ANÁLISIS DE MODELOS DE INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN Y CONTROL DE PATRONES DE CIBERSEGURIDAD

Francisco Jose Martin Aguilar
Universidad Politécnica de Cataluña

Resumen

En este proyecto se plantea una solución para la creciente complejidad de los ataques cibernéticos que comprometen a los usuarios y a empresas. Para ello se han usado tecnologías avanzadas como algoritmos de inteligencia artificial, con el objetivo de proteger activos digitales y información.

Se plantea la creación de un Sistema de Detección de Intrusiones de Red con Inteligencia Artificial (NIDS-AI - Network Intrusion Detection System with Artificial Intelligence), diseñado con el objetivo de fortalecer la seguridad tanto en entornos empresariales como personales. Este sistema se sustenta en modelos de aprendizaje automático, siguiendo un proceso que abarca desde la recolección, análisis y preprocesamiento de datos hasta el entrenamiento, selección, validación y evaluación de modelos.

El sistema contará con un entorno visual que indicará las amenazas detectadas. Además, incluirá un panel visual donde se mostrarán los datos obtenidos de la red para abordar un análisis completo del estado de la red.

1. Introducción

En el cambiante panorama digital actual, la ciberseguridad se ha convertido en un desafío crítico y en constante evolución para empresas, gobiernos e individuos. Con el incremento exponencial de los ataques cibernéticos y la creciente sofisticación de las amenazas, resulta esencial adoptar enfoques innovadores para proteger los activos digitales y salvaguardar la integridad de la información.

En este contexto, este proyecto se centra en el desarrollo e implementación de un Sistema de Detección de Intrusiones de Red con Inteligencia Artificial (NIDS-AI), diseñado para fortalecer la seguridad tanto en entornos empresariales como personales. Se presenta el NIDS-AI como una solución innovadora y adaptable a las necesidades actuales de protección de redes. Esta solución integral aprovecha técnicas avanzadas de aprendizaje automático y análisis de datos para detectar y mitigar eficazmente las amenazas cibernéticas en un entorno dinámico y desafiante.

Esta investigación radica en la creciente complejidad de los ataques cibernéticos, que pueden comprometer no solo la confidencialidad, integridad y disponibilidad de los datos, sino también la reputación y la continuidad operativa de las organizaciones. Este trabajo se enmarca dentro de las tendencias y tecnologías más avanzadas en el campo de la ciberseguridad, y busca contribuir al desarrollo de soluciones más inteligentes y adaptables para la protección de redes.

2. Objetivos

El objetivo principal de este proyecto es desarrollar y evaluar un Sistema de Detección de Intrusiones en Red con Inteligencia Artificial (NIDS-AI) que pueda identificar y mitigar amenazas cibernéticas en tiempo real.

Objetivos Generales:

1. Profundizar en la comprensión del funcionamiento de los modelos de aprendizaje automático y su capacidad para aprender a partir de grandes

volúmenes de datos.

2. Realizar un análisis exhaustivo de las necesidades y desafíos específicos de seguridad en entornos empresariales y personales. Abordar aspectos cruciales como la protección contra ataques cibernéticos y la implementación de estrategias sólidas de seguridad en la infraestructura.

3. Diseñar una arquitectura de sistema que integre el procesamiento de datos, soluciones de detección de amenazas y respuestas automáticas. Utilizar servicios en la nube para proporcionar una defensa proactiva contra las amenazas en la red. Permitir a las empresas detectar y responder rápidamente a incidentes de seguridad.

4. Establecer un sistema de monitoreo continuo de eventos de seguridad y del estado de la infraestructura. Generar informes y alertas detalladas sobre las amenazas detectadas. Proporcionar una visión completa de la postura de seguridad de la empresa.

3. Metodología

La metodología utilizada en este proyecto incluye una fase inicial de investigación y selección de tecnologías y datos adecuados , seguida de una toma de requisitos y de una propuesta de presupuesto para este proyecto.

A continuación se realiza un preprocesamiento de los datos a utilizar y de la obtención de seis conjuntos de datos con diferentes distribuciones y tamaños para la predicción en base a la etiqueta(si es un ataque o no) o en base al tipo (que tipo de ataque és). Estos seis conjuntos de datos se han usado para probar la efectividad y el rendimiento de los modelos en diferentes contextos de entrenamiento.

Se realizaron pruebas y análisis de rendimiento de los modelos y se obtuvieron unos resultados. Gracias a este resultado se pudo conocer un modelo que se encargará de nuestro sistema NIDS.

Posteriormente, se desarrolló una aplicación que conectaba con el modelo y con servicios como Zeek y Elastic Stack para probar con datos en tiempo real la efectividad del modelo.

Finalmente, se desarrolló la plataforma de monitorización, integrando Kibana como visualizador de dashboard, para controlar y visualizar los datos que entran en nuestra red, utilizando los datos que extrae Zeek.

Por ahora el sistema de alerta que tiene la aplicación es mediante la impresión de una detección sospechosa de intrusión en la propia parte visual de la aplicación.

Durante el trabajo se ha utilizado metodología Agile Scrum y un sistema de control de versiones con Git.

4. Contexto

A medida que la digitalización avanza y la generación de datos aumenta exponencialmente en nuestra vida cotidiana, la seguridad de la información emerge como una preocupación crítica. La ineludible necesidad de proteger los activos digitales y robustecer la infraestructura frente a una constante evolución de las amenazas cibernéticas se posiciona como una prioridad irrefutable. En este contexto, se propone abordar estos desafíos mediante una investigación exhaustiva, un diseño meticuloso y una implementación rigurosa de una solución integral de ciberseguridad enfocada en el ámbito de las redes y de la inteligencia artificial.

5. Origen de los datos

El dataset UNSW-NB15 fue desarrollado por el Australian Centre for Cyber Security (ACCS) en la Universidad de Nueva Gales del Sur (UNSW), Australia. Su creación tuvo lugar en 2015, con el objetivo de ofrecer un conjunto de datos realista y representativo de los entornos de red modernos. El dataset ha sido actualizado continuamente, con la última actualización realizada el 8 de febrero de 2024.

La generación del dataset UNSW-NB15 involucró la captura de tráfico de red y posteriormente fueron etiquetados manualmente por expertos en seguridad cibernética, asegurando la precisión y la relevancia de las etiquetas.

Este dataset abarca múltiples clases de ataques cibernéticos, como análisis, explotación, infiltración y denegación de servicio (DoS). Con esta diversidad nos ha permitido probar y validar la eficacia de los modelos contra una amplia gama de amenazas.

El conjunto de datasets UNSW-NB15 contiene un total de 22.339.021 registros, distribuidos en veintitrés archivos CSV. Cada registro está compuesto por 46 características que proporcionan una visión detallada del tráfico de red y son esenciales para el análisis profundo de los patrones de tráfico.

6. Preprocesado de datos

Durante el preprocesamiento de datos se explotó la ausencia de valores dentro de los datos y la duplicidad de los mismos, eliminando estos para que el entrenamiento de los modelos sea más preciso y eficiente.

Seguidamente, se comprobó que hubiera integridad en los datos revisando que todos los datos de cada campo fueran valores iguales, es decir que no existiera un campo con valores categóricos y numéricos a la vez.

A continuación se valoró la eliminación del campo temporal ya que no era relevante para nuestro entrenamiento.

Una vez preprocesado los datos se crearon seis conjunto de datos de diferentes tamaños y distribución de conexiones de ataques y benignas para predecir tanto la etiqueta de si es un ataque o no, como para predecir qué tipo de ataque es o si es una conexión normal.

Finalmente, se aplicó a cada modelo una codificación por etiqueta para modificar los valores categóricos y transformarlos en numéricos. Este proceso es requerido por el propio modelo de inteligencia artificial, ya que este solo trabaja con valores numéricos.

7. Entrenamiento de modelos

Con esos seis conjuntos de datos, se entrenaron los modelos:

- Naive Bayes
- Decision Tree
- Random Forest
- XGBoost
- LightGBM
- CatBoost
- Red Neuronal ANN

Estos modelos se han utilizado por su versatilidad a la hora de predecir clases binarias y multiclase.

8. Resultados

En la elección del modelo más apropiado, se destaca la eficacia y versatilidad de los árboles de decisión, específicamente el algoritmo Decision Tree. Este modelo se ha seleccionado por varias razones fundamentales que lo hacen ideal para nuestras necesidades:

Interpretabilidad y Transparencia: Los árboles de decisión son modelos altamente interpretables

Consistencia en Rendimiento:

- *Clasificación Binaria (Label)*:
    - Con un conjunto de datos pequeño obtuvo una precisión de 99,97% y un F1-score de 99,97% con un tiempo de entrenamiento muy corto (0:00:05.386587).
    - Con un conjunto de datos grande, el rendimiento fue perfecto con una precisión y F1-score de 100%, y un tiempo de entrenamiento razonable (0:01:12.157236).
- Clasificación *Múltiple (Type)*:
    - Con un conjunto de datos pequeño, alcanzó una precisión de 99,50% y un F1-score de 99,50% en un tiempo muy corto (0:00:06.917932).

- Con un conjunto de datos grande, mantuvo un alto rendimiento con una precisión de 99,42% y un F1-score de 99,42% con un tiempo de entrenamiento adecuado (0:00:55.299085).

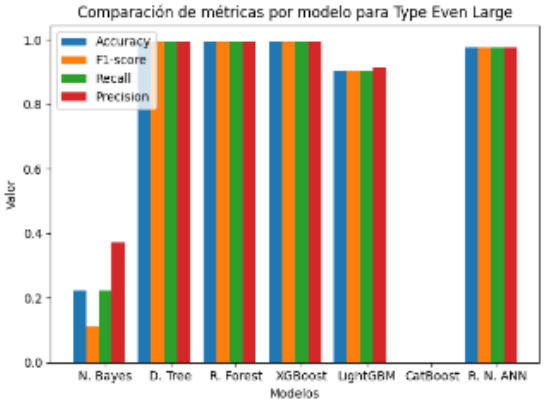

Figura 1: Gráfica de rendimiento de modelos.

9. Infraestructura

Una vez entrenados los modelos, se guardó el mejor modelo entrenado, que en nuestro caso fue un Árbol de Decisión, el cual ofreció resultados casi perfectos.

La aplicación utiliza Tkinter para proporcionar una interfaz visual y carga el modelo guardado, desarrollado utilizando programación orientada a objetos.

Los servicios de Zeek, Filebeat, Elasticsearch y Kibana se encargan de obtener y visualizar todas las conexiones de la red. Zeek se utiliza para la recolección del tráfico de red, Filebeat y Elasticsearch para indexar y mapear esos datos, y Kibana para visualizarlos en un panel interactivo.

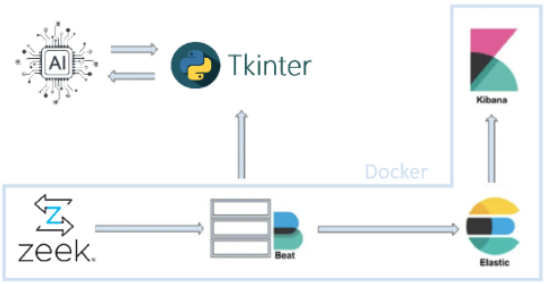

Figura 2: Infraestructura de la aplicación.

10. Interfaz

Para la parte visual del NIDS, se ha utilizado la librería de Tkinter en Python. La parte visual del proyecto nos permitirá identificar las detecciones realizadas por nuestro sistema NIDS-AI en tiempo real, mostrando información sobre las conexiones sospechosas.

Además, para dar mejores resultados, se ha implementado un botón en cada conexión destacada como sospechosa para visualizar los datos que contiene esta conexión en profundidad, como por ejemplo el porcentaje de seguridad que ha tenido el modelo de categorizar una conexión como sospechosa.

Figura 3: Pantalla secundaria de información de la conexión sospechosa.

11. Dashboard

En Kibana se ha diseñado un panel visual utilizando visualizaciones personalizadas para analizar en profundidad el estado de la red.

Este dashboard proporciona información geolocalizada de las IPs de destino en formato de mapa. Además, se representan gráficamente en un gráfico circular los datos relativos a la cantidad de paquetes, los bytes transportados y la duración de las conexiones de las IPs tanto de origen como de destino, lo que permite un mejor análisis de las conexiones de la red. También se incluye

información sobre los puertos utilizados por cada conexión.

Por último, se muestran las conexiones detectadas a lo largo del tiempo en una visualización temporal.

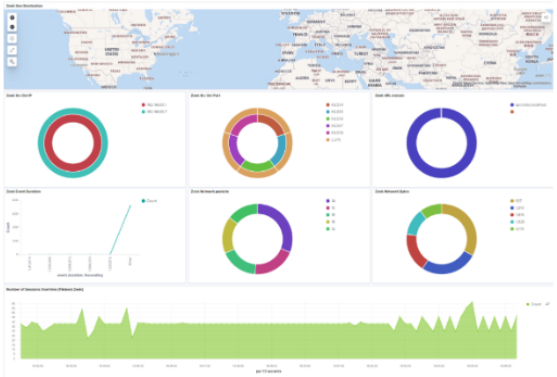

Figura 4: Panel visual de Kibana para el análisis de la red.

12. Resultado Final

El modelo dentro de la aplicación visual se ha mostrado tal y como en la validación posterior realizada. Dando resultados muy satisfactorios y mostrando correctamente resultados de conexiones sospechosas, la gran mayoría con una alta precisión.

13. Futuro Proyecto

El desarrollo del proyecto queda finalizado en este punto, ya que no hay previsión de darle una continuidad ni acabar colgando en un servidor en un entorno de producción. Sin embargo, en caso de continuidad debería mejorar el aspecto visual y adaptarse a dispositivos móviles y tabletas, así como a diferentes sistemas operativos. Actualmente, se ha diseñado solo para ordenadores, ya que es una aplicación de escritorio que une servicios lanzados en Docker.

Por otra parte, sería interesante la posibilidad de mejorar las pipelines que hacen que se conecte los diferentes servicios de Docker con la aplicación visual y utilizar únicamente un entorno, es decir unificar todo a contenedores o unificar todo en una aplicación. Un último punto a considerar por una posible continuidad es la mejora de los modelos, entrenándolos en un entorno computacional mejor y con diferentes datos. Actualmente sólo se permite el entrenamiento bajo unos pocos protocolos, lo que mejoraría drásticamente el rendimiento y la eficacia del modelo.

14. Conclusiones

La meta final marcada al principio del proyecto ha sido alcanzada satisfactoriamente, habiendo realizado la totalidad de las tareas con la funcionalidad de la prueba visual.

Durante el proyecto se ha desarrollado un sistema completo, a partir de una especificación de requisitos y de un posterior diseño, una aplicación frontend y backend con Python, utilizando diversas librerias y herramientas como Docker y Elastic Stack (https://github.com/Franmartin09/TFG-FrontBackVisual). Todas estas partes del sistema, que se comunican entre ellas, han sido verificadas, y una vez validado su comportamiento se ha dado por cerrado el desarrollo. En el mismo repositorio, en el directorio de preproceso está el Notebook de Google Colab que se ha usado para realizar el preprocesamiento de los datos.

Trabajar en este proyecto me ha brindado la oportunidad de aprender nuevas tecnologías y satisfacer mi curiosidad. Asumir y superar este reto ha sido muy satisfactorio. Enfrentarme a tecnologías desconocidas requirió un esfuerzo significativo de investigación y formación, implicando el riesgo de quedarme estancado si no lograba dominarlas.

Desarrollar el proyecto desde el inicio hasta el testing, pasando por la gestión de proyecto, especificación de requisitos, diseño del sistema y desarrollo, me permitió aplicar todos los conocimientos adquiridos durante mi grado y especialidad. Ver cómo estos conocimientos se integraron en un proyecto concreto me demostró que una idea sencilla y bien organizada puede convertirse en un gran proyecto.

15. Agradecimientos

Este trabajo no habría sido posible sin el apoyo y la colaboración de muchas personas a las que me gustaría expresar mi más sincero agradecimiento.

En primer lugar, quiero agradecer a mi director de tesis, el Sr. Masip, por su inestimable orientación, paciencia y consejos a lo largo de todo el proceso.

Agradezco también a mis profesores y compañeros de la Universidad Politécnica de Catalunya por el ambiente académico enriquecedor y por las discusiones constructivas que me han permitido aprender y crecer tanto personal como profesionalmente.

Mi gratitud se extiende a los miembros de mi familia, quienes han sido mi pilar de apoyo incondicional.

Quiero agradecer especialmente a STP GROUP por facilitarme el acceso a los recursos y herramientas necesarias para la investigación.

Finalmente, agradezco especialmente a Valentín Palonsky Guitard, un amigo y compañero, por su apoyo técnico en la utilización de diversas plataformas y tecnologías que han sido cruciales para la elaboración de este trabajo. Su disposición para ayudar y resolver mis dudas ha sido invaluable.

A todos ustedes, mi más sincero agradecimiento por estar a mi lado en este camino. Este trabajo es tanto mío como de todos aquellos que me han acompañado y apoyado en esta travesía. ¡Gracias!

16. Referencias

[1] NIDS, en Wikipedia. ¿Qué es un NIDS? Enlace: https://es.wikipedia.org/wiki/NIDS.

[2] Aprendizaje automático (ML), en Microsoft, ¿Qué es el aprendizaje automático? Enlace: https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform

[3] Inteligencia artificial (IA), en Wikipedia. ¿Qué es la inteligencia artificial? Enlace: https://es.wikipedia.org/wiki/Inteligencia_artificial

[4] Scrum, en Blog, Metodologia Scrum, Roles, Procesos y Artefactos. Enlace: https://blog.innevo.com/metodologia-scrum

[5] Control de versiones Git, en Atlassian, Gitflow workflow Enlace: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

[6] Ciberseguridad, en IBM, ¿Qué es la ciberseguridad? Enlace: https://www.ibm.com/es-es/topics/cybersecurity

[7] Algoritmos de machine learning, en Medium, Tipos de Aprendizaje Automático. Enlace: https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2LG

[8] Naive Bayes, en IBM, Naice Bayes. Enlace: https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareas%20de%20clasificaci%C3%B3n

[9] Decision Tree, en IBM, ¿Qué es árbol de decisión? Enlace: https://www.ibm.com/es-es/topics/decision-trees

[10] Random Forest, en IBM, ¿Qué son árboles aleatorios? Enlace: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems

[11] GBM, XGBoost, LightGBM y CatBoost, en Ciencia de datos, Series Temporales. Enlace: https://cienciadedatos.net/documentos/py39-forecasting-series-temporales-con-skforecast-xgboost-lightgbm-catboost

[12] Redes neuronales artificiales (ANN), en IBM, Neural Network. Enlace: https://www.ibm.com/es-es/topics/neural-network

[13] Métricas de Rendimiento, en Data Source, Métricas de Evaluación de Modelos. Enlace: https://datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

[14] UNSW-NB15, en UNSW Sydney, Datasets UNSW-NB15. Enlace: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[15] Preprocesamineto de datos, en Klippa, ¿Qué es la preparación de los datos? Enlace: https://www.klippa.com/es/blog/informativo/que-es-preparacion-datos/

[16] Zeek, en Zeek, ¿Qué es Zeek y cuales son sus funcionalidades?. Enlace: https://zeek.org/

[17] Elastic Stack, en Elastic, ¿Qué es Elastic Stack y qué componentes tiene?. Enlace: https://www.elastic.co/es/elastic-stack

[18] Tkinter, en Documentación Python, Documentación librería Tkinter. Enlace: https://docs.python.org/es/3/library/tkinter.html

[19] Índices y Mapping, en KeepCoding, ¿Para qué sirven los Indexes y Maps en ElasticSearch? Enlace: https://keepcoding.io/blog/indexes-maps-elasticsearch/