Sistema de caracterización de técnicas MITRE ATT&CK en incidentes de ciberseguridad

Carmen Sánchez-Zas, Xavier Larriva-Novo, Víctor A. Villagrá, Diego Rivera, Sonia Solera-Cotanilla
Universidad Politécnica de Madrid (UPM). DIT, ETSI Telecomunicaciones. Avda. Complutense 30, 28040 Madrid
{carmen.szas, xavier.larriva.novo, victor.villagra, diego.rivera, sonia.solera}@upm.es

***Resumen*—En la actualidad, la ciberseguridad es un aspecto crítico debido a la complejidad creciente de los incidentes. Por ello son necesarios sistemas que identifiquen y contrarresten los riesgos eficazmente. En esta propuesta se presenta un sistema de aprendizaje automático supervisado capaz de analizar registros de tráfico para caracterizar técnicas y tácticas MITRE ATT&CK. A partir de esa identificación, se almacena en una ontología que actúa como base de conocimiento y tiene la capacidad de razonamiento necesaria para la toma de decisiones y recomendación de contramedidas. Los resultados obtenidos permiten identificar correctamente una serie de tácticas y técnicas y proporcionar las recomendaciones específicas de MITRE para reaccionar frente a los incidentes, ofreciendo una respuesta optimizada ante incidentes de seguridad, fortaleciendo la seguridad en las organizaciones, contribuyendo en los procesos de gestión de riesgo.**

***Index Terms*—MITRE ATT&CK, Ciberseguridad, TTPs, Aprendizaje Automático, Contramedidas.**

Tipo de contribución: *Investigación original*

I. Introducción

En los últimos años, la evolución en las herramientas de ciberseguridad ha provocado que los ataques crezcan en complejidad, según mejoran las protecciones disponibles. En esta carrera, la inteligencia artificial juega un papel fundamental en la necesidad de desarrollar sistemas avanzados que identifiquen incidentes. En este contexto, el marco de referencia MITRE ATT&CK [1] permite entender la sofisticación de los ataques, y su implementación permite clasificar las tácticas, técnicas y procedimientos (TTPs) empleados por los atacantes.

El concepto de *Cyber Threat Hunting* (CTH) [2] define un proceso en el que activamente se identifican e interrumpen amenazas de ciberseguridad y se mejoran las medidas de seguridad para defenderse ante futuras amenazas. Para reaccionar frente a este crecimiento y las nuevas formas de intrusión se necesitan sistemas avanzados capaces no solo de identificar esas intrusiones, sino también de clasificar las tácticas y técnicas utilizadas por los atacantes y así poder recomendar contramedidas específicas para salvaguardar el sistema y preparar el entorno para reaccionar ante las posibles amenazas [3].

Enmarcado dentro del ámbito de CTH, en este artículo se aborda la necesidad de un sistema de aprendizaje automático que, analizando registros de tráfico, permita detectar técnicas MITRE ATT&CK, almacenando la información en una ontología para la recomendación óptima de contramedidas en tiempo real. Este enfoque combina la eficiencia del aprendizaje automático en la detección de amenazas con la capacidad de almacenar y estructurar la información de una ontología, proporcionando una base sólida para la toma de decisiones informadas, mejorando la capacidad de respuesta y fortaleciendo la ciberseguridad en los sistemas protegidos.

La novedad de este sistema radica en su capacidad, no solo para detectar amenazas, sino para identificar dentro de esos registros de tráfico las técnicas de la matriz MITRE ATT&CK que representan, y poder así proponer contramedidas de acuerdo a la amenaza recibida. A diferencia de las propuestas tradicionales, mediante firmas y reglas, la aplicación de modelos de aprendizaje automático supervisado multiclase en este tipo de tareas permite extraer relaciones entre los datos que no se perciben a simple vista y, por tanto, identificar técnicas en registros de tráfico que aparentemente no encajan con ninguna de las reglas establecidas, o en casos donde la definición de estas reglas no es posible. Tras probar varios modelos, los algoritmos supervisados multi-clase con mejor resultado son los de tipo árbol de decisión y, por tanto, los que se explorarán en este desarrollo.

A lo largo del documento presentaremos el entorno de la propuesta y los conceptos principales en los que se basa el desarrollo. A continuación, detallaremos la arquitectura de la propuesta, el desarrollo del sistema y los resultados. En la sección I se presenta el entorno del trabajo. A continuación, en la sección II, los trabajos relacionados. En la sección III los conceptos teóricos relacionados con el proyecto. La propuesta se define en la sección IV y se desarrolla en la sección V. Finalmente, los resultados obtenidos se presentan en la sección VI y las conclusiones y líneas futuras en la sección VII.

II. Trabajos Relacionados

En el análisis de la literatura revisada se abordan diversas facetas del marco MITRE ATT&CK. Encontramos múltiples métodos para detectar TTPs en el el ámbito de esta propuesta. Estos trabajos se resumen en esta sección. En [4], los autores presentan un conjunto de reglas de asociación para llevar a cabo procesos de atribución en inteligencia de amenazas. Uno de los problemas principales a los que se enfrenta este tipo de proyectos son las grandes cantidades de datos a analizar, que pueden impedir encontrar información relevante. Por ello, introducen un proceso de minado de datos asociado que pueda encontrar relaciones entre datasets y llevar a cabo el proceso de atribución del ciberataque en cuanto a inteligencia de amenazas, como por ejemplo identificar las tácticas y técnicas empleadas.

Otra de las aproximaciones ampliamente utilizada es la aplicación de grafos de conocimiento. En [5] se presenta un modelo de datos basado en el repositorio de MITRE (CAR) [6] que, combinado con reglas de inferencia es capaz de detectar técnicas de ataque y, mediante modelos de

aprendizaje automático entrenados sobre este grafo, puede predecir nuevas amenazas. Los autores en [7], a partir de un grafo de conocimiento buscan mejorar la eficiencia en la detección temprana de riesgos de los sistemas ciber-físicos, a partir de un método de evaluación de riesgos basado en MITRE ATT&CK. Del grafo obtienen la probabilidad de que un indicador conlleve una amenaza, y calculan el valor de riesgo causado por las distintas tácticas y técnicas. Por otro lado, los autores de RADAR [8][9] presentan un sistema basado en la ontología de TTPs de MITRE ATT&CK para identificar y clasificar comportamiento malicioso. El objetivo es entrenar modelos de aprendizaje automático para detectar y clasificar TTPs, con sus correspondientes explicaciones. Identifica cada muestra con una TTP y trata de predecir si es maliciosa o no. Está entrenado para detectar tácticas y técnicas de *Reconnaissance* (T1590), *Credential Access* (T1557.001), *Discovery* (T1124 y T1135), *Lateral Movement* (T1021.001/4, T1550.003, T1563.001/2, 1570), y *Command and Control* (T1071, T1090, T1105, T1571, T1053), con la posibilidad de extenderse. Además, aprovechando el sistema, realizan una comprobación para verificar que, al añadir información de TTPs en los datos de entrenamiento, se mejora la detección de amenazas.

Profundizando en la detección con modelos entrenados, encontramos múltiples propuestas con técnicas de aprendizaje no supervisado. En [10] los autores proponen agrupar técnicas mediante un clustering jerárquico con un 95 % de acierto al inferir las tácticas. Así, al detectar una técnica, se pueden inferir aquellas que están relacionadas. Sin embargo, en [11] se utilizan redes neuronales para la detección de ataques a través de distintas etapas del marco MITRE.

Una de las principales aplicaciones de la identificación de tácticas es la capacidad de asociarlas con las vulnerabilidades que pueden explotar. Mediante modelos de aprendizaje automático [12] se pueden identificar múltiples etiquetas asociadas a las tácticas utilizadas para explotar vulnerabilidades, lo que permite priorizar las estrategias de defensa frente a estos posibles ataques. También, los autores en [13] buscan predecir las técnicas más probables para explotar una vulnerabilidad utilizando procesamiento del lenguaje natural para mapearlas, con el objetivo de priorizar riesgos y correlacionar los eventos e incidentes que ocurren con la vulnerabilidad.

El procesamiento de lenguaje natural es una de las técnicas más empleadas [14], especialmente entrenando a partir del dataset de ENISA que permite asociar vulnerabilidades y técnicas. Sin embargo, a pesar de encontrar múltiples trabajos que aprovechan las ventajas de la inteligencia artificial, el entorno en el que se desarrolla esta propuesta es, por naturaleza, desbalanceado, ya que no todos los ataques se dan en el mismo porcentaje, y por lo tanto, los conjuntos de datos disponibles para entrenamiento tienen un tamaño de las clases muy descompensado [15]. La solución principal a este problema es el pre-procesado de los datos, como en [16], donde el método de clasificación de TTPs en modelos de procesamiento del lenguaje natural con un tratamiento correcto de los datos, permite pasar de un 61,26 % de precisión en el dataset TRAM a un 98.76 % en el mismo dataset.

De la amplitud de trabajos e investigaciones en las que se aplica el marco MITRE ATT&CK, por el contexto de esta propuesta, las referencias principales se encuentran en aquellos que permitan identificar tácticas y técnicas a partir de registros de tráfico, donde la mayoría utilizan técnicas de aprendizaje automático [17]. En concreto, encontramos un dataset de nuevo desarrollo, que no tiene mucha literatura relacionada, UWF-ZeekDataFall22 [18] con el objetivo principal de identificar las tácticas de *Resource Development* (TA0042), *Reconnaissance* (TA0043) y *Discovery* (TA0007). El objetivo principal es clasificar las tácticas, no los ataques, obteniendo un 100 % de precisión en clasificación binaria entrenando modelos individuales para detectar cada táctica, y un 99.99 % en la clasificación multiclase. Existe un antecedente a este dataset, UWF-ZeekData22 [19] que incluye menor volumen de tráfico de estas tácticas, y por tanto se centra en las principales (*Reconnaissance* y *Discovery*), que permite un 99.4 % de precisión en la detección de *Reconnaissance* y 99.95 % en *Discovery*, entrenando también modelos binarios para identificar si el registro representa o no esas tácticas [20]. Además, mediante un grafo, representa la táctica de reconocimiento en busca de patrones que permitan identificarla y etiquetarla [21].

Algunos de los trabajos relacionados con el conjunto de datos UWF-ZeekData22 se centran en el tratamiento de datasets no balanceados en el campo de la detección de ataques [22], y resaltan la necesidad de tratar estos datos con sub-muestreo y sobre-muestreo de las clases para obtener resultados coherentes. Los autores de [23] presentan un modelo que utiliza el dataset para entrenar la clasificación de malware. Dentro del pre-procesado se debe balancear el conjunto para poder realizar clasificaciones binarias e identificar con muy buenos resultados las tácticas de *Credential Access*, *Discovery*, *Lateral Movement*, *Reconnaissance*, *Resource development* y el tráfico benigno. A pesar de que las tácticas de *Exfiltration* y *Privilege Escalation* se confunden entre ellas, el resultado total es positivo.

La base del desarrollo de esta propuesta será, por tanto, a partir del dataset UWF-ZeekDataFall22, entrenar modelos de aprendizaje automático para identificar técnicas con un algoritmo multi-clase, sin perder precisión a la hora de identificar estas tácticas, que es un aspecto muy poco abordado hasta el momento.

De todos los trabajos analizados, únicamente hemos podido extraer información relacionada con el número de tácticas y técnicas en los que se recogen en la Tabla I, donde se presenta el método utilizado para la identificación y cuántas tácticas y técnicas son capaces de reconocer en los resultados de las investigaciones presentadas.

Tabla I
RESUMEN DE LOS RESULTADOS DE TRABAJOS PREVIOS

Publicación	Método	N. Tácticas	N. Técnicas
[7]	Grafo	5	5
[8], [9]	Reglas	6	15
[18]	Algoritmos sup.	3	-
[20]	Algoritmos sup.	2	-
[23]	Algoritmos sup.	8	-

Como se puede observar, no existen, dentro de nuestro conocimiento, trabajos hasta la fecha que aborden de manera

efectiva la caracterización de técnicas específicas de la matriz MITRE mediante el análisis de registros de tráfico con modelos de aprendizaje automático. Esto supone una debilidad a la hora de defender un sistema frente a los ciberataques, que están en constante evolución. Este estudio plantea una posible solución a este problema, mejorando la postura de seguridad en el ámbito digital.

III. Marco Teórico

III-A. Marco MITRE ATT&CK

MITRE creó en 2013 ATT&CK [1] para documentar TTPs que las amenazas avanzadas persistentes usan para atacar redes. Es una base de conocimiento de tácticas y técnicas basada en observaciones reales de comportamiento de adversarios y una taxonomía de acciones adversarias a lo largo del ciclo de vida. La utilidad principal es el desarrollo de modelos de amenazas y su aplicación en distintas metodologías. El contenido de ATT&CK es abierto y disponible a cualquier persona. Se creó a partir de la necesidad de documentar los comportamientos adversarios para un proyecto de investigación y sirve como lenguaje común.

Se divide en dos partes: *Enterprise*, que cubre comportamiento contra redes empresariales y nubes; y *Mobile*, que se centra en el comportamiento contra dispositivos móviles.

Las tácticas representan el "porqué" de una técnica o subtécnica, es el objetivo táctico del adversario, el motivo para realizar la acción. Por otro lado, las técnicas representan "cómo" el atacante consigue el objetivo táctico al realizar la acción.

La lista de tácticas de ATT&CK *Enterprise* está compuesta de:

- *Reconnaissance*: El atacante está intentando recoger información que usará en futuras operaciones. Las técnicas de este tipo implican acciones que activa o pasivamente permitan adaptar el ataque al objetivo.
- *Resource Development*: El adversario trata de establecer recursos que den soporte a sus operaciones. Estas técnicas implican crear o comprometer recursos que apoyen sus objetivos.
- *Initial Access*: El atacante trata de acceder a la red. Las técnicas usan distintos vectores de entrada para conseguir su punto de apoyo dentro de la red.
- *Execution*: El adversario trata de ejecutar código malicioso de forma local o remota. Normalmente suelen acompañar otras tácticas que busquen resultados más amplios.
- *Persistence*: El atacante trata de mantener su punto de acceso en casos de reinicio o cambios de credenciales.
- *Privilege Escalation*: El adversario busca conseguir permisos de mayor nivel en una red o sistema. Pueden acceder a una red sin permisos pero para conseguir sus objetivos se requieren perfiles con permisos elevados.
- *Defense Evasion*: El atacante trata de evitar ser detectado.
- *Credential Access*: El adversario trata de obtener nombres de usuario y contraseñas.
- *Discovery*: El atacante trata de conocer el entorno (sistema y red interna), lo que le permita orientarse antes de decidir cómo actuar, cómo podría entrar o qué puede controlar.
- *Lateral Movement*: El adversario busca moverse a través del entorno. Estas tácticas buscan entrar y controlar sistemas remotos de la red.
- *Collection*: El atacante trata de recoger información de interés para su objetivo. Normalmente, el siguiente objetivo es extraer los datos del sistema.
- *Command and Control*: El adversario busca comunicarse con los sistemas comprometidos dentro de la red víctima y controlarlos.
- *Exfiltration*: El atacante busca robar datos de la red.
- *Impact*: El adversario trata de manipular, interrumpir o destruir el sistema o los datos. Estas técnicas buscan interrumpir la disponibilidad o comprometer la integridad manipulando procesos operacionales.

La lista de técnicas de cada táctica es extensa y se puede consultar en la página oficial de ATT&CK [1]. Las que se utilizarán en el sistema de la propuesta se presentarán en la siguiente sección.

III-B. Aprendizaje Automático Supervisado

El aprendizaje automático [24] es una herramienta con la capacidad de extraer información de un conjunto de datos y crear un modelo de datos a partir del que extraer conclusiones. Los algoritmos utilizados para entrenar los modelos pueden clasificarse en dos tipos, en función de si los datos utilizados como entrada están etiquetados (aprendizaje supervisado) o no (aprendizaje no supervisado).

En esta propuesta se utilizarán modelos supervisados, que permiten llevar a cabo tareas de regresión y clasificación, ya que contamos con un conjunto de datos ya clasificado. Estos algoritmos [25] son programas capaces de ajustarse a los datos que reciben para mejorar los resultados. Para el desarrollo se utilizarán modelos basados en árboles como los siguientes:

- *Decision Trees* [26]: Modelo basado en árbol en el que cada nodo representa una prueba sobre un atributo y cada rama un posible resultado, hasta que se alcanza una predicción.
- *Random Forest* [25]: Este algoritmo está compuesto por múltiples árboles de decisión, cada uno creado con un conjunto de las características de los datos de entrada. Cada árbol vota la clasificación más probable según su análisis, y el algoritmo computa todos los resultados para elegir la predicción.
- *Extreme Gradient Boosting* (XGBoost) [27]: Este algoritmo consiste en una implementación de árboles de decisión de gradiente reforzado. Este método comienza con clasificadores débiles sobre un conjunto de datos para mejorar los resultados a partir de un procesamiento secuencial cuya función de pérdidas es minimizar el error con cada iteración, obteniendo al final un modelo más fuerte.

Existen otros algoritmos de aprendizaje supervisado, que utilizan metodologías distintas para alcanzar la decisión final. Sin embargo, en la literatura previa se definió el conjunto de modelos basados en árboles como los más adecuados para tratar este tipo de problemáticas.

III-C. Balanceo de datos

Los datos desequilibrados son una característica propia de los entornos de ciberseguridad, ya que se dan cuando las distintas clases no se representan en la misma proporción en el dataset, como es el caso de los ataques de ciberseguridad. La problemática principal del desequilibrio de datos surge principalmente al aplicar modelos de aprendizaje automático, ya que estos suelen sobre-ajustarse a los datos mayoritarios, ignorando las clases menos representadas.

Para corregirlo, el caso ideal sería capturar más datos, pero cuando esto no es posible, es necesario re-muestrear el dataset, añadiendo copias de los datos menos frecuentes (sobre-muestreo aleatorio), o eliminando algunos de los datos pertenecientes a la clase más frecuente (sub-muestreo aleatorio). Sin embargo, la solución más eficiente es la generación de muestras sintéticas, con el método *Synthetic Minority Oversampling Technique* (SMOTE) [28]. Este algoritmo busca muestras similares en distancia, las conecta y crea nuevas instancias a partir de los puntos entre las muestras originales.

III-D. Dataset UWF-ZeekDataFall22

En cuanto al conjunto de datos de entrenamiento, UWF-ZeekDataFall22 [18], disponible en [29], ya se presentó brevemente en la Sección II. Es un dataset nuevo, apenas explotado, que destaca entre el resto de literatura analizada ya que, sin necesidad de aplicar modelos o algoritmos propios, permite la comparación directa con otros conjuntos o estudios, estableciendo una base para el entrenamiento de modelos supervisados que puede validarse con distintos datasets.

Contiene información proporcionada por Zeek de las siguientes columnas: 'community_id', 'conn_state', 'duration', 'history', 'src_ip_zeek', 'src_port_zeek', 'dest_ip_zeek', 'dest_port_zeek', 'local_orig', 'local_resp', 'missed_bytes', 'orig_bytes', 'orig_ip_bytes', 'orig_pkts', 'proto', 'resp_bytes', 'resp_ip_bytes', 'resp_pkts', 'service', 'ts', 'uid', 'datetime', 'label_tactic', 'label_technique' y 'label_binary'.

El reparto de tráfico inicial del conjunto de datos será 346.933 entradas marcadas como tráfico malicioso ('label_binary' igual a *True*), 350.339 registros normales ('label_binary' igual a *False*) y 3.068 filas marcadas como duplicadas. En relación con las tácticas explicadas en el apartado anterior, aparecen representadas *Reconnaissance* (TA0043), *Initial Access* (TA0001), *Discovery* (TA0007), *Command and Control* (TA0011), *Lateral Movement* (TA0008), *Collection* (TA00099), *Persistence* (TA0003), *Execution* (TA0002), *Defense Evasion* (TA0005) y *Resource Development* (TA0042). El dataset no incluye información de las tácticas *Privilege Escalation*, *Credential Access*, *Exfiltration* e *Impact*, por lo que se excluyen de la caracterización. El reparto de registros para cada una se presentará en la Sección V junto al preprocesado de los datos.

Por último, las técnicas incluidas en el dataset serán las siguientes [1]:

- *Non-Standard Port* (T1571): los atacantes utilizan un protocolo y un puerto que, por defecto, no están asociados.
- *Abuse Elevation Control Mechanism* (T1548): Los adversarios pueden evitar los mecanismos de control de la elevación de privilegios.
- *Gather Victim Network Information* (T1590): Los atacantes recogen información de la red víctima.
- *Boot or Logon Autostart Execution* (T1547): Los adversarios pueden configurar un sistema para ejecutar un programa al inicio.
- *Modify Registry* (T1112): Los atacantes interactúan con el registro Windows para ocultar información de configuración en las claves del registro o eliminar información.
- *Develop Capabilities* (T1587): El adversario construye sus capacidades.
- *Network Service Discovery* (T1046): Los atacantes obtienen una lista de los servicios corriendo en equipos remotos o en dispositivos locales de la red.
- *Exploitation for Client Execution* (T1203): Los adversarios explotan vulnerabilidades de las aplicaciones para ejecutar código.
- *User Execution* (T1204): Cualquier adversario recurre a acciones específicas del usuario para conseguir la ejecución.
- *Exploitation of Remote Services* (T1210): Los atacantes explotan servicios remotos para conseguir acceso no autorizado a sistemas internos dentro de una red.
- *Application Layer Protocol* (T1071): Los adversarios se comunican usando la pila de protocolos OSI para evitar la detección al mezclarse con el resto del tráfico.
- *Create Account* (T1136): Los atacantes crean una cuenta para mantener el acceso al sistema víctima.
- *Command and Scripting Interpreter* (T1059): Los adversarios abusan de los intérpretes de comandos y *scripts* para ejecutar otros comandos o *scripts*.
- *Adversary-in-the-Middle* (T1557): Los atacantes se colocan entre dos o más dispositivos de una red para llevar a cabo comportamientos como *sniffing* o manipulación de datos.
- *Phishing* (T1566): Los adversarios envían mensajes engañosos y falsos para conseguir acceso al sistema víctima.
- *Server Software Component* (T1505): Los adversarios abusan de las características de los servidores para establecer un acceso persistente a los sistemas.
- *Exploit Public-Facing Application* (T1190): Los atacantes pueden explotar debilidades en equipos con acceso a Internet para acceder a una red.
- *Gather Victim Host Information* (T1592): Los adversarios recogen información del dispositivo de la víctima.
- *Active Scanning* (T1595): El atacante ejecuta escaneos activos de reconocimiento para obtener información.
- *Gather Victim Identity Information* (T1589): Los adversarios recogen información de la identidad de la víctima.
- *Event Triggered Execution* (T1546): Los atacantes pueden establecer persistencia o elevar privilegios utilizando mecanismos del sistema que desencadenan la ejecución basándose en eventos específicos.
- *External Remote Services* (T1133): Los adversarios aprovechan los servicios remotos externos para acceder a una red o permanecer en ella.

IV. Propuesta

Para afrontar la problemática identificada, la necesidad de detectar en registros de tráfico las distintas técnicas MITRE,

presentamos esta propuesta con el objetivo de caracterizar las tácticas y técnicas asociadas a *logs* que representan ataques según sus propiedades. Así, teniendo en cuenta la información recopilada por ATT&CK, poder ofrecer las contramedidas recomendadas por este marco frente a los incidentes. La propuesta puede integrarse con una ontología que lleve a cabo una gestión de riesgos como la presentada en [30] para aprovechar esta información y enriquecer y complementar procedimientos de análisis y gestión de riesgos.

En la Figura 1 presentamos la metodología propuesta para la solución planteada en esta investigación.

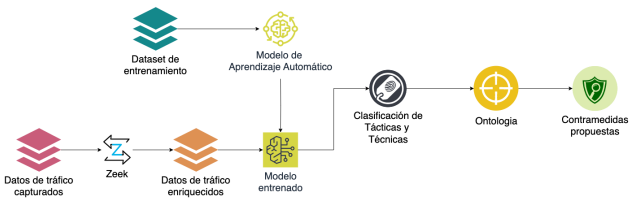

Figura 1. Metodología propuesta

El sistema captura tráfico de red en tiempo real, y complementa las propiedades identificadas aplicando un módulo Zeek [31]. Además, a partir de un modelo de aprendizaje automático entrenado, se busca caracterizar el tráfico con técnicas MITRE. Según el mismo marco, estas técnicas llevan asociadas unas contramedidas recomendadas para hacer frente a los ataques que las utilizan. Almacenando esta información en una ontología de gestión de riesgos como la mencionada anteriormente [30], se pueden plantear contramedidas concretas a los ataques identificados.

En este trabajo nos centraremos en presentar y detallar el desarrollo del sistema de aprendizaje automático que será integrado en la metodología definida anteriormente. El enriquecimiento de los datos utilizando Zeek es un trabajo previo que se realiza a partir de datos capturados en tiempo real, o a partir de capturas de otros datasets, hasta obtener las propiedades que se utilizaron en el entrenamiento, de modo que el modelo guardado para realizar la clasificación e identificación de TTPs funcione correctamente. Además, la ontología que permitiría la recomendación de contramedidas ya ha sido validada en [30].

Por tanto, el proceso de selección y entrenamiento de los modelos de aprendizaje supervisado comienza con el pre-procesado del conjunto de datos con el fin de obtener las características idóneas para el entrenamiento y la validación del modelo. A continuación se eligen los algoritmos, se optimizarán sus hiper-parámetros y se entrenarán para poder seleccionar el que mejor resultado obtenga a partir de los datos del dataset UWF-ZeekDataFall22.

V. Desarrollo

Para conseguir el objetivo propuesto de identificar tácticas y técnicas MITRE ATT&CK en registros de tráfico y obtener con ello una recomendación de contramedidas, el desarrollo realizado se divide en dos módulos: pre-procesado de los datos para adaptarlos al modelo, y evaluación de distintos algoritmos y entrenamiento del que mejor se ajuste al entorno de trabajo.

V-A. Pre-procesamiento del dataset

Como se presentó en la sección anterior, el dataset consta de información de tráfico enriquecida mediante un módulo Zeek y etiquetada con 10 tácticas distintas y las 22 técnicas presentadas. Sin embargo, por las características del entorno, no todas las clases están igualmente representadas en el conjunto de datos, y por lo tanto será necesario tratar los datos hasta que sean aptos para entrenar un modelo de aprendizaje automático.

En primer lugar, se unieron todos los archivos para obtener un dataset con los registros de todos los periodos de tiempo capturados y eliminamos los registros marcados como duplicados. Así, se obtienen finalmente 697.272 filas de tráfico benigno e incidentes. Después se eliminan los registros con valores *Not a Number (NaN)*, que no permiten entrenar los modelos y por tanto se mantienen las técnicas T1046, T1548, T1210, T1587, T1557, T1566, T1590, T1190, T1592, T1595, T1589 y T1071, que corresponden a las tácticas *Lateral Movement*, *Discovery*, *Initial Access*, *Collection*, *Command and Control*, *Resource Development*, *Reconnaissance* y *Defense Evasion*.

A continuación eliminamos las columnas con valores de identificación del registro '*uid*' y '*community_id*' y convertimos las columnas de valores no numéricos mediante codificadores de las etiquetas: *'conn_state'*, *'history'*, *'local_orig'*, *'local_resp'*, *'proto'*, *'service'*, *'label_tactic'*, *'label_technique'*, *'label_binary'*. Las columnas *'src_ip_zeek'*, *'dest_ip_zeek'* y *'datetime'* se convierten aplicando las librerías de Python correspondientes.

Dada la desproporción en la representación de clases, decidimos aplicar sobre-muestreo, primero de forma aleatoria y luego mediante la técnica SMOTE, de las clases que menos datos tienen; y sub-muestreo de las más frecuentes, eliminando las clases que únicamente contienen un registro, ya que no permiten crear muestras nuevas correctamente. El número de registros final por cada técnica se presenta en la Tabla II.

Tabla II
Registros por cada técnica tras el balanceo de datos

Categoría	Original	Sub-Muestreo	Sobre-muestreo aleatorio	Sobre-muestreo SMOTE
0 - Benigno	328026	3061	3061	3061
1 - T1046	10	10	100	200
2 - T1071	6	6	60	140
3 - T1190	3	3	30	126
4 - T1210	9	9	90	180
5 - T1548	3061	3061	3061	3061
6 - T1566	9	9	90	180
7 - T1587	113	113	113	226
8 - T1589	292	292	292	584
9 - T1590	21175	3061	3061	3061
10 - T1592	21400	3061	3061	3061
11 - T1595	37	37	75	150
T1557	1	-	-	-

Finalmente, se calcula la información que aporta cada columna para la clasificación multi-clase, manteniendo las que aportan más valor (*'ts'*, *'datetime'*, *'orig_ip_bytes'*, *'resp_ip_bytes'*, *'resp_bytes'*, *'duration'*, *'orig_bytes'*, *'orig_pkts'*. *'history'*, *'resp_pkts'*, *'conn_state'*), como se puede observar en la Figura 2.

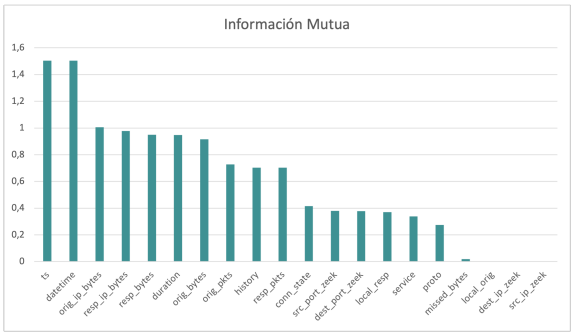

Figura 2. Información Mutua de las columnas del dataset

Finalmente se calcula la matriz de correlación del dataset, eliminando las columnas *'datetime'*, *'resp_ip_bytes'* y *'resp_pkts'* por tener un valor en la matriz de correlación mayor a 0.9, mostrada en la Figura 3.

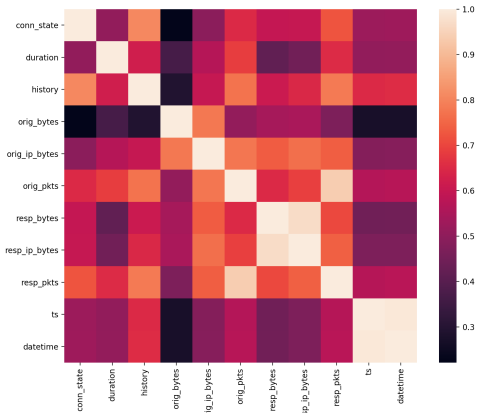

Figura 3. Matriz de correlación del dataset UWF-ZeekDataFall22

El dataset resultante con el número de muestras que se indica en la última columna de la Tabla II y las propiedades restantes se divide en un conjunto para entrenamiento y otro para validación, que se guardan en formato *Comma-Separated Values* (CSV), para utilizarlo en el entrenamiento y evaluación de los modelos junto a las columnas de etiquetas binaria, tácticas y técnicas.

V-B. Elección del algoritmo y entrenamiento del modelo

Para el entrenamiento del sistema hemos optado por entrenar modelos supervisados dado que en trabajos previos se ha demostrado el buen rendimiento que tienen en este tipo de tareas, en concreto los árboles de decisión.

Los algoritmos elegidos han sido *Decision Tree*, *Random Forest* y XGBoost. Los tres se entrenan con los mismos datos de entrada, en formato CSV, que se obtienen del preprocesamiento del apartado anterior y la columna de etiquetas correspondiente a las técnicas (*'label_technique'*).

Las librerías utilizadas serán las correspondientes a cada modelo en la clase Python de Scikit-learn [32]: *DecisionTreeClassifier*, *RandomForestClassifier* y *GradientBoostingClassifier*, respectivamente.

Los hiperparámetros de los algoritmos han sido optimizados mediante la técnica *Grid Search*. Los modelos, por tanto, han sido entrenados con las siguientes configuraciones:

- *Decision Tree*: Configuramos los hiperparámetros de la profundidad del árbol ('*max_depth*'), la aleatoriedad del estimador ('*random_state*') y la función para medir la calidad de una división ('*criterion*'). Los valores obtenidos de la optimización fueron '*max_depth*' = 10; '*random_state*' = 88; '*criterion*' = '*entropy*'.
- *Random Forest*: Para este algoritmo se configuran los hiperparámetros '*criterion*' y '*max_depth*' que se explicaron en el modelo anterior, y además el número de árboles en el bosque ('*n_estimators*'). En este caso se obtienen los valores '*criterion*' = '*entropy*'; '*max_depth*' = 9; '*n_estimators*' = 114.
- XGBoost: Este algoritmo se entrena con hiperparámetros como el número de etapas de refuerzo a realizar '*n_estimators*', y '*criterion*', que ya se mencionó en modelos anteriores. Los valores optimizados son '*n_estimators*' = 500; '*criterion*' = '*friedman_mse*'.

Los resultados obtenidos con los modelos se presentarán a continuación.

VI. Resultados

Aquí se presentan los resultados del entrenamiento de los modelos de identificación de técnicas, eligiendo el que proporcione mejor resultado.

Para comparar los tres modelos entrenados y elegir el que mejor se adapta a las condiciones de este entorno, se evaluarán mediante la exactitud de los modelos, las matrices de confusión y otras métricas como la F1 o el tiempo de ejecución.

Debido a la limitación de espacio, en la Tabla III se presentan los resultados de exactitud (*accuracy*) de entrenamiento y validación, F1 y tiempo de ejecución.

Tabla III
Comparación de los modelos entrenados

Modelo	Exactitud entrenamiento	Exactitud validación	Métrica F1	Tiempo de ejecución
Decision Tree	0.9913	0.9868	0.9869	0.7292 s
Random Forest	0.9904	0.9878	0.9879	1.7820 s
XGBoost	0.9926	0.9839	0.9840	51.1392 s

Teniendo en cuenta los datos anteriores, cualquiera de los algoritmos entrenados tiene buenos resultados, eligiendo como el modelo más adecuado para este conjunto de datos *Decision Tree* debido al tiempo de ejecución. Las métricas de exactitud completas del algoritmo se muestran en la Tabla IV, incluyendo las de entrenamiento y validación para la identificación de técnicas, y la de validación para la detección binaria y la identificación de tácticas.

En la Tabla V se presenta el informe de clasificación del modelo en la identificación de técnicas. Debido a la similitud en T1590 y T1592, que consisten en recoger información de la red y del dispositivo víctima, hemos decidido unificar los registros. Las etiquetas por tanto quedarán de esta forma: 0 - Benigno, 1 - T1046, 2 - T1071, 3 - T1190, 4 - T1210, 5 -

Tabla IV
EXACTITUD DEL MODELO *Decision Tree*

Exactitud entrenamiento	Exactitud validación	Exactitud Binaria	Exactitud Tácticas
0.9913	0.9868	1.0	0.9939

T1548, 6 - T1566, 7 - T1587, 8 - T1589, 9 - T1590/T1592, 10 - T1595.

Tabla V
INFORME DE CLASIFICACIÓN DEL MODELO *Decision Tree*

Etiqueta	Precisión	*Recall*	F1
0	1.00	1.00	1.00
1	0.98	1.00	0.99
2	0.65	0.93	0.76
3	1.00	1.00	1.00
4	0.92	0.61	0.73
5	1.00	1.00	1.00
6	1.00	1.00	1.00
7	1.00	1.00	1.00
8	0.86	0.98	0.92
9	1.00	0.98	0.99
10	1.00	1.00	1.00
accuracy			0.99
macro avg	0.95	0.96	0.95
weighted avg	0.99	0.99	0.99

En las Figuras 4, 5 y 6 se muestran las matrices de confusión de este modelo. Se puede apreciar que, a pesar del balanceo de datos, el conjunto sigue siendo muy desbalanceado. Las tácticas se identifican mediante las siguientes etiquetas: 0 - Benigno, 1 - *Command and Control*, 2 - *Defense Evasion*, 3 - *Discovery*, 4 - *Initial Access*, 5 - *Lateral Movement*, 6 - *Reconnaissance*, 7 - *Resource Development*.

En las matrices vemos que el algoritmo permite la clasificación binaria sin ningún tipo de error. En la identificación de tácticas existe confusión entre las clases de *Defense Evasion* (2) y *Reconnaissance* (6), y entre la clase *Initial Access* (4) y *Command and Control* (1). Por otro lado, en cuanto a la identificación de técnicas, existe confusión entre T1071 (Etiqueta 2 - *Application Layer Protocol*) y T1210 (Etiqueta 4 - *Exploitation of Remote Services*), y entre T1589 (Etiqueta 8 - *Gather Victim Identity Information*) y T1590/T1592 (Etiqueta 9 - *Gather Victim Network Information/Gather Victim Host Information*). En todos los casos, como se puede deducir del resto de métricas, la confusión existente en estos casos es menor y se debe principalmente a la similitud entre los procesos de las técnicas.

VII. CONCLUSIONES Y LÍNEAS FUTURAS

Dada la importancia de la protección dinámica de los sistemas frente a amenazas y riesgos, la caracterización de técnicas y tácticas permite complementar la información obtenida a partir de estudios expertos y ofrecer las protecciones óptimas en cada situación. Este proceso implica una comprensión más profunda de los elementos de la matriz MITRE ATT&CK para poder definir reglas que identifiquen estos comportamientos. Sin embargo, estos criterios para la detección no pueden abarcar las técnicas más complejas, donde la mejor oportunidad se encuentra en las ventajas de la inteligencia artificial y el aprendizaje automático.

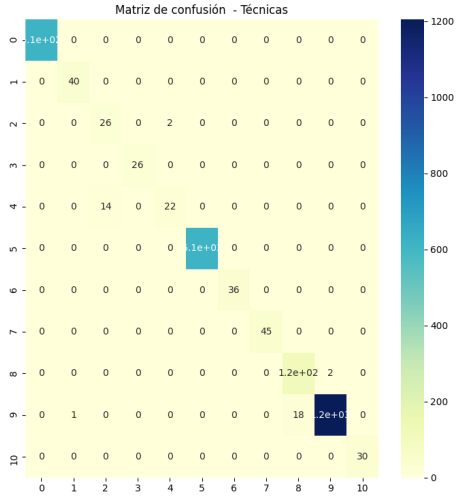

Figura 4. Matriz de confusión de la identificación de técnicas

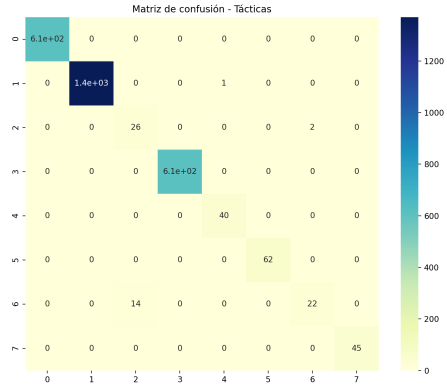

Figura 5. Matriz de confusión de la identificación de tácticas

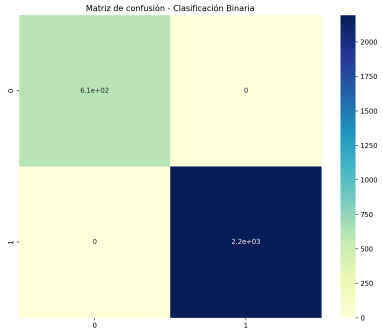

Figura 6. Matriz de confusión de la clasificación binaria

Los resultados presentados contribuyen al desarrollo de una estrategia integral que permita abordar la respuesta frente a amenazas a raíz de la caracterización de las técnicas utilizadas

en el ataque con soluciones innovadoras.

La investigación presentada aprovecha la eficacia del aprendizaje automático aplicándolo en la identificación de técnicas MITRE ATT&CK, permitiendo recomendar contramedidas específicas. Este enfoque aborda la detección de incidentes de ciberseguridad y la respuesta proactiva ante ellos gracias a la capacidad de razonamiento e inferencia de conocimiento de las ontologías basándose en la información propuesta por MITRE.

Los resultados obtenidos permiten mejorar en algunos aspectos los trabajos realizados anteriormente, como se recoge en la Tabla VI.

Tabla VI
COMPARACIÓN DE TRABAJOS PREVIOS CON ESTA PROPUESTA

Publicación	N. Tácticas	N. Técnicas	Métrica
[7]	5	5	No se indica
[8], [9]	6	15	Área bajo la curva: 0.868 (*Area Under the Curve*, AUC)
[18]	3	-	Exactitud multiclase: 0.9999
[20]	2	-	Descubrimiento: 0.9991 Reconocimiento: 0.994
[23]	8	-	Exactitud : 0.999936
Propuesta	7	11	Exactitud Técnicas: 0.9868 Exactitud Tácticas: 0.9939 AUC: 0.999

En cuanto a la detección de técnicas, esta propuesta mejora los resultados de área bajo la curva del trabajo presentado en [8], [9], a pesar de tener la capacidad de detectar 4 técnicas menos, y superando al resto de trabajos relacionados. Además, nuestro desarrollo permite la detección de tácticas, superando en número a casi todos los trabajos previos. En el caso de [23], a pesar de detectar una táctica más, no incluye la detección de técnicas, que supone una mejora del nuestro con respecto a ese trabajo.

Para continuar con el desarrollo de esta propuesta en el futuro, las siguientes líneas de trabajo consisten en la inclusión de nuevas técnicas o la posibilidad de complementar la detección mediante aprendizaje automático con la detección mediante reglas. Además, la validación del sistema utilizando otros conjuntos de datos es recomendada, ya que en el utilizado hay técnicas con muy pocas muestras y de esta forma se evalúa la robustez del modelo.

REFERENCIAS

[1] MITRE ATT&CK® en https://attack.mitre.org/ (Accedido: 20/2/2024).

[2] Nombeko Ntingi et al.: “Effective Cyebr Threat Hunting: Where and how does it fit?”, en *Proceedings of the 21st European Conference on Cyber Warfare and Security*, pp.206-213, 2022.

[3] A. Aydeger, N. Saputro and K. Akkaya: “Cloud-based Deception against Network Reconnaissance Attacks using SDN and NFV”, en *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, Sydney, NSW, Australia, pp. 279-285, 2020.

[4] Md Sahrom Abu et al.: “Formulation of Association Rule Mining (ARM) for an Effective Cyber Attack Attribution in Cyber Threat Intelligence (CTI)”, en *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 12, n. 4, 2021.

[5] S. Kriaa y Y. Chaabane: “SecKG: Leveraging attack detection and prediction using knowledge graphs”, en *12th International Conference on Information and Communication Systems (ICICS)*, Valencia, Spain, pp. 112-119, 2021.

[6] Repositorio CAR MITRE en https://car.mitre.org (Accedido: 19/3/2024).

[7] He, Tiancai y Zhihua L: “A Model and Method of Information System Security Risk Assessment based on MITRE ATT&CK.” en *2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, pp. 81-86, 2021.

[8] Yashovardhan Sharma, Simon Birnbach, y Ivan Martinovic: “ RADAR: A TTP-based Extensible, Explainable, and Effective System for Network Traffic Analysis and Malware Detection”, en *Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference (EICC ’23). Association for Computing Machinery*, pp. 159–166, 2023.

[9] Yashovardhan Sharma et al.: “To TTP or not to TTP?: Exploiting TTPs to Improve ML-based Malware Detection”, en *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 8-15, 2023.

[10] Rawan Al-Shaer, Jonathan M. Spring y Eliana Christou: “Learning the Associations of MITRE ATT&CK Adversarial Techniques”, en *Criptography and Security (arXiv)*, (2020).

[11] G. P. Singh, J. K. Sandhu y M. K. Hooda:“BiLSTM Classifier: A New Approach for Detecting Cyber-Attacks in MITRE ATTACK Framework”, en *6th International Conference on Contemporary Computing and Informatics (IC3I)*, India, 2023, pp. 1339-1343.

[12] Y. Lakhdhar y S. Rekhis: “Machine Learning Based Approach for the Automated Mapping of Discovered Vulnerabilities to Adversial Tactics”, en *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 309-317, 2021.

[13] Constantin Adam et al.: “Attack Techniques and Threat Identification for Vulnerabilities”, en en *Criptography and Security (arXiv)*, (2022).

[14] Mendsaikhan, Otgonpurev et al.: “Automatic Mapping of Vulnerability Information to Adversary Techniques.”(2020).

[15] Chung-Kuan Chen et al.: “Building Machine Learning-based Threat Hunting System from Scratch”, en *Digital Threats*, vol. 3, n. 3, 2022.

[16] Heejung Kim y Hwankuk Kim: “Comparative Experiment on TTP Classification with Class Imbalance Using Oversampling from CTI Dataset”, en *Security and Communication Networks*, vol. 2022, 2022.

[17] Shanto Roy et al.: “SoK: The MITRE ATT&CK Framework in Research and Practice”, en *Criptography and Security (arXiv)*, (2023).

[18] Bagui, S.S. et al: “Introducing the UWF-ZeekDataFall22 Dataset to Classify Attack Tactics from Zeek Conn Logs Using Spark’s Machine Learning in a Big Data Framework”, en *Electronics* vol 12, n.24, 2023.

[19] Bagui, S.S. et al.: “Introducing UWF-ZeekData22: A Comprehensive Network Traffic Dataset Based on the MITRE ATT&CK Framework”, en *Data* vol, 8, 2023.

[20] Bagui, S. et al.: “Detecting Reconnaissance and Discovery Tactics from the MITRE ATT&CK Framework in Zeek Conn Logs Using Spark’s Machine Learning in the Big Data Framework”, en *Sensors* vol.22, n. 20, 2022.

[21] Bagui, S.S. et al.: “Using a Graph Engine to Visualize the Reconnaissance Tactic of the MITRE ATT&CK Framework from UWF-ZeekData22”, en *Future Internet*, vol. 15, n. 7, 2023.

[22] Bagui, S. et al.: “Resampling Imbalanced Network Intrusion Datasets to Identify Rare Attacks”, en *Future Internet*, vol. 15, n.4, 2023.

[23] Ángel Casanova Bienzobas y Alfonso Sánchez-Macián: “Threat Trekker: An Approach to Cyber Threat Hunting”, en *Criptography and Security (arXiv)*, (2023).

[24] MSDS621 lectures, en https://github.com/parrt/msds621/tree/master/lectures (Accedido: 9/3/2024).

[25] Machine Learning Algorithms — Pathmind, en https://pathmind.com/wiki/machine-learning-algorithms (Accedido: 9/3/2024).

[26] DecisionTreeClassifer — scikit-learn 1.4.1 documentation, en https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (Accedido: 9/3/2024).

[27] Soner Yıldırım: “Gradient Boosted Decision Trees-Explained”, en https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af (Accedido: 9/3/2024).

[28] Kevin W. Bowyer et al.: “SMOTE: Synthetic Minority Over-sampling Technique”, en *CoRR abs/1106.1813* (2011). arXiv: http://arxiv.org/abs/1106.1813.

[29] Department of Computer Science, University of West Florida: “UWF-ZeekData22 Dataset”, en https://datasets.uwf.edu/index.html (Accedido: 26/2/2024).

[30] Carmen Sánchez-Zas, Víctor A. Villagrá, Mario Vega-Barbas, Xavier Larriva-Novo, José Ignacio Moreno, Julio Berrocal: “Ontology-based approach to real-time risk management and cyber-situational awareness”, en *Future Generation Computer Systems*, vol. 141, pp. 462-472, 2023.

[31] Zeek en https://zeek.org (Accedido: 26/2/2024).

[32] Scikit-learn - Supervised Learning, en https://scikit-learn.org/stable/supervised_learning.html#supervised-learning (Accedido: 28/2/2024).