Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

Título

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

2.1. CRISPDM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) 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 estas fases:

1. Entendimiento del problema: Esta es la fase inicial y se centra en alinear los objetivos técnicos con las necesidades del negocio o el problema a resolver. Implica definir requisitos, identificar métricas de éxito y comprender el contexto organizacional. Por ejemplo, en un proyecto de detección de fraude, se establecerían criterios para medir la eficacia del modelo (como reducir falsos negativos en un 20

2. Entendimiento de los datos: En esta fase se recopilan y exploran los datos disponibles para evaluar su calidad, relevancia y limitaciones. Mediante análisis descriptivo y visualizaciones, se identifican patrones preliminares, valores atípicos o sesgos.

3. Preparación de los Datos: Fase crítica donde los datos brutos se transforman en un conjunto adecuado para modelado. Incluye limpieza (manejo de valores nulos), integración de fuentes diversas, normalización y creación de características (feature engineering).

4. Modelado: Se seleccionan y entrenan algoritmos (como redes neuronales o árboles de decisión) para resolver el problema definido. La elección del modelo depende de la naturaleza de los datos y los objetivos (clasificación, regresión, etc.). Se aplican técnicas de validación (como k-fold cross-validation) para evitar sobreajuste.

5. Evaluación: Los modelos se prueban con métricas rigurosas (precisión, recall, AUC-ROC) para determinar si cumplen los criterios de negocio establecidos en la Fase 1. También se valida su robustez en escenarios realistas.

6. Despliegue: Una vez validado, el modelo se implementa en producción, integrado en sistemas operativos. Esto incluye monitorización continua, actualizaciones y documentación para usuarios finales. En agricultura inteligente, un modelo de predicción de cosechas podría desplegarse en una app móvil para que los agricultores reciban alertas en tiempo real.

CRISP-DM es 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.

Capítulo 3

Planificación

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 encontradas.

Algunos parámetros presentan valores infinitos que no son aptos para evitar que estos datos produzcan errores en el ejecución del algoritmo para entrenar al modelo han sido eliminados.

Los datos están sesgados por las direcciones IPv4 de los dispositivos origen. Solo se producen ataques desde las direcciones con máscara 175.45.176.255 por lo que este parámetro será ignorado para que los resultados del modelo no estén condicionados por dicho sesgo.

Capítulo 6

Modelos

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/`.