UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

Clasificación de tráfico Web orientadas a la identificación oportuna de ataques usando técnicas de Deep Learning.

AUTOR

Duma Silva Valeria Patricia

Proyecto De Integración Curricular

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ms. Quirumbay Yagual, Daniel Iván

Santa Elena, Ecuador

Año 2024

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez A. Msc.
DIRECTOR DE LA CARRERA

Ing. Quirumbay Yagual, Daniel Iván Mgt
TUTOR

Ing. Iván Coronel Suárez Mgt
Docente Especialista

Ing. Marjorie Coronel S. Mgti.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Valeria Patricia Duma Silva, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 04 días del mes de Diciembre del año 2024

TUTOR

Msia. Quirumbay Yagual, Daniel Iván

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
DECLARACIÓN DE RESPONSABILIDAD

Yo, **Valeria Patricia Duma Silva**

DECLARO QUE:

El trabajo de Titulación, "**Clasificación de tráfico Web orientadas a la identificación oportuna de ataques usando técnicas de Deep Learning.**" previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 04 días del mes de Diciembre del año 2024

EL AUTOR

Valeria Patricia Duma Silva

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Clasificación de tráfico Web orientadas a la identificación oportuna de ataques usando técnicas de Deep Learning.**, presentado por el estudiante Valeria Patricia Duma Silva fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 8%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

CERTIFICADO DE ANÁLISIS
magister

TI_DumaSilvaValeria2

8%
Textos sospechosos

5% Similitudes
< 1% similitudes entre comillas
1% entre las fuentes mencionadas
3% Idiomas no reconocidos
50% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: TI_DumaSilvaValeria2.docx
ID del documento: ac4a558e8fe8412b287a38418e8fa6bd1b9a275f
Tamaño del documento original: 5,19 MB
Autores: []

Depositante: DANIEL IVAN QUIRUMBAY YAGUAL
Fecha de depósito: 3/12/2024
Tipo de carga: interface
fecha de fin de análisis: 3/12/2024

Número de palabras: 18.440
Número de caracteres: 129.659

TUTOR

DANIEL IVAN QUIRUMBAY YAGUAL

Msia. Quirumbay Yagual, Daniel Iván

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Valeria Patricia Duma Silva**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 04 días del mes de Diciembre del año 2021

EL AUTOR

Valeria Patricia Duma Silva

AGRADECIMIENTO

Agradezco a Dios por ser mi guía y fortaleza en este camino. A mi tutor de tesis, el Ing. Quirumbay Yagual, Daniel Iván, le expreso mi más sincero agradecimiento por su invaluable apoyo y dedicación. Su guía y esfuerzo constante han sido clave para el desarrollo de este proyecto, y su compromiso ha dejado una huella imborrable en mi formación.

Me agradezco a mí misma por la valentía de no rendirme, a mi familia por ser mi apoyo incondicional, y a la Universidad por brindarme la oportunidad de crecer como persona y profesional.

A mis docentes, gracias por sus enseñanzas y por inspirarme a superarme. Finalmente, agradezco a todos quienes estuvieron presentes en este proceso y contribuyeron a alcanzar este logro.

De manera especial, agradezco a las personas que conocí recientemente, quienes con su apoyo incondicional y sus palabras de ánimo estuvieron allí para darme fuerzas cuando más lo necesitaba. Su confianza en mí fue un impulso invaluable en este recorrido.

.Valeria Patricia Duma Silva

DEDICATORIA

Dedico esta tesis a mi grupo de amigos ISTA, quienes siempre se preocuparon por mi bienestar y mi superación, demostrando su apoyo incondicional en los momentos más importantes. A sus familias, que también estuvieron presentes, brindándome aliento y respaldo cuando más lo necesité.

Con especial amor, dedico este logro a mi mamá, quien ha sido mi motor y mi mayor inspiración para seguir adelante. Agradezco al señor Esteban Jara, por estar a mi lado en los momentos más difíciles, ofreciéndome su apoyo incondicional.

Finalmente, dedico este trabajo a las personas que he conocido recientemente, como un testimonio de que sí es posible alcanzar los sueños con esfuerzo, determinación y la fe de no rendirse jamás.

Valeria Patricia Duma Silva

ÍNDICE GENERAL

Contenido

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El proyecto tiene como objetivo diseñar un sistema basado en aprendizaje profundo para identificar amenazas cibernéticas en el tráfico web de la Facultad de Sistema y Telecomunicaciones (FACSISTEL). Este sistema se centra en el análisis de archivos de registro (logs) capturados en la red, utilizando modelos avanzados de Deep Learning como redes neuronales recurrentes (GRU), redes recurrentes (DTL) y mapas autoorganizados (SOM). El proceso incluye el preprocesamiento de datos, entrenamiento de algoritmos y evaluación con métricas como precisión y F1-score. Los resultados demuestran una alta efectividad en la detección de tráfico anómalo, minimizando los falsos positivos y permitiendo respuestas proactivas ante accesos a sitios maliciosos. Esta solución tecnológica no solo fortalece la seguridad cibernética de la institución, sino que también establece un modelo escalable y adaptable a redes distribuidas más complejas, posicionándose como una herramienta clave en el ámbito de la ciberseguridad educativa.

.**Palabras claves:** Aprendizaje profundo, detección de anomalías, ciberseguridad

ABSTRACT

The project aims to design a system based on deep learning to identify cyber threats in the web traffic of the Faculty of System and Telecommunications (FACSISTEL). This system focuses on analyzing log files captured in the network using advanced deep learning models such as Gated Recurrent Units (GRU), Deep Transfer Learning (DTL), and Self-Organizing Maps (SOM). The process includes data preprocessing, algorithm training, and evaluation with metrics such as precision and F1-score. The results demonstrate high effectiveness in detecting anomalous traffic, minimizing false positives, and enabling initiative-taking responses to malicious site access. This technological solution not only strengthens the institution's cybersecurity but also establishes a scalable and adaptable model for more complex distributed networks, positioning itself as a key tool in the field of educational cybersecurity.

Keywords: Deep learning, anomaly detection, cybersecurity.

INTRODUCCIÓN

El presente proyecto titulado “Clasificación de tráfico Web orientadas a la identificación oportuna de ataques usando técnicas de Deep Learning” se centra en la implementación de técnicas de aprendizaje profundo para diseñar un sistema avanzado de detección de amenazas. Este enfoque busca identificar patrones anómalos en archivos de registro (logs) de tráfico web, permitiendo alertar sobre accesos a sitios maliciosos dentro de la red institucional.

El desarrollo del proyecto se basa en la captura y análisis de paquetes de red, cuyo contenido es procesado mediante algoritmos de Deep Learning diseñados específicamente para detectar anomalías en datos cifrados. A través de este proceso, el tráfico se clasifica como normal o anómalo, integrando herramientas modernas de inteligencia artificial que superan las limitaciones de los métodos tradicionales de seguridad informática.

En el Capítulo 1, se describe la problemática asociada con la detección de amenazas en entornos académicos y se detalla el marco metodológico utilizado para la construcción del algoritmo. Se incluyen explicaciones sobre las arquitecturas de redes neuronales aplicadas, las técnicas de preprocesamiento de datos y las métricas empleadas para evaluar el desempeño del modelo.

El Capítulo 2 profundiza en el contexto de la institución, la configuración de su infraestructura tecnológica y los experimentos realizados para validar el sistema. Además, se presentan los resultados del modelo entrenado, destacando su capacidad para identificar anomalías en tráfico HTTPS con alta precisión y eficiencia.

Este proyecto contribuye al fortalecimiento de la ciberseguridad en la institución, demostrando cómo el uso de aprendizaje profundo puede transformar el análisis de tráfico de red. Asimismo, genera conocimientos aplicables en diversos entornos donde la seguridad digital es crítica, impulsando la adopción de tecnologías avanzadas en la lucha contra las amenazas cibernéticas.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1 ANTECEDENTES

La Internet actual está mayoritariamente cifrada, con el 94% del tráfico web de Google usando HTTPS [1].Aunque esto mejora la privacidad y seguridad, también presenta desafíos en la era digital. El uso generalizado del protocolo HTTPS ha creado una percepción de seguridad que puede ser engañosa, ya que los ataques a través de canales HTTPS siguen siendo una amenaza significativa [2]. Los ciberdelincuentes explotan el protocolo SSL para ocultar software malicioso, con un 70% de las amenazas web empleando esta técnica. Además, el 60% de las empresas no desencriptan el tráfico HTTPS, omitiendo riesgos importantes [3]. Esto resalta la necesidad de implementar soluciones de inspección de tráfico SSL/TLS. Las empresas deben adoptar tecnologías avanzadas de análisis de tráfico cifrado para detectar y bloquear amenazas [4].

La creciente sofisticación de las amenazas cibernéticas y la rápida evolución de las técnicas de evasión utilizadas por los ciberdelincuentes dificultan la detección y mitigación de amenazas reales. Esta situación plantea un desafío significativo para los sistemas de defensa cibernéticas actuales, que a menudo carecen de la capacidad para detectar y responder eficazmente a los ataques dirigidos a través de conexiones HTTPS [5].

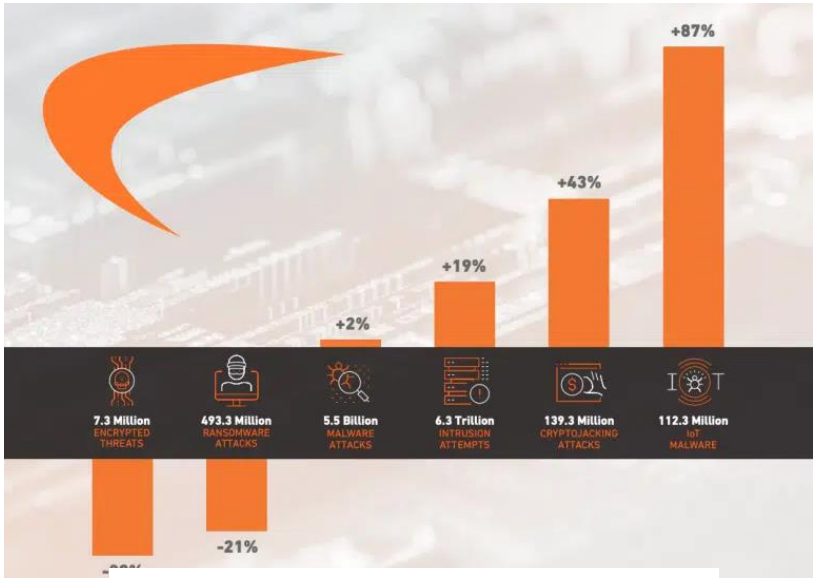


Ilustración 1 Gráfica de porcentajes de virus 2022 [13].

La Universidad Estatal Península de Santa Elena, establecida en 1998, tiene su sede principal en La Libertad, Santa Elena. Dirigida por un Rector y Vicerrector Académico, la institución alberga siete facultades[6], incluyendo la Facultad de Sistema y Telecomunicaciones (FACSISTEL), creada en 2010. Esta última ofrece programas en Tecnología de la Información, Software, Telecomunicaciones, Electrónica y Automatización. El campus cuenta con diversas áreas, como oficinas administrativas, salas para docentes y varios laboratorios, incluido uno especializado en CISCO. La Dirección de Tecnologías de la Información y Comunicación supervisa el uso de los equipos informáticos en los laboratorios de la facultad. (Anexo 1.

Actualmente, existen cuatro laboratorios destinados a actividades académicas estudiantiles. Sin embargo, no se realiza un monitoreo de los sitios web visitados por los alumnos durante sus sesiones. Esta falta de control aumenta el riesgo de acceso a páginas web potencialmente peligrosas, lo que podría comprometer la seguridad de los sistemas informáticos de la universidad.

El estudio 'Una revisión del aprendizaje profundo aplicado a la ciberseguridad'analiza la aplicación de técnicas de aprendizaje profundo en ciberseguridad, destacando su potencial para abordar la creciente complejidad de las amenazas digitales. En un contexto donde los ataques cibernéticos se proyectan alcanzar 10,5 billones de dólares para 2025, con un crecimiento anual del 15%, la investigación examina cuatro técnicas fundamentales de redes neuronales: Perceptrón Multicapa (MLP), Red Neuronal Convolucional (CNN), Red Neuronal Recurrente de Memoria a Corto Plazo (LSTM) y Aprendizaje Profundo por Transferencia (DTL). Estas arquitecturas ofrecen capacidades especializadas para diferentes desafíos de seguridad informática, como detección de intrusiones, identificación de malware, prevención de phishing y análisis de anomalías [7].

Los investigadores enfatizan la importancia crítica de la calidad de los datos en el desarrollo de soluciones de ciberseguridad basadas en inteligencia artificial. Se destaca que el rendimiento de los modelos depende significativamente de las características de los datos, subrayando la necesidad de métodos sofisticados de recopilación y preprocesamiento. A diferencia de los enfoques tradicionales como antivirus y firewalls, las técnicas de aprendizaje profundo permiten una gestión más adaptativa e inteligente,

con capacidad para detectar y clasificar amenazas en tiempo real, desarrollar criterios de clasificación automáticos y responder rápidamente a las tácticas emergentes de los cibercriminales [7]. El estudio concluye presentándose como una guía de referencia para académicos y profesionales, resaltando las continuas oportunidades de investigación en el campo de la ciencia de datos aplicada a la ciberseguridad.

El Premio Nobel de Física destacó las contribuciones de Hopfield y Hinton en inteligencia artificial. Hopfield desarrolló una memoria asociativa para almacenar y reconstruir patrones complejos, mientras que Hinton creó métodos que permiten a las máquinas identificar propiedades de datos de forma autónoma. Estas innovaciones han transformado el aprendizaje automático, permitiendo aplicaciones como el reconocimiento facial y los diagnósticos médicos, consolidando la IA en diversos ámbitos [8].

A nivel técnico, los premiados aplicaron conceptos de la física para diseñar redes neuronales artificiales inspiradas en las conexiones cerebrales humanas, sentando las bases de la IA moderna. Sin embargo, Hinton advirtió sobre los riesgos asociados a su desarrollo incontrolado, subrayando la necesidad de regulaciones éticas para gestionar su impacto [8].

El estudio realizado en el Máster Universitario en Ciberseguridad y Privacidad de la Universidad Internacional de La Rioja en el 2021, con el tema “Malicious URL detection mediante técnicas de Deep Learning” [9], presenta un sistema de detección y clasificación de URLs maliciosas utilizando redes neuronales profundas (DNN). El proyecto emplea un conjunto de datos que incluye características léxicas de las URLs y aspectos de seguridad como la longitud de código JavaScript. Sin embargo, el estudio no propone una solución que notifique automáticamente al administrador cuando se detecta una URL maliciosa.

El estudio realizado por Abbasi et al. en 2021, titulado "Aprendizaje profundo para el monitoreo y análisis del tráfico de red (NTMA): una encuesta" [10], presenta una revisión de las aplicaciones del aprendizaje profundo en el monitoreo y análisis del tráfico de red (NTMA). El proyecto examina cómo los modelos de aprendizaje profundo se aplican para la clasificación y predicción en redes modernas como IoT y celulares, que generan grandes cantidades de datos heterogéneos. Los investigadores analizan técnicas

propuestas, desafíos clave y futuras direcciones de investigación en este campo. Sin embargo, el estudio no propone una solución específica, sino que ofrece una visión general de las aplicaciones existentes y potenciales del aprendizaje profundo en NTMA.

El estudio realizado por Hwang. en 2020, titulado "Un modelo de aprendizaje profundo no supervisado para la detección temprana de anomalías en el tráfico de red"[11] , presenta D-PACK, un mecanismo de detección de tráfico anómalo para el Internet de las Cosas (IoT). El proyecto emplea redes neuronales convolucionales (CNN) y aprendizaje profundo no supervisado para analizar los primeros bytes de los primeros paquetes de cada flujo de red. Los experimentos muestran que, examinando solo dos paquetes por flujo, D-PACK logra una precisión cercana al 100% con una tasa de falsos positivos de 0.83%. Sin embargo, el estudio no aborda el tráfico web con el protocolo SLS/TLS.

En este contexto, surge la necesidad de desarrollar enfoques más avanzados y proactivos para la protección de las comunicaciones HTTPS. El uso de técnicas de aprendizaje profundo (Deep Learning) para el análisis del tráfico web se presenta como una solución prometedora para abordar estas amenazas emergentes. Estas técnicas de inteligencia artificial tienen el potencial de identificar patrones complejos y anomalías en el tráfico HTTPS que podrían pasar desapercibidos para los métodos de detección convencionales [11].

1.2 Descripción del Proyecto

Este proyecto de investigación se centra en el desarrollo de una solución para la detección de amenazas de seguridad en el tráfico HTTPS, utilizando técnicas de aprendizaje profundo (Deep Learning). El objetivo principal es reforzar la capacidad de detección y respuesta frente a ataques cibernéticos en entornos web cifrados, abordando las limitaciones de los métodos de detección convencionales.

El proyecto se desarrollará en varias fases clave: Basándose en la metodología heurística iterativa de **Juliana Martins** y **Carlos Maldonado** [12], combinada con el enfoque de desarrollo de algoritmos propuesto por **Michael Nielsen**[13], se ha creado una fusión de metodologías que se estructura en cinco fases:

1. **Fase de planificación y requisitos**:
    - Se define el problema y los objetivos del modelo.
    - Se investiga y seleccionan los algoritmos adecuados.
2. **Fase de análisis y diseño**:
    - Se recolectan y limpian los datos, llevando a cabo el preprocesamiento necesario.
    - Se diseña la arquitectura del modelo.
3. **Fase de implementación**:
    - Se entrenan los modelos seleccionados utilizando un data set de prueba.
    - Se evalúa el rendimiento inicial para determinar el mejor modelo.
4. **Fase de pruebas**:
    - Se prueba el modelo elegido con un data set real.
    - Se miden las métricas de rendimiento, incluyendo F1 score, recall, precisión y accuracy.
5. **Fase de evaluación y revisión**:
    - Se analizan los resultados y se ajusta el modelo según sea necesario.
    - Se implementa el modelo y se establece un plan de monitoreo continuo.

Este enfoque proporciona un marco estructurado y adaptable para el desarrollo de modelos de Deep learning, integrando principios de ambas metodologías.

Este proyecto contribuirá a mejorar la seguridad cibernética de la institución, generando conocimientos valiosos sobre la aplicación de técnicas de aprendizaje profundo en la detección de amenazas en tráfico web cifrado, lo cual tiene implicaciones más amplias para el campo de la seguridad informática.

1.3 Objetivos del Proyecto

1.3.1 Objetivo General

Desarrollar una solución para la detención de posibles amenazas de seguridad en el tráfico HTTPS, utilizando algoritmos de aprendizaje profundo reforzando la detección y respuesta frente a ataques cibernéticos.

1.3.2 Objetivos específicos

- Recolectar tráfico de red web utilizando herramientas de sniffing y librerías de Python para generar un conjunto de datos limpio y estandarizado, que pueda ser utilizado posteriormente en el entrenamiento de modelos de detección de tráfico malicioso.
- Preparar algoritmos de aprendizaje profundo para el análisis avanzado de los datos del tráfico HTTPS, con el fin de detectar y prevenir ataques cibernéticos dirigidos que puedan evadir los métodos de detección convencionales
- Monitorear las posibles amenazas de seguridad en conexiones cifradas para fortalecer la seguridad en línea, obtener un enfoque integral que permita identificar y responder rápidamente a riesgos cibernéticos, complementado con un dashboard para supervisar los posibles ataques.

1.4 Justificación del Proyecto

Este proyecto de investigación se justifica por varias razones fundamentales que destacan la necesidad de mejorar la seguridad cibernética en la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) de la Universidad Estatal Península de Santa Elena. En un contexto donde la digitalización avanza a pasos agigantados, las instituciones educativas se enfrentan a un panorama cada vez más complejo y amenazante en términos de ciberseguridad. La creciente dependencia de tecnologías digitales para la enseñanza, la gestión administrativa y el acceso a recursos en línea ha hecho que las universidades sean objetivos atractivos para los ciberdelincuentes.

En un contexto donde la digitalización avanza rápidamente, las instituciones educativas enfrentan un panorama cada vez más complejo y amenazante en términos de ciberseguridad. Según un informe de Cybersecurity Ventures, se estima que el costo global del cibercrimen alcanzará los $10.5 billones anuales para 2025, lo que subraya la urgencia de desarrollar mecanismos de defensa más robustos y adaptativos para proteger la integridad y confidencialidad de las comunicaciones en línea [14]

Ilustración 2 Número de víctimas que han sido atacadas en América [15]

Además, el 95% de las violaciones de datos son causadas por errores humanos, lo que resalta la necesidad de implementar sistemas que detecten y prevengan ataques antes de que ocurran[16] . Los métodos tradicionales de detección de amenazas a menudo carecen de la capacidad para identificar ataques sofisticados en el tráfico HTTPS. Actualmente, aproximadamente el 82% del tráfico web es HTTPS, lo que dificulta la inspección por parte de los sistemas convencionales[17]. Este proyecto aborda esta brecha al aplicar técnicas avanzadas de aprendizaje profundo, que tienen el potencial de detectar patrones y anomalías sutiles que podrían pasar desapercibidos por los sistemas tradicionales.

Ilustración 3 Reclamaciones y pérdida de valores a lo largo de cinco años [15]

El proyecto beneficiará directamente a FACSISTEL al proporcionar herramientas avanzadas para el análisis y monitoreo del tráfico de red. Esto permitirá al personal administrativo y técnico tomar decisiones más informadas para proteger y fortalecer la seguridad informática de la institución. La implementación de soluciones tecnológicas alineadas con las mejores prácticas en ciberseguridad es vital para mantener la integridad operativa y reputacional.

Este estudio contribuirá al cuerpo de conocimientos en el campo de la ciberseguridad, específicamente en la aplicación de técnicas de aprendizaje profundo para la detección de amenazas en tráfico web cifrado. Los hallazgos y metodologías desarrolladas podrían tener implicaciones más amplias para la comunidad de seguridad informática, como se ha discutido en investigaciones previas sobre el uso del aprendizaje automático en la identificación de malware[18].

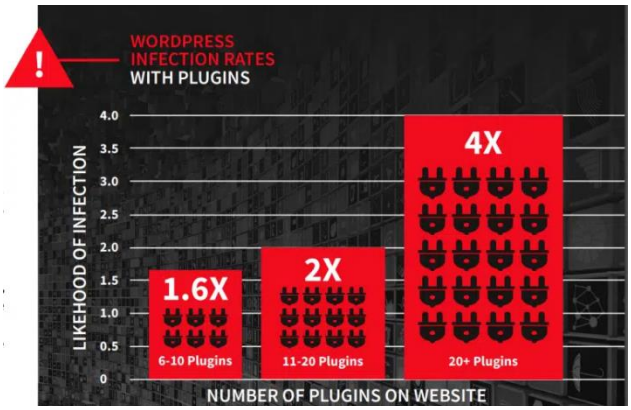


Ilustración 4Gráfica de aumentos de infección por plugins 2022 [13].

El uso de técnicas de inteligencia artificial, como el aprendizaje profundo, para abordar problemas de seguridad cibernética está en línea con las tendencias actuales en tecnología e investigación. Según Gartner, se espera que el mercado global de inteligencia artificial en ciberseguridad crezca a una tasa compuesta anual del 23% hasta 2025[19]. Esta tendencia resalta la importancia y relevancia del proyecto propuesto, que no solo busca mejorar la seguridad dentro de FACSISTEL, sino también contribuir a un enfoque más amplio sobre cómo las instituciones pueden adaptarse a un entorno digital cambiante.

El Deep learning, una subdisciplina del aprendizaje automático se inspira en la estructura y funcionamiento del cerebro humano para crear redes neuronales artificiales capaces de aprender representaciones complejas de datos [19]. Estas redes neuronales, compuestas por múltiples capas, permiten modelar patrones altamente no lineales en grandes conjuntos de datos. Según su arquitectura, los modelos de Deep learning se clasifican en diferentes categorías. Por ejemplo, las redes neuronales convolucionales (CNN) son ideales para el procesamiento de imágenes y videos, mientras que las redes neuronales recurrentes (RNN), como las LSTM y GRU, se utilizan para procesar secuencias de datos, como texto o series temporales. Las redes generativas adversarias (GAN), por su parte, son capaces de generar nuevos datos realistas, como imágenes o música [16]. Esta diversidad de arquitecturas ha permitido al Deep learning revolucionar campos como la visión por computadora, el procesamiento del lenguaje natural y la generación de contenido.

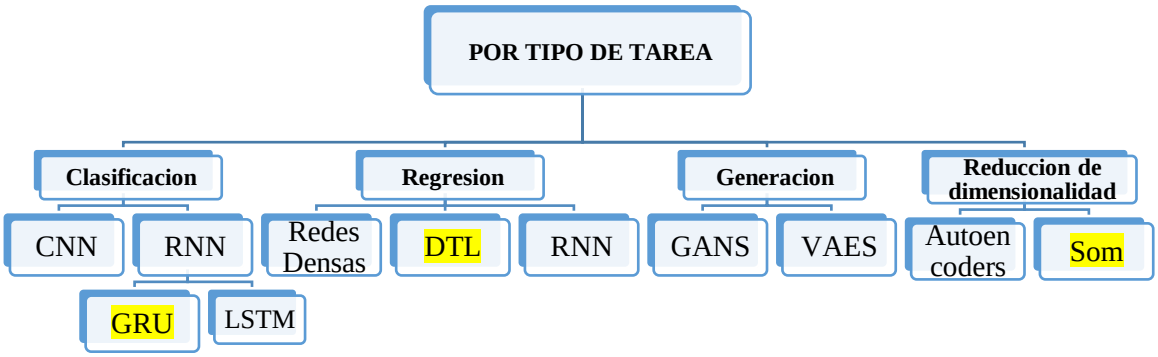


Ilustración 5 Modelos por tipo de tarea

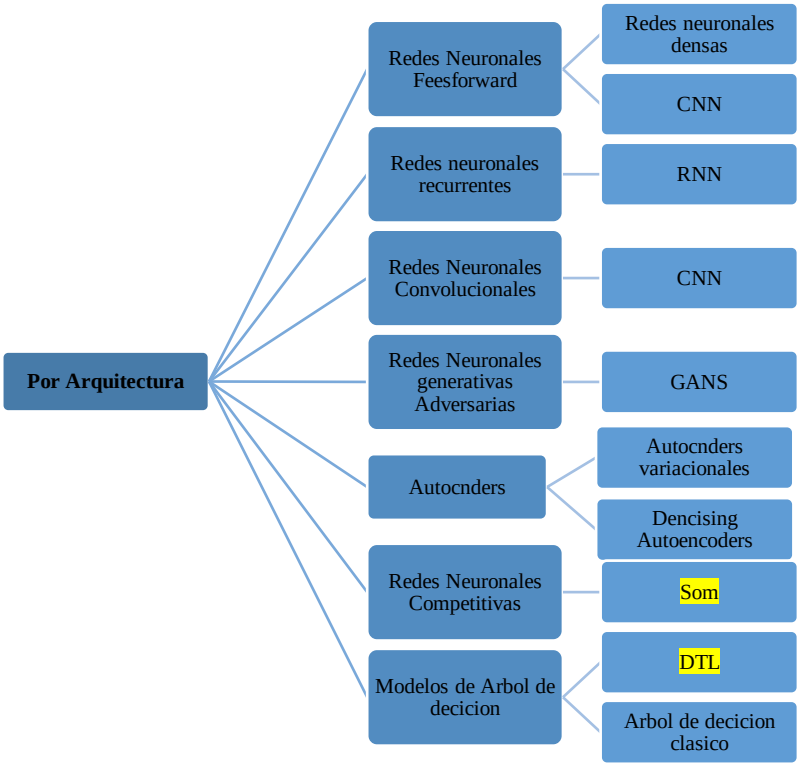


Ilustración 6Tipos de modelos por Arquitectura

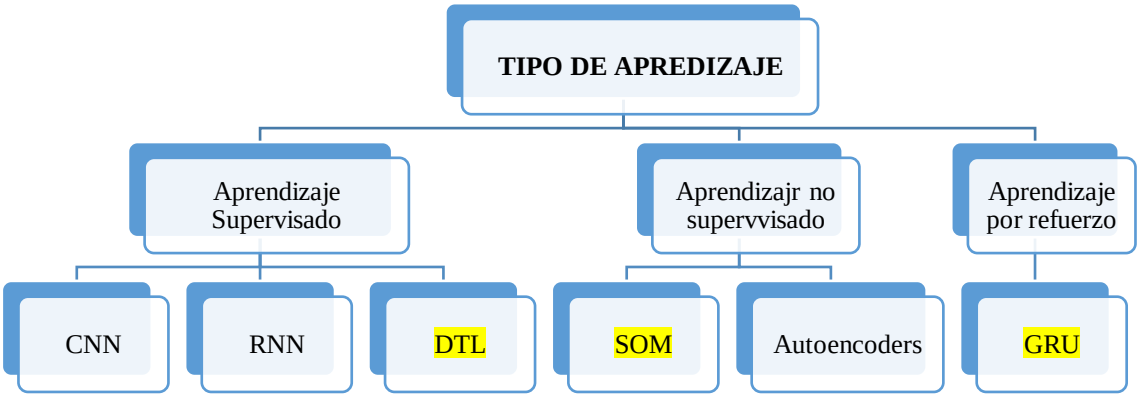


Ilustración 7Modelos por tipo de Aprendizaje

Además, las alertas generadas por nuestro algoritmo se mostrarán en un dashboard diseñado específicamente para este propósito. Este panel permitirá a los administradores visualizar las amenazas detectadas, facilitando una respuesta ante incidentes potenciales. La visualización clara y accesible de estos datos es crucial para una gestión proactiva de

la seguridad cibernética, permitiendo a los responsables tomar decisiones informadas basadas en información actualizada sobre el estado del tráfico web y las amenazas emergentes. Esto no solo mejorará la capacidad reactiva ante incidentes, sino que también proporcionará una herramienta valiosa para el análisis y mejora continua del sistema de seguridad implementado.

1.5 Alcance del Proyecto

El alcance de este proyecto de investigación se centra en el desarrollo e implementación de un sistema de detección de amenazas en tráfico web HTTPS utilizando técnicas de aprendizaje profundo. Específicamente, el proyecto abarcará los siguientes aspectos:

En la fase de planificación y requisitos, el proyecto se centra en definir el problema de detección de tráfico de red y establecer los objetivos del modelo. Esta etapa implica investigar y seleccionar los algoritmos más adecuados, como SOM, DTL y GRU, para determinar cuál se adapta mejor a los datos y al problema específico. Se realiza un análisis preliminar de los requisitos técnicos y de los recursos necesarios para el desarrollo del modelo, asegurando así una base sólida para las etapas posteriores.

Durante la fase de análisis y diseño, se procede con la recolección de datos relevantes y la limpieza de estos, lo que incluye el preprocesamiento necesario para garantizar la calidad y relevancia de los datos. Esta etapa también implica el diseño de la arquitectura del modelo de Deep learning, donde se definen aspectos clave como el número de capas, neuronas y funciones de activación, asegurando que el modelo esté optimizado para el tipo de datos que se manejará.

En la fase de implementación, se lleva a cabo el entrenamiento de los modelos seleccionados utilizando un data set de prueba. Durante este proceso, se ajustan los pesos de la red neuronal a través de técnicas de optimización como el backpropagation. Al finalizar el entrenamiento, se evalúa el rendimiento inicial de cada modelo para identificar cuál muestra el mejor desempeño y es más adecuado para continuar con el desarrollo.

La fase de pruebas se enfoca en validar el modelo elegido utilizando un data set real de tráfico de red. En esta etapa, se miden diversas métricas de rendimiento, incluyendo el F1 score, recall, precisión y accuracy. Estas métricas permiten evaluar la efectividad del

modelo en un entorno más cercano a la realidad, asegurando que sea capaz de generalizar correctamente a datos no vistos.

Finalmente, en la fase de evaluación y revisión, se analizan los resultados obtenidos en la fase de pruebas. Esta etapa incluye la revisión de las métricas de rendimiento, la identificación de áreas de mejora y la realización de ajustes necesarios en el modelo. Una vez optimizado, se implementa el modelo final en un entorno de producción, estableciendo un plan de monitoreo continuo para supervisar su rendimiento y realizar actualizaciones según sea necesario para mantener su efectividad.

1.6 Metodología del Proyecto

1.6.1 Metodología de Investigación

Se utiliza una técnica de investigación exploratoria [20], para buscar datos y proyectos relacionados con el control y monitoreo de la red en relación con las solicitudes HTTPS para la identificación de errores en páginas web para comparar métodos y recursos para llevar a cabo este tipo de análisis.
La investigación de diagnóstico [21], se lleva a cabo mediante un análisis de observación. dentro del Departamento de Sistemas y Telecomunicaciones, para determinar los factores involucrados, las variables a medir y el conocimiento de la situación de control y monitoreo del tráfico de red con respecto a peticiones en línea.

1.6.2 Beneficiarios del Proyecto

1.6.2.1 Beneficiario Directo

El personal administrativo y docentes de la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) de la Universidad Estatal Península de Santa Elena son los beneficiarios de este proyecto de investigación. El desarrollo del algoritmo les permitirá analizar y monitorear el tráfico de red de la facultad para detectar sitios web maliciosos o anómalos. Permitiendo al personal administrativo tomar decisiones razonables sobre cómo proteger y mejorar la seguridad informática de la institución.

1.6.2.2 Beneficiarios indirectos

Los estudiantes de la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) se beneficiarán indirectamente de esta investigación mediante la implementación de un

algoritmo de detección de amenazas que fortalece la seguridad informática. Esto les permitirá realizar sus tareas académicas con mayor confianza, protegiendo la integridad de su información. Además, el personal administrativo contará con herramientas avanzadas para analizar el tráfico de red, lo que facilitará decisiones informadas sobre la seguridad de la institución. El personal técnico de TI desarrollará habilidades en aprendizaje profundo, mejorando sus estrategias de defensa. Así, el entorno académico se vuelve más seguro y eficiente, lo que también puede impactar positivamente en el sector de ciberseguridad en general.

1.6.3 Variables

La presente investigación tiene como objetivo cuantificar el impacto de la implementación de un algoritmo de Deep Learning en la capacidad de detección de paquetes web maliciosos. Específicamente, se busca determinar si el cambio del método de detección actual, basado en revisión manual, a uno automatizado mediante Deep Learning, conlleva un aumento significativo en la detección de paquetes maliciosos y una reducción en la tasa de falsos positivos.

1.6.4 Técnicas de recolección de información

En el proceso de recopilación de datos para la investigación, se emplearon varias técnicas e instrumentos. La observación y la recopilación de datos son dos métodos utilizados. Estas variables proporcionan un marco para medir y analizar el rendimiento y la eficacia del sistema de detección de amenazas basado en aprendizaje profundo. Permitirán evaluar cómo los diferentes factores afectan la capacidad del sistema para identificar amenazas en el tráfico HTTPS y ayudarán a optimizar su funcionamiento.

1.6.5 Análisis de recolección de datos

La recolección de datos para este estudio se llevará a cabo mediante el procesamiento de archivos log proporcionados por el usuario, los cuales contienen registros históricos del tráfico de red web HTTPS. Los datos contenidos en estos logs incluirán tanto tráfico normal como anómalo, los cuales serán transformados en un data set limpio y estandarizado. Para ello, se emplearán librerías de Python como Pandas, Numpy y Scikit-learn para realizar las tareas de preprocesamiento, las cuales incluyen la limpieza, normalización y codificación "one-hot" de los datos. Posteriormente, se realizará un

análisis exhaustivo de los datos para identificar patrones y características relevantes que indiquen la presencia de actividad maliciosa. El conjunto de datos será procesado utilizando diferentes modelos de aprendizaje profundo que han sido previamente entrenados para detectar amenazas en el tráfico HTTPS, permitiendo la clasificación de los registros del log entre tráfico normal y anómalo.

Este proceso permitirá contar con una base de datos estructurada y adecuada para el entrenamiento de los modelos de detección de amenazas, asegurando la calidad y relevancia de los datos utilizados.

1.7 Metodología de Desarrollo

Para este proyecto de detección de amenazas, se utilizará un enfoque heurístico iterativo, el cual integra los cinco pasos del proceso de Juliana Martins, alineados con los principios de Carlos Maldonado[12] para optimizar la detección y clasificación de tráfico malicioso. Este enfoque permite la refinación constante del sistema a través de pruebas y ajustes, garantizando la mejora continua.

1. Fase de planificación y requisitos

En esta fase se definirán los objetivos específicos del sistema de detección de amenazas, estableciendo métricas claras para evaluar el rendimiento, como la precisión, el recall y el F1-score. Alineado con la fase de Exploración Activa de Maldonado, se investigarán las técnicas de detección más recientes, así como nuevos enfoques no convencionales para ampliar el marco de trabajo.

2. Fase de análisis y diseño

El análisis abarcará la evaluación de técnicas de aprendizaje profundo aplicables al sistema de detección. Se diseñará la arquitectura inicial del sistema, donde la Creatividad e Innovación serán fundamentales, fomentando ideas originales y la experimentación con diferentes algoritmos y modelos. Esta fase permite explorar nuevas combinaciones y fomentar soluciones novedosas en la construcción del sistema.

3. Fase de implementación

La implementación comenzará con el desarrollo del prototipo inicial del sistema de detección, aplicando las técnicas seleccionadas durante el diseño. Se mantendrá la Flexibilidad y Adaptabilidad en cada iteración, permitiendo ajustes basados en los

resultados de las pruebas. Esto incluye mejoras en el preprocesamiento de datos, optimización de hiperparámetros y selección de mejores algoritmos.

4. Fase de pruebas

Se realizarán pruebas exhaustivas del prototipo usando datos de tráfico web reales para evaluar el rendimiento del sistema. Las métricas definidas en la fase de planificación se utilizarán para medir la eficacia del sistema. La fase de pruebas estará alineada con el principio de Aprendizaje Continuo, identificando áreas de mejora en el sistema y documentando las lecciones aprendidas para iterar sobre el diseño y la implementación.

5. Fase de evaluación y revisión

La evaluación consistirá en analizar los resultados de las pruebas y recopilar feedback de expertos en seguridad. Con base en este análisis, se planificarán nuevas iteraciones del sistema, refinando tanto la arquitectura como los algoritmos utilizados. Este ciclo de Aprendizaje Continuo se repetirá hasta alcanzar un rendimiento óptimo del sistema.

Este enfoque garantiza que el sistema se desarrolle de manera iterativa y adaptativa, utilizando métricas cuantitativas para evaluar y ajustar continuamente el rendimiento, con el objetivo de optimizar la detección de amenazas en entornos reales.

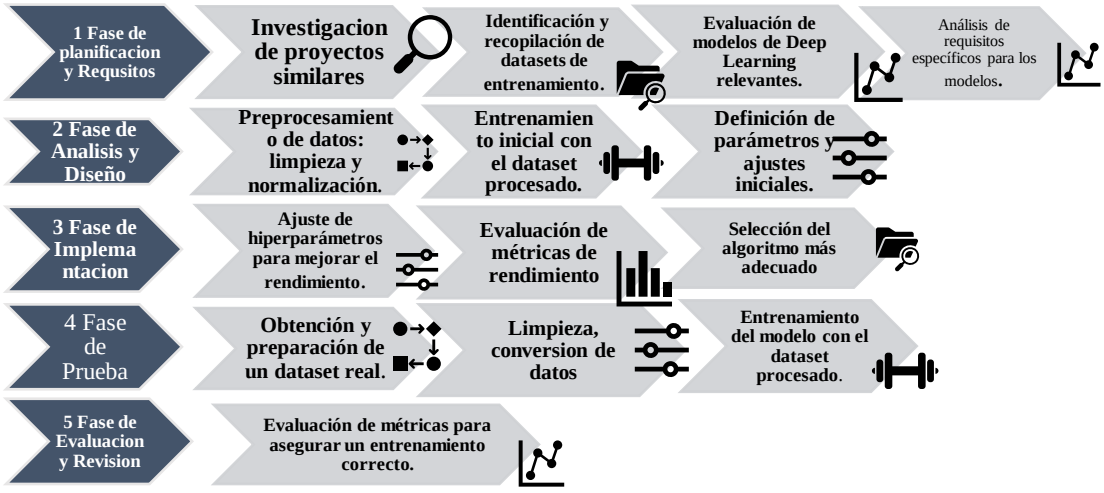


Ilustración 8 Proceso de metodología

2 CAPÍTULO 2. Propuesta

2.1 Marco Contextual

Universidad Estatal Península de Santa Elena

La Universidad Estatal Península de Santa Elena (UPSE), ubicada en la provincia de Santa Elena en el Ecuador, es una institución educativa con amplia trayectoria en la formación de profesionales en diversas áreas del conocimiento. La UPSE fue creada el 02 julio de 1998 mediante la Ley N° 110, y publicada en el suplemento del Registro Oficial N° 366 de 22 de julio de 1998. Su sede se encuentra en la avenida principal de La Libertad-Santa Elena, en el cantón La Libertad. La universidad cuenta con una infraestructura moderna y tecnológica que permite a sus estudiantes y docentes desarrollar sus actividades académicas de manera eficiente y efectiva[22] .

MISIÓN

Formar profesionales que aportan al desarrollo sostenible, contribuye a la solución de los problemas de la comunidad y promueve la cultura [23].

VISIÓN

Ser reconocida por su calidad académica, impacto de sus investigaciones y su aporte al desarrollo de la sociedad [23].

Ubicación

Ilustración 9 Fig. 6. Vista del mapa de evacuación de la Facultad de Sistemas y Telecomunicaciones

2.1.1 Base Legal

2.1.1.1 Constitución de la Republica del Ecuador

Artículo 66.- Derecho a la protección de datos de carácter personal Garantiza a todas las personas el derecho a la protección de sus datos personales. Este derecho implica que el individuo tiene el control y la decisión sobre la información y los datos que le conciernen, así como su correspondiente protección. Cualquier recolección, archivo, procesamiento, distribución o difusión de estos datos o información requerirán la autorización del titular o el mandato de la ley [24]

2.1.1.2 Código Orgánico Integral Penal

Sección sexta

Artículo 178.- Violación a la intimidad

La intromisión no consentida en la privacidad de una persona, a través del acceso o divulgación de datos personales o comunicaciones privadas, será castigada con pena privativa de libertad de uno a tres años[26].

Sección Novena

Artículo 190.- Apropiación fraudulenta por medios electrónicos

El uso fraudulento de sistemas informáticos para cometer delitos como el robo o la transferencia no autorizada de fondos será sancionado con pena privativa de libertad de uno a tres años. Se incluye en esta sanción el uso de técnicas como el hackeo o la falsificación de datos de acceso[26].

La misma sanción se impondrá si la infracción se comete con inutilización de sistemas de alarma o guarda, descubrimiento o descifrado de claves secretas o encriptadas, utilización de tarjetas magnéticas o perforadas, utilización de controles o instrumentos de apertura a distancia, o violación de seguridades electrónicas, informáticas u otras semejantes [26].

Sección tercera

Artículo 230.- Intersección ilegal de datos

"Será sancionada con pena privativa de libertad de tres a cinco años:" [26]

1) La persona que, sin orden judicial previa, en provecho propio o de un tercero, intercepte, escuche, desvíe, grabe u observe, en cualquier forma un dato informático en su origen, destino o en el interior de un sistema informático, una señal o una transmisión de datos o señales con la finalidad de obtener información registrada o disponible [26].

2) Quienes intercepten comunicaciones electrónicas, diseñen sitios web fraudulentos o clonen tarjetas de crédito serán penados con prisión de tres a cinco años. También se sancionará a quienes fabriquen o distribuyan herramientas para cometer este delito [26].

3) La persona que a través de cualquier medio copie, clone o comercialice información contenida en las bandas magnéticas, chips u otro dispositivo electrónico que esté soportada en las tarjetas de crédito, débito, pago o similares [26].

4) La persona que produzca, fabrique, distribuya, posea o facilite materiales, dispositivos electrónicos o sistemas informáticos destinados a la comisión del delito descrito en el inciso anterior [26].

Artículo 234.- Acceso no consentido a un sistema informático

El hackeo o la intrusión en sistemas informáticos para fines ilícitos, como la explotación de servicios o la modificación de información, será sancionado con pena privativa de libertad de tres a cinco años. [26].

2.1.1.3 Ley Orgánica de datos personales

Artículo 37.- Seguridad de datos personales

EL RESPONSABLE O ENCARGADO DEL TRATAMIENTO DE DATOS PERSONALES, SEGÚN SEA EL CASO, DEBERÍA SUJETARSE AL PRINCIPIO DE SEGURIDAD DE DATOS PERSONALES, PARA LO CUAL DEBERÁ TOMAR EN CUENTA LAS CATEGORÍAS Y VOLUMEN DE DATOS PERSONALES, EL ESTADO DE LA TÉCNICA, MEJORES PRÁCTICAS DE SEGURIDAD INTEGRAL Y LOS COSTOS DE APLICACIÓN DE ACUERDO CON LA NATURALEZA, ALCANCE, CONTEXTO Y LOS FINES DEL TRATAMIENTO, ASÍ COMO IDENTIFICAR LA PROBABILIDAD DE RIESGOS [27].

El responsable o encargado del tratamiento de datos personales, deberá implementar un proceso de verificación, evaluación y valoración continua y permanente de la eficiencia y efectividad de las medidas de carácter técnico, organizativo y de cualquier otra índole, implementadas con el objeto de garantizar y mejorar la seguridad del tratamiento de datos personales [27].

Entre otras medidas, se podrán incluir las siguientes:

1) Medidas de anonimización, seudonimización o cifrado de datos personales.

2) Medidas dirigidas a mantener la confidencialidad, integridad y disponibilidad permanentes de los sistemas y servicios del tratamiento de datos personales y el acceso a los datos personales, de forma rápida en caso de incidentes.

3) Medidas dirigidas a mejorar la residencia técnica, física, administrativa y jurídica.

4) Los responsables y encargados del tratamiento de datos personales podrán acogerse a estándares internacionales para una adecuada gestión de riesgos enfocada a la protección de derechos y libertades, así como para la implementación y manejo de sistemas de seguridad de la información o a códigos de conducta reconocidos y autorizados por la Autoridad de Protección de Datos Personales.

Artículo 40.- Análisis de riesgo, amenazas y vulnerabilidades

Para el análisis de riesgo, amenazas y vulnerabilidades, el responsable y el encargo del tratamiento de los datos personales deberán utilizar una metodología que considere, entre otros [27]:

1) Las particularidades del tratamiento.

2) Las particularidades de las partes involucradas

3) Las categorías y el volumen de datos personales objeto de tratamiento

Articulo 41.- Determinación de medidas de seguridad aplicables

Para determinar las medidas de seguridad, aceptadas por el estado de la técnica, a las que están obligadas el responsable y el encargado del tratamiento de los datos personales se deberán tomas en consideración, entre otros [27]:

1) Los resultados del análisis del riesgos, amenazas y vulnerabilidades.

2) La naturaleza de los datos personales.

3) Las características de las partes involucradas.

4) Los antecedentes de destrucción de datos personales, la perdida, alteración, divulgación o impedimento de acceso a los mismos por parte del titular, sean accidentales

e intencionales, por acción u omisión, así como los antecedentes de trasferencia, comunicación o de acceso no autorizado o acceso de autorización de tales datos.

2.2 Marco Conceptual

2.2.1 Sistema de Información

Un sistema de Información (SI) es un conjunto de elementos organizados, relacionados y coordinados entre sí, con el fin de encargarse de facilitar el funcionamiento de una empresa o de cualquier otra actividad humana para conseguir sus objetivos [28].

Estos elementos son:

1. Recursos
2. Equipo Humano
3. Información
4. Actividades

2.2.2 Sistema de informático

Un sistema informático está formado por una mezcla de componentes físicos, como ordenadores, dispositivos periféricos y redes, junto con elementos no tangibles, como sistemas operativos y software de aplicaciones. Todos estos elementos colaboran para llevar a cabo tareas como el procesamiento, almacenamiento y transmisión de datos, y su funcionamiento eficiente depende de la intervención de profesionales capacitados. [29].

2.2.3 Seguridad informática

Esta es una disciplina encargada del diseño de reglas, procedimiento, métodos y técnicas para crear sistemas de información seguros y confiables [30].

Para establecer este sistema de seguridad es necesario conocer lo siguiente

1. Cuáles son los elementos
2. Cuáles son los peligros
3. Cuáles son las medidas

2.2.4 Ataques Informáticos

Un ataque informático es una agresión dirigida a sistemas computacionales, que busca causar daño o disrupción en sus operaciones normales. Los ciber atacantes aprovechan

las vulnerabilidades existentes para obtener acceso no autorizado a sistemas y datos confidenciales, con el fin de obtener ganancias económicas o causar perjuicios a organizaciones y personas. [31].

2.2.4.1 Pharming

El pharming o suplantación de dominio es un delito cibernético muy similar al phishing que consiste en manipular el tráfico del sitio web para obtener información confidencial. Este utiliza el proceso de navegación en Internet, específicamente el mecanismo mediante el cual se transforma una cadena de caracteres, como "www.google.com", en una dirección IP a través de un servidor DNS, permitiendo así establecer la conexión [32].

El exploit ataca de dos maneras:Primero, se puede introducir un virus o troyano en la computadora de un usuario que modifique el archivo de hosts, desviando el tráfico de su destino original hacia un sitio web fraudulento [61], el hacker puede contaminar un servidor DNS para que redireccione a los usuarios a sitios falso [33].

2.2.4.2 Phishing

El phishing es un delito informático en donde por medio de la ingeniería social y medios técnicos para robar los datos de identificación personal de los consumidores y credenciales de cuentas financieras. Los esquemas de ingeniería social utilizan correos electrónicos engañosos que pretenden ser de empresas u organizaciones legítimas, diseñados para dirigir a los consumidores a sitios web falsos que engañan a los destinatarios para que revelen datos financieros, como nombres de usuario y contraseñas [33].

2.2.4.3 Troyanos

Un troyano es un tipo de malware que se disfraza de programa para persuadir a los usuarios y obtener acceso no autorizado a sus sistemas. Al ejecutarse, este software malicioso puede robar información confidencial, controlar el dispositivo a distancia, instalar otros programas dañinos o incluso cifrar los archivos del usuario para extorsionar un pago (ransomware). Los troyanos suelen propagarse a través de correos electrónicos, descargas falsas o sitios web comprometidos, aprovechando la curiosidad o la confianza de los usuarios. Para protegerse, es fundamental mantener actualizado el software, utilizar

un antivirus confiable y ser cauteloso al abrir archivos o hacer clic en enlaces desconocidos [34].

2.2.4.4 Spyware

El spyware es un software malicioso que se infiltra en dispositivos para recopilar información confidencial sin el consentimiento del usuario. Puede robar contraseñas, rastrear hábitos de navegación y causar problemas de rendimiento en el sistema. Se propaga a través de descargas de software infectado, archivos adjuntos maliciosos y sitios web comprometidos. Para protegerte, evita descargar software de fuentes desconocidas, mantén tu sistema actualizado y utiliza un software antivirus confiable con protección contra spyware [35].

2.2.4.5 Adware

El adware es un software intrusivo que se instala en dispositivos sin autorización, mostrando anuncios no solicitados para generar ingresos. A menudo se oculta en programas gratuitos o se introduce a través de vulnerabilidades del sistema. Estos programas pueden rastrear tu actividad en línea, ralentizar tu dispositivo y, en algunos casos, abrir puertas a malware más peligroso. Para protegerte, evita descargar software de fuentes desconocidas, mantén tu sistema operativo y aplicaciones actualizados, y utiliza un software antivirus confiable [36].

2.2.4.6 Malware

Malware es un término general que engloba todo tipo de software malicioso creado para infiltrarse en dispositivos, redes o servicios con el fin de causar daño. Los ciberdelincuentes utilizan el malware para sustraer información sensible, como datos financieros, registros médicos o credenciales de acceso, con el propósito de extorsionar a las víctimas o lucrarse de diversas maneras. La gama de datos que pueden ser comprometidos por el malware es prácticamente ilimitada [37].

2.2.4.7 Ransomware

El ransomware es un tipo de malware que cifra los archivos de un dispositivo, impidiendo el acceso a ellos hasta que se paga un rescate. Este ciberdelito se propaga a través de correos electrónicos fraudulentos, sitios web maliciosos o descargas infectadas. Una vez dentro del sistema, el ransomware cifra los archivos más importantes, como documentos,

fotos y bases de datos, haciendo que sean inutilizables. Los atacantes suelen exigir el pago en criptomonedas para proporcionar la clave de descifrado y restaurar el acceso a los datos. Para protegerse, es fundamental mantener actualizado el software, utilizar un antivirus confiable y evitar abrir archivos o enlaces sospechosos [38]

2.2.4.8 Rootkits

Los rootkits son programas maliciosos diseñados para ocultarse en sistemas operativos, otorgando a los hackers acceso remoto y control total sobre los equipos infectados. Estos programas pueden robar información confidencial, desactivar software de seguridad y manipular el sistema a su antojo. Se propagan a través de vulnerabilidades en software, archivos infectados y dispositivos externos. Detectar y eliminar un rootkit es complejo y, en algunos casos, puede requerir reinstalar el sistema operativo [39]

2.2.4.9 Cryptojacking

El cryptojacking es una ciberamenaza que utiliza los dispositivos de las víctimas para minar criptomonedas sin su consentimiento. Los atacuentes infectan computadoras y dispositivos móviles con malware que consume recursos del sistema para resolver complejos problemas matemáticos. Esto puede provocar un rendimiento lento, sobrecalentamiento y, en casos extremos, daños en el hardware. Para protegerse, es fundamental mantener actualizado el software, utilizar un antivirus confiable, desconfiar de enlaces sospechosos y evitar sitios web no seguros. Además, el uso de bloqueadores de anuncios y extensiones de navegador especializadas puede ayudar a prevenir este tipo de ataques [40]

2.2.5 Redes Distribuidas

Son aquellas que se componen de varios nodos que están interconectados y trabajan de forma coordinada para proporcionar un servicio en conjunto. En el contexto de la detección de anomalías en tráfico URL, las redes distribuidas permiten que se monitoree el tráfico de múltiples puntos de la red, lo que permite detectar patrones de comportamiento anómalos con mayor eficacia [41].

2.2.6 **Análisis de Tráfico de Red**

Es la inspección de los paquetes de datos que se transmiten a través de una red para extraer información sobre su contenido, origen y destino. En el caso de la detección de anomalías

en tráfico URL, el análisis de tráfico de red permite identificar patrones de comportamiento anómalo en las solicitudes de URL y detectar posibles amenazas [38].

Durante el análisis del tráfico de red, se pueden encontrar paquetes con contenido maliciosos o anómalos. Estos paquetes tendrán que ser analizados con más detalle para ver si se tratan de paquetes espontáneos o paquetes con un objetivo oculto [42].

2.2.6.1 Direcciones IP

El protocolo de Internet es un protocolo no orientado a conexión y que funciona a través de una red conmutada de paquetes. Es, por tanto, un protocolo de máximo esfuerzo de entrega de paquetes no confiable. Es uno de los protocolos de Internet más importantes, ya que permite el transporte de paquetes de datos a pesar de que se haga sin garantías [43].

2.2.6.2 Análisis de Paquete

El análisis de paquetes de red permite examinar el tráfico de la red en un nivel granular, evaluando los paquetes individualmente. Proporciona una visión detallada de la información contenida en cada paquete, lo que facilita la comprensión y el análisis específico del tráfico. Por otro lado, el análisis de flujo se enfoca en recopilar metadatos o información resumida sobre el tráfico de red. Esta información incluye detalles como direcciones IP, puertos y protocolos utilizados, permitiendo un análisis estadístico del tráfico en general. Ambos enfoques son complementarios y proporcionan perspectivas útiles para entender y gestionar el tráfico de red de manera efectiva [44].

2.2.6.3 Análisis por Flujo

El análisis de flujo tiene como objetivo recopilar metadatos o información sobre el tráfico de una red. Un flujo de IP se refiere a un conjunto de paquetes con atributos específicos de paquetes IP, donde cada paquete es direccionado y procesado por un conmutador o enrutador, y se incluye la siguiente información [45]:

- IP de origen
- IP de destino
- Puerto de origen
- Puerto de destino
- Clase de servicio

- Tipo de protocolo
- Interfaz

2.2.6.4 Análisis de Cabecera

La cabecera es el inicio de la red donde se procesa la información que se va a enviar a los abonados. Las cabeceras contienen las direcciones de las máquinas de origen y destino, direcciones IP, direcciones que serán usadas por los conmutadores de paquetes, switches y los enrutadores, routers para decidir el tramo de red por el que reenviarán los paquetes [46].

2.2.6.5 IP

Es la parte del direccionamiento de Internet y se encarga de intercambiar paquetes de datos de distintos dispositivos en la red, a todos los dispositivos conectados se les asigna una IP el cual es un número que los identifica en Internet [43].

2.2.6.6 IP v4

Este es uno de los principales protocolos de internet, utiliza 32 bits, teniendo un total de hasta 4300 millones de direcciones IP [43]

2.2.7 Seguridad por capas

La seguridad en capas es una estrategia que combina varios elementos de seguridad, como software antivirus, firewalls y herramientas de evaluación de vulnerabilidades, para crear una barrera defensiva integral y más robusta que la suma de sus partes individuales [45]. Este enfoque aumenta significativamente el costo y la dificultad para que un atacante pueda penetrar en un sistema, lo que reduce la probabilidad de que se convierta en objetivo de ataques. Al implementar la seguridad en capas, se disuade a los atacantes de intentar asediar una institución debido al nivel adicional de protección y complejidad que deben superar [47].

2.2.8 Sitio Web

Es una estructura de información, como muchas otras, donde la peculiaridad de la hipertextualidad y su papel en diferentes escenarios, acceso múltiple y a gran escala, como el ciberespacio[48] .

2.2.9 Protocolos de Transferencia de hipertexto

El Protocolo de Transferencia de Hipertexto (HTTP) es un protocolo de aplicación sin estado que opera sobre la capa de transporte TCP, empleando el método cliente-servidor. Diseñado para permitir la comunicación entre un cliente web (generalmente un navegador). Las respuestas del servidor incluyen un código de estado que indica si la solicitud fue exitosa y el contenido del recurso solicitado. El HTTP es fundamental para la World Wide Web, ya que es el protocolo que subyace a la mayoría de las interacciones en internet [49].

2.2.10 Protocolo HTTPS

El Protocolo de Transferencia de Hipertexto Seguro (HTTPS) es una extensión del HTTP que utiliza el protocolo TLS/SSL para cifrar la comunicación entre un cliente y un servidor. Esto significa que los datos que se transmiten entre ambos están encriptados, lo que hace imposible que terceros los intercepten y lean [50].

2.2.11 Protocolo TLS/SSL

TLS/SSL son protocolos criptográficos diseñados para proporcionar comunicaciones seguras a través de una red, especialmente Internet. TLS (Transport Layer Security) es el sucesor de SSL (Secure Sockets Layer) y se utiliza para cifrar los datos transmitidos entre un cliente y un servidor, garantizando así la confidencialidad, integridad y autenticidad de la información. Estos protocolos establecen un canal seguro a través del cual se pueden intercambiar datos de forma privada, protegiendo contra la interceptación y manipulación de estos. TLS/SSL utilizan certificados digitales para autenticar la identidad de los servidores y establecer claves de cifrado simétricas que permiten una comunicación segura y eficiente [50].

2.2.12 Trafico anómalo

Son actividades en la red que se desvía significativamente del comportamiento normal y esperado. Esto puede incluir patrones inusuales de acceso, volúmenes de datos atípicos, o tipos de tráfico no reconocidos. En otras palabras, es cualquier señal que indique una posible amenaza o actividad sospechosa en la red, como un ataque cibernético, un fallo en el sistema o un error humano [51].

2.2.13 **Página anómala**

Una página anómala se refiere a aquellas que presentan comportamiento o resultado inusual o atípico en comparación con el patrón esperado, siendo su mayor característica, verse con total normalidad en términos de contenido, estructura o comportamiento. La detección de este tipo de páginas es esencial, debido a que permite identificar actividades maliciosas como ataques cibernéticos, phishing o distribución de malware [52].

2.2.14 Modelos de detección de anomalías

La detección de anomalías es una técnica que identifica patrones inusuales en datos. En lugar de buscar ejemplos específicos de irregularidades, estos modelos aprenden el comportamiento normal de los datos y señalan cualquier desviación. Esta técnica es valiosa en campos como la detección de fraudes, donde los patrones pueden cambiar constantemente. Al agrupar datos similares y calcular un índice de desviación para cada punto de datos, se pueden identificar valores atípicos. Sin embargo, es esencial recordar que no todos los valores marcados como anomalías son necesariamente problemáticos, y se requiere un análisis más profundo para confirmar su relevancia [53].

2.2.14.1 Detección basada en firmas.

La detección de firmas es un método que identifica actividades maliciosas en una red comparando el tráfico con una base de datos de patrones conocidos de amenazas. Cuando un software detecta una coincidencia entre el tráfico y una firma almacenada, se activa un mecanismo de seguridad para bloquear el acceso y prevenir daños. Esta técnica es ampliamente utilizada para proteger sistemas informáticos de una variedad de amenazas cibernéticas [54].

2.2.14.2 Detección basada en Heurística

El análisis heurístico es una técnica que detecta virus examinando el comportamiento de un programa en busca de actividades sospechosas. A diferencia de la detección de firmas, que busca coincidencias exactas con virus conocidos, el análisis heurístico identifica patrones y características típicas de malware, incluso en muestras nuevas o modificadas. Esta técnica es crucial para combatir las amenazas emergentes, ya que los cibercriminales constantemente desarrollan nuevos virus. El análisis heurístico funciona tanto de forma estática, examinando el código fuente, como de forma dinámica [55], ejecutando el

programa en un entorno seguro para observar su comportamiento. Al combinar estas técnicas, las soluciones de seguridad pueden detectar y bloquear una amplia gama de amenazas antes de que causen daños.

2.2.14.3 Detección basada en Deep Learning

La detección de comportamientos anómalos ha sido un objetivo clave en el campo de la minería de datos. Si bien las técnicas tradicionales han demostrado su eficacia, el surgimiento del aprendizaje profundo ha abierto nuevas posibilidades para identificar patrones complejos y sutiles, lo que ha mejorado significativamente la detección de actividades maliciosas. [56].

2.2.15 Detección basada en Listas

Existen dos enfoques principales usados en los sistemas de detección de páginas web fraudulentas: listas blancas y listas negras. Los sistemas de detección basados en listas blancas recopilan un conjunto de páginas web consideradas de confianza. Cada página web que no esté incluida en la lista blanca se considera sospechosa. Estos sistemas confían en que las páginas web legítimas se encuentren en la lista blanca y, por lo tanto, se les permite el acceso, mientras que las páginas no listadas se consideran potencialmente fraudulentas [57].

Por otro lado, los sistemas de detección basados en listas negras, también conocidas como blacklists, contienen URLs conocidas de páginas fraudulentas. Estas listas proporcionan un método de control de acceso para evitar que los usuarios visiten estas páginas. Si una URL coincide con la lista negra, se bloquea el acceso a la página web correspondiente [57].

Tanto las listas blancas como las listas negras son utilizadas en los sistemas de detección para clasificar y categorizar las páginas web con el objetivo de identificar posibles amenazas de phishing. Sin embargo, es importante tener en cuenta que estos enfoques tienen limitaciones, ya que las listas deben mantenerse actualizadas constantemente para adaptarse a las nuevas páginas fraudulentas o legítimas que surjan. Además, pueden generar falsos positivos o negativos, dependiendo de la precisión y exhaustividad de las listas utilizadas.

2.2.16 Redes Neuronales

Una red neuronal es un modelo computacional recrea en funcionamiento del cerebro humano, que está diseñado para aprender de la experiencia y realizar tareas complejas a través de la interconexión de múltiples unidades de procesamiento simples, llamadas neuronas artificiales. Al igual que las neuronas biológicas, estas unidades se comunican entre sí y ajustan sus conexiones sinápticas para representar y procesar información de manera eficiente [58].

2.2.16.1 Algoritmo de detección

Los algoritmos univariantes operan utilizando únicamente una señal o un sensor. Generalmente, estos algoritmos crean un modelo por cada señal, el cual se emplea para detectar anomalías en el sensor o la señal correspondiente. El servicio Anomaly Detection permite entrenar un modelo único que abarque múltiples señales dentro de un conjunto de datos, gestionando internamente la relación entre cada sensor o señal y el modelo [59].

2.2.17 Deep Learning

El Deep learning es una rama avanzada del machine learning que permite a las computadoras aprender y tomar decisiones de manera similar a los humanos. A través de redes neuronales artificiales con múltiples capas, estos sistemas pueden reconocer patrones complejos en grandes cantidades de datos, como imágenes, texto o sonido. Esta capacidad ha revolucionado campos como el reconocimiento de voz, la visión por computadora y el procesamiento del lenguaje natural. A diferencia de los métodos tradicionales, el Deep learning no requiere programar reglas específicas para cada tarea, sino que aprende de forma autónoma a partir de los datos. Este progreso se ha visto impulsado por avances en algoritmos, hardware especializado, disponibilidad de datos y la creciente demanda de interfaces más intuitivas [60].

2.2.17.1 Modelo SOM

Los Mapas Autoorganizados (SOM) ofrecen una herramienta poderosa para complementar las arquitecturas de Deep learning. Al proporcionar una forma de visualizar y comprender las representaciones latentes aprendidas por las redes neuronales profundas, los SOM permiten a los investigadores y desarrolladores obtener una mayor comprensión de sus modelos y mejorar su rendimiento. La integración de los SOM en los

pipelines de Deep learning ha abierto nuevas y emocionantes posibilidades para la investigación y la aplicación de la inteligencia artificial [61].

2.2.17.2 Modelo DTL

El Deep Transfer Learning (DTL) es una técnica revolucionaria en el campo del aprendizaje profundo que permite a los modelos aprovechar el conocimiento adquirido en una tarea para resolver problemas en otras áreas relacionadas. En esencia, es como si enseñáramos a una computadora a reconocer gatos y luego, con unos pocos ajustes, la capacitáramos para identificar perros. Esta capacidad de transferir conocimiento preexistente reduce drásticamente el tiempo y los recursos necesarios para entrenar nuevos modelos. Un ejemplo claro de esto se encuentra en el campo de la visión por computadora, donde los modelos reentrenados en grandes conjuntos de datos de imágenes (como Imágenes) se utilizan como punto de partida para tareas más específicas, como la detección de objetos en imágenes médicas o la clasificación de productos en tiendas en línea. El DTL ha demostrado ser especialmente útil en situaciones donde los datos disponibles para una tarea específica son limitados, ya que permite aprovechar el conocimiento adquirido en tareas con grandes cantidades de datos [61].

2.2.17.3 Modelo GRU

Las Unidades Recurrentes Gated (GRU) son un tipo de red neuronal recurrente especialmente diseñada para procesar secuencias de datos. A diferencia de otras redes neuronales, las GRU incorporan un mecanismo de "puertas" que les permite controlar el flujo de información a lo largo de la secuencia. Esto las hace particularmente eficientes en tareas que requieren recordar información a largo plazo, como el procesamiento del lenguaje natural y el reconocimiento del habla. Las GRU consisten en dos puertas principales: una puerta de actualización que decide qué parte de la información de la entrada se actualizará en el estado oculto, y una puerta de restablecimiento que determina qué parte del estado oculto anterior se "olvidará". Gracias a esta arquitectura, las GRU son capaces de aprender representaciones más precisas y complejas de las secuencias, superando a otros modelos en muchas tareas [61].

2.2.18 Herramientas

2.2.18.1 Python

Python es un lenguaje de programación caracterizado por ser potente y fácil de aprender, teniendo una estructura de datos de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos [65]. Su sintaxis elegante y la tipificación dinámica de Python, en conjunto con su naturaleza interpretada, lo convierten en un lenguaje ideal para secuencias de comandos y desarrollo rápido de aplicaciones en muchas áreas en la mayoría de las plataformas. [62]

2.2.18.2 Jupyter

Jupyter Notebook es una herramienta interactiva y de código abierto que permite a científicos de datos, ingenieros y analistas crear y compartir documentos que combinan código ejecutable, visualizaciones y texto enriquecido. Esta plataforma, altamente versátil, facilita la exploración de datos, el desarrollo de modelos y la creación de prototipos. Al soportar múltiples lenguajes de programación, Jupyter Notebook se ha convertido en una herramienta esencial en el campo de la ciencia de datos y la computación científica, permitiendo a los usuarios experimentar y visualizar sus ideas de manera eficiente [63].

2.2.18.3 RStudio

RStudio es un entorno de desarrollo integrado (IDE) diseñado específicamente para el lenguaje de programación R. Imagina que es un taller de herramientas completo para un carpintero, pero en este caso, las herramientas están diseñadas para trabajar con datos y crear análisis estadísticos [64].

2.2.18.4 Dashboard

Un dashboard es una herramienta visual que facilita la comprensión y análisis de datos complejos. A través de una interfaz intuitiva, permite a los usuarios interactuar con información relevante y tomar decisiones de manera más eficaz. Esta herramienta es altamente adaptable, lo que la convierte en una solución personalizada para cada proyecto.[65].

2.2.19 Base de datos

Una Base de datos (BD) es un conjunto de datos ordenado y estructurado que representa la realidad objetiva y que está organizado independientemente de las aplicaciones, por lo que puede ser utilizado y compartido por diferentes usuarios y aplicaciones [66]

2.2.20 API

Las Application Programming Interface (API) son interfaces o zonas de contacto de un conjunto de bibliotecas o paquetes de software, ser visto y ejecutados por otros software o programas. Es decir, las Api son herramientas que permiten que diferentes programas se comuniquen entre sí. La importancia del uso de API radica en su capacidad para permitir que diferentes programas, dispositivos y aplicaciones trabajen en conjunto y compartan información, creando de esta manera una conectividad denominada internet [67].

2.3 Marco Teórico

En el estudio "Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification" [68], se exploró la aplicación del aprendizaje profundo para identificar el tráfico DNS over HTTPS (DoH) encriptado. Los investigadores emplearon una variedad de algoritmos de aprendizaje automático, destacando las redes neuronales recurrentes (LSTM) por su capacidad de manejar secuencias de datos. Los experimentos se realizaron utilizando el conjunto de datos CIRA-CIC-DoHBrw-2020, y los resultados obtenidos demostraron una alta precisión en la clasificación del tráfico DoH, especialmente al utilizar modelos de stacking.

La investigación profundizó en la capacidad del aprendizaje profundo para extraer características relevantes del tráfico encriptado, permitiendo así discriminar entre el tráfico DoH y otros tipos de tráfico HTTPS. Los autores resaltaron la importancia de utilizar modelos de aprendizaje profundo para abordar el desafío creciente de la clasificación de tráfico en entornos de red cada vez más encriptados. Sin embargo, también identificaron la necesidad de realizar más investigaciones para evaluar el rendimiento de estos modelos en diferentes escenarios de red y con conjuntos de datos más diversos [68] .

La patente US12063248B2, propiedad de Google [69], presenta un enfoque innovador para la clasificación de URLs maliciosas empleando técnicas de aprendizaje profundo. Este trabajo propone un marco de "inocente hasta que se demuestre lo contrario" (IUPG), donde las URLs se consideran inicialmente benignas y solo se clasifican como maliciosas si cumplen con ciertos criterios establecidos por un modelo de aprendizaje profundo. Los investigadores de Google desarrollaron y entrenaron este modelo utilizando grandes conjuntos de datos de URLs etiquetadas, lo que permitió al sistema aprender a identificar patrones y características distintivas de las URLs maliciosas.

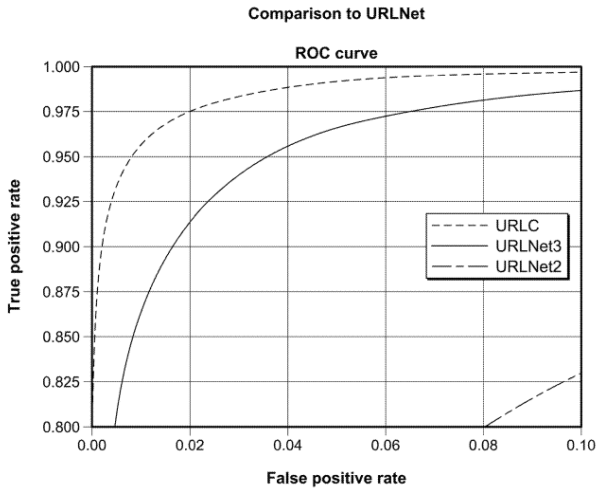


Ilustración 10 Curva de Roc modelo de Google

Malal Aljabri realizo un estudio exhaustivo sobre la detección de URLs maliciosas empleando un conjunto diverso de características y modelos de aprendizaje automático y profundo [70]. La investigación se centró en evaluar la eficacia de combinar características léxicas, de red y basadas en contenido para identificar URLs maliciosas. Los experimentos se llevaron a cabo utilizando un conjunto de datos de gran escala, y se empleó una metodología de validación cruzada de 10 pliegues para garantizar la robustez de los resultados.

Los hallazgos del estudio revelaron que la combinación de múltiples características, especialmente aquellas relacionadas con la estructura de la URL y el contenido de la página web, mejora significativamente la precisión de los modelos de detección. Los modelos de aprendizaje automático tradicionales, como SVM y Random Forest, demostraron un buen desempeño, mientras que los modelos de aprendizaje profundo,

como las redes neuronales recurrentes, mostraron un potencial aún mayor, especialmente en la detección de URLs que contienen código JavaScript. Sin embargo, los autores identificaron algunas limitaciones, como la dependencia de la calidad de los datos de entrenamiento y la dificultad de generalizar los resultados a nuevos tipos de ataques [70].

El artículo sobre el "Agrupación eficiente de correos electrónicos mediante la aplicación de algoritmos de aprendizaje automático supervisado." aborda la creciente relevancia de los servicios de correo electrónico como herramienta clave de comunicación, proyectando un aumento significativo en su uso y, con ello, en los riesgos de ciberseguridad, como el phishing. Este tipo de ataques, que en 2020 generaron pérdidas globales superiores a 54 millones de dólares, resaltan la necesidad de estrategias efectivas para proteger la información sensible. Ante esta problemática, el estudio explora soluciones basadas en aprendizaje automático para clasificar correos electrónicos y mitigar amenazas [71].

La investigación destaca algoritmos supervisados como Naive Bayes, árboles de decisión y Random Forest, que han demostrado alta precisión en la detección de spam y no spam. Al combinar estas técnicas, se lograron métricas sobresalientes en precisión, sensibilidad y puntaje F1. Este sistema se integra en servidores de correo electrónico como Zimbra, permitiendo una clasificación automatizada y eficiente basada en metodologías estructuradas como KDD para el manejo de grandes volúmenes de datos [71].

2.4 Requerimientos

2.4.1 Requerimientos Funcionales

Carga de Archivos	
Código	**Descripción**
RF-1	El sistema debe aceptar únicamente archivos en formato .log para procesar y analizar los datos.
RF-2	Al recibir un archivo, el sistema debe verificar que cumpla con la estructura definida, incluyendo la presencia de campos

	obligatorios y el formato adecuado de datos, antes de iniciar el procesamiento.
RF-3	Tras la validación y carga del archivo, el sistema debe proporcionar un mensaje de confirmación que indique si la operación fue exitosa o si ocurrió algún error específico.
RF-4	Durante la carga del archivo, el sistema debe mostrar una barra de progreso visual en la interfaz para indicar el avance de la operación, mejorando la experiencia del usuario y permitiéndole conocer el estado actual de la carga.

Tabla 1 Requerimientos Funcionales- Carga de Archivos

Preprocesamiento de Datos	
Código	**Descripción**
RF-5	El sistema debe identificar y eliminar automáticamente las columnas que no aportan valor al análisis, de acuerdo con los criterios establecidos.
RF-6	Durante el preprocesamiento, se deben eliminar los registros que contengan datos vacíos o inconsistentes que puedan afectar la precisión de los análisis posteriores.
RF-7	Es necesario reclasificar ciertos campos críticos para garantizar que los datos estén en un formato adecuado para los modelos de detección, optimizando su interpretabilidad y relevancia.
RF-8	Una vez preprocesados, los datos en formato .log deben ser transformados a formato .CSV, estandarizando el archivo para facilitar su integración y análisis en etapas posteriores.

Tabla 2 Requerimientos Funcionales - Preprocesamiento de Datos

Selección de Algoritmos	
Código	**Descripción**
RF-10	El sistema debe proporcionar tres opciones de algoritmos de procesamiento de datos: Self-Organizing Maps (SOM), Gated Recurrent Unit (GRU), y Decision Tree Learning (DTL), cada uno optimizado para diferentes tipos de análisis y patrones de datos
RF-11	La interfaz debe permitir al usuario seleccionar solo un algoritmo a la vez, evitando la ejecución simultánea de múltiples modelos y garantizando un procesamiento optimizado.

Tabla 3 Requerimientos Funcionales - Selección de Algoritmos

Análisis de Datos	
Código	**Descripción**
RF-13	El sistema debe ejecutar el proceso de análisis utilizando el algoritmo previamente seleccionado por el usuario, optimizando el flujo de trabajo según el modelo aplicado.
RF-14	El sistema debe categorizar el tráfico de red en dos clases: normal y anómalo, utilizando los criterios definidos por el algoritmo para identificar posibles patrones o irregularidades.
RF-15	Durante y después del análisis, el sistema debe calcular y reportar métricas de rendimiento como precisión, sensibilidad, y especificidad para evaluar la efectividad y confiabilidad del modelo aplicado.
RF-16	La interfaz debe mostrar un indicador de progreso en tiempo real, proporcionando al usuario visibilidad sobre el avance del análisis y el tiempo estimado de finalización.

Tabla 4 Requerimientos Funcionales -Análisis de Datos

Visualización de Resultados	
Código	**Descripción**
RF-17	El sistema debe generar y mostrar visualizaciones comparativas que permitan al usuario observar y analizar las proporciones entre tráfico normal y anómalo, destacando posibles patrones y anomalías detectadas.
RF-18	Debe incluir un panel de estadísticas detalladas que resuma los resultados del análisis, incluyendo métricas clave como número total de registros, tasas de detección de anomalías, y métricas de rendimiento del algoritmo aplicado como la Ocurrencia, F1score y Precisión.
RF-19	El sistema debe mostrar un gráfico de líneas que represente los datos de tráfico normal y anómalo a lo largo del tiempo. Esta visualización permitirá al usuario identificar tendencias y fluctuaciones en ambos tipos de tráfico para facilitar el análisis de patrones anómalos.

Tabla 5 Requerimientos Funcionales - Visualización de Resultados

2.4.2 Requerimientos Técnicos

Código	Descripción
RT-1	Configuración Mínima
RT-2	Configuración Recomendada 1
RT-3	Configuración Recomendada 2

Tabla 6 Requerimientos Técnicos

RT-1 Configuración Mínima	
Código	**Descripción**
RTM-1	Procesador: Intel Core i5 7ma generación o superior

RTM-2	Memoria RAM: 8GB mínimo
RTM-3	Almacenamiento: 1TB

Tabla 7 Requerimientos Técnicos - Configuración Mínima

RT-2 Configuración Recomendada 1	
Código	**Descripción**
RTR1-1	GPU: Arquitectura Ampere • 2048 núcleos CUDA • 64 núcleos Tensor
RTR1-2	CPU: ARM Cortex-A78AE de 12 núcleos
RTR1-3	Memoria: 64 GB LPDDR5
RTR1-4	Rendimiento de IA: 275 TOPS
RTR1-5	Aceleradores: TPUs y Deep Learning Accelerator (DLA)

Tabla 8 Requerimientos Técnicos- Configuración Recomendada 1

RT-2 Configuración Recomendada 2	
Código	**Descripción**
RTR2-1	CPU: Procesador ARM Cortex de última generación (8 núcleos)
RTR2-2	GPU: Compatible con gráficos avanzados y aceleración de video
RTR2-3	Memoria RAM: 8 GB LPDDR4

RTR2-4	TPU especializado para aprendizaje profundo
RTR2-5	Rendimiento de IA: 4 TOPS
RTR2-6	Optimizado para procesamiento en el borde

Tabla 9 Requerimientos Técnicos - Configuración Recomendada 2

2.5 Componente de la Propuesta Tecnológica

En esta sección se procederá a demostrar la parte práctica de la propuesta tecnológica, donde se desarrolló un sistema de detección de amenazas en tráfico HTTPS mediante algoritmos de aprendizaje profundo. El desarrollo sigue un enfoque heurístico iterativo basado en la metodología de Juliana Martins y los principios de Carlos Maldonado [10], dividiéndose en las siguientes fases:

Fase 1: Planificación y Requisitos

Fase 2: Análisis y Diseño

Fase 3: Implementación

Fase 4: Pruebas

Fase 5: Evaluación y Revisión

*2.5.1.1 **Fase 1: Planificación y Requisitos***

Durante esta fase inicial, se realizaron las siguientes actividades clave:

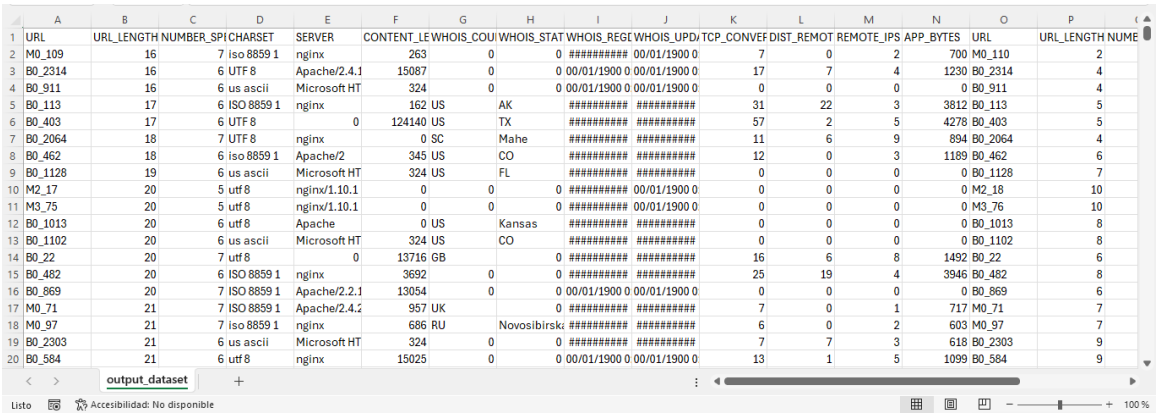
2.5.1.1.1 Adquisición del Data set de Estudio

Ilustración 11 Plataforma Kanggle para la recopilación de un data set limpio

Para este proyecto, se utilizó un conjunto de datos proveniente de Kaggle, una plataforma colaborativa que reúne a expertos en ciencia de datos, análisis y limpieza de información. Esta plataforma permite a investigadores y profesionales compartir data sets de alta calidad para el entrenamiento de modelos de aprendizaje automático.

Específicamente, se trabajó con un conjunto de datos sobre sitios web maliciosos y benignos, el cual fue desarrollado como parte de un proyecto de investigación en seguridad web. Este data set es particularmente valioso debido a la escasez de conjuntos de datos que contengan características detalladas de sitios web tanto maliciosos como benignos.

El conjunto de datos seleccionado incluye información extraída de más de 63,000 URLs, con características tanto de la capa de aplicación como de la capa de red. Los datos fueron recopilados utilizando diversas fuentes verificadas y un honeypot de cliente de baja interacción para el aislamiento del tráfico de red. Este data set constituye la base para el entrenamiento de nuestros modelos de clasificación [72].



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	URL	URL_LENGTH	NUMBER_SPI	CHARSET	SERVER	CONTENT_LE	WHOIS_COU	WHOIS_STAT	WHOIS_REGI	WHOIS_UPD/	TCP_CONVEF	DIST_REMOT	REMOTE_IPS	APP_BYTES	URL	URL_LENGTH	NUME
2	M0_109	16	7	iso 8859 1	nginx	263	0	0	##########	00/01/1900 0	7	0	2	700	M0_110	2	
3	B0_2314	16	6	UTF 8	Apache/2.4.1	15087	0	0	00/01/1900 0	00/01/1900 0	17	7	4	1230	B0_2314	4	
4	B0_911	16	6	us ascii	Microsoft HT	324	0	0	00/01/1900 0	00/01/1900 0	0	0	0	0	B0_911	4	
5	B0_113	17	6	ISO 8859 1	nginx	162	US	AK	##########	##########	31	22	3	3812	B0_113	5	
6	B0_403	17	6	UTF 8	0	124140	US	TX	##########	##########	57	2	5	4278	B0_403	5	
7	B0_2064	18	7	UTF 8	nginx	0	SC	Mahe	##########	##########	11	6	9	894	B0_2064	4	
8	B0_462	18	6	iso 8859 1	Apache/2	345	US	CO	##########	##########	12	0	3	1189	B0_462	6	
9	B0_1128	19	6	us ascii	Microsoft HT	324	US	FL	##########	##########	0	0	0	0	B0_1128	7	
10	M2_17	20	5	utf 8	nginx/1.10.1	0	0	0	##########	00/01/1900 0	0	0	0	0	M2_18	10	
11	M3_75	20	5	utf 8	nginx/1.10.1	0	0	0	##########	00/01/1900 0	0	0	0	0	M3_76	10	
12	B0_1013	20	6	utf 8	Apache	0	US	Kansas	##########	##########	0	0	0	0	B0_1013	8	
13	B0_1102	20	6	us ascii	Microsoft HT	324	US	CO	##########	##########	0	0	0	0	B0_1102	8	
14	B0_22	20	7	utf 8	0	13716	GB	0	##########	##########	16	6	8	1492	B0_22	6	
15	B0_482	20	6	ISO 8859 1	nginx	3692	0	0	##########	##########	25	19	4	3946	B0_482	8	
16	B0_869	20	7	ISO 8859 1	Apache/2.2.1	13054	0	0	00/01/1900 0	00/01/1900 0	0	0	0	0	B0_869	6	
17	M0_71	21	7	ISO 8859 1	Apache/2.4.2	957	UK	0	##########	##########	7	0	1	717	M0_71	7	
18	M0_97	21	7	iso 8859 1	nginx	686	RU	Novosibirsk	##########	##########	6	0	2	603	M0_97	7	
19	B0_2303	21	6	us ascii	Microsoft HT	324	0	0	##########	##########	7	7	3	618	B0_2303	9	
20	B0_584	21	6	utf 8	nginx	15025	0	0	00/01/1900 0	00/01/1900 0	13	1	5	1099	B0_584	9	

output_dataset

Listo Accesibilidad: No disponible 100 %

lustración 12 CVS extraído de Kanggle [72]

2.5.1.1.2 Investigación de Algoritmos

- Se realizó un análisis comparativo de diferentes algoritmos de aprendizaje profundo

	Modelos de algoritmo Deep Learning		
Característica	**SOM (Self-Organizing Maps)**	**DTL (Deep Transfer Learning)**	**GRU (Gated Recurrent Unit)**
Procesamiento de Texto	• Bueno para clustering y visualización de datos textuales. • Útil para análisis de similitud semántica • Limitado en secuencias largas	• Excelente para transferir conocimiento de modelos pre-entrenados • Muy efectivo en tareas NLP • Mantiene contexto semántico	• Especializado en secuencias de texto • Excelente memoria a corto y largo plazo • Ideal para traducción y análisis secuencial
Velocidad	• Entrenamiento relativamente rápido • Inferencia muy rápida • Eficiente en datasets pequeños/medianos	• Entrenamiento lento inicial • Fine-tuning rápido • Inferencia moderadamente rápida	• Entrenamiento moderadamente rápido • Inferencia rápida
Adaptabilidad	• Limitada a datos similares	• Alta adaptabilidad	• Buena adaptabilidad

	• Requiere reentrenamiento para nuevos dominios • Mejor en datos estáticos	• Excelente en transfer learning • Funciona bien en múltiples dominios	• Flexible con diferentes longitudes de secuencia • Requiere ajustes para nuevos dominios
Uso de Recursos	• Bajo consumo de memoria • Eficiente en CPU • Escalable horizontalmente	• Alto consumo de memoria • Requiere GPU para entrenamiento • Pesado en producción	• Consumo moderado de memoria • Beneficiado por GPU
Facilidad de Entrenamiento	• Relativamente simple de configurar • Pocos hiperparámetros Requiere normalización de datos	• Complejo de entrenar desde cero • Simple en fine-tuning Requiere experiencia en deep learning	• Moderadamente complejo • Sensible a hiperparámetros
Aplicaciones Óptimas	• Clustering de documentos • Visualización de datos • Reducción de dimensionalidad	• Clasificación de texto Tareas NLP complejas	• Generación de texto Traducción • Análisis secuencial
Interpretabilidad	• Alta interpretabilidad visual • Fácil de explicar • Mapeo claro de relaciones	• Baja interpretabilidad • Caja negra • Difícil de explicar decisiones	• Interpretabilidad moderada • Puertas visibles • Flujo de información trazable
Escalabilidad	• Buena para datasets pequeños/medianos • Limitada en datos muy grandes • Fácil paralelización	• Excelente para grandes datasets • Requiere infraestructura robusta	• Buena para secuencias largas • Escalable con hardware adecuado • Paralelización moderada
Mantenimiento	• Bajo mantenimiento • Fácil de actualizar	• Alto mantenimiento	• Mantenimiento moderado • Actualizaciones ocasionales

	• Robusto a cambios	• Requiere actualizaciones frecuentes • Sensible a cambios en datos	• Estable en producción

Tabla 10 Comparación de los modelos escogidos

- Se evaluaron las capacidades de cada algoritmo para la detección de anomalías

En el presente estudio comparativo, se implementaron tres modelos de Deep Learning (SOM, GRU y DTL) para la clasificación de URLs maliciosas y benignas. Los experimentos se realizaron sobre un conjunto de datos uniforme de 1,782 muestras, manteniendo una arquitectura consistente con 80 épocas para todos los modelos, lo que permite una comparación equitativa de su rendimiento. La evaluación se fundamentó en métricas estándar de clasificación binaria, incluyendo precisión, sensibilidad y F1-Score, complementadas con el tiempo de ejecución como medida de eficiencia computacional. Los modelos fueron entrenados y validados bajo condiciones idénticas, utilizando la misma distribución de datos y configuración de hiperparámetros, lo que posibilita una evaluación objetiva de sus capacidades intrínsecas en la tarea de detección de URLs maliciosas.

Tabla 11 Métricas de presión después del entrenamiento modelo SOM

SOM	
Métricas	Resultados
Precisión	0,82
Sensibilidad	0,76
F1 Score	0,79
Tiempo de Ejecución	1 minuto

Tabla 12 Métricas de presión después del entrenamiento modelo GRU

GRU	
Métricas	Resultados
Precisión	0,93
Sensibilidad	0,93
F1 Score	0,94
Tiempo de Ejecución	3 minutos

DTL	
Métricas	Resultados
precisión	0,93
Sensibilidad	0,93
F1 Score	0,94
Tiempo de Ejecución	4 minutos

Tabla 13 Métricas de presión después del entrenamiento modelo DTL

Tabla 14 Matriz de confusión de modelo SOM

Matrix de confusión SOM	
Verdaderos Positivos	76
Verdaderos Negativos	597
Falsos Positivos	16
Falsos Negativos	24

Matrix de confusión GRU	
Verdaderos Positivos	82
Verdaderos Negativos	588
Falsos Positivos	25
Falsos Negativos	18

Tabla 15 Matriz de confusión de modelo GRU

Matrix de confusión DTL	
Verdaderos Positivos	76
Verdaderos Negativos	595
Falsos Positivos	18
Falsos Negativos	24

Tabla 16 15 Matriz de confusión de modelo DTL

Tras un análisis exhaustivo de los algoritmos candidatos, se seleccionó el modelo Self-Organizing Map (SOM) como la arquitectura óptima para la implementación final del sistema de detección de URLs maliciosas. Esta decisión se fundamenta en tres criterios críticos de evaluación: primero, su eficiencia computacional superior, demostrada por un tiempo de ejecución de 1 minuto en comparación con los 3 y 5 minutos requeridos por GRU y DTL respectivamente; segundo, su capacidad de adaptación inherente para el reconocimiento de patrones topológicos en datos de red, facilitando la identificación de nuevas variantes de amenazas mediante su arquitectura de mapeo no supervisado; y tercero, su paradigma de aprendizaje simplificado que minimiza la complejidad de hiperparametrización mientras mantiene una precisión aceptable de 0.82. Estas características hacen del SOM una solución escalable y sostenible para la detección de amenazas en tiempo real, equilibrando efectivamente el rendimiento con la eficiencia operativa.

*2.5.1.2 **Fase 2: Análisis y Diseño***

2.5.1.2.1 Preparación del Data set Estudio

- Limpieza inicial de datos
- Estandarización de formatos
- Eliminación de datos inconsistentes

- Normalización de variables

```python
def limpiar_datos(log_content):
    log_data = log_content.splitlines()
    log_list = []

    # Barra de progreso en pantalla para procesar las líneas del log
    st.write("Procesando líneas del log...")
    progress_bar = st.progress(0)
    for i, line in enumerate(tqdm(log_data, desc="Procesando líneas del log", unit="línea")):
        matches = re.findall(r'(\w+)="?([^"\s]+)"?', line)
        log_list.append(dict(matches))
        # Actualizar barra de progreso en pantalla
        progress_bar.progress((i + 1) / len(log_data))

    log_df = pd.DataFrame(log_list)
    log_df.columns = [transformar_cabecera(col) for col in log_df.columns]

    filas_iniciales = log_df.shape[0]
    log_df = log_df.dropna(subset=['Duration', 'Sentbyte', 'Rcvdbyte'])
    filas_despues = log_df.shape[0]

    porcentaje_eliminadas = ((filas_iniciales - filas_despues) / filas_iniciales) * 100
    print(f"Porcentaje de filas eliminadas: {porcentaje_eliminadas:.2f}%")

    columnas_a_eliminar = [
        'Shapingpolicyid', 'Shapingpolicyname', 'Shapersentname', 'Shaperpeipname',
        'type', 'Subtype', 'Vd', 'Srcintfrole', 'Action', 'Srccountry', 'Dstcountry',
        'Dstcity', 'Dstreputation', 'Policytype', 'Trandisp', 'Shaperdropsentbyte',
        'Shaperperipdropbyte', 'Vwlid', 'Wanin', 'Wanout', 'Lanin', 'Countweb',
        'Crscore', 'Craction', 'Crlevel', 'Msg', 'Devtype', 'Srcfamily', 'Srchwversion',
        'Srcswversion', 'Srcserver', 'Countapp', 'Tranip', 'Tranport', 'Dsthwvendor',
        'Masterdstmac', 'Dstmac', 'Dstserver', 'Dstregion', 'Sentdelta', 'Rcvddelta',
        'Durationdelta', 'Sentpktdelta', 'Rcvdpktdelta', 'Identifier', 'Vpntype',
        'Countssl', 'Dstdevtype', 'Dstosname', 'Dstswversion', 'User', 'Group',
        'Dstfamily', 'Dsthwversion', 'Countdns'
    ]

    log_df = log_df.drop(columns=columnas_a_eliminar, errors='ignore')
    print(f"Columnas eliminadas: {len(columnas_a_eliminar)}")

    apprisk_mapping = {'critical': 4, 'elevated': 3, 'high': 2, 'medium': 1, 'low': 0}
    utmaction_mapping = {'allow': 0, 'block': 1}
    level_mapping = {'notice': 0, 'warning': 1}

    log_df['Apprisk'] = log_df['Apprisk'].fillna('elevated').map(apprisk_mapping)
    log_df['Utmaction'] = log_df['Utmaction'].fillna('allow').map(utmaction_mapping)
    log_df['Level'] = log_df['Level'].map(level_mapping)

    columnas_a_convertir = ['Eventtime', 'Srcport', 'Dstport', 'Apprisk', 'Proto', 'Level', 'Utmaction']
    log_df[columnas_a_convertir] = log_df[columnas_a_convertir].fillna(0).astype(int)

    log_df = log_df[log_df['Dstintfrole'] != 'lan']
    log_df['Apprisk'] = log_df['Apprisk'].replace({0: 1, 1: 1, 3: 3, 4: 3})
    log_df = log_df[log_df['Proto'].isin([6, 17])]

    cleaned_csv_file_path = "cleaned_data.csv"
    log_df.to_csv(cleaned_csv_file_path, index=False)
    st.write(f"Archivo CSV guardado exitosamente")

    return cleaned_csv_file_path
```

Ilustración 13 Código de limpieza de los datos

Para garantizar la calidad y coherencia del conjunto de datos utilizado, se llevó a cabo un exhaustivo análisis mediante el empleo de dos herramientas especializadas: Virus Total y Api Criminal . Este proceso tuvo como objetivo principal verificar que los datos no contuvieran elementos maliciosos, fraudulentos o inconsistentes que pudieran comprometer la fiabilidad del análisis posterior. La integración de estas Apis permitió cruzar información de manera efectiva, asegurando que las entradas de la data set cumplieran con criterios de integridad y autenticidad antes de proceder a cualquier otra etapa del flujo de trabajo.

Con los resultados obtenidos de ambas Apis, que analizaron las direcciones IP presentes en el data set, se generó una nueva columna que clasificaba cada IP según su naturaleza. Este enfoque permitió etiquetar con un 0 aquellas direcciones IP consideradas normales y con un 1 las identificadas como anómalas o maliciosas. Esta columna adicional se integró como una característica clave para el entrenamiento de los modelos, proporcionando un criterio sólido y verificable para diferenciar comportamientos legítimos de posibles amenazas. Este proceso no solo mejora la eficacia del entrenamiento, sino que también garantiza que el modelo esté optimizado para detectar y manejar de manera precisa posibles riesgos de seguridad.

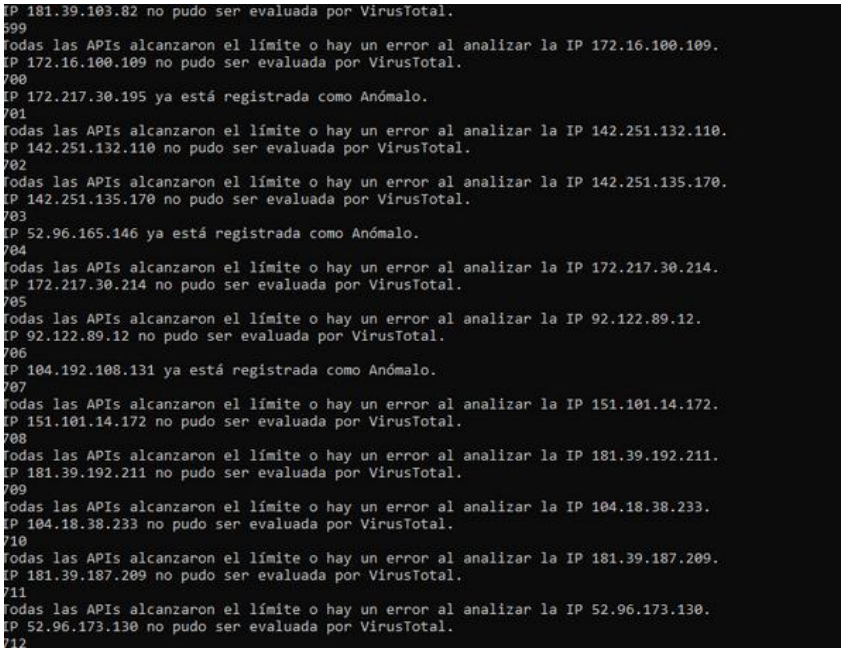


Ilustración 14 Ejecución de Script por la Api de Virus Total

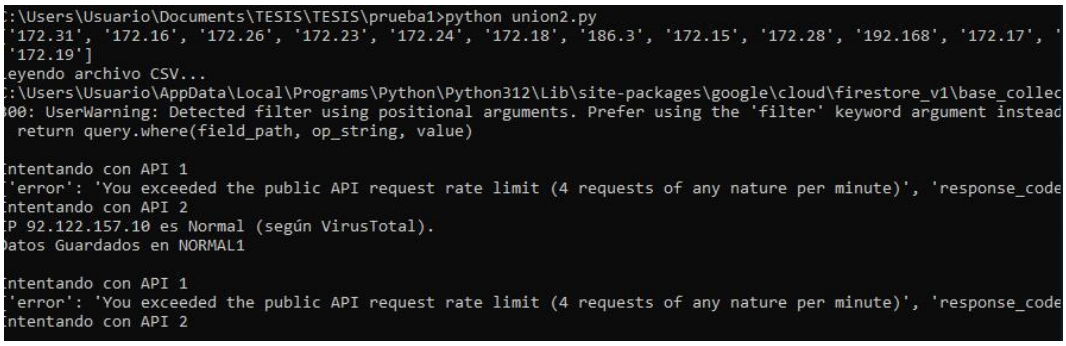


Ilustración 15 Ejecución de Script por la Api de Api Criminal

isk	Applist	Duration	Sentbyte	Rcvdbyte	Sentpkt	Rcvdpkt	Lanout	Utmaction				Ut		Resultado
3		210	2988	3544	70	100	0	0	Hf	W	c0: c0	56875	DI	0
3		208	12545	13932	230	230	0	0		W	f4: f4	56875	DI	0
3		61	606	373	60	40	2010	0		W	f0: f0	56875	UI	0
3	AC_Estudiantes	5	25667	112	4850	20	0	0		W	b8: b8	56875	DI	0
3		974	86857	96742	12050	12280		0	De	W	c0: c0		AF	1
3		1	527	734	60	50	4660	0	TF		30: 30	56875	TL	0
3		1	495	414	50	50	1460	0	TF		30: 30	56875	TL	1
3		240	5506	7566	340	450		0	GI	W	d8: d8		DI	1
3		180	2990	4059	70	60	0	0		W	b8: b8	56875	DI	0
3		180	6501	7724	140	180	0	0		W	ac: ac	56875	SH	1
3		180	6259	9586	120	140		0	TF		d8: d8			0
3		180	4399	5417	110	130		0	TF		d8: d8			0
3		180	6294	7743	110	130	0	0	Hf	W	c0: c0	56875	DI	0
3	AC_Estudiantes	199	0	871	0	130		0		Ar	32: 32		In	0
3	AC_TI	25	260	0	50	0		0	TF		14: 14		TL	0
3	AC_Estudiantes	720	56792279	62549315	6334590	6549950		0		Ar	76: 76			0
3		180	4323	5015	100	130	0	0		Ar	e4: e4	56875	14	0
3		180	8149	7411	120	140	0	0		W	e8: e8	56875	DI	0
3		190	9790	7017	250	310	0	0	GI	W	d8: d8	56875	DI	0

Ilustración 16 Resultados de la columna agregada por el análisis de las Apis

2.5.1.2.2 Diseño de la Arquitectura

El pipeline de entrenamiento sigue una secuencia estándar para problemas de Deep learning:

- Balanceo de Clases: Se utiliza la técnica SMOTE (Synthetic Minority Over-sampling Technique) para equilibrar las clases en el conjunto de datos, lo que ayuda a manejar problemas de desbalanceo.
- División de Datos: Los datos se dividen en conjuntos de entrenamiento y prueba utilizando train_test_split, con proporciones ligeramente diferentes en cada modelo (70-30 o 80-20).
- Normalización: Se aplica StandardScaler para normalizar los datos, lo que ayuda a que todas las características tengan una escala similar.
- Preparación para Modelos: Los datos se redimensionan para ser compatibles con las arquitecturas de cada modelo específico.

Característica	Modelo DTL	Modelo GRU	Modelo SOM
Tipo de Modelo	Deep Transformer Learning	Recurrent Neural Network	Self-Organizing Map
Arquitectura	MultiHeadAttention + Dense	GRU (64+32 unidades)	Mapa 5x5

Número de Neuronas	96 (capa densa)	64 + 32	25 (mapa 5x5)
Sigma	N/A	N/A	1.0
Capas Dropout	0.9	0.9 y 0.6	N/A
Optimizador	Adam	Adam	N/A
Normalización	LayerNormalization	StandardScaler	N/A
Épocas de Entrenamiento	50	50	50
Tamaño de Batch	120	500	N/A
Función de Activación	Sigmoid	Sigmoid	N/A
Tipo de Salida	Probabilidad binaria	Probabilidad binaria	Clúster asignado
Balanceo de Datos	SMOTE	SMOTE	SMOTE
Métricas Evaluación	Accuracy, F1, Precision, Recall	Accuracy, F1, Precision, Recall	Accuracy, F1, Precision, Recall

Tabla 17 Características de los modelos SOM,DTL y GRU

2.5.1.2.3 Códigos de los entrenamientos de los modelos

Los modelos implementados para este proyecto fueron desarrollados y evaluados utilizando Jupyter Notebook como entorno de trabajo. Este entorno fue seleccionado debido a su capacidad para integrar procesamiento eficiente con visualización clara de los resultados, facilitando la depuración de código y el análisis de datos. A continuación, se presentan los códigos de los modelos GRU, DTL, y SOM, cada uno optimizado para la tarea de clasificación binaria de tráfico de red. El modelo GRU utiliza una arquitectura basada en redes recurrentes para capturar patrones temporales, mientras que el modelo DTL emplea un enfoque Transformer que integra mecanismos de atención y

normalización para manejar relaciones complejas en los datos. Por último, el modelo SOM (Self-Organizing Map) se diseñó como un modelo no supervisado que combina el aprendizaje competitivo y la visualización de mapas topológicos. Estos modelos fueron cuidadosamente configurados y ajustados para maximizar su rendimiento en términos de precisión, recall y otras métricas clave.

```
# Cargar el archivo CSV
file_path = 'archivo_combinado31_10_2024.csv'
df = pd.read_csv(file_path, sep=',', low_memory=False)

# Selección de características relevantes
relevant_features = ['Eventtime','Srcport', 'Dstport', 'Duration', 'Sentbyte', 'Rcvdbyte', 'Proto','Level'
                     ,'Apprisk','Utmaction']

X = df[relevant_features].copy()
y = df['Utmaction'].copy()
# Verificar la distribución de clases en 'Resultado'
print("Distribución de clases antes del balanceo:", np.bincount(y))
# Balanceo de clases
smote = SMOTE(random_state=5)
X_balanced, y_balanced = smote.fit_resample(X, y)
# División de datos en entrenamiento y prueba
x_train, x_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.5, random_state=250)
# Normalizar los datos
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
# Definir y entrenar el modelo SOM con monitoreo de precisión
som = MiniSom(x=5, y=5, input_len=x_train_scaled.shape[1], sigma=1.0, learning_rate=0.3)
som.random_weights_init(x_train_scaled)
```

```
def classify(som, data, labels):
    win_map = som.win_map(data)
    labels_map = {}
    for k, values in win_map.items():
        indices = []
        for value in values:
            idx = np.where((data == value).all(axis=1))[0]
            if idx.size > 0:
                indices.append(idx[0])  # Agregar índice válido

        if indices:  # Solo si hay índices válidos
            cluster_labels = labels.iloc[indices].tolist()
            labels_map[k] = np.bincount(cluster_labels).argmax()

    return np.array([labels_map.get(som.winner(x), -1) for x in data])

if not isinstance(y_train, pd.Series):
    y_train = pd.Series(y_train)
if not isinstance(y_test, pd.Series):
    y_test = pd.Series(y_test)

# Listas para almacenar la precisión en cada iteración
train_accuracies = []
validation_accuracies = []
```

```
num_iterations = 20
for i in range(num_iterations):
    som.train_random(x_train_scaled, num_iteration=1)
    y_pred_train = classify(som, x_train_scaled, y_train)
    train_accuracy = accuracy_score(y_train, y_pred_train)
    train_accuracies.append(train_accuracy)
    y_pred_test = classify(som, x_test_scaled, y_test)
    validation_accuracy = accuracy_score(y_test, y_pred_test)
    validation_accuracies.append(validation_accuracy)

    if (i + 1) % 1 == 0:
        print(f"Iteración {i + 1} de {num_iterations} - Precisión en entrenamiento: {train_accur
```

```
# Evaluación del modelo
y_pred_test = classify(som, x_test_scaled, y_test)
accuracy = accuracy_score(y_test, y_pred_test)
precision = precision_score(y_test, y_pred_test, average='weighted')
recall = recall_score(y_test, y_pred_test, average='weighted')
f1 = f1_score(y_test, y_pred_test, average='weighted')
cm = confusion_matrix(y_test, y_pred_test)
```

Ilustración 17 Código Modelo Som

```
# Cargar el archivo CSV
file_path = 'archivo_combinado31_10_2024.csv'
df = pd.read_csv(file_path, sep=',', low_memory=False)

# Selección de características relevantes
relevant_features = ['Eventtime','Srcport'
                     , 'Dstport', 'Duration', 'Sentbyte', 'Rcvdbyte', 'Proto','Level'
                     ,'Apprisk','Utmaction']

X = df[relevant_features].copy()
y = df['Utmaction'].copy()
# Verificar la distribución de clases en 'Resultado'
print("Distribución de clases antes del balanceo:", np.bincount(y))

# Convertir características a enteros y manejar valores nulos en etiquetas
X = X.astype(int)
y = y.fillna(y.median()).astype(int)

# Balanceo de clases
smote = SMOTE(random_state=50)
X_balanced, y_balanced = smote.fit_resample(X, y)

# División de datos en entrenamiento y prueba
x_train, x_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.5, random_state=123)

# Normalizar los datos
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

```
# Redimensionar los datos para que sean compatibles con las capas Transformer (agregar una dimensión de tiempo)
x_train_scaled = np.expand_dims(x_train_scaled, axis=1)
x_test_scaled = np.expand_dims(x_test_scaled, axis=1)

# Definir el modelo DTL
input_shape = (x_train_scaled.shape[1], x_train_scaled.shape[2])
inputs = Input(shape=input_shape)

# MultiHeadAttention y normalización
attention = MultiHeadAttention(num_heads=10, key_dim=99)(inputs, inputs)
attention = Dropout(0.9)(attention)
attention = Add()([inputs, attention])
attention = LayerNormalization(epsilon=1e-6)(attention)

# Capa densa y normalización
dense = Dense(96, activation='relu')(attention)
dense = Dropout(0.9)(dense)
dense = Flatten()(dense)

# Capa de salida
outputs = Dense(1, activation='sigmoid')(dense)
```

```
# Crear y compilar el modelo
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Entrenar el modelo
history = model.fit(x_train_scaled, y_train, epochs=5, batch_size=120, validation_data=(x_test_scaled, y_test))

# Evaluación del modelo
y_pred_test = (model.predict(x_test_scaled) > 0.5).astype(int)

accuracy = accuracy_score(y_test, y_pred_test)
precision = precision_score(y_test, y_pred_test, average='weighted')
recall = recall_score(y_test, y_pred_test, average='weighted')
f1 = f1_score(y_test, y_pred_test, average='weighted')
cm = confusion_matrix(y_test, y_pred_test)
```

Ilustración 18 Código Modelo DTL

```
# Cargar el archivo CSV
file_path = 'archivo_combinado31_10_2024.csv'
df = pd.read_csv(file_path, sep=',', low_memory=False)

# Selección de características relevantes
relevant_features = ['Eventtime','Srcport'
                     , 'Dstport', 'Duration', 'Sentbyte', 'Rcvdbyte', 'Proto','Level'
                     ,'Apprisk','Utmaction']

X = df[relevant_features].copy()
y = df['Utmaction'].copy()
# Verificar la distribución de clases en 'Resultado'
print("Distribución de clases antes del balanceo:", np.bincount(y))


# Convertir características a enteros y manejar valores nulos en etiquetas
X = X.astype(int)
y = y.fillna(y.median()).astype(int)

# Balanceo de clases
smote = SMOTE(random_state=25)
X_balanced, y_balanced = smote.fit_resample(X, y)

# División de datos en entrenamiento y prueba
x_train, x_test, y_train, y_test = train_test_split(X_balanced, y_balanced, test_size=0.5, random_state=500)

# Normalizar los datos
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

```
# Redimensionar los datos para que sean compatibles con las capas GRU
x_train_scaled = np.expand_dims(x_train_scaled, axis=1)
x_test_scaled = np.expand_dims(x_test_scaled, axis=1)

# Definir el modelo GRU
model = Sequential([
    GRU(64, input_shape=(x_train_scaled.shape[1], x_train_scaled.shape[2]), return_sequences=True),
    Dropout(0.9),
    GRU(32),
    Dropout(0.6),
    Dense(1, activation='sigmoid')  # Activación sigmoide para clasificación binaria
])

# Crear un DataFrame con los resultados
resultados = pd.DataFrame({
    'Métrica': ['Accuracy', 'Precision', 'Recall', 'F1 Score'],
    'Valor': [ accuracy_score, precision_score, recall_score, f1_score]
})
```

```
# Compilar el modelo
model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Entrenar el modelo
history = model.fit(x_train_scaled, y_train, epochs=10, batch_size=500, validation_data=(x_test_scaled, y_test))

# Evaluación del modelo
y_pred_test = (model.predict(x_test_scaled) > 0.5).astype(int)

accuracy = accuracy_score(y_test, y_pred_test)
precision = precision_score(y_test, y_pred_test, average='weighted')
recall = recall_score(y_test, y_pred_test, average='weighted')
f1 = f1_score(y_test, y_pred_test, average='weighted')
cm = confusion_matrix(y_test, y_pred_test)
```

Ilustración 19 Código de Modelo GRU

*2.5.1.3 **Fase 3: Implementación***

2.5.1.3.1 Entrenamiento con Data set Estudio

- Implementación de los algoritmos seleccionados SOM, GRU y DTL,
- Entrenamiento inicial con datos educativos
- Validación de métricas preliminares:

- ✓ Precisión
- ✓ Recall
- ✓ F1-score

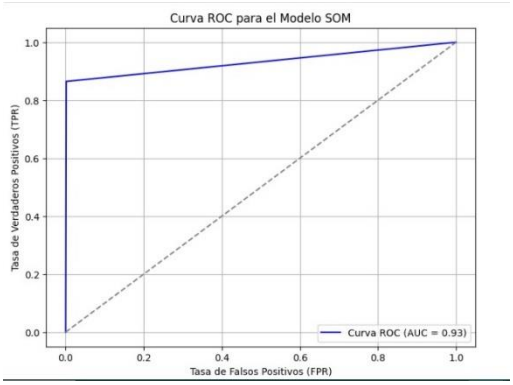


Ilustración 20 Gráfica ROC modelo SOM

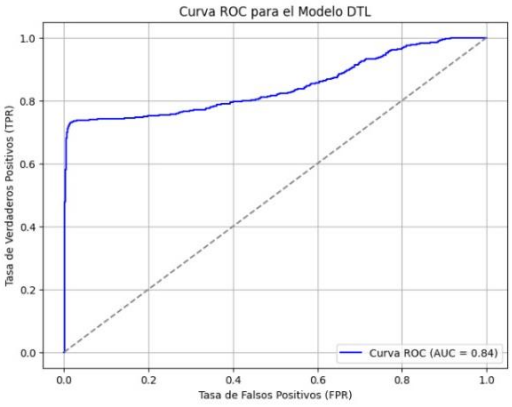


Ilustración 21 Gráfica ROC modelo DTL

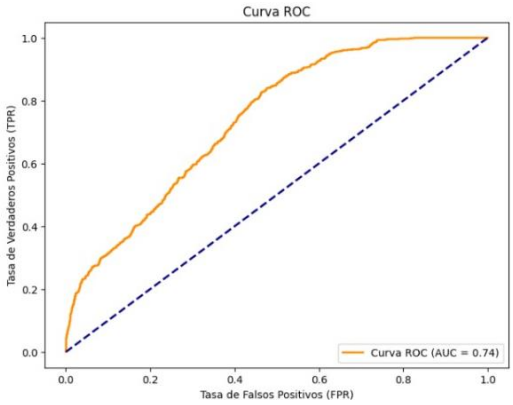


Ilustración 22 Gráfica ROC modelo GRU

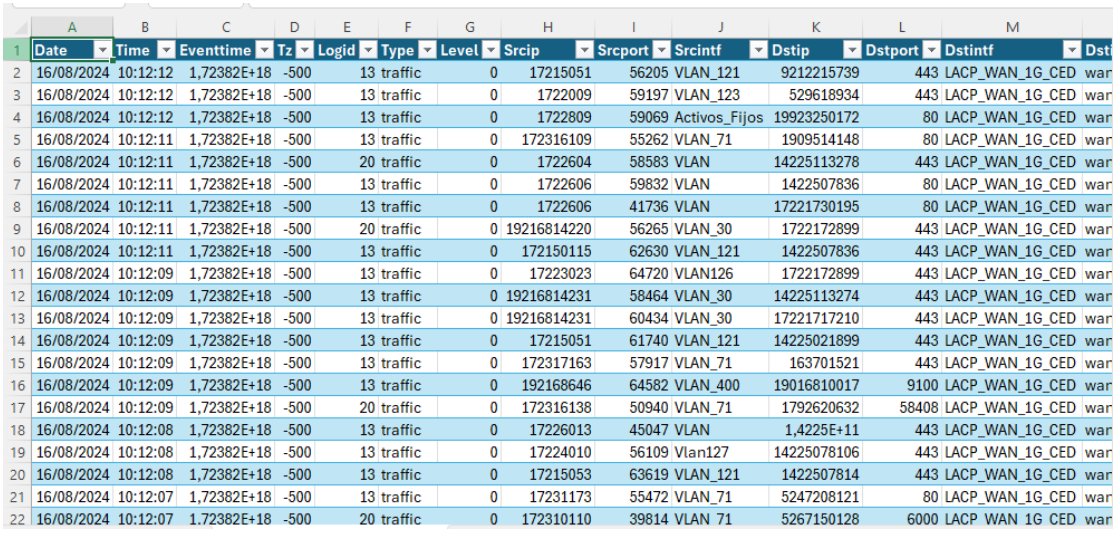
2.5.1.3.2 Procesamiento de Datos Reales

Para la validación del modelo con datos reales, se procesó un archivo de registro (log) proporcionado por el área de Tics, proveniente del firewall Fortinet implementado en la infraestructura universitaria. El data set inicial contenía 506,306 registros de eventos de seguridad. Se ejecutó un proceso sistemático de preprocesamiento de datos que incluyó:

1. Transformación del formato original de logs a una estructura CSV estandarizada
2. Depuración de columnas no relevantes para el análisis predictivo
3. Eliminación de registros con valores nulos o inconsistentes

4. Normalización de variables para optimizar el entrenamiento del modelo

Como resultado del proceso de limpieza y normalización, se obtuvo un conjunto de datos refinado de 454,300 registros, lo que representó una reducción del 25% de los datos originales. Esta depuración fue crucial para garantizar la calidad y consistencia de los datos de entrenamiento, eliminando ruido y redundancias que podrían afectar el rendimiento del modelo.



	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	Date	Time	Eventtime	Tz	Logid	Type	Level	Srcip	Srcport	Srcintf	Dstip	Dstport	Dstintf	Dst
2	16/08/2024	10:12:12	1,72382E+18	-500	13	traffic	0	17215051	56205	VLAN_121	9212215739	443	LACP_WAN_1G_CED	war
3	16/08/2024	10:12:12	1,72382E+18	-500	13	traffic	0	1722009	59197	VLAN_123	529618934	443	LACP_WAN_1G_CED	war
4	16/08/2024	10:12:12	1,72382E+18	-500	13	traffic	0	1722809	59069	Activos_Fijos	19923250172	80	LACP_WAN_1G_CED	war
5	16/08/2024	10:12:11	1,72382E+18	-500	13	traffic	0	172316109	55262	VLAN_71	1909514148	80	LACP_WAN_1G_CED	war
6	16/08/2024	10:12:11	1,72382E+18	-500	20	traffic	0	1722604	58583	VLAN	14225113278	443	LACP_WAN_1G_CED	war
7	16/08/2024	10:12:11	1,72382E+18	-500	13	traffic	0	1722606	59832	VLAN	1422507836	80	LACP_WAN_1G_CED	war
8	16/08/2024	10:12:11	1,72382E+18	-500	13	traffic	0	1722606	41736	VLAN	17221730195	80	LACP_WAN_1G_CED	war
9	16/08/2024	10:12:11	1,72382E+18	-500	20	traffic	0	19216814220	56265	VLAN_30	1722172899	443	LACP_WAN_1G_CED	war
10	16/08/2024	10:12:11	1,72382E+18	-500	13	traffic	0	172150115	62630	VLAN_121	1422507836	443	LACP_WAN_1G_CED	war
11	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	17223023	64720	VLAN126	1722172899	443	LACP_WAN_1G_CED	war
12	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	19216814231	58464	VLAN_30	14225113274	443	LACP_WAN_1G_CED	war
13	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	19216814231	60434	VLAN_30	17221717210	443	LACP_WAN_1G_CED	war
14	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	17215051	61740	VLAN_121	14225021899	443	LACP_WAN_1G_CED	war
15	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	172317163	57917	VLAN_71	163701521	443	LACP_WAN_1G_CED	war
16	16/08/2024	10:12:09	1,72382E+18	-500	13	traffic	0	192168646	64582	VLAN_400	19016810017	9100	LACP_WAN_1G_CED	war
17	16/08/2024	10:12:09	1,72382E+18	-500	20	traffic	0	172316138	50940	VLAN_71	1792620632	58408	LACP_WAN_1G_CED	war
18	16/08/2024	10:12:08	1,72382E+18	-500	13	traffic	0	17226013	45047	VLAN	1,4225E+11	443	LACP_WAN_1G_CED	war
19	16/08/2024	10:12:08	1,72382E+18	-500	13	traffic	0	17224010	56109	Vlan127	14225078106	443	LACP_WAN_1G_CED	war
20	16/08/2024	10:12:08	1,72382E+18	-500	13	traffic	0	17215053	63619	VLAN_121	1422507814	443	LACP_WAN_1G_CED	war
21	16/08/2024	10:12:07	1,72382E+18	-500	13	traffic	0	17231173	55472	VLAN_71	5247208121	80	LACP_WAN_1G_CED	war
22	16/08/2024	10:12:07	1,72382E+18	-500	20	traffic	0	172310110	39814	VLAN_71	5267150128	6000	LACP_WAN_1G_CED	war

Ilustración 23 Parte de la data set normalizado

2.5.1.4 ***Fase 4: Pruebas***

2.5.1.4.1 Evaluación con Datos Reales

Para la validación del modelo con datos reales, se implementó un proceso de verificación dual utilizando las Apis de CriminalIP y Virus Total como herramientas de referencia externa. Esta validación se centró en tres campos críticos del data set: Apprisk, Level y Utmaction, los cuales son indicadores de alertas en los registros del firewall. El análisis comparativo con estas Apis de seguridad reveló que el campo Utmaction presentaba la correlación más alta con la detección de tráfico de red anómalo, estableciéndose como el indicador más fiable para la clasificación de amenazas.

Utilizando este data set validado, se procedió al reentrenamiento de los tres modelos con 50 épocas, obteniendo los siguientes resultados:

Los resultados demuestran una mejora significativa en la precisión de todos los modelos al ser entrenados con datos reales validados, manteniendo el patrón de eficiencia temporal observado en las pruebas iniciales, donde SOM continúa destacando por su menor tiempo de ejecución mientras alcanza métricas de rendimiento competitivas.

Dado que los tiempos de procesamiento iniciales resultaron sig

nificativamente elevados para un entorno de producción, se implementó un proceso de optimización de los modelos. Esta optimización se centró en dos aspectos fundamentales: la reducción del número de épocas de entrenamiento y el ajuste fino de la arquitectura neuronal. El objetivo principal fue mantener la eficacia predictiva mientras se mejoraba sustancialmente la eficiencia computacional.

Los resultados post-optimización demostraron una mejora extraordinaria en los tiempos de ejecución:

El modelo SOM mantuvo su precisión de 0.93 mientras redujo su tiempo de ejecución de 45 minutos a solo 4 minutos

GRU conservó su precisión de 0.98 con una reducción de tiempo de 1 hora a 6 minutos

DTL sostuvo su precisión de 0.98 mientras mejoró de 1 hora y 20 minutos a 6 minutos

Esta optimización representa una reducción del tiempo de procesamiento de aproximadamente 90% sin comprometer las métricas de rendimiento (Precisión, Sensibilidad y F1-Score). La significativa mejora en la eficiencia temporal hace que estos modelos sean ahora viables para implementación en entornos de producción, permitiendo respuestas más ágiles a las necesidades de detección de amenazas en tiempo real mientras se mantiene la robustez del análisis predictivo.

Tabla 18 Métricas de presión después de la optimización SOM

SOM	
Métricas	Resultados
Precisión	0,93
Sensibilidad	0,93
F1 Score	0,93
Tiempo de Ejecución	4 minutos

DTL	
Métricas	Resultados
Precisión	0,98
Sensibilidad	0,98
F1 Score	0,98
Tiempo de Ejecución	6 minutos

Tabla 19 Métricas de presión después de la optimización del modelo DTL

GRU	
Métricas	Resultados
Precisión	0,98
Sensibilidad	0,98
F1 Score	0,98
Tiempo de Ejecución	6 minutos

Tabla 20 Métricas de presión después de la optimización del modelo GRU

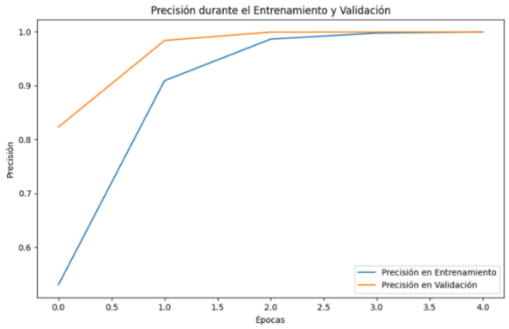


Ilustración 24 Gráfica de Validación y entrenamiento modelo SOM

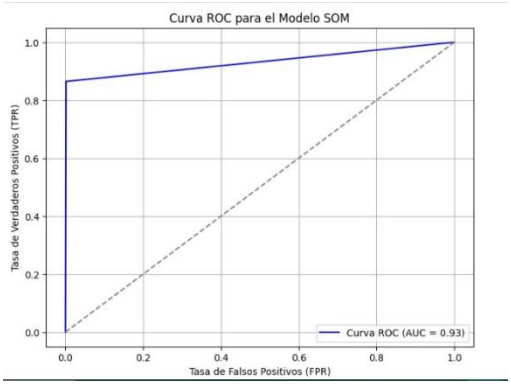


Ilustración 27 Gráfica ROC modelo SOM

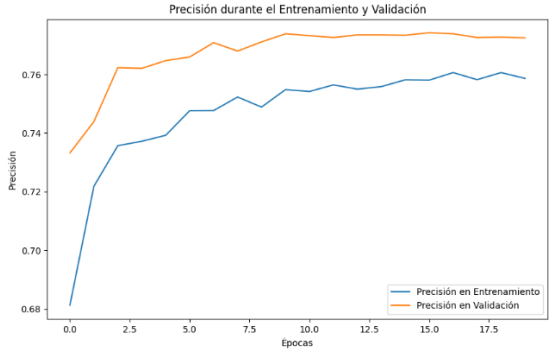


Ilustración 26 Gráfica de entrenamiento y validación modelo DTL

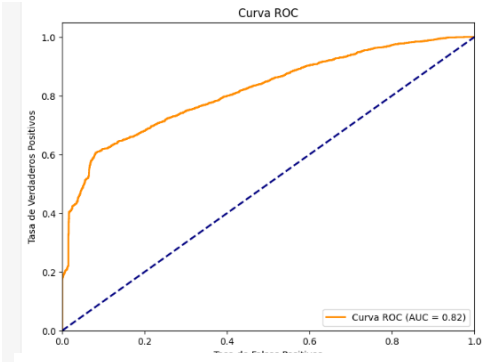
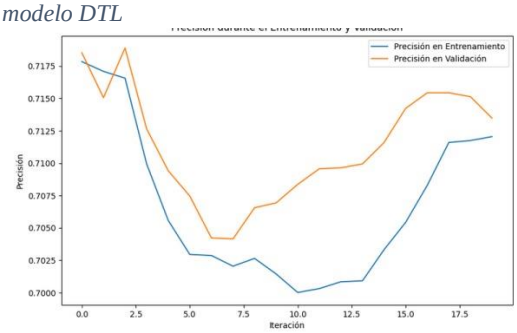
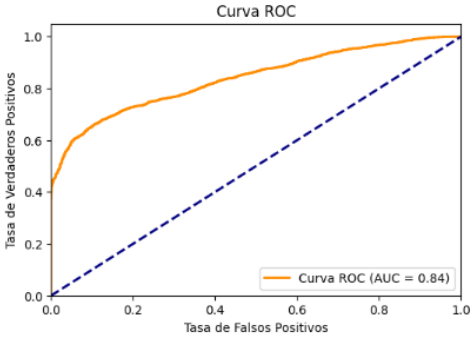


Ilustración 25 Gráfica ROC modelo DTL





2.5.1.4.2 Desarrollo del Dashboard

- Implementación de interfaz de visualización
- Integración de gráficos de procesamiento
- Desarrollo de visualizaciones para:
  - ✓ Proceso de limpieza de datos
  - ✓ Rendimiento de algoritmos
  - ✓ Detección de anomalías
  - ✓ Métricas de rendimiento

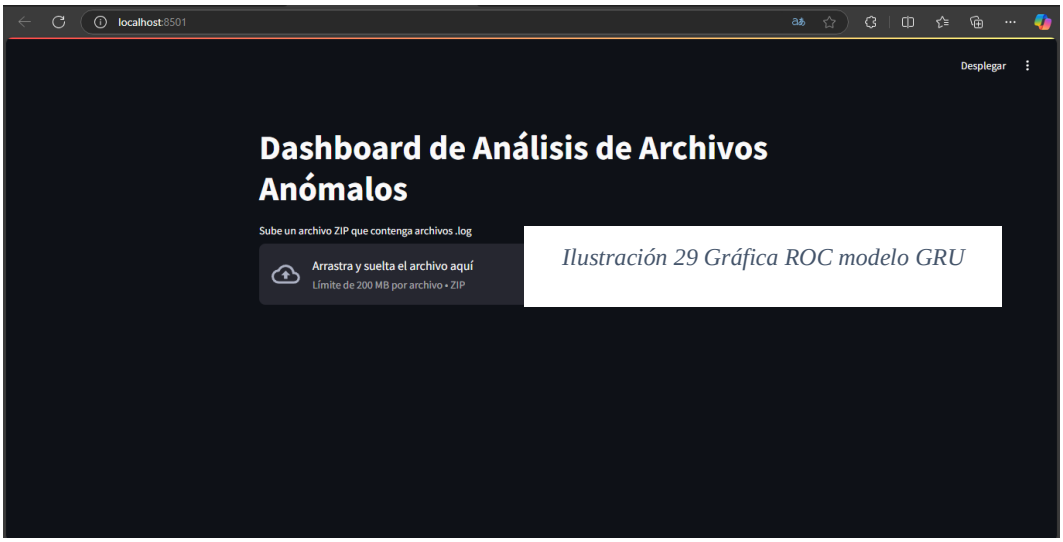


Ilustración 29 Gráfica ROC modelo GRU

Ilustración 30 Barra de progreso para la carga del archivo log al programa

Ilustración 31 Pantalla de inicio del Dashboard

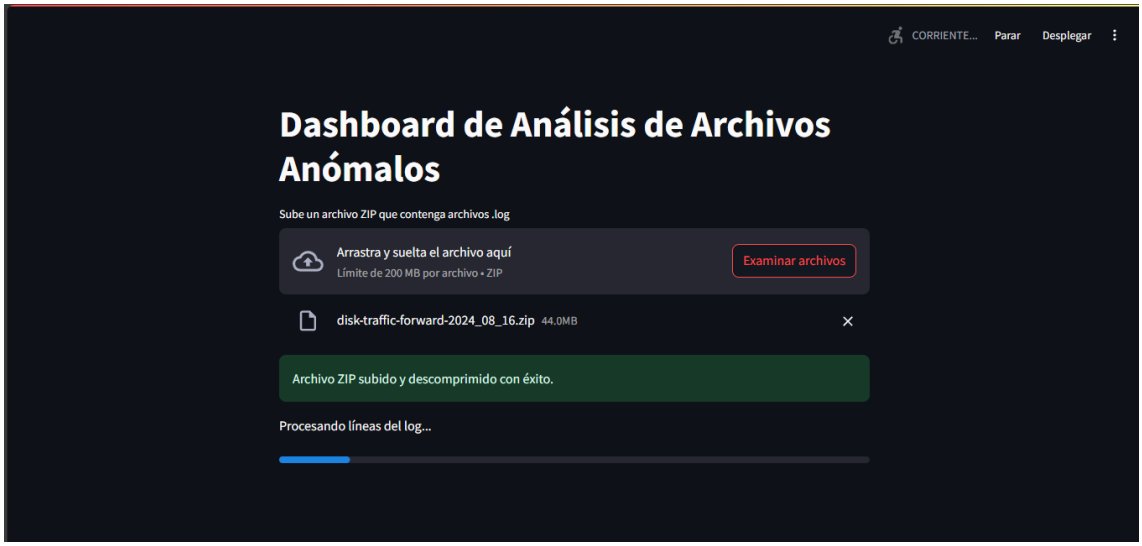
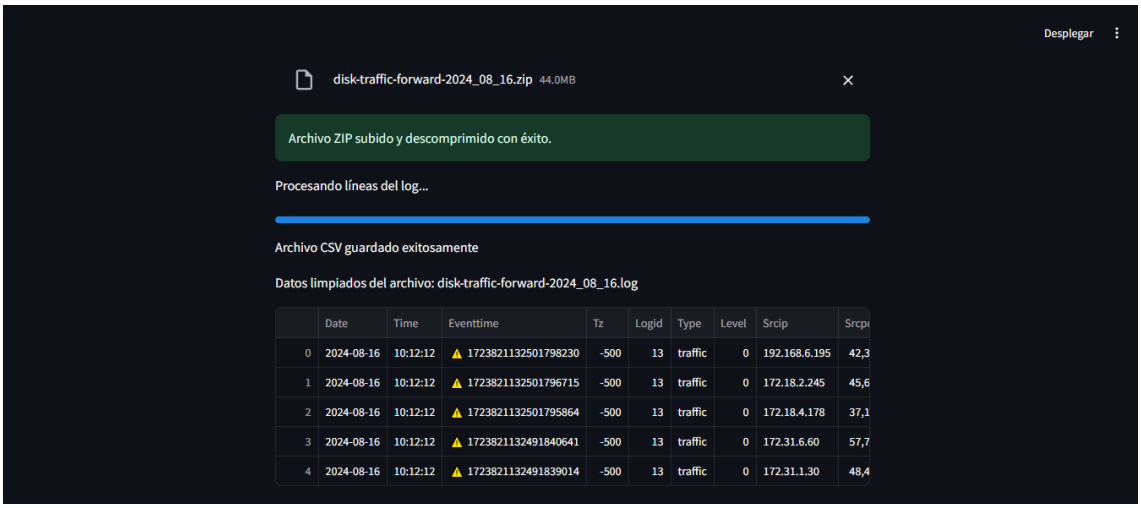


Ilustración 32 Mensaje de éxito de carga del archivo y barra de progreso de la limpieza del archivo



	Date	Time	Eventtime	Tz	Logid	Type	Level	Srcip	Srcp
0	2024-08-16	10:12:12	1723821132501798230	-500	13	traffic	0	192.168.6.195	42,3
1	2024-08-16	10:12:12	1723821132501796715	-500	13	traffic	0	172.18.2.245	45,6
2	2024-08-16	10:12:12	1723821132501795864	-500	13	traffic	0	172.18.4.178	37,1
3	2024-08-16	10:12:12	1723821132491840641	-500	13	traffic	0	172.31.6.60	57,7
4	2024-08-16	10:12:12	1723821132491839014	-500	13	traffic	0	172.31.1.30	48,4

Ilustración 33 Carga de información para validación de limpieza exitosa de los datos

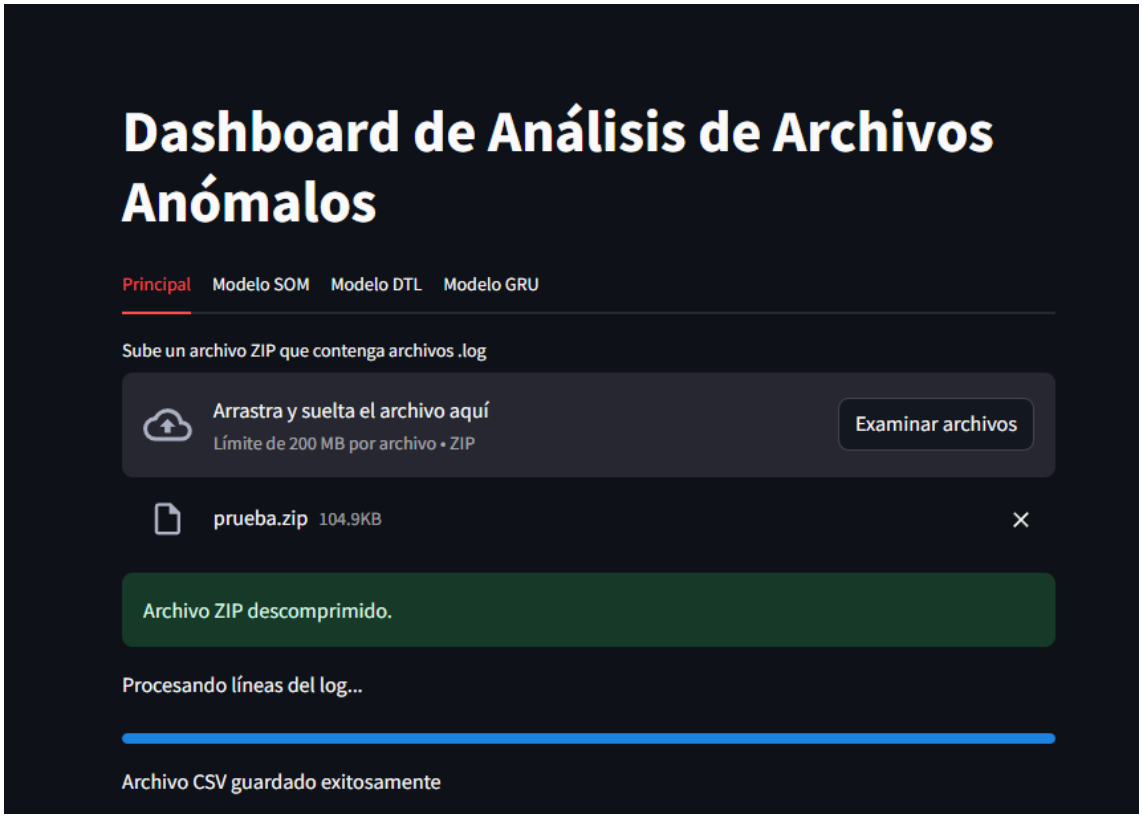


Ilustración 35 Elección del algoritmo4

Ilustración 34 Elección y evaluación de los modelos

2.5.1.4.3 Evaluación Final

- Análisis completo de resultados
- Validación de cumplimiento de objetivos
- Medición de métricas finales

2.5.2 Arquitectura del Sistema

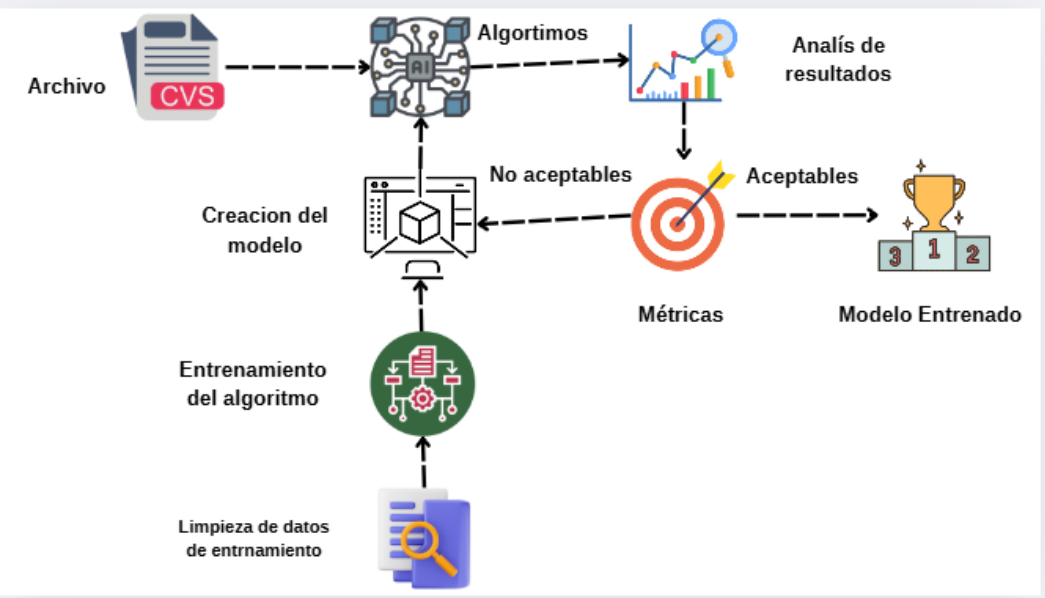


Ilustración 36 Arquitectura del proceso de creación de los modelos de Deep Learnig

El proceso representado en esta imagen muestra un flujo de trabajo completo para el desarrollo y aplicación de un modelo de aprendizaje automático. Se parte de un archivo CVS que contiene los datos de entrenamiento. Estos datos pasan por un proceso de limpieza y preparación antes de ser utilizados para el entrenamiento del modelo. Posteriormente, se crea el modelo utilizando algoritmos de aprendizaje automático, los cuales se entrenan con los datos preparados. Durante esta etapa, se monitorean métricas clave para evaluar el desempeño del modelo.

Una vez entrenado el modelo, este se somete a un análisis de resultados, donde se identifican los datos "aceptables" y "no aceptables" de acuerdo con el rendimiento del modelo. Esto permite refinar y ajustar el modelo para mejorar su precisión y capacidad de generalización. Finalmente, se obtiene el modelo entrenado, listo para ser

implementado y utilizado en aplicaciones o sistemas que requieran sus predicciones o clasificaciones.

Este flujo de trabajo integra diversas etapas fundamentales, como la preparación de datos, el entrenamiento del modelo, la evaluación de métricas y la retroalimentación para mejorar el rendimiento.

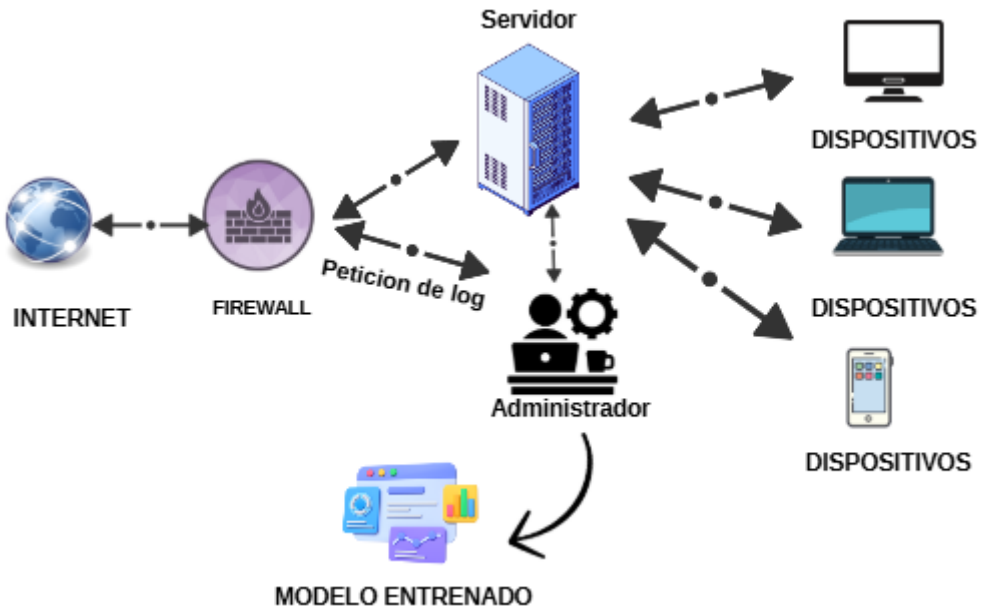


Ilustración 37 Arquitectura del Programa

La arquitectura de seguridad de red propuesta representa un avance significativo en la protección de infraestructuras tecnológica universitaria. El sistema se estructura como una sofisticada línea de defensa que va más allá de los métodos tradicionales de seguridad, integrando inteligencia computacional para analizar y comprender el tráfico de red de manera profunda y contextualizada. El firewall perimetral, como primera barrera, implementa reglas estáticas de seguridad que filtran inicialmente el tráfico, estableciendo un perímetro inicial de protección basado en configuraciones predefinidas por los administradores de red.

El administrador de red juega un papel crucial en este ecosistema de seguridad, actuando como un supervisor que recopila y gestiona los registros generados por el firewall. Su labor implica un análisis preliminar del tráfico, identificando posibles anomalías y patrones sospechosos mediante herramientas tradicionales de monitoreo. Sin embargo, la limitación de estos métodos radica en su capacidad para detectar amenazas cada vez más

sofisticadas y mutantes, lo que demanda soluciones más avanzadas e inteligentes que puedan adaptarse dinámicamente a los nuevos vectores de ataque.

Es precisamente en este contexto esta solución, posicionando como una tercera capa de seguridad revolucionaria. Utilizando modelos computacionales complejos como GRU, DTL y SOM, el sistema trasciende el análisis superficial de logs, profundizando en el análisis de características de red mediante técnicas de aprendizaje profundo. Estos algoritmos son capaces de descomponer y comprender patrones intrincados en el tráfico de red, identificando con alta precisión comportamientos potencialmente maliciosos que podrían pasar desapercibidos para sistemas tradicionales de detección de intrusiones.

2.6 Diagramas de casos de uso

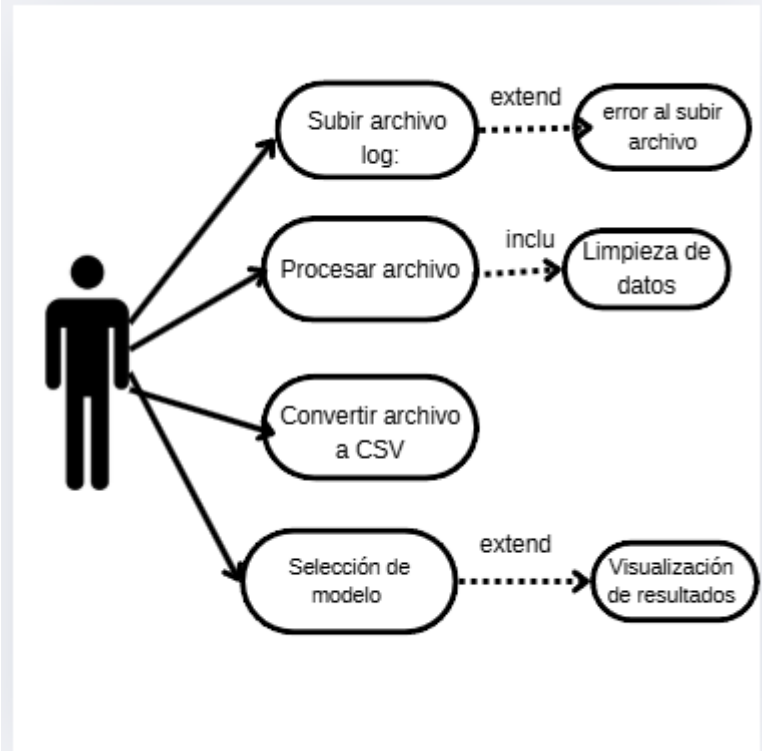


Ilustración 38 Diagrama de uso del procesamiento de archivos log

2.7 Resultados

A continuación, se presenta el análisis de los resultados obtenidos en las diferentes fases de la tesis. En primera instancia, se recopiló un archivo de registro (log) que contenía un total de 506,387 registros. Posteriormente, se realizó un proceso de limpieza y depuración de los datos, el cual permitió reducir el número de registros a 454,300. Este proceso de

limpieza eliminó aproximadamente un 10.3% de los registros iniciales, lo que evidencia la necesidad de contar con datos depurados y de calidad para su posterior análisis.

Una vez obtenido el conjunto de datos limpio y procesado, se procedió a enviar dicha información a los modelos de aprendizaje profundo SOM, DTL y GRU, con el objetivo de analizar en profundidad el comportamiento y las características de los registros. Estos modelos permitieron identificar patrones, tendencias y posibles anomalías dentro de los datos, brindando valiosa información para la toma de decisiones y la optimización de los procesos.

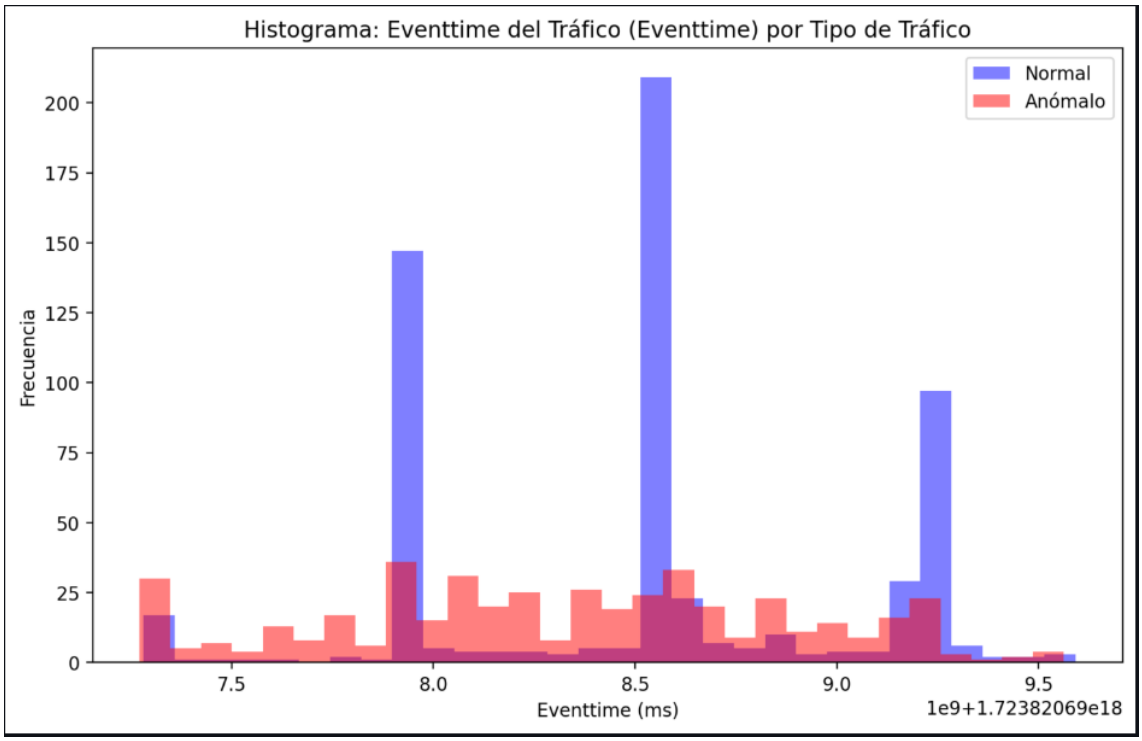


Ilustración 39 Gráfica de histograma de predicción modelo SOM

En esta imagen se presenta un histograma que muestra el Eventtime del Tráfico (Eventtime) por Tipo de Tráfico (Type of Traffic), comparando los escenarios de tráfico normal y anómalo. El eje horizontal representa el Eventtime en milisegundos (ms), mientras que el eje vertical indica la Frecuencia (Frecuencia) de los eventos.

En el escenario de tráfico normal, representado por las barras azules, se observa una distribución relativamente uniforme con un pico pronunciado alrededor de los 8,5 ms.

Esto sugiere que, durante las condiciones normales de tráfico, la mayoría de los eventos se agrupan en torno a este rango de tiempo.

En contraste, el escenario de tráfico anómalo, representado por las barras rojas, muestra un patrón más disperso y con mayores fluctuaciones. Se destacan dos picos principales: uno alrededor de los 9,0 ms y otro en 9,5 ms. Estas diferencias en la distribución del Eventtime entre los escenarios normal y anómalo indican que el modelo es capaz de detectar y diferenciar los patrones de tráfico típicos de aquellos que se desvían de lo esperado.

Además, se puede observar que el tráfico anómalo presenta una mayor frecuencia en los valores extremos, tanto en los rangos más bajos (alrededor de 7,5 ms) como en los más altos (cerca de 10 ms y 11 ms). Esto sugiere que el modelo puede identificar eventos inusuales o atípicos que se alejan significativamente de la distribución normal.

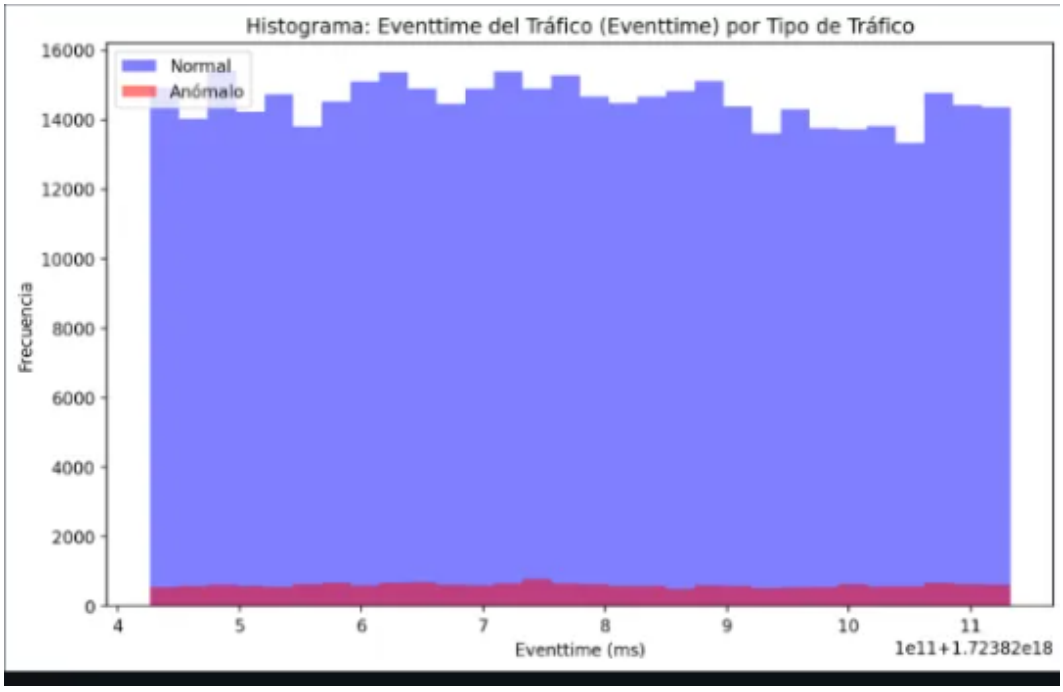


Ilustración 40 Gráfica de histograma de predicción modelo DTL

En este histograma del modelo DTL, el escenario de tráfico normal se representa en azul y el escenario de tráfico anómalo se representa mediante las barras rojas. Durante condiciones normales de tráfico, los valores de Duración muestran un patrón relativamente estable, con una distribución coherente en todo el rango de Eventtime. Sin

embargo, en el escenario anómalo, los valores de Duración muestran fluctuaciones más significativas, con picos y valles distintos en todo el espectro de Eventtime.

Los patrones contrastantes entre los escenarios de tráfico normal y anómalo sugieren que el modelo DTL es capaz de capturar diferencias notables en las características de duración del tráfico. Estas variaciones podrían ser indicativas de diversos eventos de tráfico, como congestión, accidentes u otros factores perturbadores que pueden afectar el flujo y la duración general del tráfico.

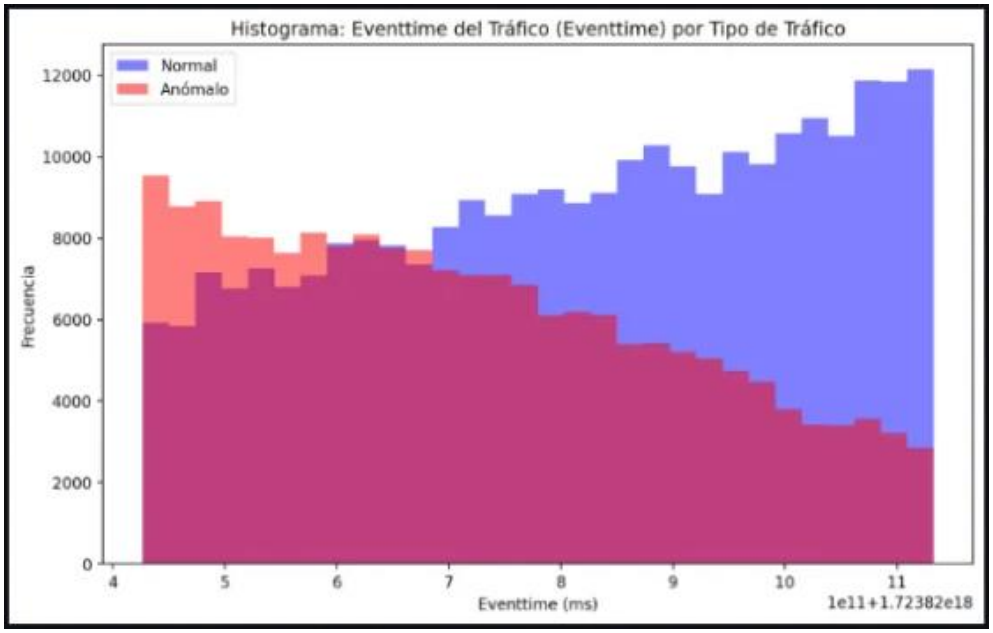


Ilustración 41 Gráfica de histograma de predicción modelo GRU

En este histograma del modelo GRU, podemos observar varios puntos clave. Las barras azules representan el escenario de tráfico normal y las barras rojas representan el escenario de tráfico anómalo. En condiciones normales de tráfico, los valores de Eventtime suelen ser más altos, alcanzando un máximo de entre 10.000 y 11.000 milisegundos. Sin embargo, durante condiciones anómalas, los valores de Eventtime son significativamente más bajos, con una caída brusca alrededor de la marca de 8.000 milisegundos. Esto sugiere que las condiciones de tráfico anómalas se caracterizan por duraciones de Eventtime significativamente más cortas en comparación con los patrones de tráfico normales.

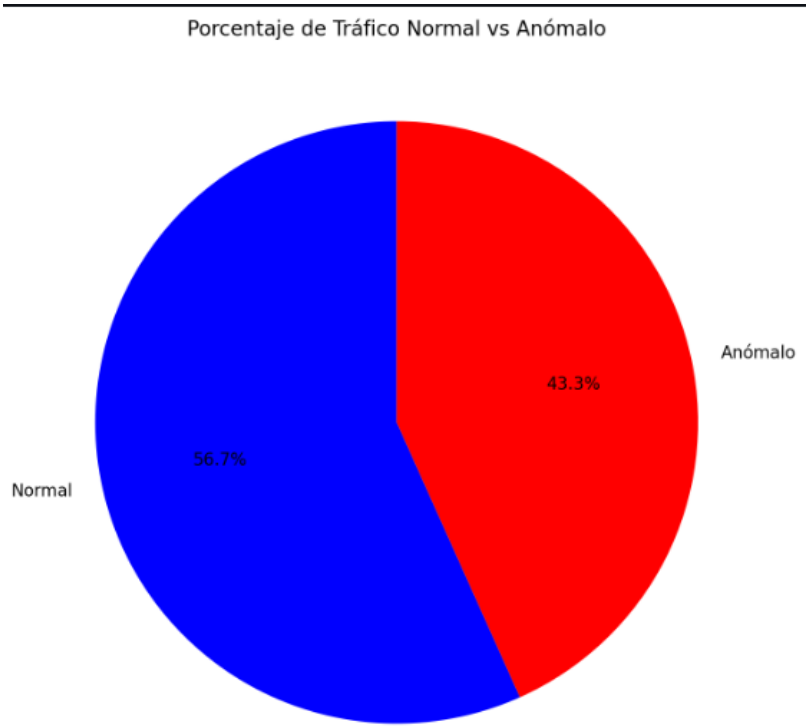


Ilustración 42 Gráfica de pastel de predicción modelo SOM

La imagen muestra un gráfico circular que representa la distribución porcentual del tráfico normal y anómalo.La sección azul ocupa el 56.7% del gráfico, representando el porcentaje de tráfico considerado normal. Por otro lado, la sección roja abarca el 43.3% del círculo, indicando el porcentaje de tráfico clasificado como anómalo según el modelo SOM (Mapa Auto-Organizado).

Este tipo de gráfico circular proporciona una representación visual clara y concisa de la composición del tráfico entre sus estados normal y anómalo. Permite a los analistas y gestores de tráfico comprender rápidamente la proporción relativa de cada categoría, lo cual puede ser valioso para la toma de decisiones y la implementación de estrategias de gestión del tráfico.

La predominancia del tráfico normal, aunque ligeramente superior al tráfico anómalo, sugiere que el sistema o red de tráfico se encuentra en un estado general saludable. Sin embargo, el porcentaje de tráfico anómalo, que representa casi la mitad del total, también es un dato relevante que debe ser analizado en profundidad para identificar las causas subyacentes y desarrollar medidas para mitigar o reducir ese tipo de eventos.

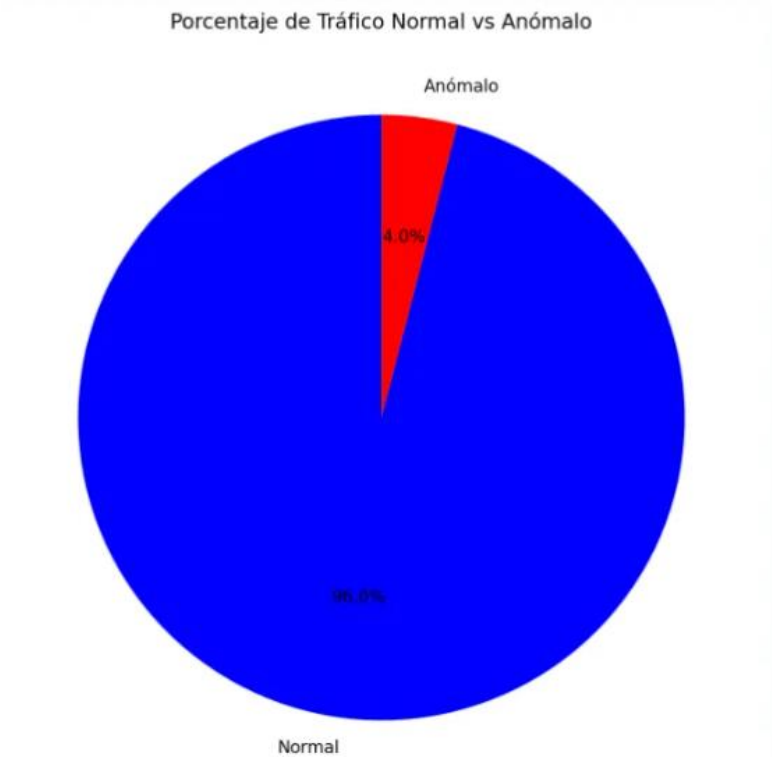


Ilustración 43 Gráfica de pastel de predicción modelo DTL

En este gráfico circular, el segmento azul representa el 96,0% del tráfico total, que el modelo DTL clasifica como normal. El 4,0% restante del tráfico se muestra en rojo, indicando la proporción de tráfico anómalo.

La clasificación de las condiciones de tráfico del modelo DTL revela un porcentaje significativamente mayor de tráfico normal (96,0%) en comparación con el tráfico anómalo (4,0%). Esto sugiere que el modelo DTL es altamente efectivo en identificar y distinguir los patrones de tráfico normales de los anómalos.

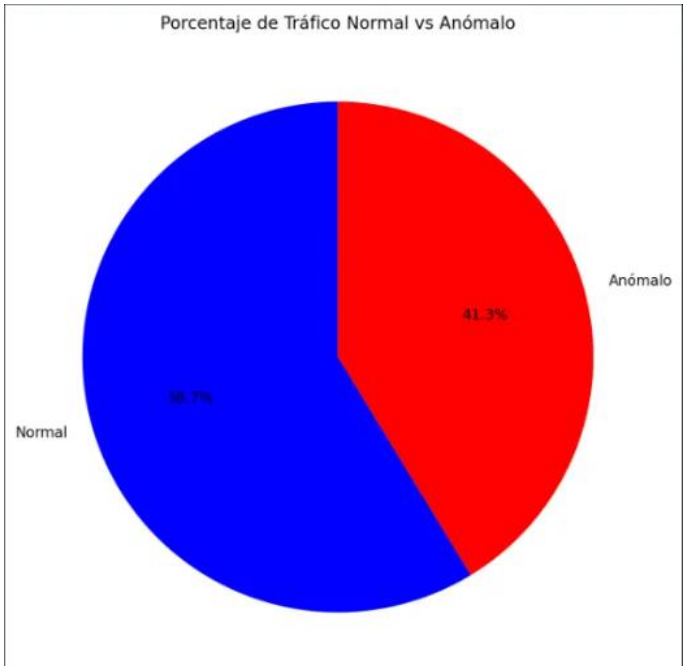


Ilustración 44 Gráfica de pastel de predicción modelo GRU

En este gráfico circular, el segmento azul representa el 58,7% del tráfico total, que el modelo GRU clasifica como normal. El 41,3% restante del tráfico se muestra en rojo, indicando la proporción de tráfico anómalo.

La capacidad del modelo GRU para diferenciar entre las condiciones de tráfico normales y anómalas se demuestra claramente en esta visualización. El porcentaje significativo de tráfico anómalo (41,3%) sugiere que el modelo GRU es hábil en identificar y caracterizar las desviaciones de los patrones de tráfico típicos.

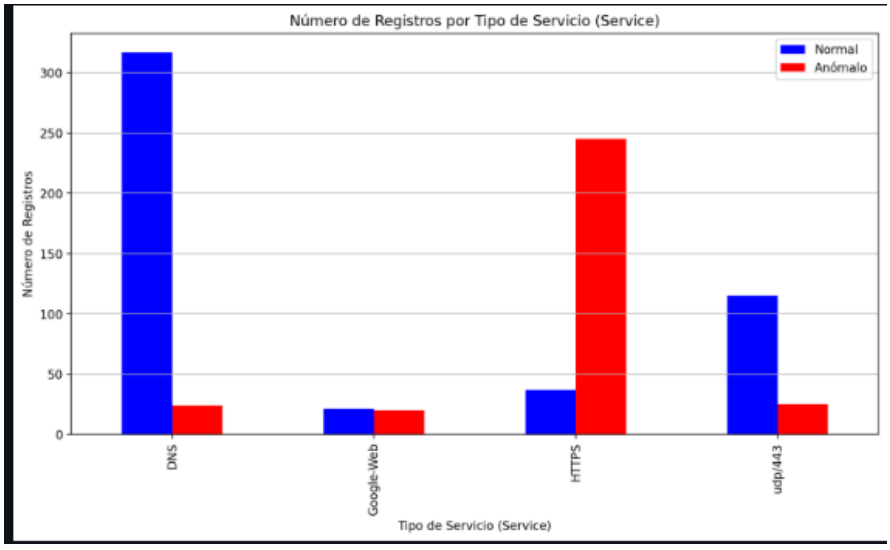


Ilustración 45 Gráfica de barras de predicción modelo SOM

Esta gráfica representa el Número de Registros por Tipo de Servicio (Service), comparando los escenarios de tráfico normal y anómalo utilizando el modelo SOM (Self-Organizing Map).

En el eje horizontal se muestran los diferentes Tipos de Servicio, mientras que en el eje vertical se indica el Número de Registros.

En el escenario de tráfico normal, representado por las barras azules, se observa una distribución bastante uniforme con algunos picos más pronunciados. Por ejemplo, el servicio "DVS" tiene el mayor número de registros en condiciones normales, seguido por "Servicio Local" y " Highway ".

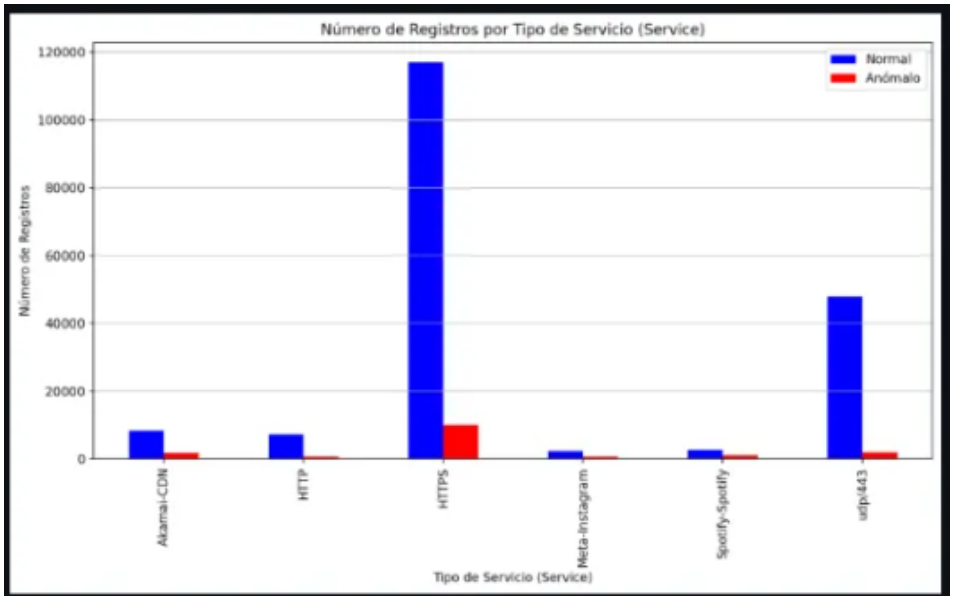


Ilustración 46 Gráfica de barras de predicción modelo DTL

Por otro lado, en el escenario de tráfico anómalo, representado por las barras rojas, se aprecia un patrón muy diferente. Para la mayoría de los tipos de servicio, el número de registros anómalos es significativamente menor que el número de registros normales. Sin embargo, hay algunas excepciones, como en el caso de "HTTPS" y "Seguridad", donde los registros anómalos superan a los normales.

Esta imagen representa un gráfico de barras que muestra el Número de Registros por Tipo de Servicio para escenarios normales y anómalos. El eje x muestra el Tipo de Servicio, mientras que el eje y representa el Número de Registros.

En este histograma del modelo DTL (Aprendizaje de Árbol de Decisión), podemos observar varias ideas clave. Las barras azules representan el escenario de tráfico normal, mientras que las barras rojas representan el escenario de tráfico anómalo. Durante las condiciones de tráfico normales, el número de registros es significativamente mayor para ciertos tipos de servicio, como " Highway " y "Servicio Local". Sin embargo, en el escenario anómalo, el número de registros para estos tipos de servicio es considerablemente menor, lo que indica una posible interrupción o cambio en los patrones de tráfico normales.

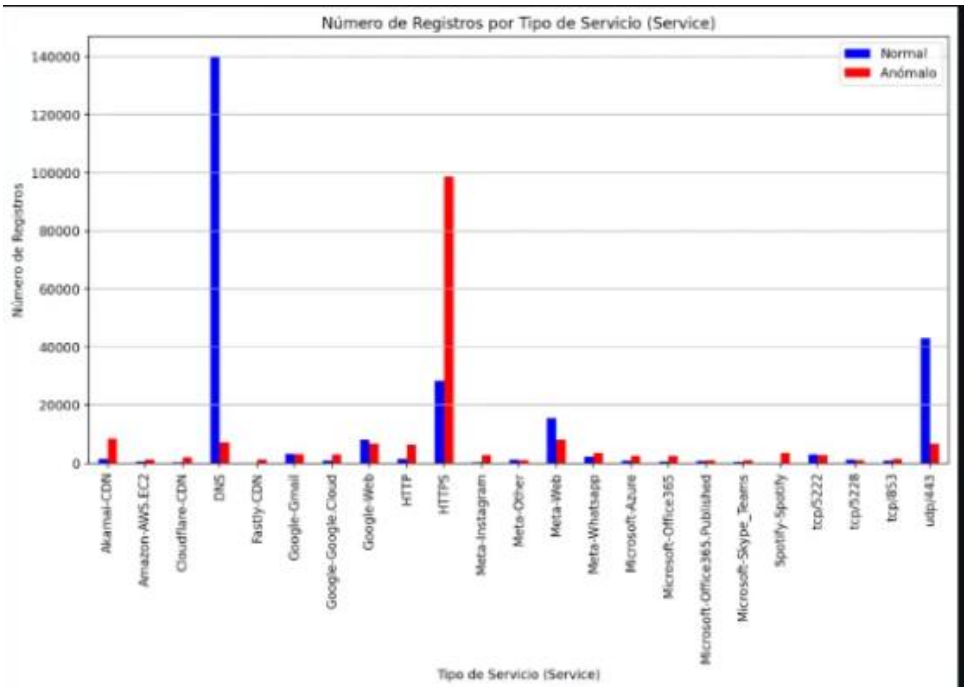


Ilustración 47 Gráfica de barras de predicción modelo GRU

En este histograma del modelo GRU, podemos observar una diferencia más pronunciada entre las condiciones de tráfico normal y anómalo. El escenario de tráfico normal exhibe números de registro significativamente más altos para ciertos tipos de servicio, como " Highway " y "Servicio Local", mientras que el escenario anómalo muestra una caída dramática en el número de registros para estos tipos de servicio.

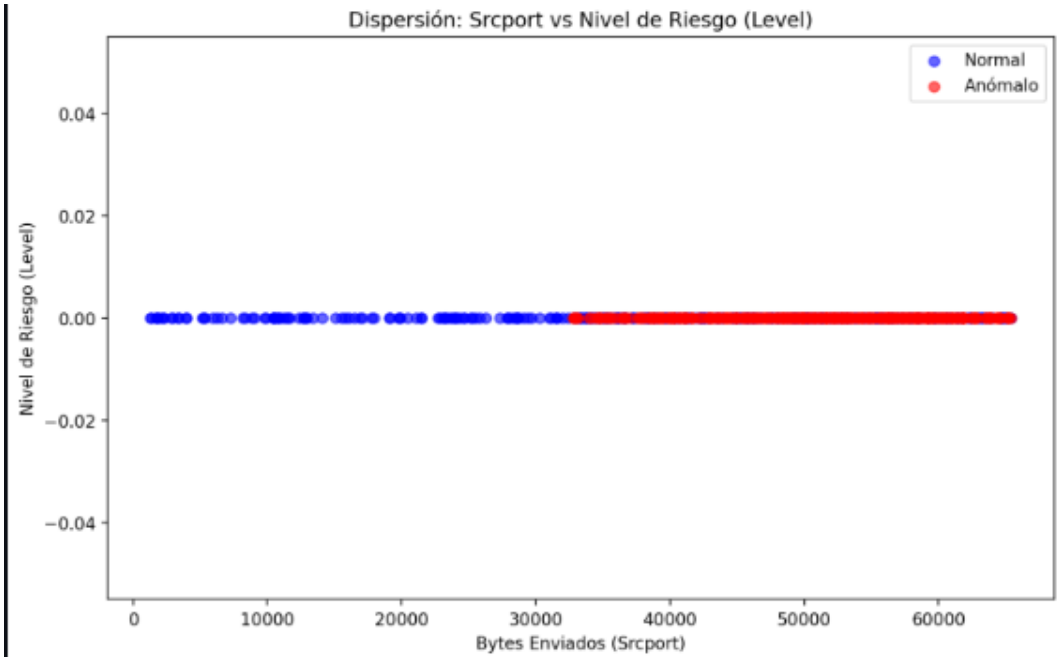


Ilustración 48 Gráfica de dispersión de predicción modelo SOM

Esta imagen muestra un gráfico que representa la Dispersión: Puerto de Origen vs. Nivel de Riesgo tanto para escenarios de tráfico normal como anómalo. El eje x muestra los Bytes Enviados (Puerto de Origen), mientras que el eje y representa el Nivel de Riesgo.

En el escenario de tráfico normal, representado por la línea azul, el Nivel de Riesgo se mantiene relativamente constante a través del rango de Bytes Enviados. Esto sugiere una relación estable y predecible entre el número de bytes enviados y el nivel de riesgo asociado en condiciones normales de operación.

Sin embargo, el escenario de tráfico anómalo, representado por la línea roja, exhibe un patrón marcadamente diferente. A medida que aumentan los Bytes Enviados (Puerto de Origen), el Nivel de Riesgo también aumenta a un ritmo mucho más rápido en comparación con el escenario normal. Esto indica que el modelo es capaz de identificar y caracterizar desviaciones significativas del comportamiento de tráfico esperado, lo que podría indicar posibles amenazas de seguridad o anomalías de red.

Las tendencias contrastantes entre las condiciones de tráfico normal y anómalo permiten a los profesionales de gestión de tráfico y seguridad comprender mejor la relación dinámica entre el número de bytes enviados y los niveles de riesgo asociados. Esta

información puede aprovecharse para desarrollar mecanismos de monitoreo y detección más robustos, permitiéndoles identificar y responder rápidamente a patrones de tráfico anómalo que puedan requerir una mayor investigación o mitigación.

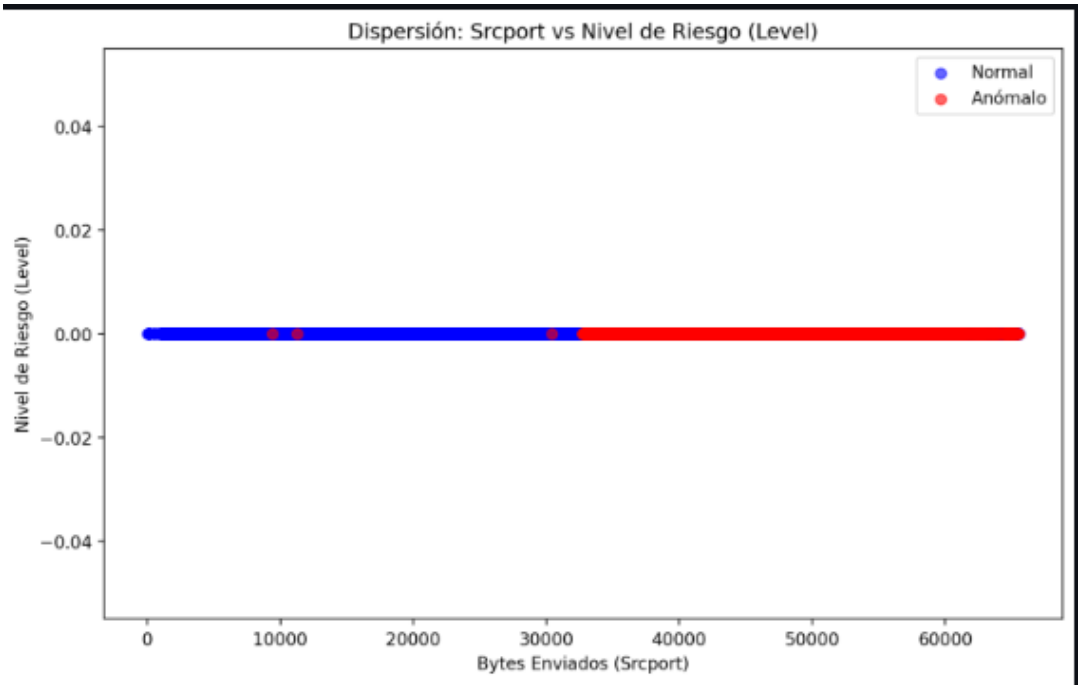


Ilustración 49 Gráfica de dispersión de predicción modelo DTL

La imagen muestra un gráfico que muestra la Dispersión: Bytes Enviados vs Nivel de Riesgo para escenarios normales y anómalos. El eje x representa los Bytes Enviados (Srcport), mientras que el eje y muestra el Nivel de Riesgo (Level).

En este gráfico del modelo DTL (Aprendizaje de Árbol de Decisión), la línea azul representa el escenario de tráfico normal, mientras que la línea roja representa el escenario de tráfico anómalo. Durante las condiciones de tráfico normales, el Nivel de Riesgo permanece relativamente constante en todo el rango de Bytes Enviados, indicando una relación estable y predecible entre los bytes enviados y el nivel de riesgo.

Sin embargo, en el escenario anómalo, la línea roja exhibe un patrón más volátil e impredecible. A medida que aumentan los Bytes Enviados (Srcport), el Nivel de Riesgo también aumenta a un ritmo mucho más alto en comparación con el escenario normal. Esto sugiere que el modelo DTL puede identificar y caracterizar desviaciones

significativas de los patrones de tráfico esperados, lo que podría ser indicativo de posibles amenazas de seguridad o anomalías de red.

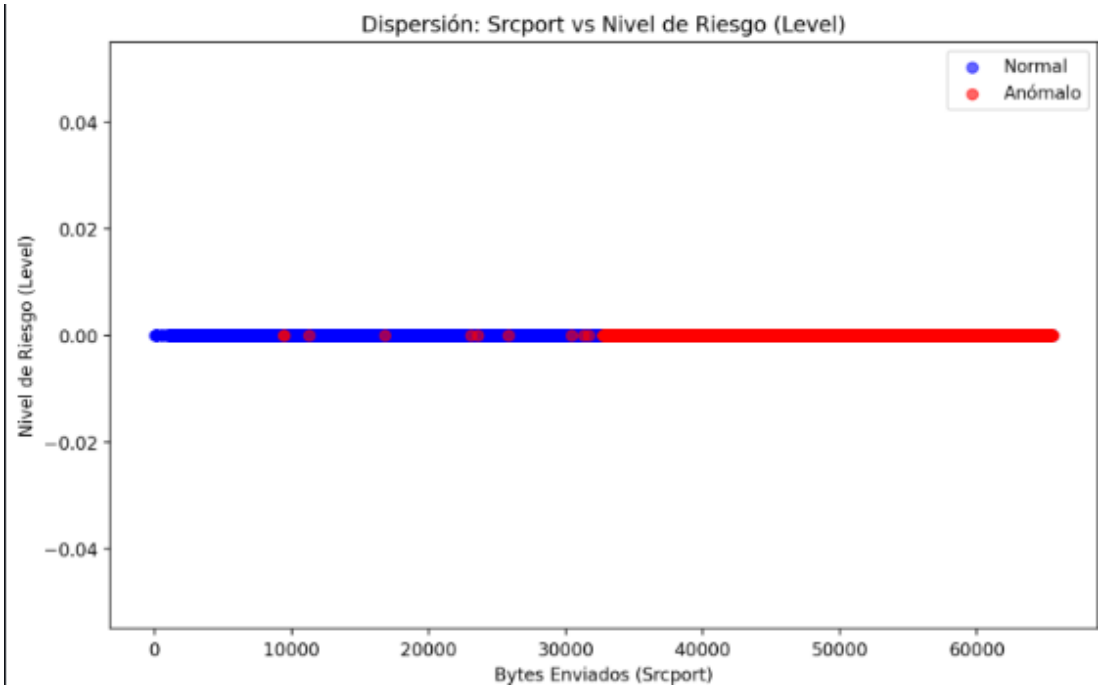


Ilustración 50Gráfica de dispersión de predicción modelo GRU

En este gráfico del modelo GRU, el escenario de tráfico normal exhibe una relación relativamente plana y estable entre los Bytes Enviados (Srcport) y el Nivel de Riesgo (Level). Esto sugiere que el modelo GRU es efectivo para capturar el comportamiento típico o esperado, donde el nivel de riesgo permanece constante independientemente del número de bytes enviados.

Por otro lado, el escenario de tráfico anómalo, representado por la línea roja, muestra un aumento mucho más pronunciado y abrupto en el Nivel de Riesgo a medida que aumentan los Bytes Enviados (Srcport). Este marcado contraste entre los patrones de tráfico normal y anómalo indica que el modelo GRU es adecuado para identificar y marcar anomalías potenciales que se desvían significativamente de la dinámica de tráfico esperada.

CONCLUCIONES

La implementación de los modelos SOM, GRU y DTL permitió analizar 454,300 registros, tras depurar un conjunto inicial de 506,306 con una limpieza del 10.3% que eliminó ruido y redundancias. Entrenados en 100 épocas, alcanzaron un F1-Score de hasta 98% para clasificar tráfico normal y anómalo. SOM fue el más eficiente, con 93% de precisión en solo 4 minutos, mientras GRU y DTL lograron 98% en 6 minutos, destacando en entornos dinámicos. Usando datos del log de un firewall Fortinet, GRU detectó el 41.3% de tráfico anómalo, frente al 4.0% de DTL y 41.3% de SOM, demostrando su capacidad para manejar patrones temporales complejos, mientras que DTL y SOM ofrecieron enfoques complementarios según el tipo de tráfico.

Los resultados muestran que, tras una optimización de hiperparámetros y reducción de épocas, se logró una disminución promedio del 90% en los tiempos de ejecución sin comprometer la precisión de los modelos. Este avance permite su integración en entornos de producción con tiempos de respuesta optimizados para análisis en tiempo real. El modelo SOM se presenta como una solución altamente eficiente y escalable para sistemas con limitaciones de recursos computacionales, mientras que GRU y DTL son más adecuados para entornos donde la precisión y la capacidad de aprendizaje transferido son prioritarias.

En términos operativos, la arquitectura propuesta logró combinar efectivamente el análisis tradicional con inteligencia artificial avanzada, estableciendo un sistema que integra un firewall perimetral como primera barrera de defensa con modelos de aprendizaje profundo para la identificación y caracterización de patrones maliciosos en tráfico HTTPS. Este enfoque permitió no solo la detección de amenazas conocidas, sino también la identificación proactiva de nuevos vectores de ataque, marcando un avance significativo en la protección de redes universitarias frente a ciber amenazas emergentes.

Estos hallazgos demuestran que la solución no solo es técnicamente sólida, sino también adaptable y eficiente, posicionándola como una herramienta clave para la ciberseguridad moderna en entornos académicos y potencialmente en otras aplicaciones corporativas. Con métricas cuantitativas sobresalientes, el proyecto valida la efectividad del uso de algoritmos de aprendizaje profundo en la detección avanzada de anomalías en tráfico de red cifrado.

RECOMENDACIONES

Una de las principales recomendaciones para mejorar este proyecto es realizar pruebas con diversos tipos de archivos de logs generados por diferentes firewalls y dispositivos de red, incluyendo formatos cómo `.apk` y otros que puedan capturar información relevante para análisis avanzados. Esto permitiría evaluar la capacidad de los modelos para adaptarse a diferentes fuentes y denominaciones de datos, incrementando la robustez y generalización del sistema. Además, estas pruebas ampliarían el alcance del análisis, reflejando escenarios más representativos de entornos reales de ciberseguridad.

Asimismo, se recomienda implementar un proceso manual y detallado de revisión de los archivos CSV resultantes del preprocesamiento, verificando la calidad, integridad y consistencia de los datos. Aunque la automatización es eficiente, este paso manual es crucial para identificar posibles anomalías o errores en los datos de entrada, que pueden afectar el desempeño del modelo. Este enfoque garantizará un entrenamiento más confiable, mejorando la precisión de la detección de anomalías en el tráfico de red.

Por último, es aconsejable emplear dispositivos especializados en el procesamiento de datos, como equipos con GPUs de arquitectura avanzada o TPUs, optimizados para tareas de aprendizaje profundo. Estas plataformas no solo aceleran los tiempos de entrenamiento de los modelos, sino que también permiten manejar conjuntos de datos más grandes y complejos, maximizando la eficiencia y el rendimiento del sistema en contextos de alta demanda computacional.

Referencias

[1] "Cifrado HTTPS en la Web – Informe de transparencia de Google." Accessed: Oct. 13, 2024. [Online]. Available: https://transparencyreport.google.com/https/overview?hl=es

[2] G. I. Cruz Lucas, R. E. Galarza Espinoza, R. S. Delgado De La Cruz, and M. J. Marcillo Merino, "Aplicación de protocolos SSL y TSL para él envió de información," *Journal TechInnovation*, vol. 1, no. 2, pp. 4–9, Jul. 2022, doi: 10.47230/journal.techinnovation.v1.n2.2022.4-9.

[3] "La ciberamenaza que podría estar pasando por alto: ataques de tráfico cifrado | Avast Business." Accessed: Oct. 13, 2024. [Online]. Available: https://www.avast.com/es-ww/business/resources/infographic/encrypted-traffic-attacks#pc

[4] "Principales ciberamenazas y cómo enfrentarlas - The Bridge | Digital Talent Accelerator." Accessed: Oct. 13, 2024. [Online]. Available: https://thebridge.tech/blog/principales-ciberamenazas

[5] "(25) El Panorama Completo de la Ciberseguridad en América Latina: Retos, Avances y Perspectivas | LinkedIn." Accessed: Oct. 13, 2024. [Online]. Available: https://www.linkedin.com/pulse/el-panorama-completo-de-la-ciberseguridad-en-am%C3%A9rica-alvaro-ji7jf/

[6] "UPSE TRABAJA EN UN NUEVO ESTATUTO PARA POTENCIAR SU FUTURO." Accessed: Oct. 13, 2024. [Online]. Available: https://upse.edu.ec/index.php?option=com_content&view=article&id=976:upse-trabaja-en-un-nuevo-estatuto-para-potenciar-su-futuro-2&catid=10&Itemid=178

[7] D. I. Q. Yagual, C. C. Yagual, and I. C. Suárez, "Una revisión del Aprendizaje profundo aplicado a la ciberseguridad," *Revista Científica y Tecnológica UPSE*, vol. 9, no. 1, pp. 57–65, Jun. 2022, doi: 10.26423/RCTU.V9I1.671.

[8] "Nobel de Física: John Hopfield y Geoffrey Hinton ganan el premio por hacer que 'las máquinas aprendan' y sentar las bases de la inteligencia artificial - BBC

News Mundo.” Accessed: Nov. 25, 2024. [Online]. Available: https://www.bbc.com/mundo/articles/c07njpdypn5o

[9] Ó. Francés Luesma, “Malicious URL detection mediante técnicas de Deep Learning,” 2022, Accessed: Oct. 13, 2024. [Online]. Available: https://openaccess.uoc.edu/handle/10609/138066

[10] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey,” *Comput Commun*, vol. 170, pp. 19–41, Mar. 2021, doi: 10.1016/J.COMCOM.2021.01.021.

[11] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, “An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection,” *IEEE Access*, vol. 8, pp. 30387–30399, 2020, doi: 10.1109/ACCESS.2020.2973023.

[12] C. E. Maldonado and U. El Bosque, “Sistema de Información Científica,” 2001. [Online]. Available: http://www.redalyc.org/articulo.oa?id=41400504

[13] L. M. Pehrson, M. B. Nielsen, and C. A. Lauridsen, “Automatic Pulmonary Nodule Detection Applying Deep Learning or Machine Learning Algorithms to the LIDC-IDRI Database: A Systematic Review,” *Diagnostics 2019, Vol. 9, Page 29*, vol. 9, no. 1, p. 29, Mar. 2019, doi: 10.3390/DIAGNOSTICS9010029.

[14] “Estadísticas de uso del protocolo predeterminado https para sitios web, octubre de 2024.” Accessed: Oct. 13, 2024. [Online]. Available: https://w3techs.com/technologies/details/ce-httpsdefault

[15] “50+ Estadísticas, datos y cifras de ciberseguridad para 2024.” Accessed: Oct. 21, 2024. [Online]. Available: https://www.techopedia.com/cybersecurity-statistics

[16] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, Apr. 2016, doi: 10.1109/COMST.2015.2494502.

[17] “Perspectivas globales de ciberseguridad 2024 | Foro Económico Mundial.” Accessed: Oct. 13, 2024. [Online]. Available:

https://www.weforum.org/publications/global-cybersecurity-outlook-2024/?utm_source=google&utm_medium=ppc&utm_campaign=cybersecurity&gad_source=1&gclid=Cj0KCQjwgrO4BhC2ARIsAKQ7zUkY6z-wB2R9atLWPjl9AWw30Ldzc7gujqA6v4D4TgaJ0rF6kJ03JU8aAj4sEALw_wcB

[18] "ISO/IEC 27001:2022 - Sistemas de gestión de la seguridad de la información." Accessed: Oct. 13, 2024. [Online]. Available: https://www.iso.org/standard/27001

[19] J. D. Kelleher, "The Deep Learning," *MIT Press*, p. 269, 2019, Accessed: Oct. 13, 2024. [Online]. Available: https://books.google.com/books/about/Deep_Learning.html?hl=es&id=b06qDwAAQBAJ

[20] "investigacion diagnostica | PDF." Accessed: Oct. 13, 2024. [Online]. Available: https://es.slideshare.net/slideshow/investigacion-diagnostica-236627991/236627991

[21] E. Militar de Cadetes, G. José María Córdova, and C. Zafra Galvis, "Tipos de Investigación," *Revista Científica General José María Córdova*, vol. 4, no. 4, pp. 13–14, 2006, Accessed: Oct. 13, 2024. [Online]. Available: https://www.redalyc.org/articulo.oa?id=476259067004

[22] "HISTORIA." Accessed: Oct. 13, 2024. [Online]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=10&Itemid=166

[23] "MISIÓN - VISIÓN." Accessed: Oct. 13, 2024. [Online]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=12&Itemid=167

[24] "Constitución de la República del Ecuador | Descargar PDF Constitución de la República del Ecuador | Actualizado 2024." Accessed: Oct. 13, 2024. [Online]. Available: https://www.lexis.com.ec/biblioteca/constitucion-republica-ecuador

[25] R. Oficial Suplemento, "CÓDIGO ORGÁNICO INTEGRAL PENAL, COIP", Accessed: Oct. 13, 2024. [Online]. Available: www.lexis.com.ec

[26] “Código Orgánico Integral Penal (COIP), 3 de febrero del 2014 - Informática Jurídica.” Accessed: Dec. 01, 2024. [Online]. Available: https://www.informatica-juridica.com/codigo/codigo-organico-integral-penal-coip-10-agosto-del-2014/

[27] “Decreto-Ejecutivo-No.-904”.

[28] “Sistema de Información: definición, características y ejemplos.” Accessed: Oct. 13, 2024. [Online]. Available: https://rockcontent.com/es/blog/que-es-un-sistema-de-informacion/

[29] “¿Qué son los sistemas informáticos? | Blog UI1.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.ui1.es/blog-ui1/sistemas-informaticos-si-que-son-caracteristicas-y-tipos

[30] A. A. Varón Quimbayo, “Seguridad en aplicaciones WEB,” 2021, Accessed: Oct. 13, 2024. [Online]. Available: localhost/xmlui/handle/11506/2401

[31] “Análisis de tráfico de red | ¿Cómo analizar el tráfico de red? - ManageEngine NetFlow Analyzer.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.manageengine.com/latam/netflow/analisis-de-trafico-de-red.html

[32] “¿Qué es el pharming y cómo puedes protegerte?” Accessed: Oct. 13, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/pharming?srsltid=AfmBOopXFwdgVGVVJcaxV4DwsikTyvLXhnwqQaNfrEGU-e6t7O2Zffzv

[33] “Phishing: reconocer y evitar las estafas de phishing.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.malwarebytes.com/es/phishing

[34] “¿Qué es un troyano? Virus troyanos y software malware | Fortinet.” Accessed: Oct. 21, 2024. [Online]. Available: https://www.fortinet.com/lat/resources/cyberglossary/trojan-horse-virus

[35] “Spyware: ¿Qué es y cómo protegerse?” Accessed: Oct. 21, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/threats/spyware?srsltid=AfmBOopQjptKVdubCxqFj9SVht6uQo_fN_LsynPoccRpKXy_SJ1wJlFi

[36] "¿Qué es el adware?" Accessed: Oct. 21, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/threats/adware?srsltid=AfmBOopX8cJcaStQb4ExDMKtKbxMzKPZHLdKzZERUL6oYVKGjpnrKiLV

[37] "¿Qué es el malware? | McAfee." Accessed: Nov. 24, 2024. [Online]. Available: https://www.mcafee.com/es-mx/antivirus/malware.html

[38] "¿Qué es el ransomware? | Protección contra el ransomware | Kaspersky." Accessed: Oct. 21, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/threats/ransomware?srsltid=AfmBOoqKvIUb_ZIKoWgkVJzXfZRjRokkx651X019XrvusIUGXW_q7uN4

[39] "Qué es un rootkit | Blog oficial de Kaspersky." Accessed: Oct. 21, 2024. [Online]. Available: https://www.kaspersky.es/blog/que-es-un-rootkit/594/

[40] "¿Qué es el cryptojacking? ¿Cómo funciona?" Accessed: Oct. 21, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/what-is-cryptojacking?srsltid=AfmBOorwtSx3dve4ygU7iEOBKrykcLdtPdXqbCTNIDAeP29r_7OuKq7R

[41] "¿Qué es la red distribuida y qué ventajas tiene? - Ikusi." Accessed: Oct. 13, 2024. [Online]. Available: https://www.ikusi.com/mx/blog/que-es-la-red-distribuida-y-que-ventajas-tiene-2/

[42] "Herramienta de análisis del tráfico de red - Analice su red | SolarWinds".

[43] "¿Qué es una dirección IP y qué significa?" Accessed: Oct. 13, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/what-is-an-ip-address

[44] "Android .apk: Se ha producido un error al analizar el paquete - Stack Overflow en español." Accessed: Oct. 13, 2024. [Online]. Available: https://es.stackoverflow.com/questions/360085/android-apk-se-ha-producido-un-error-al-analizar-el-paquete

[45] “¿Qué es un FIA (Análisis por Inyección en Flujo) | Glosario.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.unilabs.es/glosario/fia

[46] “Analiza la seguridad de las cabeceras HTTP de tu sitio web con esta herramienta.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.redeszone.net/2016/09/11/analiza-la-seguridad-las-cabeceras-http-sitio-web-esta-herramienta/

[47] “¿Qué es la seguridad por capas? - ADM Cloud Services,” *https://admcloudservices.com/*, Accessed: Oct. 13, 2024. [Online]. Available: https://admcloudservices.com/seguridad-por-capas/

[48] “Qué es un sitio web, para qué sirve y cuáles son sus elementos.” Accessed: Oct. 13, 2024. [Online]. Available: https://blog.hubspot.es/website/que-es-sitio-web

[49] “¿Qué es HTTPS? | Cloudflare.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.cloudflare.com/es-es/learning/ssl/what-is-https/

[50] “¿Qué son SSL, TLS y HTTPS? | DigiCert.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.digicert.com/es/what-is-ssl-tls-and-https

[51] “Detección de Anomalías en el Tráfico de Red: - ¿Qué es Anomaly Detection System?” Accessed: Oct. 13, 2024. [Online]. Available: https://www.internationalit.com/post/detecci%C3%B3n-de-anomal%C3%ADas-en-el-tr%C3%A1fico-de-red-qu%C3%A9-es-anomaly-detection-system?lang=es

[52] “Controlando el caos: automatizando reacciones ante anomalías en el tráfico web - Transparent Edge.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.transparentedge.eu/blog/automatizando-reacciones/

[53] “Nodo Detección de anomalías - Documentación de IBM.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-anomaly-detection-node

[54] “¿Qué es la detección basada en firmas? | phoenixNAP Glosario de TI.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.phoenixnap.mx/glosario/detecci%C3%B3n-basada-en-firmas

[55] "¿Qué es un análisis heurístico?" Accessed: Oct. 13, 2024. [Online]. Available: https://latam.kaspersky.com/resource-center/definitions/heuristic-analysis

[56] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, Accessed: Oct. 13, 2024. [Online]. Available: http://arxiv.org/abs/1412.3555

[57] "¿Qué es un sistema de detección de intrusiones? - Palo Alto Networks." Accessed: Oct. 13, 2024. [Online]. Available: https://www.paloaltonetworks.lat/cyberpedia/what-is-an-intrusion-detection-system-ids

[58] C. Alberto Ruiz Marta Susana Basualdo Autor and D. Jorge Matich, "Cátedra: Informática Aplicada a la Ingeniería de Procesos-Orientación I Redes Neuronales: Conceptos Básicos y Aplicaciones".

[59] "Algoritmos de detección de anomalías." Accessed: Oct. 13, 2024. [Online]. Available: https://docs.oracle.com/es-ww/iaas/Content/anomaly/using/kernels.htm

[60] "¿Qué es deep learning? | SAS." Accessed: Oct. 13, 2024. [Online]. Available: https://www.sas.com/es_es/insights/analytics/deep-learning.html

[61] A. Mosavi, S. Ardabili, and A. R. Várkonyi-Kóczy, "List of Deep Learning Models," *Lecture Notes in Networks and Systems*, vol. 101, pp. 202–214, 2020, doi: 10.1007/978-3-030-36841-8_20.

[62] "¿Qué es Python? - Explicación del lenguaje Python - AWS." Accessed: Oct. 13, 2024. [Online]. Available: https://aws.amazon.com/es/what-is/python/

[63] "Jupyter Notebook: la herramienta de Python para procesar datos - IONOS." Accessed: Oct. 21, 2024. [Online]. Available: https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/

[64] C. Corporación Centro de Investigación en Palma de Aceite, L. E. Vargas, and E. Mesa-Fúquen, "Introducción al análisis de datos con RStudio," *Cenipalma*, pp. 1–65, Aug. 2021, Accessed: Oct. 13, 2024. [Online]. Available: https://repositorio.fedepalma.org/handle/123456789/141281

[65] “Qué es Dashboard - Definición, significado y ejemplos.” Accessed: Oct. 13, 2024. [Online]. Available: https://www.arimetrics.com/glosario-digital/dashboard

[66] “Conceptos básicos sobre bases de datos - Soporte técnico de Microsoft.” Accessed: Oct. 13, 2024. [Online]. Available: https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204

[67] “¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS.” Accessed: Oct. 13, 2024. [Online]. Available: https://aws.amazon.com/es/what-is/api/

[68] A. R. Alzighaibi, “Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification,” *Computers 2023, Vol. 12, Page 47*, vol. 12, no. 3, p. 47, Feb. 2023, doi: 10.3390/COMPUTERS12030047.

[69] Brody James KuttPeng PengFang LiuII William Redington Hewlett, “ Deep learning for malicious URL classification (URLC) with the innocent until proven guilty (IUPG) learning framework,” Google Patents.

[70] M. Aljabri *et al.*, “An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models,” *Comput Intell Neurosci*, vol. 2022, no. 1, p. 3241216, Jan. 2022, doi: 10.1155/2022/3241216.

[71] D. Q. Yagual, B. S. Méndez, and V. C. Ruiz, “Efficient clustering of e-mails by applying supervised machine learning algorithms.,” *Journal of Applied Research and Technology*, vol. 22, no. 4, pp. 560–566, Aug. 2024, doi: 10.22201/ICAT.24486736E.2024.22.4.2383.

[72] “Malicious and Benign Websites.” Accessed: Nov. 11, 2024. [Online]. Available: https://www.kaggle.com/datasets/xwolf12/malicious-and-benign-websites/data

ANEXOS

Anexo 1. Recopilación de información de los laboratorios

<table>
<tr><td colspan="4">UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE TECNOLOGIAS DE LA INFORMACION</td></tr>
<tr><td>Realizado por:</td><td>Duma Silva Valeria</td><td rowspan="2">Nombre del reporte:</td><td rowspan="2">Comprensión organizacional</td></tr>
<tr><td>Fecha</td><td>10 de octubre del 2024</td></tr>
<tr><td>Hora</td><td colspan="3">12 y 30</td></tr>
<tr><td colspan="4">Metodología: Observación</td></tr>
<tr><td colspan="4">Objetivos del informe:
• Documentar las actividades realizadas en los laboratorios.
• Registrar el patrón de uso de los laboratorios por parte de la comunidad estudiantil.</td></tr>
<tr><td colspan="4">Resultados
Como resultado se obtuvo la siguiente información:
• En total de ordenados en los laboratorios 1,2,3 se encuentra 50 máquinas.
• En laboratorio especializado en Cisco 0se encuentra 10 maquinas
• Los ordenadores se utilizan tanto en la jornada matutina como en la vespertina.
• Los estudiantes utilizan los ordenadores principalmente para actividades de investigación y desarrollo.
• Se observo que muchos estudiantes acceden a las primeras páginas recomendadas por el navegador al buscar información en internet.</td></tr>
</table>

- La institución cuenta con tres páginas web verificadas, siendo la de ambiente virtual de aprendizaje la más usada por la comunidad estudiantil para actividades académicas, seguida del SGA UPSE y la página principal de la Universidad Estatal Península de santa Elena.
- Los alumnos suelen consultar blogs y sitios web alternativos en sus procesos de búsqueda de información.
  - Algunas páginas requieren que los estudiantes creen cuentas e ingresen sus datos personales para acceder a la información.
  - La búsqueda de información en sitios web, suele estar condicionada a la aceptación de términos de uso y al registro de datos personales por parte del usuario.
  - En caso de que un ordenador sea infectado con virus debe ser notificado al técnico docente para que este comunique al personal de tics.
  - No se realiza un monitoreo del acceso a páginas con contenido anómalo por parte del administrador.
  - Los horarios con mayor cantidad de usuarios conectados a la red son de 9 a 10 am y 3 a 4 pm
  - Durante los horarios con actividades educativas, se observa un mayor número de usuarios navegando por internet en los laboratorios de la Facultad