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

Agradecimientos **III**

Resumen **V**

Abstract **VII**

Lista de figuras **XI**

Lista de tablas **XIII**

1. Introducción **1**

1.1. Explicación del problema 1

1.2. Motivación 2

1.3. Objetivos 3

1.4. Estrucutra de la memoria 3

2. Metodología **5**

2.1. CRISPDM 5

3. Planificación **7**

4. Entendimiento del problema **9**

5. Entendimiennto de los datos **11**

6. Modelos **13**

7. Test **15**

8. Despliegue **17**

9. Tecnologías usadas **19**

10.Seguimiento del proyecto **21**

11.Conclusiones **23**

A. Manuales **25**

A.1. Manual de despliegue e instalación 25

A.2. Manual de mantenimiento 25

A.3. Manual de usuario 25

B. Resumen de enlaces adicionales **27**

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.

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. Comprensión del Negocio: Esta fase inicial se centra en alinear los objetivos técnicos con las necesidades del negocio o 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. Comprensión de los Datos: Aquí 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. En un sistema de recomendación, esto incluiría analizar el historial de compras de usuarios para garantizar que los datos reflejen comportamientos reales.

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). Por ejemplo, en predicción de fallos de hardware, se podrían generar variables derivadas como "horas de uso continuo.ª partir de registros crudos.

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. En diagnóstico médico, podrían compararse modelos de random forest y SVM para predecir enfermedades con mayor precisión.

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. Un modelo de churn de clientes, por ejemplo, debería demostrar que identifica correctamente al 90

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: 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

Capítulo 5

Entendimiennto de los datos

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/`.