superando ampliamente el millar de observaciones en todos los casos, acercándose a las decenas de miles de observaciones en la primera variante SMOTE.

Respecto al rendimiento de cada variante, se observa que la variante c) es la que mejores resultados obtiene, con un porcentaje del 78% en la predicción de la precisión media de las variables y de un 81% de AUC. En relación con la positividad de las variables observadas (véase Figura 39), las dos primeras variantes reducen la positividad en aquellas variables con mayor presencia de unos, mientras que la variante c) mantiene e incluso aumenta ligeramente la positividad en las variables con mayor presencia de unos, pero por el contrario aumenta la positividad en aquellas variables con menor presencia de unos.

Para analizar las opciones mostradas y su utilidad, además de comprobar el sobremuestreo y submuestreo, se aplicará un algoritmo de regresión logística a cada una de las variables analizadas, tomando ésta como objetivo y el resto como predictoras. Parece interesante escoger aquella opción que ofrezca la precisión más elevada, pero que tenga resultados suficientemente homogéneos entre todas las variables analizadas. Por ello, se analizará también el índice de variación que pueda existir en las opciones (desviación estándar/media aritmética) para tratar de escoger el menor índice posible en conjunción con la mayor precisión.

En la Figura 41 pueden observarse los resultados obtenidos en relación con la precisión y el AUC para cada caso particular. Ha de tenerse presente que el algoritmo no se ha ajustado en lo referente a hiperparámetros, lo que puede tener gran influencia en el resultado. Así mismo, ha de contemplarse que los resultados con otro tipo de algoritmos (*Support Vector Machines*, Redes Neuronales, etc.) pueden ser superiores. En este sentido, dado que la variante c), NCR, ofrece los mejores resultados de precisión y AUC, y teniendo en cuenta que el aumento del volumen de observaciones es más que aceptable, se considera la mejor opción.

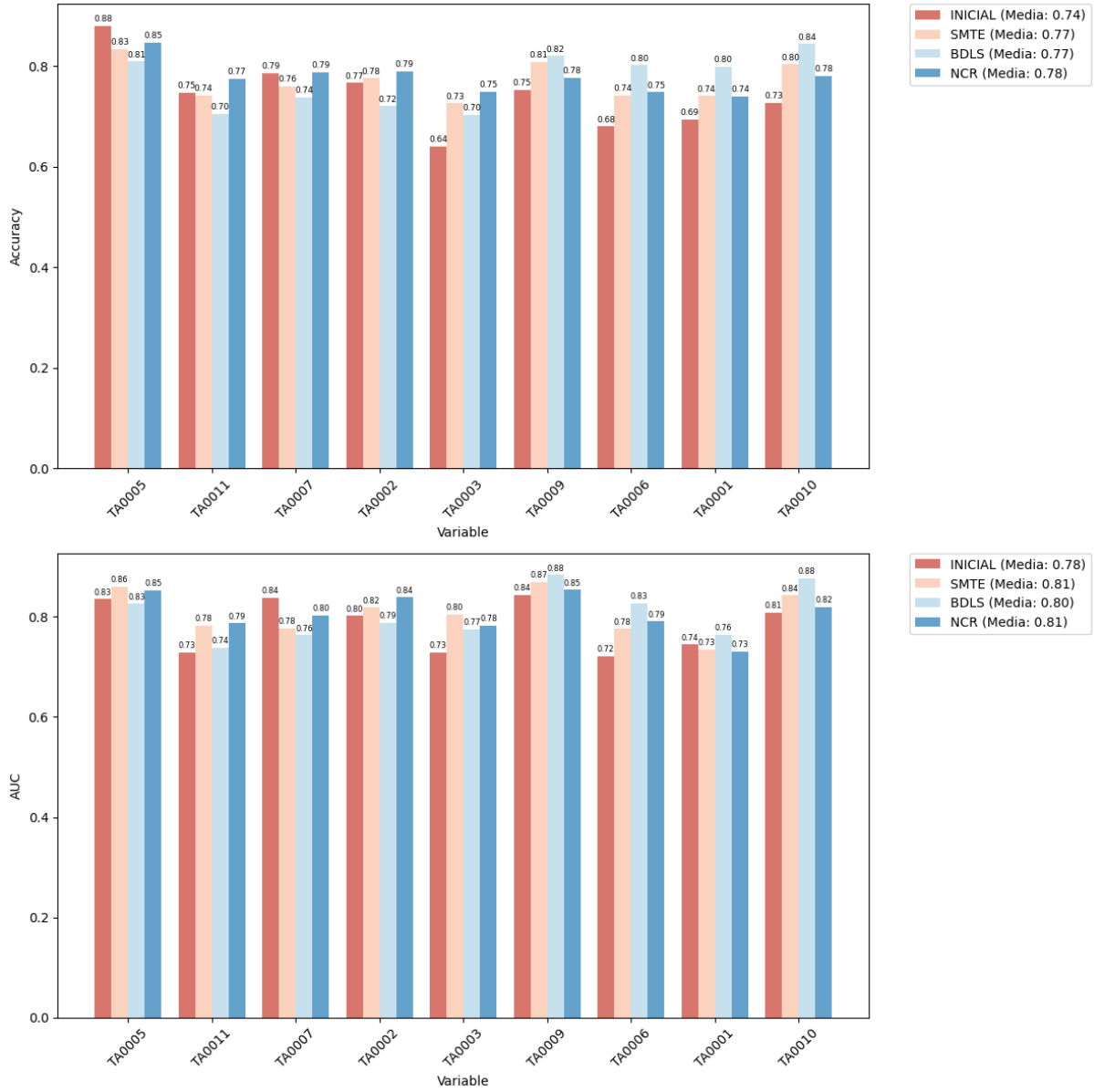

Figura 41. Comparativa de la precisión y AUC en cada una de las variables

9 Conclusiones

Tras llevar a cabo un análisis de varias opciones para el aumento de un conjunto de datos binarios en el marco Mitre Att&ck. Se observa que la aplicación de la regla de limpieza Neighborhood Cleaning Rule es el método que ofrece mejores resultados, ya que permite equilibrar y aumentar los datos manteniendo la homogeneidad en las distintas variables analizadas. Por otro lado, los resultados obtenidos han sido analizados utilizando un único algoritmo, y sin el ajuste de los propios hiperparámetros, lo que puede permitir que futuros trabajos aumenten la precisión reflejada en este análisis.

Capítulo 7. Aplicación y análisis de las estrategias y algoritmos de Machine Learning

1 Planteamiento del problema

Los ciberataques se encuentran en la actualidad en constante evolución y son cada vez más sofisticados y tecnificados, por lo que robustecer los sistemas de ciberdefensa e investigación permitirá adaptarse más rápidamente a las cambiantes estrategias de los ciberdelincuentes. En base a ello, el problema concreto que se plantea consiste en poder determinar la presencia o no de una determinada táctica o fase en un ciberataque cuando se dispone de información parcial por parte del investigador, tal y como sucede en la mayoría de las ocasiones en el marco de análisis forenses. En este modelo se consideran las tácticas como las acciones específicas dentro de un ataque.

Así pues, se dispone de un conjunto de observaciones que tienen asociado un vector de características, que son los indicadores binarios que indican la presencia o no de una determinada táctica Mitre Att&ck. De este modo, siendo T un vector de tácticas de longitud n que representa un ciberataque:

$$T = [TA_1, TA_2, \dots, TA_n]$$

El problema se puede reducir al hecho de que no todas las tácticas o elementos de vector T son conocidos para un determinado hecho. Habrá determinadas tácticas o elementos del vector que sí se hayan empleado (representadas con un 1), otras tácticas no se hayan empleado (representadas con un 0) y otras que se desconoce si pueden estar presentes o no. Por ello, ha de buscarse inferir los valores que puedan desconocerse, denominados "?" en base a los valores de los que sí se disponga información, de modo que será necesario aprender relaciones y patrones a partir de lo conocido para poder inferir valores razonables para un determinado ciberataque.

$$T = [1,0,?,0,\dots,?]$$

Por tanto, se busca determinar una táctica objetivo teniendo presente la posibilidad de que existan tácticas adicionales que no se conocen desde el inicio pero que se irán determinando a lo largo del modelo. Así pues, el objetivo a alcanzar es transformar estos vectores de manera que todos los elementos desconocidos (?) se reemplacen ya sea por 0 o 1, en función de si la táctica correspondiente se utilizó o no.

Este problema puede expresarse formalmente como el hecho de determinar una función f tal que $f(x_i) = y_i$ donde $x_i = [v_1, v_2, \dots, v_M]$ y por otro lado $y_i = [w_1, w_2, \dots, w_M]$ tomando valores w_j del conjunto $\{0,1\}$.

En el marco de las investigaciones forenses, se considera imprescindible que el analista disponga del mayor número posible de información valiosa, pero es crucial que no se obvie la posible existencia de una táctica en detrimento de sufrir en determinadas ocasiones falsos positivos. Es por ello por lo que se prestará atención especial a la métrica sensibilidad o *recall* para la evaluación, en tanto en cuanto se va a buscar minimizar los falsos negativos, pero sin obviar la información que aportan la precisión del modelo, el AUC (o Área bajo la curva) y el f-1 Score. Así pues, se va a crear para este modelo una métrica *ad hoc*, que puede ser una buena estrategia para tener una visión de conjunto, pero focalizando especialmente la evaluación en el *recall*. Se propone, por tanto:

$$\textbf{Métrica Compuesta (MC)} = w_1\textbf{Recall} + w_2\textbf{f-1 Score} + w_3\textbf{Precisión} + w_4\textbf{AUC}$$

Donde w son pesos que se asignan a cada una de las métricas según la importancia que se le quiere asignar, y que van a sumar la unidad. Para este caso, se toman: $w_1 = \frac{1}{2}$, $w_2 = \frac{1}{6}$, $w_3 = \frac{1}{6}$, $w_4 = \frac{1}{6}$; de modo que el *recall* tendrá el doble de peso del resto de métricas.

2 Propuesta

Para solventar el problema mencionado, y por tanto predecir el conjunto de tácticas de la cadena de ataque se plantea el uso de algoritmos de *Machine Learning* sobre la base del siguiente razonamiento:

Una vez definido el conjunto total de tácticas que pueden existir en un ataque como:

$$T = [TA_1, TA_2, \dots, TA_n]$$

El conjunto de tácticas conocidas en un determinado momento como:

$$U = [u_1, u_2, \dots, u_b]$$

El conjunto de tácticas desconocidas en un determinado momento como:

$$D = [d_1, d_2, \dots, d_a]$$

$$D \subseteq T, D \cap U = \emptyset$$

Y el número de tácticas desconocidas en un determinado momento como a, donde $a \in \{0, 1, 2, 3\}$, que lleva a definir $b = n - a$, siendo b es el número de tácticas conocidas y n el número total de tácticas.

Se parte de un conjunto de datos de entrada X que es el subconjunto de tácticas de T que solo va a incluir las U tácticas conocidas y se busca obtener el conjunto de tácticas desconocidas D o conjunto de datos de salida Y.

Para poner en marcha estos algoritmos se optará por el estudio de dos posibles estrategias:

2.1 Estrategia a): Classifier Chains (o cadenas de clasificadores) CC

Se considerará la permutación θ de las posibles etiquetas de $D = Y = [d_1, d_2, \dots, d_a]$, correspondiente con el vector D a la par que se define el conjunto $U = X = [u_1, u_2, \dots, u_b]$ correspondiente con el vector U.

Para esta posibilidad, se entrena una secuencia de clasificadores $C_1, C_2, \dots, C_k$, donde cada clasificador C_i opera con el conjunto de datos de entrada X y la etiqueta y_i (que es la i-ésima etiqueta en el orden θ). A continuación, para cada $y > 1$, el clasificador C_i se entrena con el conjunto de datos de entrada, pero ampliado con las predicciones de los clasificadores previos: $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{i-1}$.

Así pues, dada una nueva muestra x, la predicción $\hat{y}$ se realiza secuencialmente mediante el uso de los clasificadores $C_1, C_2, \dots, C_k$ en el orden θ.

De cara a determinar el mejor orden posible de las etiquetas, θ^*, se realiza un proceso de búsqueda entre todas las permutaciones posibles, de modo que se va a determinar el modelo con el mayor rendimiento posible.

$$\theta^* = argmax_\theta MC(\theta)$$

Donde $MC(\theta)$ es la métrica compuesta calculada para el orden de etiquetas.

$$MC(\theta) = \sum_{i=1}^{l} w_i M_i(\theta)$$

Donde, ha de recordarse que:

$$\sum_{i=1}^{l} w_i = 1$$

Así pues, se buscará determinar el conjunto de clasificadores asociados a un determinado algoritmo (Support Vector Machines, Regresión Logística, Decision Tree, AdaBoost o Random Forest) que, para cada número de variables ausentes o táctica ofrezcan la mejor métrica compuesta en su conjunto.

> **Ejemplo**: Considerando en un momento específico el conjunto de tácticas desconocidas $D = Y$:
>
> $$D = [TA_1, TA_2, TA_6]$$
>
> El número de tácticas conocidas es $b = 9 - 3 = 6$, por lo que el modelo se alimenta con $X = U = [TA_3, TA_4, TA_{5,}, TA_7, TA_8, TA_{9,}]$, y siendo $Y = [TA_1, TA_2, TA_6]$.
>
> Se definen los clasificadores C_1, C_2, C_3 como:
>
> - C_1 se emplea para la predicción $\hat{y}_1$, esto es TA_1, (empleando en su caso $\hat{y}_2$ o $\hat{y}_3$ como entrada adicional)

- C_2 se emplea para la predicción de $\hat{y}_2$, esto es TA_2, (empleando en su caso $\hat{y}_1$ o $\hat{y}_3$ como entrada adicional)
- C_3 se emplea para la predicción de $\hat{y}_3$, esto es TA_3, (empleando en su caso $\hat{y}_1$ o $\hat{y}_2$ como entrada adicional)

Para determinar el mejor orden posible para realizar las predicciones, se emplearán los algoritmos indicados y se considerará como el orden óptimo aquel que produzca la mayor métrica conjunta MC.

Las posibles cadenas posibles serán, por tanto:

1. $C_1 \rightarrow C_2 \rightarrow C_3$
2. $C_1 \rightarrow C_3 \rightarrow C_2$
3. $C_2 \rightarrow C_1 \rightarrow C_3$
4. $C_2 \rightarrow C_3 \rightarrow C_1$
5. $C_3 \rightarrow C_1 \rightarrow C_2$
6. $C_3 \rightarrow C_2 \rightarrow C_1$

Al final del proceso, se seleccionará una cadena de tres clasificadores y un algoritmo, el cual será el que haya demostrado proporcionar la mejor métrica conjunta MC en comparación con los demás. Este enfoque garantiza que se ha realizado una búsqueda exhaustiva para encontrar la mejor combinación de cadena de clasificadores y algoritmo para predecir las tácticas desconocidas D a partir del conjunto conocido U.

2.2 Estrategia b): One-Vs-Rest (o uno contra todos) OVR

Siendo $U = X = [u_1, u_2, \dots, u_b]$ el conjunto de datos de entrada y C el clasificador base, para cada táctica desconocida d_i en $D = Y = [d_1, d_2, \dots, d_a]$. Durante el entrenamiento, la táctica específica d_i se etiqueta como "1" (clase positiva), y todas las demás tácticas en D que no son d_i se van a etiquetar como "0" (clase negativa).

$$C_i : X \rightarrow \{0,1\}$$

Donde:

$C_i(X) = 1$ si la táctica es d_i

$C_i(X) = 0$ si la táctica no es d_i y es cualquier otra dentro de D

Para una nueva muestra x del conjunto de datos X, cada clasificador C_i hace una predicción.

$$\hat{y}_i = C_i(X);\ i = 1,2, \dots, a$$

Donde $\hat{y}_i$ es la predicción del clasificador C_i para la muestra x. La clase final predicha $\hat{d}$ será aquella para la cual el clasificador correspondiente va a tener la mayor confianza.

$$\hat{d} = arg\max_i C_i(X)$$

Así pues, se buscará determinar el conjunto de clasificadores asociados a un determinado algoritmo (Support Vector Machines, Regresión Logística, Decision Tree, AdaBoost o

Random Forest) que, para cada número de variables ausentes o táctica, ofrezcan la mejor métrica compuesta en su conjunto.

> **Ejemplo**: Considerando en un momento específico el conjunto de tácticas desconocidas $D = Y$:
>
> $$D = [TA_1, TA_2]$$
>
> El número de tácticas conocidas es $b = 9 - 2 = 7$, por lo que el modelo se alimenta con $X = U = \left[TA_3, TA_4, TA_{5,}, TA_{6,} TA_7, TA_8, TA_{9,}\right]$, y siendo $Y = [TA_1, TA_2]$.
>
> Se definen los clasificadores C_1, C_2 en el siguiente sentido:
>
> - C_1 se emplea para la predicción $\hat{y}_1$, esto es TA_1, con los datos de entrada $X = U$. Así pues, durante el entrenamiento, TA_1 se etiqueta como positivo y todas las demás tácticas, incluyendo TA_2 se etiqueta como negativo.
> - C_2 se emplea para la predicción de $\hat{y}_2$, esto es TA_2, con los datos de entrada $X = U$. Así pues, durante el entrenamiento, TA_2 se etiqueta como positivo y todas las demás tácticas, incluyendo TA_1 se etiqueta como negativo.
>
> Al final del proceso, se seleccionará el algoritmo que haya demostrado proporcionar la mejor métrica conjunta MC en comparación con los demás.

3 Metodología

Para la elaboración del modelo matemático, y tras fijar las dos estrategias y los cinco algoritmos a analizar, se procederá a entrenar y testear sobre el conjunto de prueba. Para ello se aplicará la técnica *cross validation*, que se empleará para el análisis de los resultados que ofrecerá el modelo de *Machine Learning*, de tal modo que pueda garantizarse que los datos del entrenamiento y la validación son independientes y puedan generalizarse en otras observaciones. Para reducir la variabilidad, se llevan a cabo múltiples rondas de *cross validation* con diversos grupos y se promedian los resultados obtenidos.

Los distintos tipos de *cross validation* que se emplean más comúnmente en la actualidad son *k-fold*, *Repeated K-Fold*, *Leave One Out (LOO)* y *Leave P Out (LPO)*. No obstante, uno de ellos es empleado por encima del resto y será el escogido para la investigación, el *k-fold*. El *k-fold* divide los datos del entrenamiento en k grupos para validar el modelo en k-1 grupos un total de k ocasiones. Posteriormente el error obtenido se promedia y se llama error de validación cruzada. Generalmente el valor de k es de 5 o 10, lo que genera grupos del 20% o 10% de muestras. En esta investigación se empleará un valor k de 5.

Previamente a ejecutar los algoritmos, se ha llevado a cabo un *hyperparameter tunning* mediante el método *grid search* siguiendo el flujo de trabajo reflejado en la Figura 42, que ha arrojado los hiperparámetros mostrados en los siguientes párrafos para cada algoritmo. Es reseñable que, de cara a reducir el coste computacional, se va a realizar la búsqueda de forma aleatoria en vez de ejecutar una evaluación de todas las combinaciones de hiperparámetros. Así mismo se reducirá en la medida de lo posible a un conjunto finito y manejable las opciones que ofrecen los hiperparámetros que permitirá tener unos valores en escasos minutos.

Support Vector Machines (SVM): *C: 1.0, gamma: 0.01, kernel: 'rbf'*. Es decir, C=1.0 es un valor típico y es el valor predeterminado en muchas implementaciones de SVM, empleando también RBF o Radial Basis Function, uno de los núcleos más comunes, y un valor de gamma que puede ayudar a evitar el sobreajuste

Regresión Logística (RL): *penalty='l2', C=1.0, solver='lbfgs', max_iter=1000, class_weight='balanced'*. Es decir, una regresión logística que va a emplear una regularización L2 *Ridge* (que va a penalizar coeficientes grandes pero que no los anula necesariamente), un valor de C=1 y un *solver* como método adecuado para problemas de tamaño mediano. A su vez incluye una limitación máxima de iteraciones y un ajuste automático de pesos.

Decision Tree (DT): *min_samples_split=5, min_samples_leaf=1, max_depth=10.* Es decir, que un nodo tendrá al menos cinco muestras para que se considere para ejecutar una división, que no hay límite impuesto en el número mínimo de muestras que debe tener un nodo hoja (al menos una muestra) y que la profundidad máxima o camino del árbol será de 10 para evitar el sobreajuste. Por defecto se empleará el Índice de Gini como método de toma de decisión.

AdaBoost *(AB): learning_rate: 1.0, n_estimators=150, min_samples_split': 5, min_samples_leaf': 1, max_depth': 8.* Es decir, un AdaBoost va a emplear una tasa de aprendizaje de uno, ciento cincuenta árboles de decisión, con al menos cinco muestras para considerar una decisión con al menos una muestra cada nodo hoja y con una profundidad máxima de ocho para cualquier árbol del bosque.

Random Forest (RF): *n_estimators=100, min_samples_split=5, min_samples_leaf=2, max_depth=10*. Es decir, un bosque aleatorio que va a emplear cien árboles de decisión, con al menos cinco muestras para considerar una decisión con al menos dos muestras cada nodo hoja y con una profundidad máxima de diez para cualquier árbol del bosque.

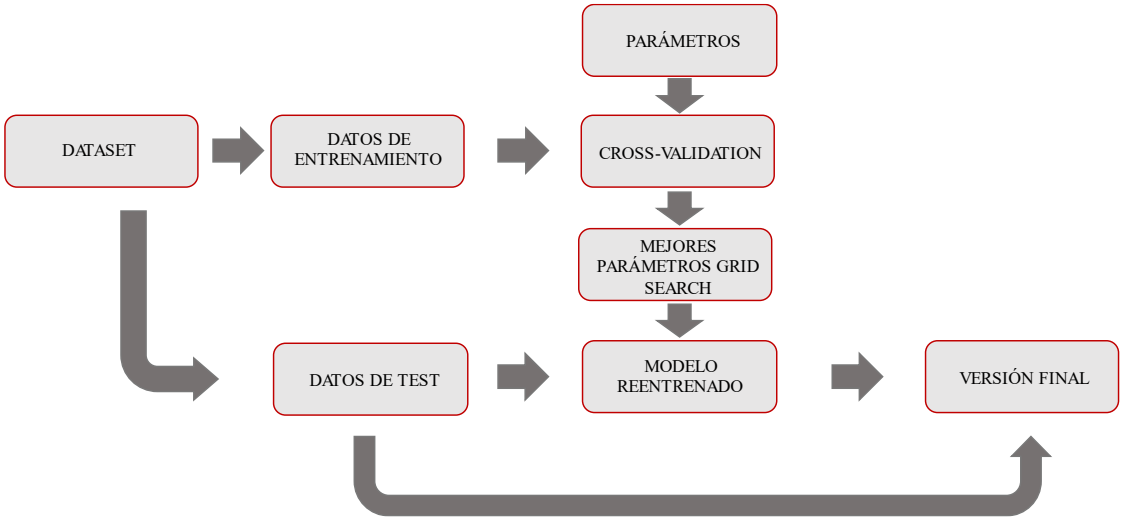

Figura 42. Flujo de trabajo para la determinación de los hiperparámetros

4 Resultados

En los siguientes apartados se puede observar los resultados obtenidos y su detalle en base a;

- Número de variables ausentes
- Variable o técnica objetivo de la predicción
- Estrategia empleada
- Algoritmo empleado

Los resultados se estructuran en tablas con una estructura tipo en la que pueden observarse las siguientes posibles columnas:

- N.º variables desconocidas: Se representa el número de variables que se encuentran ausentes en el modelo (1, 2, o 3).
- Algoritmo: Identifica el tipo de algoritmo de *Machine Learning* empleado (SVM, RL, DT, AB o RF).
- Precisión: El valor de instancias clasificadas correctamente (realizando la media para las observaciones del número de variables ausentes fijado en la columna pertinente).
- AUC: Esta columna informa acerca de la métrica Área bajo la curva ROC, que representa la capacidad del modelo para clasificar correctamente (realizando la media para las observaciones del número de variables ausentes fijado en la columna pertinente).
- F1-score: En esta columna se recoge información consistente en la media armónica de la precisión y la sensibilidad (realizando la media para las observaciones del número de variables ausentes fijado en la columna pertinente).
- Recall: La fracción de instancias positivas que fueron identificadas correctamente (realizando la media para las observaciones del número de variables ausentes fijado en la columna pertinente).

- Métrica compuesta: Esta métrica, explicada en el capítulo correspondiente, representa de forma integral las cuatro anteriores métricas, ponderando en función de su interés para la investigación.

Tal y como se ha indicado, y de cara a reducir la complejidad de la investigación y los costos computacionales, se ha optado por analizar los 129 modelos que resultan mediante combinatoria al entrar en juego 9 variables y las posibles ausencias (0, 1 o 2 variables ausentes), mediante valores promedio.

4.1 Análisis en base al número de variables ausentes

Conforme el planteamiento previamente reseñado, se han considerado tres posibles escenarios; el desconocimiento de una, dos y hasta tres variables de la cadena o vector de un ciberataque. Los resultados que arrojan los algoritmos son los siguientes:

Tabla 23. Resultados en función del número de variables desconocidas

Número de variables desconocidas	Estrategia	Algoritmo	Precisión	AUC	F1-score	Recall	Métrica compuesta
1	CC	SVM	0.7503	0.6208	0.7029	0.7961	0.7437
		RL	0.7323	0.7411	0.7320	0.7369	0.7360
		DT	0.8143	0.7615	0.7974	0.8251	0.8081
		AB	0.8149	0.7612	0.7981	0.8274	**0.8094**
		RF	0.8144	0.7602	0.7970	0.8268	0.8087
1	OVR	SVM	0.7503	0.6208	0.7029	0.7961	0.7437
		RL	0.7323	0.7411	0.7320	0.7369	0.7360
		DT	0.8143	0.7615	0.7974	0.8251	0.8081
		AB	0.8142	0.7587	0.7977	0.8288	0.8095
		RF	0.8135	0.7602	0.7982	0.8301	**0.8104**
2	CC	SVM	0.7447	0.6107	0.6932	0.7821	0.7325
		RL	0.7198	0.7293	0.7216	0.7314	0.7274
		DT	0.7900	0.7272	0.7766	0.8175	0.7911
		AB	0.7900	0.7262	0.7771	0.8196	**0.7920**
		RF	0.7899	0.7264	0.7768	0.8191	0.7917
2	OVR	SVM	0.7467	0.6130	0.6862	0.7700	0.7260
		RL	0.7216	0.7314	0.7249	0.7391	0.7325
		DT	0.7950	0.7284	0.7727	0.7961	0.7807
		AB	0.7948	0.7270	0.7732	0.7984	0.7817
		RF	0.7948	0.7272	0.7733	0.7987	**0.7819**
3	CC	SVM	0.7387	0.6012	0.6849	0.7729	0.7239
		RL	0.7078	0.7186	0.7129	0.7299	0.7215
		DT	0.7675	0.6945	0.7611	0.8209	0.7810
		AB	0.7674	0.6938	0.7613	0.8219	0.7814
		RF	0.7675	0.6941	0.7613	0.8221	**0.7815**
3	OVR	SVM	0.7430	0.6030	0.6664	0.7436	0.7072
		RL	0.7101	0.7213	0.7140	0.7295	0.7223
		DT	0.7778	0.6961	0.7492	0.7745	0.7578
		AB	0.7777	0.6953	0.7494	0.7759	0.7584
		RF	0.7777	0.6956	0.7497	0.7766	**0.7588**

Se desprenden las siguientes conclusiones:

- Para el caso particular de un número de **variables desconocidas = 1**, las estrategias CC y OVR ofrecen resultados muy parejos en términos de métrica

compuesta para cada algoritmo. El algoritmo que tiene el mejor rendimiento en términos de métrica compuesta es el RF (**Random Forest**) con **0.8104** para **OVR** y el AB (AdaBoost) con 0.8094 para CC, y el algoritmo con el menor rendimiento en esta categoría es la RL (Regresión Logística) con una métrica compuesta de 0.7360 para ambas estrategias.

- Para un número de **variables desconocidas = 2**, nuevamente, las estrategias CC y OVR muestran resultados similares en términos de métrica compuesta para cada algoritmo. El algoritmo con el mejor rendimiento es el AB (**AdaBoost**) con una métrica de **0.7920** para **CC**, seguido del RF con una métrica compuesta de 0.7819 para OVR. RL tiene el rendimiento más bajo con 0.7325 para OVR y 0.7274 para CC.
- En el caso de un número de **variables desconocidas = 3**, las diferencias entre CC y OVR comienzan a ser un poco más notables, pero aún son pequeñas. RF (**Random Forest**) es el algoritmo de mejor rendimiento con una métrica compuesta de 0.7588 para OVR y **0.7815** para **CC**. RL continúa siendo el algoritmo con menor rendimiento con 0.7223 para OVR y 0.7215 para CC.

En general, las estrategias CC y OVR tienen rendimientos muy similares para cada algoritmo y número de variables desconocidas, en especial para el caso de una y dos variables desconocidas en valores de métrica compuesta. No obstante, en el caso de tres variables desconocidas las diferencias comienzan a incrementar.

Respecto a los algoritmos empleados, tanto AdaBoost como Random Forest presentan en todos los casos valores muy parejos, destacando ligeramente AB en CC y RF en OVR, mientras que SVM y Regresión Logística tienen valores inferiores, en especial en el caso de SVM para tres variables ausentes en la estrategia OVR.

En conclusión, y tal y como puede observarse en la Figura 43, a medida que aumenta el número de variables desconocidas, parece que hay una disminución en el rendimiento en términos de métrica compuesta para todos los algoritmos y estrategias, en especial a partir de la tercera variable desconocida. Este hecho podría indicar que el modelo tiene dificultades para manejar un mayor número de incógnitas (algo lógico al disponer de menor entrada de datos).

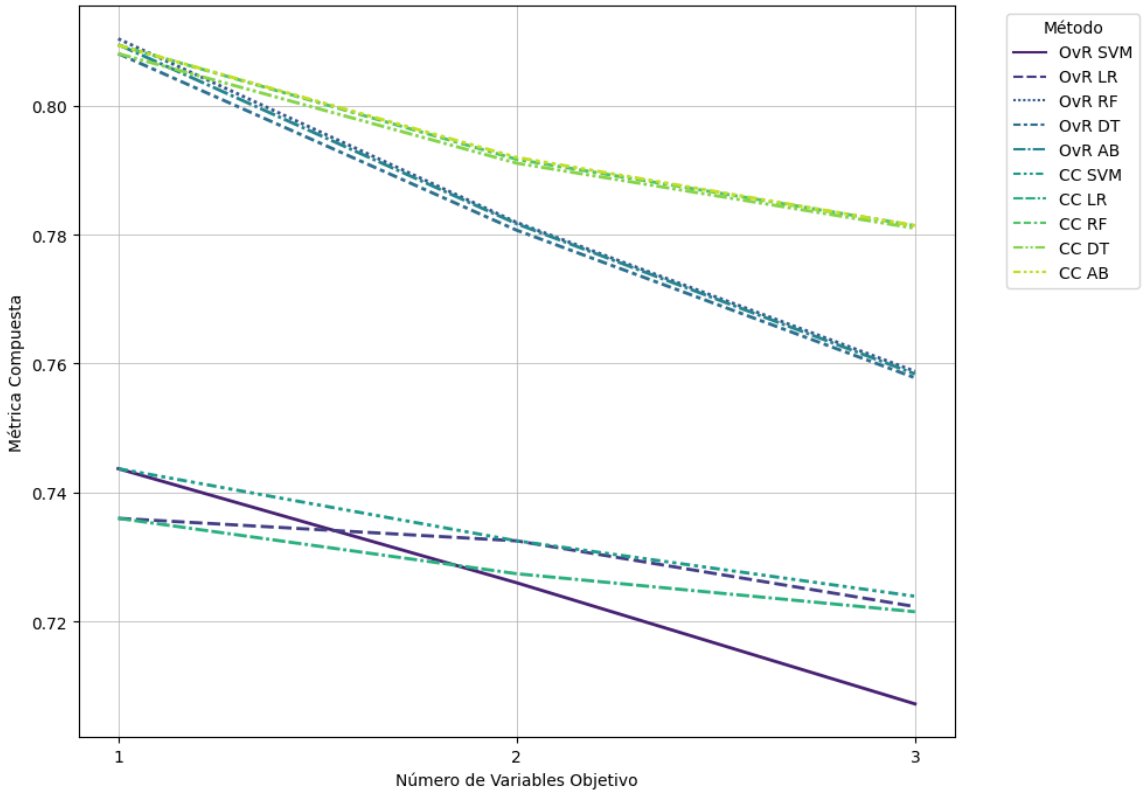

Figura 43. Evolución de la métrica compuesta en función del número de variables ausentes

4.2 Análisis en base a la variable objetivo

Los resultados conforme a la variable o táctica objetivo se recogen tabulados a continuación, estructurados a su vez conforme cada una de las dos estrategias.

Estrategia a): Classifier Chains (o cadenas de clasificadores)

Tabla 24. Resultados en función del número de variables desconocidas para CC

Variable Objetivo	Algoritmo	Precisión	AUC	F1-score	Recall	Métrica compuesta
TA0001	SVM	0.7400	0.5770	0.4786	0.5385	0.5685
	RL	0.6970	0.7083	0.6589	0.7201	0.7041
	DT	0.7696	0.6955	0.6873	0.7114	0.7144
	AB	0.7695	0.6949	0.6878	0.7128	**0.7151**
	RF	0.7691	0.6960	0.6876	0.7123	0.7150
TA0002	SVM	0.7497	0.6159	0.7412	0.8273	0.7648
	RL	0.7177	0.7299	0.7387	0.7188	0.7238
	DT	0.7850	0.7091	0.7986	0.8470	0.8056
	AB	0.7850	0.7085	0.7988	0.8482	**0.8061**
	RF	0.7847	0.7079	0.7985	0.8481	0.8059
TA0003	SVM	0.7302	0.6136	0.7253	0.8336	0.7616
	RL	0.7135	0.7160	0.7325	0.7395	0.7301
	DT	0.7690	0.7090	0.7823	0.8417	0.7976
	AB	0.7687	0.7081	0.7825	0.8429	0.7980
	RF	0.7691	0.7091	0.7829	0.8431	**0.7984**
TA0005	SVM	0.7572	0.5553	0.7541	0.8617	0.7753
	RL	0.7280	0.7399	0.7532	0.7318	0.7361
	DT	0.7965	0.6881	0.8112	0.8614	0.8133
	AB	0.7964	0.6861	0.8115	0.8633	**0.8140**
	RF	0.7964	0.6888	0.8117	0.8616	0.8136
TA0006	SVM	0.7225	0.6346	0.6599	0.7551	0.7137
	RL	0.6999	0.7155	0.6885	0.7512	0.7263
	DT	0.7554	0.7081	0.7291	0.7973	0.7641
	AB	0.7555	0.7078	0.7295	0.7986	0.7648
	RF	0.7554	0.7089	0.7295	0.7988	**0.7650**
TA0007	SVM	0.7404	0.5763	0.7404	0.8509	0.7683
	RL	0.7020	0.7181	0.7264	0.7072	0.7114
	DT	0.7798	0.6877	0.7966	0.8522	0.8034
	AB	0.7796	0.6864	0.7968	0.8542	**0.8043**
	RF	0.7797	0.6876	0.7968	0.8533	0.8040
TA0009	SVM	0.7377	0.6370	0.6793	0.7040	0.6943
	RL	0.7191	0.7245	0.7186	0.7183	0.7195
	DT	0.7643	0.7234	0.7576	0.7932	0.7708
	AB	0.7642	0.7233	0.7579	0.7944	0.7715
	RF	0.7637	0.7227	0.7581	0.7951	**0.7717**
TA0010	SVM	0.7426	0.6483	0.6639	0.7573	0.7211
	RL	0.7111	0.7250	0.6853	0.7667	0.7369
	DT	0.7681	0.7294	0.7316	0.8265	0.7848
	AB	0.7682	0.7292	0.7318	0.8273	0.7852
	RF	0.7683	0.7297	0.7319	0.8271	**0.7852**
TA0011	SVM	0.7428	0.5764	0.7420	0.8511	0.7691
	RL	0.7112	0.7168	0.7359	0.7197	0.7205
	DT	0.7752	0.6803	0.7946	0.8515	0.8008
	AB	0.7749	0.6790	0.7946	0.8525	**0.8010**
	RF	0.7750	0.6806	0.7946	0.8515	0.8008

Estrategia b): One-Vs-Rest (o uno contra todos)

Tabla 25. Resultados en función del número de variables desconocidas para OvR

Variable Objetivo	Algoritmo	Precisión	AUC	F1-score	Recall	Métrica compuesta
TA0001	SVM	0.7408	0.5764	0.4696	0.5276	0.5616
	RL	0.6946	0.7089	0.6603	0.7290	**0.7084**
	DT	0.7754	0.6929	0.6709	0.6747	0.6939
	AB	0.7754	0.6924	0.6716	0.6768	0.6950
	RF	0.7753	0.6929	0.6721	0.6778	0.6956
TA0002	SVM	0.7535	0.6232	0.7308	0.8039	0.7532
	RL	0.7212	0.7330	0.7422	0.7242	0.7282
	DT	0.7917	0.7077	0.7904	0.8137	0.7885
	AB	0.7916	0.7070	0.7907	0.8151	0.7891
	RF	0.7914	0.7072	0.7911	0.8164	**0.7898**
TA0003	SVM	0.7381	0.6264	0.7231	0.8180	0.7569
	RL	0.7154	0.7177	0.7328	0.7362	0.7291
	DT	0.7756	0.7081	0.7734	0.8076	0.7800
	AB	0.7755	0.7072	0.7735	0.8090	0.7805
	RF	0.7753	0.7072	0.7736	0.8092	**0.7806**
TA0005	SVM	0.7674	0.5773	0.7516	0.8389	0.7688
	RL	0.7275	0.7420	0.7527	0.7312	0.7360
	DT	0.8033	0.6980	0.8057	0.8366	0.8028
	AB	0.8028	0.6954	0.8056	0.8381	0.8030
	RF	0.8030	0.6964	0.8066	0.8402	**0.8045**
TA0006	SVM	0.7257	0.6349	0.6503	0.7373	0.7038
	RL	0.7039	0.7181	0.6897	0.7477	0.7258
	DT	0.7686	0.7091	0.7140	0.7426	0.7366
	AB	0.7682	0.7085	0.7141	0.7440	0.7371
	RF	0.7680	0.7084	0.7142	0.7449	**0.7376**
TA0007	SVM	0.7420	0.5776	0.7363	0.8438	0.7645
	RL	0.7058	0.7193	0.7299	0.7128	0.7156
	DT	0.7851	0.6957	0.7928	0.8291	0.7935
	AB	0.7850	0.6943	0.7930	0.8310	**0.7942**
	RF	0.7850	0.6947	0.7929	0.8305	0.7940
TA0009	SVM	0.7395	0.6269	0.6357	0.6554	0.6614
	RL	0.7234	0.7289	0.7191	0.7123	0.7181
	DT	0.7803	0.7217	0.7487	0.7406	0.7454
	AB	0.7802	0.7214	0.7492	0.7422	**0.7463**
	RF	0.7802	0.7214	0.7491	0.7420	0.7461
TA0010	SVM	0.7444	0.6339	0.6161	0.6970	0.6809
	RL	0.7132	0.7282	0.6862	0.7657	0.7374
	DT	0.7810	0.7201	0.7123	0.7517	0.7448
	AB	0.7809	0.7195	0.7129	0.7540	0.7459
	RF	0.7811	0.7203	0.7134	0.7549	**0.7466**
TA0011	SVM	0.7444	0.5747	0.7316	0.8346	0.7591
	RL	0.7135	0.7199	0.7390	0.7266	0.7254
	DT	0.7819	0.6902	0.7920	0.8282	0.7915
	AB	0.7818	0.6888	0.7922	0.8297	0.7920
	RF	0.7817	0.6893	0.7923	0.8296	**0.7920**

Desde el punto de vista de la variable objetivo, y con el apoyo visual que ofrece el *heatmap* de la Figura 44, pueden conformarse dos grupos de variables que presentan un comportamiento muy parejo:

Un grupo 1, con tácticas de puntuación más elevada: TA0002 (Ejecución), TA0005 (Evasión de Defensas), TA0007 (Descubrimiento) y TA0011 (Mando y Control) muestran valores nunca inferiores al 80% en las mejores puntuaciones de la métrica compuesta (las

referentes a CC-RF, CC-DT y CC-AB) y siendo muy parejo el rendimiento de estas cuatro variables (alrededor del 80-81%). No se observa relación aparente entre estas tácticas, más allá de que se encuentran repartidas a lo largo de toda la cadena de ataque de la matriz mitre, y no se concentran en la fase inicial, ni final ni intermedia.

Un grupo 2, con tácticas de puntuación más reducida: TA0001 (Acceso Inicial), TA0003 (Persistencia), TA0006 (Acceso a Credenciales), TA0009 (Recolección) y TA0010 (Exfiltración) ofrecen peores valores de métrica compuesta, pero aún por encima del 71% de puntuación en todo momento para aquellas referentes a CC-DT, CC-RF Y CC-AB). Es reseñable que en este grupo hay un mayor escalonamiento de resultados en las tácticas, observándose incrementos de alrededor de un 1% de forma progresiva salvo en el caso de la variable TA0001, que presenta valores bastante inferiores al resto. Al igual que en el grupo anterior no se observa relación aparente entre estas variables más allá de lo expuesto.

En conclusión, se observan dos grupos de tácticas con rendimientos parejos, si bien la variable TA0001 presenta unos valores muy inferiores al resto.

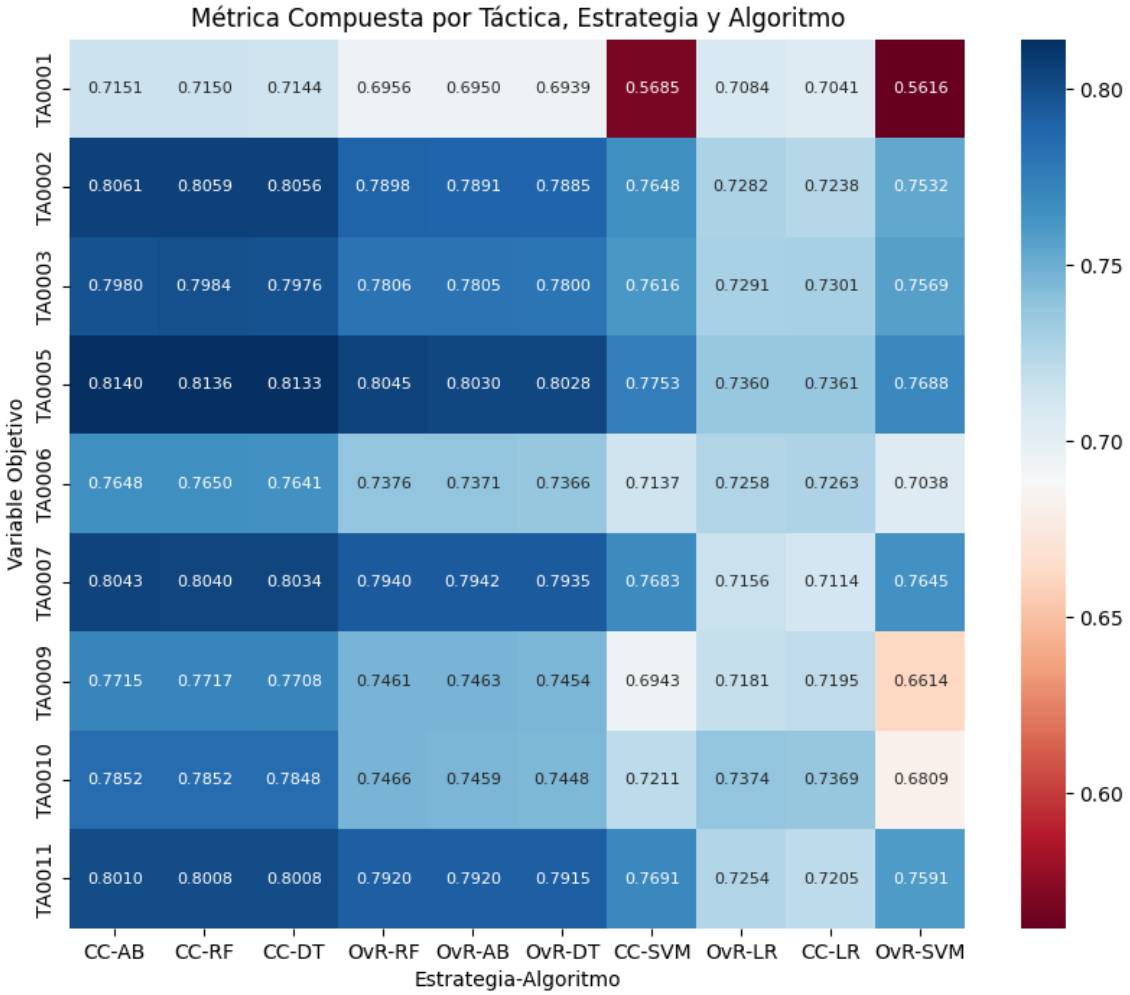

Figura 44. Valores de métrica compuesta por Estrategia y algoritmo para cada una de las tácticas

4.3 Análisis en base a la estrategia

En relación con la influencia de la estrategia a seguir en el modelo se observan los siguientes hechos:

La aplicación de Cadenas de Clasificadores (CC) ofrece en todas las variables los mejores resultados, en especial con buenos resultados para tres algoritmos concretos; el AB, el DT y el RF, mientras que esa misma estrategia con los algoritmos SVM y RL ofrece resultados mucho peores en comparación e incluso inferiores a los valores de la estrategia One-Vs-Rest (OVR) en los algoritmos AB, DT y RF.

Los mejores valores de la estrategia OVR siempre se encuentran por debajo de los mejores valores de CC para cada variable salvo para la RL, el único caso en el que OVR supera a la estrategia CC. Este hecho indica que, si bien es importante la elección de una correcta estrategia, sin duda lo es más la elección del correcto algoritmo al tener una mayor influencia en la puntuación.

En definitiva, la estrategia CC ofrece resultados máximos superiores a OVR, pero en términos medios ambas estrategias presentan valores muy similares de la métrica compuesta. Hay que tener en cuenta que la métrica compuesta es una métrica integral que tiene en cuenta varias métricas de rendimiento, por lo que un valor más alto indica un rendimiento más robusto del modelo en varios frentes.

4.4 Análisis en base al algoritmo

El rendimiento de SVM es generalmente aceptable en casi todas las variables (puntuaciones por encima del 75%), pero se observa un destacado hecho que es la profunda bajada de rendimiento (se registran valores por debajo del 57%) en la variable TA0001, que conlleva a descartar este algoritmo al entenderse que no es capaz de captar información para predecir correctamente esta variable.

La Regresión Logística muestra un desempeño bastante homogéneo para ambas estrategias (CC y OVR) que ofrece valores muy parejos, no habiendo diferencias significativas en la métrica compuesta. En todas las variables se observan rendimientos inferiores al 74%, lo que provoca que, si no fuera por el mal rendimiento de SVM en la TA0001, fuese, de lejos, el algoritmo con peor puntuación.

En el caso de los Árboles de Decisión, los Bosques Aleatorios y AdaBoost, el comportamiento es muy similar y ofrece rendimientos muy interesantes. En todos los casos se ven igual de afectados por la elección de una estrategia u otra, tienen relativas dificultades (no comparables con lo sucedido en SVM) en la variable TA0001, y la elección del algoritmo se decide por milésimas. En conclusión, estos tres modelos están ofreciendo un buen equilibrio entre precisión, AUC, F1-score y *recall*, pero los valores medios de AdaBoost son mejores que los de los Árboles de Decisión y Bosques aleatorios, y, por supuesto, de la Regresión Logística y de las Máquinas de Soporte de Vectores.

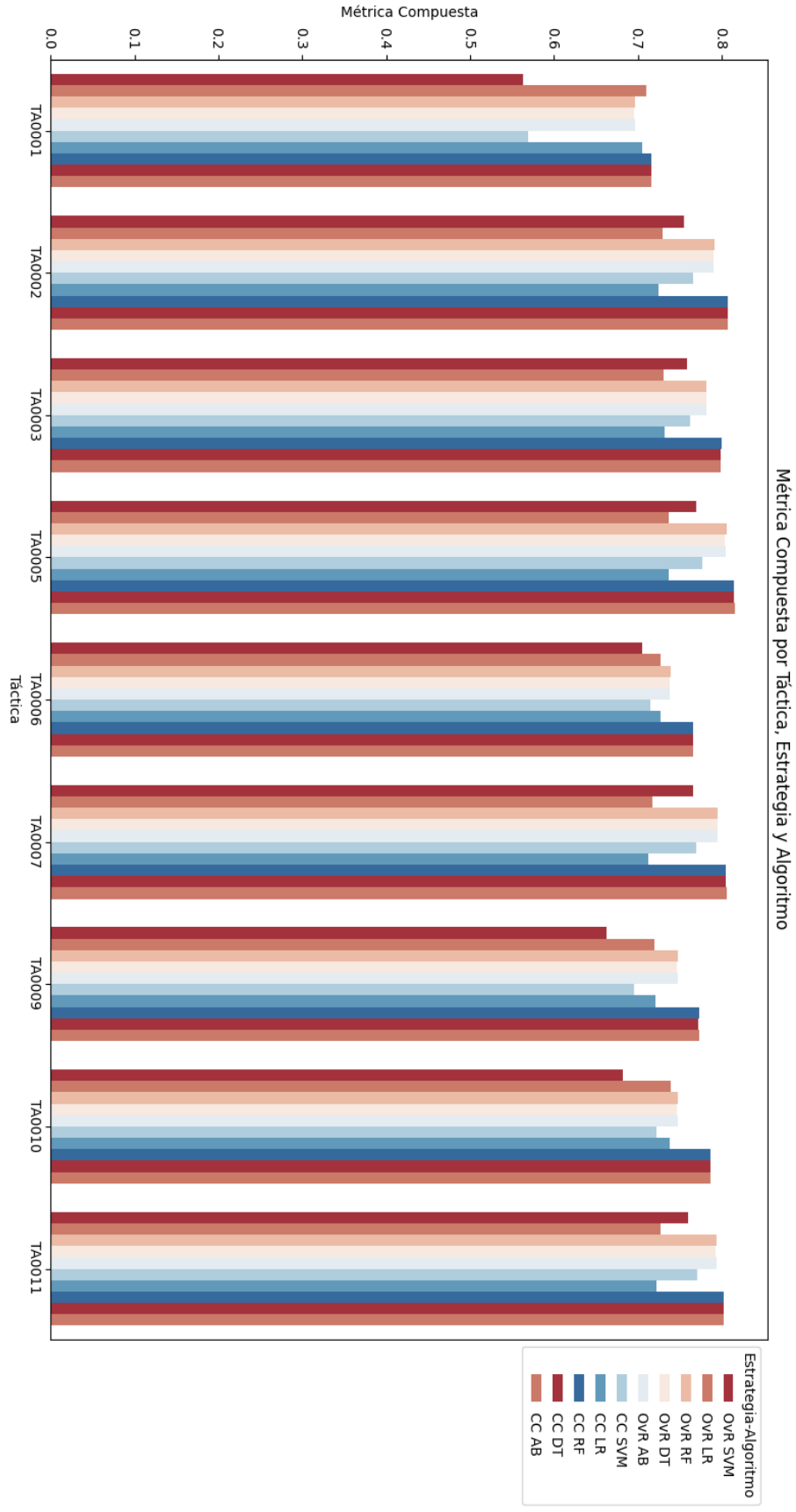

Figura 45. Desglose de métrica compuesta por algoritmo, estrategia y variable

5 Conclusiones

En este capítulo se ha presentado un modelo matemático para afrontar la problemática objeto de investigación (predecir la presencia de una, dos o tres determinadas tácticas en un ciberataque). Para ello se propone el uso de dos estrategias; por un lado, Classifier Chains, que va a emplear las predicciones sucesivas para conformar una cadena de valores para las variables desconocidas, y por otro OneVsRest, estrategia que va a confrontar cada variable ausente con el resto de las variables que sí están disponibles. A su vez, el modelo se analizará empleando los algoritmos Support Vector Machines, Regresión Logística, Decision Tree, AdaBoost y Random Forest.

Para obtener una visión integral de los resultados, pero teniendo en cuenta la importancia que se le otorga al *recall* en el problema, se empleará la Métrica Compuesta definida en este capítulo.

Tras ejecutar un estudio de los hiperparámetros óptimos, los datos analizados muestran un mejor rendimiento de los algoritmos Random Forest y AdaBoost. Estos algoritmos muestran los valores medios más elevados en la métrica compuesta si se observa el número de variables ausentes, al alcanzar unos valores medios de un 81.04% (RF OVR), 79.20% (AB CC) y 78.15% (RF CC) para los casos respectivos de una dos y tres variables ausentes. Así mismo, y tal y como era previsible, a medida que aumenta el número de variables desconocidas, se observa una disminución en el rendimiento en términos de métrica compuesta para todos los algoritmos y estrategias, en especial a partir de la tercera variable desconocida.

En relación con el comportamiento concreto de las tácticas o variables analizadas, se observa la existencia de dos grupos de tácticas con comportamientos semejantes entre ellas, si bien la variable TA0001, relativa a los estadios iniciales de un ataque, presenta mayores dificultades para su predicción en comparación con el resto, y en especial con la táctica TA0005, de modo que se observa una diferencia de casi diez puntos en el rendimiento.

Así pues, partiendo de los trabajos realizados en capítulo anteriores (por los que el *framework* Mitre Att&ck se ha codificado, adaptado mediante su reducción dimensional e incrementado y corregido sus observaciones), se concluye que la mejor solución es la propuesta consistente en un modelo de predicción de tácticas de un ciberataque empleando la estrategia de Cadenas de Clasificadores y el algoritmo AdaBoost (véase Figura 46), dado que ha mostrado un rendimiento destacado en términos de la métrica compuesta, especialmente en escenarios con un número limitado de variables ausentes y alcanzando un valor medio de 79.42%. Si bien es cierto que podría emplearse en determinados casos puntuales el algoritmo Random Forest e incrementar muy ligeramente la métrica hasta un 79.46%, se opta por emplear únicamente AdaBoost con Cadenas de Clasificadores como solución de compromiso para simplificar el modelo.

5.1 Ventajas

El método propuesto en este estudio se destaca por su notable flexibilidad, permitiendo la adaptación de modelos en circunstancias donde puede haber ausencia de hasta tres variables durante un ciberataque, hecho particularmente útil en análisis forense cuando algunos datos son desconocidos. Su fundamentación en el *framework* Mitre Att&ck, que

goza de amplia aceptación en la comunidad de ciberseguridad, añade credibilidad y robustez al método.

Una de las fortalezas primordiales es la reducción dimensional, que simplifica el enfoque al pasar de 14 a 9 tácticas, optimizando la eficiencia computacional sin sacrificar información esencial debido al estudio que se ha realizado acerca de las variables. Finalmente se trata de un modelo que va a aportar información de mucho interés al analista forense en las primeras fases de investigación de un ataque, al permitir dirigir los esfuerzos de búsqueda hacia unos activos u otros en función de la presencia o no de una determinada táctica en un ataque, reduciendo por tanto tiempo y esfuerzos de analista.

5.2 Inconvenientes

La reducción dimensional, aunque eficiente, conlleva el riesgo de pérdida de información, aunque sea pequeña. Además, el hecho de analizar las tácticas y no las técnicas (téngase presente el elevado número de técnicas que se cifran en casi doscientas) sirve para ofrecer información rápida y útil en los primeros estadios frente a un ciberataque, pero puede ser insuficiente ante un estudio detallada de técnicas puntuales.

Finalmente, el mantenimiento y actualización del modelo pueden conllevar esfuerzos en términos de tiempo y recursos, y su estructura secuencial, aunque clara, podría no ser idónea para todos los escenarios.

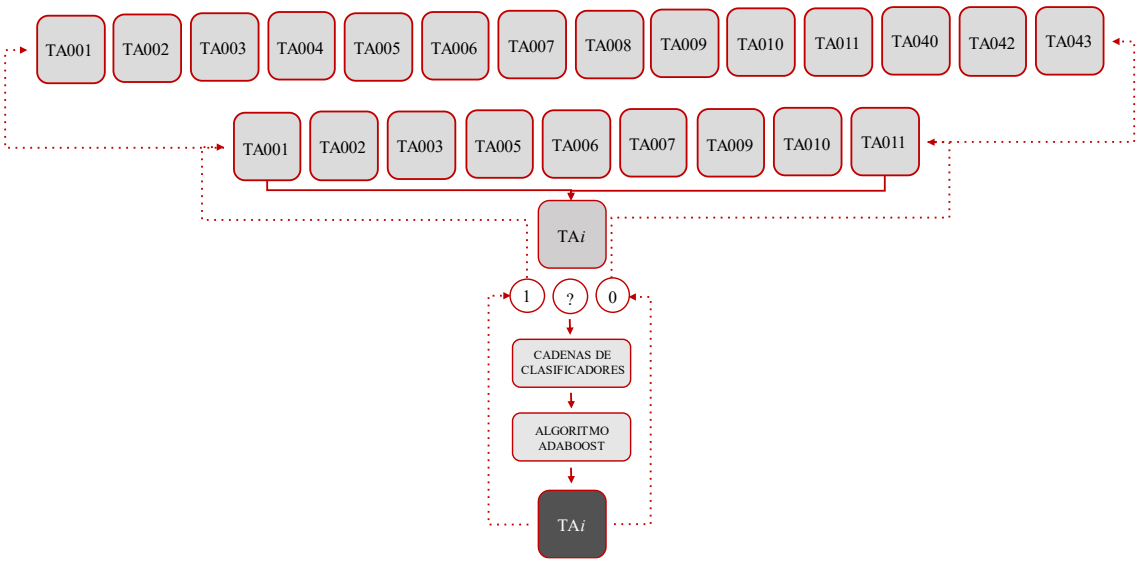

Figura 46. Modelo de predicción de tácticas empleando Cadenas de clasificadores y el algoritmo AdaBoost

Capítulo 8. Conclusiones y futuras líneas de investigación

1 Introducción

A lo largo del presente capítulo se ofrecen las principales conclusiones extraídas tras la elaboración de la investigación. En este sentido, se ha estructurado el contenido para detallar las aportaciones más relevantes a juicio del autor, así como expresar las futuras líneas de investigación que pudieran derivarse de este estudio en materia de ciberseguridad e inteligencia artificial.

2 Análisis de las principales contribuciones realizadas en la tesis doctoral

- Hito 1. Estudio de la actual problemática en materia de cibercriminalidad. Correspondiente con el Objetivo específico 1.

  De cara a poder comprender los trabajos realizados en la investigación, se lleva a cabo una síntesis de los principales conceptos que rodean a la ciberseguridad y a la cibercriminalidad, en especial detallando la metodología de un ciberataque que servirá de base para entender cómo funciona un ciberataque.

- Hito 2. Estudio de los *frameworks* de ciberinteligencia para su uso en AI. Correspondiente con el Objetivo específico 2.

  Se ha llevado a cabo un estudio detallado acerca de los tres principales *frameworks* de ciberinteligencia, de modo que mediante diecisiete ítems se ha establecido un marco que ha permitido comparar el comportamiento de estos marcos con la vista puesta en su utilidad en el ulterior uso de inteligencia artificial.
  Además, se presenta un estudio de campo acerca de un ciberataque de gran impacto que tuvo lugar en una entidad crítica de España y que sirve para ilustrar cómo se comportan cada uno de los tres modelos en un caso real.

- Hito 3. Desarrollo de un tratamiento de los datos del *framework* Mitre att&ck para su procesado Correspondiente con el Objetivo específico 3.

  Ante el reto que supone la transformación de un conjunto de datos nominal en otro que sea manejable y asimilable por los algoritmos se plantaron varias posibilidades (*one hot*, *hashing*, etc) y tras el análisis necesario se optó por una codificación *one hot*. Mediante este tipo de codificación se conformó un conjunto de datos en el que se plasman cada una de las 14 tácticas que Mitre Att&ck utiliza para representar un ciberataque.
  Profundizando aún más en la necesidad de tener un mejor *dataset* se lleva a cabo una propuesta de reducción dimensional, de modo que se establece un número final de nueve tácticas o variables, tras estimar que las cinco variables restantes no aportan información suficientemente relevante y podrían llegar a generar ruido

innecesario al conjunto. Así pues, mediante este hito se consigue un novedoso dataset binario de nueve variables

- Hito 4. Propuesta de soluciones ante la escasez de datos y desbalanceo de datos. Correspondiente con el Objetivo específico 3.

  Actualmente en el ámbito de la ciberseguridad uno de los principales problemas a los que se enfrentan los analistas es la escasez de datos. Las entidades no están dispuestas a compartir información que pueda perjudicar a su reputación y, al menos hasta el día de hoy, parece que esa tendencia no se va a invertir. Por ello, y ante el reducido número de observaciones que tiene el *dataset* conformado, y la existencia de gran número de observaciones en determinadas tácticas en relación con otras, se lleva a cabo un estudio acerca de cómo solucionar esa problemática, concluyendo que mediante el uso de la estrategia Neighbourhood Cleaning Rule adaptada para cada variable como objetivo, y realizando la oportuna limpieza posterior, se dispone de un dataset de mayor calidad, con cerca de cinco mil observaciones y con una reducción, aunque escasa, del desbalanceo inicial.

- Hito 5. Propuesta de modelo de predicción de tácticas sobre la base del estudio de dos estrategias y cuatro algoritmos. Correspondiente con el Objetivo específico 4.

  Una vez se dispone de un dataset de Mitre Att&ck con suficiente calidad, se analizan las estrategias para afrontar la problemática planteada en la investigación y se concluye que el uso de las cadenas de clasificadores ofrece una muy interesante opción ante el hecho de presentarse una, dos y hasta tres variables o tácticas ausentes. Así mismo se analiza en detalle los algoritmos que ofrecen mejor resultados, constatándose que AdaBoost junto con las cadenas de clasificadores arroja unos resultados de alrededor del 80% de puntuación en la métrica empleada en la investigación.

Así pues, mediante la consecución de estos hitos, asociados a cada uno de los objetivos específicos planteados, se llega a la consecución del objetivo principal de la investigación; la obtención de un modelo que sirve de apoyo en la determinación de patrones de comportamiento y en la predicción de tácticas empleadas en determinados ataques a sistemas de información u operación

3 Futuras líneas de investigación

- Línea 1. Nuevos marcos de ciberinteligencia

  Para futuros trabajos, pudiera ser interesante el desarrollo de un marco de ciberinteligencia global que integre las características de cada uno de los tres marcos analizados en esta investigación para su uso en IA, incluyendo especialmente los aspectos más clásicos de la inteligencia que ofrece el Modelo Diamante, junto con los aspectos más técnicos y automatizados del modelo de Mitre.

- Línea 2. Actualización del marco Mitre Att&ck

El marco Mitre Att&ck está en constante actualización, habiendo variado en numerosas ocasiones su estructura y contenidos, si bien es cierto que la nomenclatura y gran número de las tácticas se mantienen a lo largo del tiempo. Esta circunstancia genera la necesidad de disponer de un programa de actualización del modelo a desarrollar que permita adaptarse a los cambios del *framework.*

- Línea 3. Mejora de los dataset

  Los datos contenidos en el repositorio de Mitre Att&ck no son representativos de todas las técnicas posibles utilizadas por los actores asociados con los datos observados, sino un subconjunto de lo que de lo que ha estado disponible a través de informes públicos y de código abierto. Este hecho, junto con la necesidad de disponer de un mayor volumen de datos de carácter real frente a los artificiales generados en el marco de este trabajo, hace necesario contemplar el tratamiento de datos como una interesante línea de investigación de cara al futuro.

- Línea 4. Inclusión de las técnicas y subtécnicas

  La presencia de técnicas y subtécnicas se ha obviado en la investigación ante la complejidad que requeriría un tratamiento de cientos de variables, pero la necesaria tecnificación de los ataques hará necesario disponer de información más precisa más allá de las tácticas de los atacantes.

- Línea 5. Integración de metodologías de análisis de campañas de desinformación

  Disarm[7], con 18 tácticas y 134 técnicas, se ha establecido como un estándar a nivel europeo en materia de lucha contra la desinformación. Basa su estructura en la matriz de Mitre, lo que lo hace susceptible de recibir un tratamiento semejante al que se ha realizado en esta investigación, de modo que lo tratado pudiera aplicarse en la predicción de tácticas empleadas en campañas de desinformación.

- Línea 6. Implementación de algoritmos de Deep Learning.

  Finalmente, la proliferación de algoritmos de Deep Learning, no tratados en esta investigación, pueden colaborar en el incremento del rendimiento del modelo, si bien sería imprescindible disponer de un mayor volumen de datos para evitar el sobreajuste de las redes neuronales empleadas.

[7] https://github.com/DISARMFoundation/DISARMframeworks

Referencias

[1] S. Gordon y R. Ford, «On the definition and classification of cybercrime», *J Comput Virol*, vol. 2, n.º 1, pp. 13-20, ago. 2006, doi: 10.1007/s11416-006-0015-z.
[2] P. M. Tomás *et al.*, «Informe sobre la cibercriminalidad en España 2022», 2022.
[3] European Union Agency for Law Enforcement Cooperation., *IOCTA 2021: internet organised crime threat assessment 2021.* LU: Publications Office, 2021. Accedido: 29 de junio de 2023. [En línea]. Disponible en: https://data.europa.eu/doi/10.2813/113799
[4] F. Almeida, J. Duarte Santos, y J. Augusto Monteiro, «The Challenges and Opportunities in the Digitalization of Companies in a Post-COVID-19 World», *IEEE Eng. Manag. Rev.*, vol. 48, n.º 3, pp. 97-103, sep. 2020, doi: 10.1109/EMR.2020.3013206.
[5] European Union Agency for Cybersecurity (ENISA), «NIS Investments 2022». Accedido: 29 de junio de 2023. [En línea]. Disponible en: https://www.enisa.europa.eu/publications/nis-investments-2022
[6] C. S. D. Brown, «Investigating And Prosecuting Cyber Crime: Forensic Dependencies And Barriers To Justice», ago. 2015, doi: 10.5281/ZENODO.22387.
[7] C. Wagner, A. Dulaunoy, G. Wagener, y A. Iklody, «MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform», en *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, Vienna Austria: ACM, oct. 2016, pp. 49-56. doi: 10.1145/2994539.2994542.
[8] Filigran, «Open Cyber Threat Intelligence Platform (OpenCTI)». Francia. Accedido: 1 de agosto de 2023. [En línea]. Disponible en: https://www.filigran.io/en/solutions/products/opencti/
[9] J. Andress y S. Winterfeld, *Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners: Second Edition*. 2013, p. 306.
[10] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, y M. Marchetti, «On the effectiveness of machine and deep learning for cyber security», en *2018 10th International Conference on Cyber Conflict (CyCon)*, Tallinn: IEEE, may 2018, pp. 371-390. doi: 10.23919/CYCON.2018.8405026.
[11] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, y A. Ng, «Cybersecurity data science: an overview from machine learning perspective», *J Big Data*, vol. 7, n.º 1, p. 41, dic. 2020, doi: 10.1186/s40537-020-00318-5.
[12] H. Cong y W. Chao, «Network Security Situation Awareness Based on the Optimized Dynamic Wavelet Neural Network», p. 8.
[13] N. Kaloudi y J. Li, «The AI-Based Cyber Threat Landscape: A Survey», *ACM Comput. Surv.*, vol. 53, n.º 1, pp. 1-34, ene. 2021, doi: 10.1145/3372823.
[14] Z. Jadidi y Y. Lu, «A Threat Hunting Framework for Industrial Control Systems», *IEEE Access*, vol. 9, pp. 164118-164130, 2021, doi: 10.1109/ACCESS.2021.3133260.
[15] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, y C. B. Thomas, «Mitre att&ck: Design and philosophy», en *Technical report*, The MITRE Corporation, 2018.
[16] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, y X. Niu, «TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources», en *Proceedings of the 33rd Annual Computer Security Applications Conference*, Orlando FL USA: ACM, dic. 2017, pp. 103-115. doi: 10.1145/3134600.3134646.
[17] M. Li, R. Zheng, L. Liu, y P. Yang, «Extraction of Threat Actions from Threat-related Articles using Multi-Label Machine Learning Classification Method», en *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, Chongqing, China: IEEE, nov. 2019, pp. 428-431. doi: 10.1109/IICSPI48186.2019.9095885.

[18] Y. Ghazi, Z. Anwar, R. Mumtaz, S. Saleem, y A. Tahir, «A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources», en *2018 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan: IEEE, dic. 2018, pp. 129-134. doi: 10.1109/FIT.2018.00030.
[19] V. Legoy, A. Peter, C. Seifert, y M. Caselli, «Retrieving ATT&CK tactics and techniques in cyber threat reports», p. 45.
[20] R. Al-Shaer, J. M. Spring, y E. Christou, «Learning the Associations of MITRE ATT&CK Adversarial Techniques», *arXiv:2005.01654 [cs]*, may 2020, Accedido: 28 de febrero de 2022. [En línea]. Disponible en: http://arxiv.org/abs/2005.01654
[21] Y. Shin, K. Kim, J. J. Lee, y K. Lee, «ART: Automated Reclassification for Threat Actors based on ATT&CK Matrix Similarity», en *2021 World Automation Congress (WAC)*, Taipei, Taiwan: IEEE, ago. 2021, pp. 15-20. doi: 10.23919/WAC50355.2021.9559514.
[22] K. Kim, Y. Shin, J. Lee, y K. Lee, «Automatically Attributing Mobile Threat Actors by Vectorized ATT&CK Matrix and Paired Indicator», *Sensors*, vol. 21, n.º 19, p. 6522, sep. 2021, doi: 10.3390/s21196522.
[23] A. Nisioti, G. Loukas, S. Rass, y E. Panaousis, «Game-Theoretic Decision Support for Cyber Forensic Investigations», *Sensors*, vol. 21, n.º 16, p. 5300, ago. 2021, doi: 10.3390/s21165300.
[24] S. Choi, J.-H. Yun, y B.-G. Min, «Probabilistic Attack Sequence Generation and Execution Based on MITRE ATT&CK for ICS Datasets», en *Cyber Security Experimentation and Test Workshop*, Virtual CA USA: ACM, ago. 2021, pp. 41-48. doi: 10.1145/3474718.3474722.
[25] Davara, Fernando, «Las TIC y las amenazas a la seguridad nacional; ciberseguridad», *Monografías de la Editorial Tirant lo Blanch*, vol. 833, pp. 145-160, 2013.
[26] R. Ottis y P. Lorents, «Cyberspace: Definition and Implications», *Proceedings of the 5th International Conference on Information Warfare and Security, Dayton, OH, US, 8-9 April*, pp. 267-270, 2010.
[27] M. Chapple, J. M. Stewart, y D. Gibson, *CISSP certified information systems security professional: official study guide*, Eighth edition. Indianapolis, Indiana: Sybex, a Wiley brand, 2018.
[28] Centro Criptológico Nacional, «GUÍA DE SEGURIDAD DE LAS TIC (CCN-STIC-820) GUÍA DE PROTECCIÓN CONTRA DENEGACIÓN DE SERVICIO». 2013. Accedido: 15 de mayo de 2022. [En línea]. Disponible en: https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/528-ccn-stic-820-proteccion-contra-denegacion-de-servicio/file.html
[29] Centro Criptológico Nacional, «GUÍA DE SEGURIDAD DE LAS TIC (CCN-STIC-480A) SEGURIDAD EN EL CONTROL DE PROCESOS Y SCADA GUIA DE BUENAS PRÁCTICAS». febrero de 2010. Accedido: 1 de mayo de 2022. [En línea]. Disponible en: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/209-ccn-stic-480a-seguridad-en-sistemas-scada-guia-de-buenas-practicas/file.html
[30] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, y A. Hahn, «Guide to Industrial Control Systems (ICS) Security», National Institute of Standards and Technology, NIST SP 800-82r2, jun. 2015. doi: 10.6028/NIST.SP.800-82r2.
[31] «Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas.», p. 11.
[32] FireEye, «Cybersecurity trends 2020». 2020. Accedido: 15 de mayo de 2022. [En línea]. Disponible en: https://www.fireeye.com/offers/rpt-cyber-trends.html
[33] «McAfee Labs COVID-19 Threats Report, July 2020», p. 40.
[34] N. A. Hassan, *Ransomware Revealed: A Beginner's Guide to Protecting and Recovering from Ransomware Attacks*. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-4255-1.

[35] O. Skulkin y R. Rezvukhin, «RANSOMWARE UNCOVERED 2020—2021», Group-IB, 2021. Accedido: 1 de mayo de 2022. [En línea]. Disponible en: https://explore.group-ib.com/ransomware-reports/ransomware_uncovered_2020
[36] «ESET Threat report 2020 Q4», ESET, 2022. Accedido: 2 de mayo de 2022. [En línea]. Disponible en: https://www.welivesecurity.com/wp-content/uploads/2021/02/ESET_Threat_Report_Q42020.pdf
[37] K. D. Mitnick y W. L. Simon, *The art of deception: controlling the human element of security*. Ed. Indianapolis: Wiley Publishing, 2002.
[38] Sophos, «Sophos 2021. Threat Report». Accedido: 15 de mayo de 2022. [En línea]. Disponible en: https://www.sophos.com/en-us/medialibrary/pdfs/technical-papers/sophos-2021-threat-report.pdf
[39] K. Yan, L. Liu, Y. Xiang, y Q. Jin, «Guest Editorial: AI and Machine Learning Solution Cyber Intelligence Technologies: New Methodologies and Applications», *IEEE Transactions on Industrial Informatics*, vol. 16, n.º 10, pp. 6626-6631, oct. 2020, doi: 10.1109/TII.2020.2988944.
[40] R. A. Kemmerer, «Cybersecurity», en *25th International Conference on Software Engineering, 2003. Proceedings.*, may 2003, pp. 705-715. doi: 10.1109/ICSE.2003.1201257.
[41] S. H. Kim, Q.-H. Wang, y J. B. Ullrich, «A comparative study of cyberattacks», *Commun. ACM*, vol. 55, n.º 3, pp. 66-73, mar. 2012, doi: 10.1145/2093548.2093568.
[42] M. Ludwick, J. McAllister, A. O. Mellinger, K. A. Sereno, y T. Townsend, «Cyber Intelligence Tradecraft Project: Summary of Key Findings», *Software Engineering Institute*, 2013. [En línea]. Disponible en: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=40201
[43] D. Preuveneers, W. Joosen, J. Bernal Bernabe, y A. Skarmeta, «Distributed Security Framework for Reliable Threat Intelligence Sharing», *Security and Communication Networks*, vol. 2020, pp. 1-15, ago. 2020, doi: 10.1155/2020/8833765.
[44] L. Gordon, M. Loeb, y L. Zhou, «Integrating cost–benefit analysis into the NIST Cybersecurity Framework via the Gordon–Loeb Model», *Journal of Cybersecurity*, vol. 6, ene. 2020, doi: 10.1093/cybsec/tyaa005.
[45] «International Organization of Standards ISO/IEC 27000:2018, "Information security, cybersecurity and privacy protection». 2018. [En línea]. Disponible en: https://www.iso.org/standard/73906.htmlg
[46] Julian López-Muñoz, *Manual de inteligencia*. Tirant Lo Blanch, 2019.
[47] Wilson Bautista, *Practical Cyberintelligence*. Packt Publishing, 2018.
[48] C. M. Bishop, *Pattern recognition and machine learning*. en Information science and statistics. New York: Springer, 2006.
[49] M. Glassman y M. J. Kang, «Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT)», *Computers in Human Behavior*, vol. 28, n.º 2, pp. 673-682, 2012, doi: https://doi.org/10.1016/j.chb.2011.11.014.
[50] T. M. Corporation, «Structured Threat Information eXpression (STIX™)», Accedido: 22 de junio de 2023. [En línea]. Disponible en: https://makingsecuritymeasurable.mitre.org/docs/stix-intro-handout.pdf
[51] T. M. Corporation, «Trusted Automated eXchange of Indicator Information — TAXII™», Accedido: 22 de junio de 2023. [En línea]. Disponible en: https://makingsecuritymeasurable.mitre.org/docs/taxii-intro-handout.pdf
[52] T. Mattern, J. Felker, R. Borum, y G. Bamford, «Operational Levels of Cyber Intelligence», *null*, vol. 27, n.º 4, pp. 702-719, dic. 2014, doi: 10.1080/08850607.2014.924811.
[53] P. Cichonski, T. Millar, T. Grance, y K. Scarfone, «Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology», National Institute of Standards and Technology, NIST SP 800-61r2, ago. 2012. doi: 10.6028/NIST.SP.800-

61r2.
[54] P. M. Mell, T. Bergeron, y D. Henning, «Creating a patch and vulnerability management program», National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-40ver2, 2005. doi: 10.6028/NIST.SP.800-40ver2.
[55] N. Naik, P. Jenkins, P. Grace, y J. Song, «Comparing Attack Models for IT Systems: Lockheed Martin's Cyber Kill Chain, MITRE ATT&CK Framework and Diamond Model», en *2022 IEEE International Symposium on Systems Engineering (ISSE)*, oct. 2022, pp. 1-7. doi: 10.1109/ISSE54508.2022.10005490.
[56] S. Caltagirone, A. Pendergast, y C. Betz, «The Diamond Model of Intrusion Analysis», *Hanover, MD: Center for Cyber Threat Intelligence and Threat Research 5 July 2013*, p. 62.
[57] Y. Shin *et al.*, «Alert correlation using diamond model for cyber threat intelligence», en *Proceedings of the European Conference on Cyber Warfare and Security*, Academic Conferences International Limited Oxfordshire, UK, 2019, pp. 444-450.
[58] E. M. Hutchins, M. J. Cloppert, y R. M. Amin, «Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains», *Leading Issues in Information Warfare & Security Research, 1(1), 80*, p. 14.
[59] C. D. Means, «Applying Cognitive Work Analysis to Time Critical Targeting Functionality», p. 161.
[60] Y. Ahmed, A. Taufiq, y R. Md Arafatur, «A cyber kill chain approach for detecting advanced persistent threats», *Computers, Materials and Continua*, vol. 67, n.º 2, pp. 2497-2513, 2021.
[61] H. Alavizadeh, J. Jang-Jaccard, T. Alpcan, y S. A. Camtepe, «A Markov Game Model for AI-based Cyber Security Attack Mitigation», *arXiv:2107.09258 [cs]*, jul. 2021, Accedido: 28 de febrero de 2022. [En línea]. Disponible en: http://arxiv.org/abs/2107.09258
[62] National Institute of Standards and Technology, «Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1», National Institute of Standards and Technology, Gaithersburg, MD, NIST CSWP 04162018, abr. 2018. doi: 10.6028/NIST.CSWP.04162018.
[63] A. Shostack, «Experiences Threat Modeling at Microsoft», ene. 2008.
[64] J. Wynn *et al.*, «Threat assessment and remediation analysis (tara)», *MITRE Corporation*, 2014, Accedido: 4 de agosto de 2023. [En línea]. Disponible en: https://apps.dtic.mil/sti/pdfs/AD1016629.pdf
[65] K. Kanakogi *et al.*, «Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information», *Applied Sciences*, vol. 12, n.º 7, 2022, doi: 10.3390/app12073400.
[66] P. E. Kaloroumakis y M. J. Smith, «Toward a Knowledge Graph of Cybersecurity Countermeasures», p. 11.
[67] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, y J. Disso, «Cyber-Attack Modeling Analysis Techniques: An Overview», en *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, ago. 2016, pp. 69-76. doi: 10.1109/W-FiCloud.2016.29.
[68] Centro Criptológico Nacional, «Resultados informe INES CCN». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://www.ccn-cert.cni.es/soluciones-seguridad/ines/resultado-general.html
[69] «ThreatConnect IOCs Wizard Spider». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://threatconnect.com/blog/threatconnect-research-roundup-threat-intelligence-update/
[70] Mandiant, «Going ATOMIC: Clustering and Associating Attacker Activity at Scale». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://www.mandiant.com/resources/clustering-and-associating-attacker-activity-at-scale

[71] ANSSI, «Ryuk Ransomware». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-006.pdf
[72] CISA, «Alert. Ransomware Activity Targeting the Healthcare and Public Health Sector». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://www.cisa.gov/uscert/ncas/alerts/aa20-302a
[73] «Fact sheet: Trickbot malware». Accedido: 29 de marzo de 2022. [En línea]. Disponible en: https://www.cisa.gov/uscert/sites/default/files/publications/TrickBot_Fact_Sheet_508.pdf
[74] Thai CERT EDTA, «THREAT GROUP CARDS: A THREAT ACTOR ENCYCLOPEDIA». junio de 2019.
[75] M. Tatam, B. Shanmugam, S. Azam, y K. Kannoorpatti, «A review of threat modelling approaches for APT-style attacks», *Heliyon*, vol. 7, n.º 1, p. e05969, ene. 2021, doi: 10.1016/j.heliyon.2021.e05969.
[76] L. Hernández-Álvarez, E. Barbierato, S. Caputo, L. Mucchi, y L. Hernández Encinas, «EEG Authentication System Based on One- and Multi-Class Machine Learning Classifiers», *Sensors*, vol. 23, n.º 1, Art. n.º 1, ene. 2023, doi: 10.3390/s23010186.
[77] H. Yang, Y. Jia, W.-H. Han, Y.-P. Nie, S.-D. Li, y X.-J. Zhao, «Calculation of Network Security Index Based on Convolution Neural Networks», en *Artificial Intelligence and Security*, X. Sun, Z. Pan, y E. Bertino, Eds., en Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 530-540. doi: 10.1007/978-3-030-24271-8_47.
[78] N. Marir, H. Wang, G. Feng, B. Li, y M. Jia, «Distributed Abnormal Behavior Detection Approach Based on Deep Belief Network and Ensemble SVM Using Spark», *IEEE Access*, vol. 6, pp. 59657-59671, 2018, doi: 10.1109/ACCESS.2018.2875045.
[79] L. Kong, G. Huang, K. Wu, Q. Tang, y S. Ye, «Comparison of Internet Traffic Identification on Machine Learning Methods», en *2018 International Conference on Big Data and Artificial Intelligence (BDAI)*, jun. 2018, pp. 38-41. doi: 10.1109/BDAI.2018.8546682.
[80] S. J. (Stuart J. Russell, *Artificial intelligence : a modern approach*. Third edition. Upper Saddle River, N.J. : Prentice Hall, [2010] ©2010, 2010. [En línea]. Disponible en: https://search.library.wisc.edu/catalog/9910082172502121
[81] M. I. Jordan y T. M. Mitchell, «Machine learning: Trends, perspectives, and prospects», *Science*, vol. 349, n.º 6245, pp. 255-260, 2015.
[82] C. Janiesch, P. Zschech, y K. Heinrich, «Machine learning and deep learning», *Electronic Markets*, vol. 31, n.º 3, pp. 685-695, sep. 2021, doi: 10.1007/s12525-021-00475-2.
[83] C. Roadknight, U. Aickelin, G. Qiu, J. Scholefield, y L. Durrant, «Supervised Learning and Anti-learning of Colorectal Cancer Classes and Survival Rates from Cellular Biology Parameters», p. 6.
[84] P. Geurts, A. Irrthum, y L. Wehenkel, «Supervised learning with decision tree-based methods in computational and systems biology», p. 18.
[85] A. E. Ezugwu *et al.*, «A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects», *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104743, 2022, doi: https://doi.org/10.1016/j.engappai.2022.104743.
[86] R. S. Sutton y A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[87] C. J. C. Burges, «A tutorial on support vector machines for pattern recognition», *Data Mining and Knowledge Discovery*, vol. 2, n.º 2, pp. 121-167, 1998, doi: 10.1023/A:1009715923555.
[88] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.
[89] H. I. Rhys, *Machine Learning with R, the tidyverse, and mlr*. Manning Publications, 2020. [En línea]. Disponible en: https://books.google.es/books?id=BoeryQEACAAJ

[90] P. A. Devijver y J. Kittler, *Pattern Recognition: A Statistical Approach*. Prentice/Hall International, 1982. [En línea]. Disponible en: https://books.google.es/books?id=Em9QAAAAMAAJ
[91] K. Duan, S. S. Keerthi, y A. N. Poo, «Evaluation of simple performance measures for tuning SVM hyperparameters», *Neurocomputing*, vol. 51, pp. 41-59, 2003, doi: 10.1016/S0925-2312(02)00601-X.
[92] L. Yang y A. Shami, «On hyperparameter optimization of machine learning algorithms: Theory and practice», *Neurocomputing*, vol. 415, pp. 295-316, nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
[93] J. R. Quinlan, «Induction of decision trees», *Mach Learn*, vol. 1, n.º 1, pp. 81-106, mar. 1986, doi: 10.1007/BF00116251.
[94] M. Alloghani, D. Al-Jumeily, A. J. Hussain, J. Mustafina, T. Baker, y A. J. Aljaaf, «Implementation of Machine Learning and Data Mining to Improve Cybersecurity and Limit Vulnerabilities to Cyber Attacks», en *Nature-Inspired Computation in Data Mining and Machine Learning*, 2020. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:202670695
[95] A. V. Joshi, *Machine Learning and Artificial Intelligence*. Springer International Publishing, 2019. [En línea]. Disponible en: https://books.google.es/books?id=rsqExgEACAAJ
[96] Y. Freund y R. E. Schapire, «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting», *Journal of Computer and System Sciences*, vol. 55, n.º 1, pp. 119-139, ago. 1997, doi: 10.1006/jcss.1997.1504.
[97] L. Breiman, «Random forests», *Machine learning*, vol. 45, pp. 5-32, 2001.
[98] J. Read, B. Pfahringer, G. Holmes, y E. Frank, «Classifier Chains for Multi-label Classification», en *Machine Learning*, ago. 2009, pp. 254-269. doi: 10.1007/978-3-642-04174-7_17.
[99] T. Fawcett, «An introduction to ROC analysis», *Pattern Recognition Letters*, vol. 27, n.º 8, pp. 861-874, jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[100] «Working with ATT&CK | MITRE ATT&CK®». Accedido: 11 de abril de 2023. [En línea]. Disponible en: https://attack.mitre.org/resources/working-with-attack/
[101] Y. Shin, K. Kim, J. J. Lee, y K. Lee, «Focusing on the Weakest Link: A Similarity Analysis on Phishing Campaigns Based on the ATT&CK Matrix», *Security and Communication Networks*, vol. 2022, p. 1699657, abr. 2022, doi: 10.1155/2022/1699657.
[102] J. Cohen, P. Cohen, S. G. West, y L. S. Aiken, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Taylor & Francis, 2013. [En línea]. Disponible en: https://books.google.es/books?id=fAnSOgbdFXIC
[103] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, y J. Attenberg, «Feature hashing for large scale multitask learning», en *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1113-1120.
[104] K. P. F.R.S, «LIII. On lines and planes of closest fit to systems of points in space», *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, n.º 11, pp. 559-572, 1901, doi: 10.1080/14786440109462720.
[105] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, 2005. [En línea]. Disponible en: https://books.google.es/books?id=O_qHDLaWpDUC
[106] H. Cramér, *Mathematical methods of statistics*, vol. 26. Princeton university press, 1999.
[107] H. He y E. A. Garcia, «Learning from Imbalanced Data», *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, n.º 9, pp. 1263-1284, 2009, doi: 10.1109/TKDE.2008.239.
[108] R. Goorbergh, M. van Smeden, D. Timmerman, y B. Van Calster, «The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression», *Journal of the American Medical Informatics Association : JAMIA*, vol. 29, jun.

2022, doi: 10.1093/jamia/ocac093.
[109] N. V. Chawla, N. Japkowicz, y A. Kotcz, «Special issue on learning from imbalanced data sets», *ACM SIGKDD explorations newsletter*, vol. 6, n.º 1, pp. 1-6, 2004.
[110] N. V. Chawla, K. W. Bowyer, L. O. Hall, y W. P. Kegelmeyer, «SMOTE: synthetic minority over-sampling technique», *Journal of artificial intelligence research*, vol. 16, pp. 321-357, 2002.
[111] H. He, Y. Bai, E. A. Garcia, y S. Li, «ADASYN: Adaptive synthetic sampling approach for imbalanced learning», en *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 1322-1328.
[112] H. Han, W.-Y. Wang, y B.-H. Mao, «Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning», en *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, Springer, 2005, pp. 878-887.
[113] A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, y H. Kashima, «Improving imbalanced classification using near-miss instances», *Expert Systems with Applications*, vol. 201, p. 117130, 2022.
[114] «Two Modifications of CNN», *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, n.º 11, pp. 769-772, 1976, doi: 10.1109/TSMC.1976.4309452.
[115] «An Experiment with the Edited Nearest-Neighbor Rule», *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, n.º 6, pp. 448-452, 1976, doi: 10.1109/TSMC.1976.4309523.
[116] J. Laurikkala, «Improving identification of difficult small classes by balancing class distribution», en *Artificial Intelligence in Medicine: 8th Conference on Artificial Intelligence in Medicine in Europe, AIME 2001 Cascais, Portugal, July 1–4, 2001, Proceedings 8*, Springer, 2001, pp. 63-66.
[117] N. Rout, D. Mishra, y M. K. Mallick, «Handling imbalanced data: a survey», en *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications: ASISA 2016*, Springer, 2018, pp. 431-443.
[118] G. E. Batista, A. L. Bazzan, y M. C. Monard, «Balancing training data for automated annotation of keywords: a case study.», en *WOB*, 2003, pp. 10-18.
[119] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge: Cambridge University Press, 2012. doi: 10.1017/CBO9780511973000.
[120] J. West, D. Ventura, y S. Warnick, «Spring research presentation: A theoretical foundation for inductive transfer», *Brigham Young University, College of Physical and Mathematical Sciences*, vol. 1, n.º 08, 2007.
[121] I. Goodfellow *et al.*, «Generative adversarial networks», *Communications of the ACM*, vol. 63, n.º 11, pp. 139-144, 2020.
[122] H. Yu, I. Mineyev, L. R. Varshney, y J. A. Evans, «Learning from One and Only One Shot», *arXiv preprint arXiv:2201.08815*, 2022.
[123] N. Lunardon, G. Menardi, y N. Torelli, «ROSE: a Package for Binary Imbalanced Learning», *R Journal*, vol. 6, pp. 79-89, jun. 2014, doi: 10.32614/RJ-2014-008.
[124] S. AL y M. DENER, «STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment», *Computers & Security*, vol. 110, p. 102435, ago. 2021, doi: 10.1016/j.cose.2021.102435.
[125] L. Torgo, P. Branco, R. P. Ribeiro, y B. Pfahringer, «Resampling strategies for regression», *Expert Systems*, vol. 32, n.º 3, pp. 465-476, 2015.
[126] A. Dal Pozzolo, O. Caelen, R. A. Johnson, y G. Bontempi, «Calibrating probability with undersampling for unbalanced classification», en *2015 IEEE symposium series on computational intelligence*, IEEE, 2015, pp. 159-166.
[127] «scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation».

Accedido: 11 de abril de 2023. [En línea]. Disponible en: https://scikit-learn.org/stable/
[128] L. Yu y H. Liu, «Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution», en *Proceedings, Twentieth International Conference on Machine Learning*, ene. 2003, pp. 856-863.