Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

Diseño de un modelo neuronal para la detección y la clasificación de intrusiones en redes informáticas

Alumno:
Hugo López Álvarez

Tutores:
Diego García Álvarez

...

Agradecimientos

...

Resumen

Resumen

Abstract

Abstract

Índice general

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

Este documento corresponde con la memoria del Trabajo de Fin de Grado (TFG) del grado en Informática de la Universidad de Valladolid. Este trabajo se centra en la creación de un modelo neuronal capaz de detectar intrusiones en una red informática. La principal ventaja de utilizar un modelo neuronal para la detección de intrusiones en una red, frente a los algoritmos tradicionales (como firmas basadas en reglas o análisis estadísticos), radica en su capacidad para aprender patrones complejos y no lineales en los datos, lo que le permite identificar amenazas desconocidas o variantes de ataques existentes (zero-day attacks). Mientras que los métodos tradicionales dependen de reglas predefinidas y actualizaciones manuales para detectar intrusiones (limitándose a ataques conocidos), las redes neuronales pueden analizan grandes volúmenes de tráfico de red, detectando anomalías sutiles y correlaciones ocultas mediante capas de abstracción.

1.1. Explicación del problema

En la actualidad, los sistemas informáticos reciben muchos más ataques de denegación de servicio y de intrusión que hace unos años, esto se debe en parte a los avances en los modelos de IA.

Los sistemas informáticos enfrentan actualmente graves amenazas debido al uso malintencionado de la Inteligencia Artificial (IA) por parte de ciberdelincuentes. Una de las principales problemáticas es la automatización de ataques, donde herramientas basadas en IA permiten ejecutar campañas de ataques informáticos con mayor precisión y escala. Estas IAs pueden generar mensajes convincentes, imitar patrones de comportamiento legítimos y evadir medidas de seguridad tradicionales, lo que incrementa la frecuencia y sofisticación de los ataques.

Otro desafío crítico es la explotación de vulnerabilidades mediante IA, que acelera la identificación de fallos en sistemas sin intervención humana. Existen algoritmos de machine

learning que analizan grandes volúmenes de datos para descubrir brechas de seguridad en tiempo récord, facilitando ataques dirigidos incluso contra infraestructuras críticas como hospitales.

La IA también complica la defensa, ya que los sistemas de detección tradicionales no siempre pueden anticipar tácticas adaptativas generadas por algoritmos hostiles. Esto obliga a las organizaciones y empresas a invertir en soluciones de IA defensiva, como sistemas de respuesta autónoma. Sin embargo, esto genera una carrera tecnológica desigual donde actores maliciosos aprovechan herramientas accesibles y de bajo costo. La falta de regulación global agrava este escenario, dificultando la mitigación de riesgos asociados.

Además, los modelos neuronales son adaptativos: mejoran su precisión con el tiempo al entrenarse con nuevos datos, lo que es crucial en entornos dinámicos donde los ciberataques evolucionan rápidamente. Por ejemplo, pueden distinguir entre comportamientos legítimos inusuales (como un empleado accediendo a recursos fuera de horario) y actividades maliciosas (como filtración de datos), reduciendo falsos positivos. En cambio, los enfoques tradicionales suelen ser rígidos y requieren ajustes manuales frecuentes para mantener su eficacia.

Sin embargo, el uso de modelos nueronales para la defensa de los sistemas conlleva grandes desafíos, como la necesidad de grandes conjuntos de datos etiquetados y recursos computacionales intensivos. Aun así, en escenarios donde la sofisticación de los ataques supera las capacidades de detección convencionales, los modelos neuronales representan un salto cualitativo en proactividad y escalabilidad.

https://www.wsj.com/articles/the-ai-effect-amazon-sees-nearly-1-billion-cyber-threats-a-day-15434edd

1.2. Motivación

Durante mi formación universitaria en el Grado en Ingeniería Informática, como alumno de la mención de tecnologías de la información, he aprendido a administrar grandes sistemas de computación en aspectos como: la seguridad, la garantía de la información, la evaluación de dichos sistemas y el almacenamiento de los datos. Además de cierto componente de desarrollo de software.

Revisar

Sin embargo, uno de los conocimientos que no he podido adquirir durante mis estudios, es uno de los temas más importantes en la actualidad, la Inteligencia Artificial. Con el objetivo de expandir mis conocimientos sobre este tema, decidí implementar un modelo neuronal que facilitase la detección de ataques a redes informáticas que tantas complicaciones está generando a los encargados de la administración de estos sistemas.

1.3. Objetivos

- Aprender como funcionan los modelos neuronales y los diferentes tipos de ellos que existen
- Investigar las mejores opciones de arquitectura y de elección de hiperparámetros.
- Entendimiento de los problemas que enfrentan los sistemas informáticos en la actualidad
- Generación de modelos basados en Deep Learning.
- Mitigar riesgos de seguridad, reduciendo los tiempos de respuesta ante incidentes.

1.4. Estrucutra de la memoria

Este documento se estructura de la siguiente forma:

Capítulo 1 Introducción:

Capítulo 2 Metodología:

Capítulo 3 Planificación:

Capítulo 4 Entendimiento del problema:

Capítulo 5 Entendimiento de los datos:

Capítulo 6 Modelos:

Capítulo 7 Test:

Capítulo 8 Despliegue:

Capítulo 9 Tecnologías utilizadas:

Capítulo 10 Seguimiento del proyecto:

Capítulo 11 Conclusiones:

Anexo A Manuales:

Anexo B Resumen de enlaces adicionales:

Capítulo 2

Metodología

En este capítulo se explica la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), que se utiliza en el desarrollo del resto del proyecto para alcanzar los objetivos propuestos.

La adopción de metodologías estructuradas es fundamental en el desarrollo de proyectos informáticos, puesto que proporcionan un marco sistemático para garantizar la calidad, eficiencia y trazabilidad del proyecto. En particular, metodologías como CRISP-DM, permiten: alinear objetivos técnicos con necesidades de negocio,reducir riesgos mediante fases iterativas y documentadas, y facilitar la colaboración entre equipos multidisciplinares.

Según algunos estudios, los proyectos que utilizan metodologías estandarizadas incrementan un 35 % su probabilidad de éxito, frente a aproximaciones ad-hoc, al minimizar desviaciones en costes y plazos [**?**]. En el ámbito de la ciberseguridad, donde los requisitos legales y técnicos son críticos, este enfoque metodológico resulta indispensable para asegurar soluciones robustas y auditables

2.1. CRISPDM

La metodología CRISP-DM, es un marco de trabajo estandarizado para guiar proyectos de minería de datos y aprendizaje automático. Su estructura cíclica y flexible la hace aplicable en diversos dominios, desde marketing hasta ciberseguridad. Está compuesta por las siguientes fases:

1. Comprensión del negocio: La primera fase de CRISP-DM establece los cimientos estratégicos del proyecto mediante un proceso de alineación entre los objetivos técnicos y las necesidades organizacionales. Para lograr establecer los cimientos, se lleva a cabo un análisis exhaustivo del contexto empresarial para identificar los problemas clave que el proyecto debe abordar, así como las oportunidades de mejora que podrían aprovecharse. Se realiza

un proceso de recopilación y documentación de requisitos que involucra a todas las partes interesadas relevantes. El resultado de esta fase es una definición precisa del alcance del proyecto, que incluye no solo los objetivos cuantificables sino también los criterios de éxito que permitirán evaluar el impacto real de la solución propuesta. Además, se establecen las limitaciones operativas y estratégicas que condicionarán el desarrollo del proyecto, asegurando que todas las fases posteriores se ejecuten dentro de un marco bien definido y alineado con las prioridades organizacionales.

2. Comprensión de los datos: Esta fase se centra en el análisis detallado de los datos disponibles para el proyecto, con el objetivo de evaluar su idoneidad y calidad para abordar los problemas identificados en la fase anterior. Este proceso implica un examen minucioso de las diversas fuentes de información, su estructura y sus características fundamentales. Durante esta etapa, se identifican y documentan aspectos críticos como la complejidad de los datos, la presencia de posibles sesgos y la representatividad de la información en relación con los objetivos del proyecto. La comprensión profunda de los datos permite anticipar desafíos potenciales y establecer estrategias adecuadas para su tratamiento en fases posteriores. Además, esta fase proporciona perspectivas que pueden influir en decisiones técnicas importantes, como la selección de algoritmos o el diseño de características. El resultado es un conocimiento del potencial y las limitaciones de los datos disponibles, que sirve como base para las transformaciones que se realizan en la siguiente fase.

3. Preparación de los Datos: Se trata de una fase crítica donde los datos brutos se transforman en un conjunto adecuado para modelado. Esta etapa implica una serie de operaciones fundamentales que garantizan la calidad y consistencia de los datos que alimentan a los modelos analíticos. Las actividades realizadas en esta fase son cruciales para el éxito del proyecto, ya que determinan en gran medida la capacidad de los algoritmos para extraer patrones significativos y generar resultados confiables. Se aplican técnicas especializadas para abordar problemas comunes en los datos, asegurando que la información sea representativa, completa y se encuentre adecuadamente estructurada para los análisis posteriores. Cualquier deficiencia en la preparación de los datos puede comprometer significativamente la efectividad de las siguientes fases. Al finalizar este proceso, se obtiene un conjunto de datos optimizado que conserva la esencia de la información original mientras elimina ruido y distorsiones que podrían afectar negativamente a los resultados del modelado.

4. Modelado: Constituye el núcleo técnico del proceso CRISP-DM, donde se desarrollan y evalúan los algoritmos diseñados para extraer conocimiento de los datos preparados. Esta etapa comienza con la selección cuidadosa de las técnicas de modelado más apropiadas para los objetivos específicos del proyecto y las características de los datos disponibles. Durante el proceso de modelado, se exploran diferentes enfoques algorítmicos, ajustando meticulosamente sus parámetros para optimizar su rendimiento. En esta fase se incluyen procesos de validación diseñados para garantizar que los modelos desarrollados sean robustos y generalizables, capaces de mantener su efectividad cuando se enfrenten a datos nuevos y no vistos previamente. El modelado es un proceso iterativo que puede requerir volver a fases anteriores para refinar la preparación de datos o incluso reconsiderar algunos aspectos del planteamiento inicial del problema. El resultado de esta fase es uno o varios modelos validados que cumplen con los criterios de calidad establecidos y están listos para su evaluación en el contexto de los objetivos empresariales definidos inicialmente.

5. Evaluación: Esta fase representa un examen exhaustivo de los modelos desarrollados, contrastando su desempeño técnico con los objetivos empresariales establecidos en la primera fase del proyecto. Este proceso va más allá de las métricas estadísticas tradicionales para incorporar una valoración del impacto potencial de la solución propuesta. Durante la evaluación, se analiza minuciosamente la capacidad de los modelos para resolver el problema de negocio original, considerando tanto su precisión técnica como su aplicabilidad práctica en el contexto organizacional. Se identifican y documentan las limitaciones de los modelos, así como los posibles riesgos asociados a su implementación. Esta fase también incluye la validación de los resultados con las partes interesadas clave, asegurando que la solución cumpla con las expectativas y requisitos operativos. La evaluación termina con una decisión fundamentada sobre la idoneidad de los modelos para su implementación, junto con recomendaciones para su posible mejora o adaptación a escenarios futuros. También se valida su robustez en escenarios realistas.

6. Despliegue: Se trata de la fase final de CRISP-DM, esta se centra en la transición del modelo analítico desde un entorno de desarrollo a un sistema operativo donde pueda generar valor tangible para la organización. Este proceso implica una serie de actividades cuidadosamente planificadas que garantizan la integración efectiva de la solución en los procesos empresariales existentes. El despliegue incluye aspectos técnicos como la implementación de la infraestructura necesaria, el desarrollo de interfaces adecuadas y la creación de mecanismos de monitoreo continuo. También se ha de tener en cuenta la capacitación de los usuarios finales y la documentación exhaustiva de la solución, asegurando su adopción efectiva y su uso óptimo. La fase de despliegue también establece procesos para el mantenimiento y actualización periódica del modelo, puesto que las soluciones analíticas requieren evolución continua para mantener su relevancia y efectividad. Como en el resto de metodologçias, se implementan mecanismos para medir el impacto real de la solución una vez en producción, cerrando el ciclo al proporcionar retroalimentación valiosa que puede ser la base de futuros proyectos analíticos.

Como se ha explicado, CRISP-DM es una metodología iterativa, esto significa que los resultados de fases posteriores pueden revelar la necesidad de ajustes en etapas anteriores (como recolectar más datos o redefinir objetivos). Su enfoque estructurado minimiza riesgos y maximiza el valor entregado, siendo especialmente útil en proyectos complejos donde la alineación entre técnica y negocio es esencial.

Figura 2.1: Esquema del ciclo CRISP-DM estándar.

Capítulo 3

Planificación

Este capítulo aborda la organización detallada de un Trabajo de Fin de Grado, cubriendo desde su diseño inicial hasta la implementación y el seguimiento durante su desarrollo. Una planificación rigurosa resulta fundamental para sentar las bases del proyecto, ya que permite definir con claridad los objetivos, los recursos necesarios, los plazos de entrega y las actividades clave para alcanzar los resultados esperados.

En primer lugar, se establece una planificación temporal preliminar, donde se estiman los tiempos requeridos para cada etapa. Este cronograma se estructura en torno a las fases de la metodología CRISP-DM, complementadas con etapas específicas propias de un Trabajo de Fin de Grado. A continuación, se realiza un análisis de riesgos exhaustivo, evaluando tanto la probabilidad como el impacto de cada posible contingencia.

Además, se elabora un presupuesto detallado para las tareas del proyecto, abordado desde dos perspectivas. Por un lado, se incluye una estimación realista de los costes asociados a la ejecución del trabajo en el ámbito académico. Por otro lado, se plantea una proyección teórica de los gastos que implicaría un proyecto equivalente en un contexto profesional.

Por último, se contrasta la planificación inicial con el desarrollo real del trabajo, lo que permite evaluar posibles desviaciones y los aprendizajes obtenidos durante el proceso.

3.1. Planificación temporal

La planificación temporal constituye un elemento fundamental en la ejecución de un proyecto fin de grado, ya que permite estructurar de manera sistemática todas las actividades necesarias para alcanzar los objetivos propuestos. En el contexto de un trabajo académico que combine el desarrollo de software con una metodología de investigación, como es el caso de CRISP-DM para el proceso analítico y SCRUM para la gestión del proyecto, una adecuada planificación garantiza la distribución equilibrada del tiempo disponible entre las distintas

fases del trabajo. Esta organización temporal resulta especialmente relevante cuando se deben coordinar aspectos teóricos, desarrollo técnico y validación de resultados, asegurando que cada componente reciba la atención necesaria sin comprometer la calidad global del proyecto.

El empleo de un diagrama de Gantt como herramienta de planificación ofrece ventajas significativas para visualizar la secuencia de actividades y su superposición temporal. Este tipo de representación gráfica facilita la identificación de hitos críticos y dependencias entre tareas, aspectos particularmente importantes cuando se combinan metodologías diferentes como CRISP-DM y SCRUM. La primera, con sus fases bien definidas, proporciona la estructura para el desarrollo del núcleo analítico del proyecto, mientras que SCRUM, con sus sprints iterativos, permite adaptar el trabajo a los descubrimientos que vayan surgiendo durante la investigación. La integración de ambas aproximaciones en un único cronograma exige una cuidadosa coordinación que el diagrama de Gantt ayuda a materializar de forma clara y comprensible.

3.2. Gestión de riesgos

3.3. Estimación de costes

3.3.1. Costes materiales

3.3.2. Costes humanos

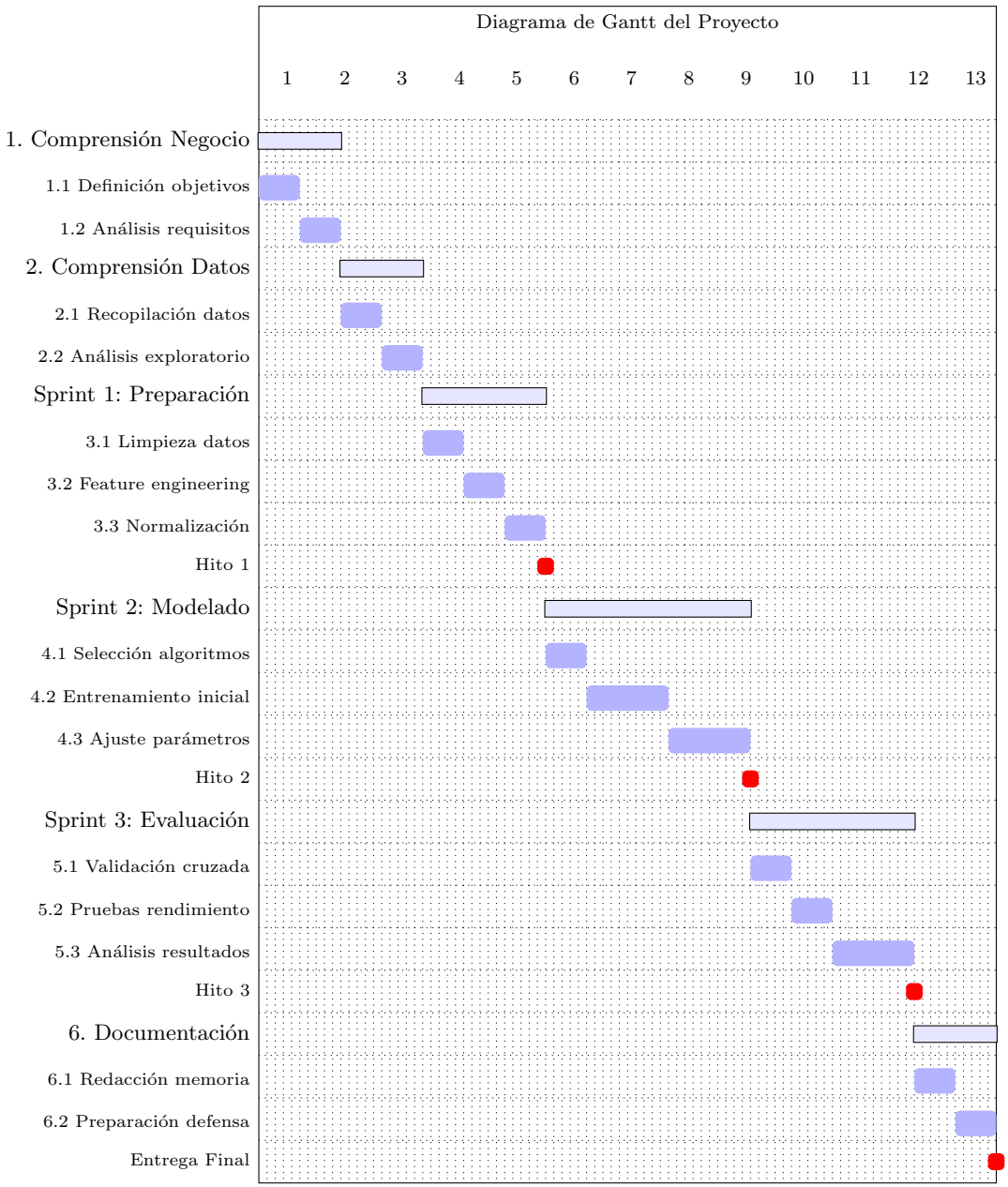

Figura 3.1: Diagrama de Gantt con planificación semanal y detalle diario

Capítulo 4

Entendimiento del problema

En este capítulo trataremos en el entendimiento del problema. Se trata de la fase inicial de la metología CRISP-DM. A continuación, se alinean los objetivos técnicos con las necesidades del negocio o el problema a resolver. Se definen requisitos, se identifican métricas de éxito y se trata de dar comprensión sobre el contexto organizacional.

4.1. Requisitos

Como se ha comentado en el punto 1.3 Objetivos, el principal objetivo del proyecto es desarrollar un modelo neuronal que detecte la presencia de ataques en una red informática y los clasifique según su tipo. Para cumplir con dicho objetivo, se considera imprescindible cumplir con los requisitos que se listan a continuación.

4.1.1. Requisitos Funcionales

Primera versión de requisitos, no me convencen mucho

- **RF-1**: El sistema deberá detectar cuales de las conexiones podrían ser potenciales intrusiones en la red.
- **RF-2**: El sistema deberá clasificará las conexiones en 10 categorías predefinidas en Tipos de ataques registrados en los datos.
- **RF-3**: El sistem deberá ser capaz de procesar formatos estándar de logs como son Syslog, NetFlow y PCAP.
- **RF-4**: El sistema deberá diferenciar entre ataques conocidos (basados en firmas) y desconocidos (basados en anomalías).

- **RF-5**: El sistema deberá ofrecer API REST para conexión con SIEMs (Splunk, IBM QRadar)
- **RF-6**: Generar alertas automatizadas con nivel de criticidad (bajo/medio/alto).
- **RF-7**: Proveer recomendaciones de mitigación básicas (ej. bloquear IPs maliciosas)

  ¿Debería integrar el modelo en algún sistema o crear un script o alguna forma para comunicarme con él?

4.1.2. Requisitos No Funcionales

- **RNF-1**: Latencia ¡50 ms en redes de 10Gbps (requisito crítico para SOC [**?**]).
- **RNF-2**: Interfaz accesible para usuarios no técnicos (evaluado con test SUS [**?**]).

4.1.3. Reglas de Negocio

- **RB-1**: Coste operativo mensual no superará $10,000 (aprobado por Comité de Seguridad).
- **RB-2**: Alertas de ransomware requerirán confirmación humana antes de aislamiento de red.

4.2. Contexto organizacional

4.3. Objetivos del proyecto

Capítulo 5

Entendimiennto de los datos

Este capítulo se corresponde con la segunda etapa de la metodología CRISP-DM, En el se explicará la naturaleza de los datos y sus características, así como los valores atípicos que presentan y sus sesgos.

5.1. Origen de los datos

Los datos que se han utilizado para desarrollar este trabajo, se han obtenido de conjuntos de datos diseñados para entrenar Sistemas de Detección de Intrusión de Red (NIDS) basados en el aprendizaje automático. El dataset en cuenstión forma parte de un análisis realizado en la Universidad de Queensland, Australia.[**?**]

El dataset utilizado es NF-UNSW-NB15-v3, este es una versión basada en NetFlow del conocido conjunto de datos UNSW-NB15, mejorada con características adicionales de NetFlow y etiquetada de acuerdo con sus respectivas categorías de ataque.

5.2. Tipos de ataques registrados en los datos

El conjunto de datos consiste en un total de 2.365.424 flujos de datos, donde 127.639 (5,4 %) son muestras de ataque y 2.237.731 (94,6 %) son benignos. Los flujos de ataque se clasifican en nueve clases, cada una representando una amenaza a la red distinta. La siguiente tabla proporciona una distribución detallada del conjunto de datos:

Clase	Cantidad	Descripción
Benigno	2.237.731	Flujos normales no maliciosos.
Fuzzers	33.816	Tipo de ataque en el que el atacante envía grandes cantidades de datos aleatorios que hacen que un sistema se bloquee y también apuntan a descubrir vulnerabilidades de seguridad en un sistema.
Analysis	2.381	Un grupo que presenta una variedad de amenazas que se dirigen a aplicaciones web a través de puertos, correos electrónicos y scripts.
Backdoor	1.226	Una técnica que tiene como objetivo eludir los mecanismos de seguridad respondiendo a aplicaciones específicas de clientes construidos.
DoS	5.980	La denegación de servicio es un intento de sobrecargar los recursos de un sistema informático con el objetivo de evitar el acceso o la disponibilidad de sus datos.
Exploits	42.748	Son secuencias de comandos que controlan el comportamiento de un host a través de una vulnerabilidad conocida.
Generic	19.651	Un método que se dirige a la criptografía y causa una colisión con cada cifrado de bloques.
Reconnaissance	17.074	Una técnica para recopilar información sobre un host de red, también se conoce como sonda.
Shellcode	4.659	Un malware que penetra en un código para controlar el host de una víctima.
Worms	158	Ataques que se replican y se extienden a otros sistemas.

Tabla 5.1: Clasificación de amenazas de seguridad

5.3. Parámetros de los datos

Los datos tienen en cuenta un total de 55 parámetros entre los que destacan:

¿Debería explicar todas las columnas del dataset o solo las más importantes?

`https://arxiv.org/pdf/2503.04404`

- **Label**: indica si cada dato es un ataque (valor = 1) o si es una conexión legítima (valor = 0).
- **Attack**: especifica el tipo de conexión, diferenciando entre los tipos mencionados anteriormente en **??**.
- **FLOW_START_MILISECONDS**: timestamp en el que se inicia la conexión entre los sistemas.

- **FLOW_END_MILISECONDS**: timestamp en el que se finaliza la conexión entre los sistemas.
- **L4_SRC_PORT**: puerto de origen desde el que se inicia la conexión.
- **L4_DST_PORT**: puerto de destino al que se quiere conectar.
- **PROTOCOL**: protocolo que que define cómo los dispositivos interactúan para comunicarse, transmitir datos y compartir recursos.
- **IN_BYTES**: número de bytes que envía el dispositivo que inicia la conexión.
- **OUT_BYTES**: número de bytes que devuelve el dispositivo objetivo de la conexión.
- **TCP_FLAG**: suma de los indicadores TCP.

5.4. Patrones preliminares, valores atípicos y sesgos

Tras analizar los datos originales del dataset, se han encontrado características que afectarían de forma negativa al entrenamiento del modelo y por lo tanto, a su correcto funcionamiento posteriormente. A continuación, se mencionan cuales han sido las característcas problematicas de los datos encontradas.

Algunos parámetros presentan valores infinitos que no son aptos. Para evitar que estos datos produzcan errores en la ejecución del algoritmo que entrena al modelo, se ha optado por eliminarlos.

En un principio, puede parecer que los datos están sesgados por las direcciones IPv4 de los dispositivos origen. Esto se debe a que solo se producen ataques desde las direcciones con máscara 175.45.176.255. Tras realizar pruebas excluyendo este parámetro del entrenamiento del modelo, se ha llegado a la conclusión de que este parámetro no recibe un peso muy alto y no altera los resultados de las métricas del modelo.

Capítulo 6

Modelos

6.1. Métricas

6.1.1. Matriz de confusión

Para evaluar el desempeño del modelo de detección y clasificación de ataques, se utilizan las siguientes métricas derivadas de la matriz de confusión.

	Predicción Positiva	**Predicción Negativa**
Real Positivo	Verdaderos Positivos (VP)	Falsos Negativos (FN)
Real Negativo	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Tabla 6.1: Matriz de confusión para clasificación binaria.

6.1.2. Fórmulas e Interpretación

- **Exactitud (*Accuracy*)**:

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + VN + FN} \tag{6.1}$$

  Interpretación: Proporción de predicciones correctas sobre el total. Útil cuando las clases están balanceadas, pero sensible a distribuciones desiguales.

- **Precisión (*Precision*)**:

$$\text{Precision} = \frac{VP}{VP + FP} \tag{6.2}$$

Interpretación: Capacidad del modelo de no etiquetar como positivo un caso negativo. Crítica en escenarios donde los falsos positivos son costosos (ej.: bloquear tráfico ilegítimo).

- **Sensibilidad (*Recall*)**:

$$\text{Recall} = \frac{VP}{VP + FN} \tag{6.3}$$

Interpretación: Capacidad de detectar todos los casos positivos. Prioritario en seguridad, donde los falsos negativos (ataques no detectados) pueden tener consecuencias catastróficas.

- **Puntuación F1 (*F1-Score*)**:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.4}$$

Interpretación: Media armónica de precisión y recall. Balancea ambos métricas, ideal para clases desbalanceadas.

- **Puntuación F2 (*F2-Score*)**:

$$F2 = 5 \times \frac{\text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}} \tag{6.5}$$

Interpretación: Versión ponderada del F1 que da más peso al recall (útil cuando omitir un ataque es más grave que generar falsas alertas).

6.1.3. Aplicación en Seguridad

En el contexto de detección de intrusiones:

- Un recall alto (¿95 %) asegura que pocos ataques pasan desapercibidos.
- La precisión debe optimizarse para reducir la carga operativa de analistas (falsos positivos ¡10 %).
- El F2-Score es preferible al F1 cuando la prioridad es minimizar riesgos de ataques no detectados.

Capítulo 7

Test

Capítulo 8

Despliegue

Capítulo 9

Tecnologías usadas

Capítulo 10

Seguimiento del proyecto

Capítulo 11

Conclusiones

Apéndice A

Manuales

A.1. Manual de despliegue e instalación

A.2. Manual de mantenimiento

A.3. Manual de usuario

Apéndice B

Resumen de enlaces adicionales

Los enlaces útiles de interés en este Trabajo Fin de Grado son:

- Repositorio del código: `https://gitlab.inf.uva.es/`.