UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

DETECCIÓN DE SPAM EN CORREOS ELECTRÓNICOS MEDIANTE EL USO DE SISTEMA INTELIGENTE PARA LA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES DE LA UPSE.

AUTOR

Franco Avila, Dalemberg Derian

EXAMÉN COMPLEXIVO

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

TUTOR

Lsi. DANIEL QUIRUMBAY YAGUAL, MSIA.

Santa Elena, Ecuador

Año 2024

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ AQUINO

Ing. Jose Sanchez A. Msc.
DIRECTOR DE LA CARRERA

Firmado electrónicamente por:
DANIEL IVAN QUIRUMBAY YAGUAL

Lsi. Daniel Quirumbay Yagual, Msia.
TUTOR

Firmado electrónicamente por:
IVAN ALBERTO CORONEL SUAREZ

Ing. Iván Coronel Suárez, Msia
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MARJORIE ALEXANDRA CORONEL SUAREZ

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 Dalemberg Derian Franco Avila, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 8 días del mes de diciembre del año 2023

TUTOR

Firmado electrónicamente por:
DANIEL IVAN
QUIRUMBAY YAGUAL

Lsi. DANIEL QUIRUMBAY, MSIA.

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

Yo, **Dalemberg Derian Franco Avila**

DECLARO QUE:

El trabajo de Titulación, "**Detección de spam en correos electrónicos mediante el uso de sistema inteligente para la facultad de sistemas y telecomunicaciones de la UPSE",** 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 8 días del mes de diciembre del año 2023

AUTOR

DALEMBERG FRANCO A.

Dalemberg Derian Franco Avila

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 **"Detección de spam en correos electrónicos mediante el uso de sistema inteligente para la facultad de sistemas y telecomunicaciones de la UPSE"**, presentado por el estudiante, Franco Avila Dalemberg Derian fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 10%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

INFORME DE ANÁLISIS
magister

TI_FRANCO_DALEMBERG_COMPLEXIVOF

10% Textos sospechosos

10% Similitudes
< 1% similitudes entre comillas
< 1% Idioma no reconocido
0% Textos potencialmente generados por la IA

Nombre del documento: TI_FRANCO_DALEMBERG_COMPLEXIVOF.docx
ID del documento: 796cc959aba3169d1fe2d0b3d8a22bfc97be5a1d
Tamaño del documento original: 1,41 MB

Depositante: DANIEL IVAN QUIRUMBAY YAGUAL
Fecha de depósito: 11/12/2023
Tipo de carga: interface
fecha de fin de análisis: 11/12/2023

Número de palabras: 7025
Número de caracteres: 48.357

TUTOR

Firmado electrónicamente por:
DANIEL IVAN QUIRUMBAY YAGUAL

Lsi. Daniel Quirumbay, Msia

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Dalemberg Derian Franco Avila**

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 8 días del mes de diciembre del año 2021

EL AUTOR

DALEMBERG FRANCO A.

Dalemberg Derian Franco Avila

AGRADECIMIENTO

Agradezco a mis padres, quienes siempre estuvieron en todo momento pendiente a mí, a mis hermanos, quienes me apoyaron en lo que más necesitaba durante todo el proceso estudiantil.

A los docentes por brindarnos las herramientas y compartir sus conocimientos, para que de esta manera se pueda realizar este trabajo.

Dalemberg Derian, Franco Avila

DEDICATORIA

Este trabajo va dedicado a mis padres, a mis hermanos, quienes fueron los que siempre estuvieron apoyándome y fueron un pilar fundamental durante mi formación como profesional.

A mi familia en general, quienes de alguno u otra forma me brindaron su ayuda cuando la necesité.

A los docentes quienes fueron los guías en las aulas, y en el desarrollo de este trabajo para la finalización del proceso universitario.

Dalemberg Derian, Franco Avila

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INDICE ANEXOS

RESUMEN

Este proyecto se enfoca en implementar un Sistema Inteligente para la detección de correos spam en la Facultad de Sistemas y Telecomunicaciones de la UPSE, mediante el empleo de algoritmos de Deep Learning, se utilizaron algoritmos de aprendizaje profundo para analizar patrones y características asociadas a correos no deseados. La conclusión destaca la efectividad de la inteligencia artificial en fortalecer la ciberseguridad de la institución, proporcionando un sistema robusto y adaptativo para la detección temprana de correos electrónicos maliciosos.

Palabras claves: Algoritmos, Deep Learning, Spam

ABSTRACT

This project focuses on implementing an Intelligent System for the detection of spam emails at the Faculty of Systems and Telecommunications of the UPSE, using Deep Learning algorithms. Deep learning algorithms were used to analyze patterns and characteristics associated with unsolicited emails. The conclusion highlights the effectiveness of artificial intelligence in strengthening the institution's cybersecurity, providing a robust and adaptive system for early detection of malicious emails.

Keywords: Algorithms, Deep Learning, Spam

INTRODUCCIÓN

El presente proyecto realizado se ha denominado “Detección de tráfico de correos electrónicos en la red mediante el uso de sistema inteligente para la Facultad de Sistema y Telecomunicaciones (FACSISTEL) de la Universidad Estatal Península de Santa Elena”, el cual uno de sus principales propósitos es crear un script que detecte si un correo es spam o no.

Esta capturará los datos del correo, la cual será enviada a un análisis por el algoritmo de forma automática, dará como respuesta el detalle de este, luego se clasificará como spam o no basándonos en el resultado obtenido, de esta manera se propone brindar una capa más de seguridad a la red de la institución.

En el capítulo 1 se explica la problemática la cual abarca el proyecto, del mismo modo se detalla las herramientas y la metodología a utilizar en el desarrollo del algoritmo, dando una breve descripción del contenido de las fases llevadas a cabo dentro de esta propuesta.

En el capítulo 2 se detallan las respectivas pruebas del funcionamiento del algoritmo, los conceptos y las teorías empleadas en el proyecto. Se presenta la metodología y lo que se realizó en cada una de las fases dentro de ellas con su respectiva evidencia. Por último, se indican los resultados de la ejecución del algoritmo.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1. Antecedentes

El aumento en el número y la sofisticación de los ataques cibernéticos en los últimos años requiere innovaciones más avanzadas en la estrategia de defensa actual. Los métodos tradicionales de detección de intrusiones, detección de anomalías e inspección profunda de paquetes ya no son suficientes para satisfacer las tendencias actuales de amenazas de seguridad en evolución. Los costos de hardware y software están disminuyendo a medida que aumenta la potencia informática. [1]

Hoy en día, las empresas modernas generan una enorme cantidad de datos, lo que brinda varias oportunidades con respecto a su explotación, pero también representa graves desafíos con respecto a su protección. En esta dirección, los registros de red se consideran una fuente inmensamente rica de información que puede ser explotada para muchos propósitos, como la estimación del estado de funcionamiento actual de la red, y la investigación y prevención de actividades potencialmente maliciosas. [2]

El 30 de agosto de 1995, el Abg. Xavier Tomalá Montenegro, en su calidad de director ejecutivo del comité de gestión, presenta en el seno de dicho comité la exposición de motivos y el Proyecto de Ley para crear la Universidad a nivel estatal que se denomina UNIVERSIDAD DEL PACÍFICO EN LA PENÍNSULA DE SANTA ELENA. El referido proyecto fue aprobado por todos los miembros del comité de gestión y fue presentado en el Congreso Nacional en septiembre de 1995 [3].

Para el conocimiento de cómo se maneja el área de TI dentro de la institución se entrevistó al Ing. Fabricio Ramos, director de Tecnologías y Sistemas de Información (véase anexo 1), nos explica que para el manejo del análisis de datos no existe ninguna persona encargada, esta carencia implica un obstáculo para el aprovechamiento efectivo de la información recopilada, lo que limita la capacidad de la institución en poder tomar decisiones de manera rápida y oportuna.

También se da a conocer la gran falta de sistemas informáticos que ayuden a la seguridad y análisis de la red en la institución, esto puede tener graves consecuencias, como ser

vulnerable a ataques de programa maligno, ransomware y otros tipos de ataques cibernéticos, esto puede provocar la interrupción de los servicios y perdida de datos, paralización de operaciones académicas y administrativas.

Al contar con un solo firewall FORTINET como única línea de defensa, deja vulnerable a un único punto de fracaso, ya que, si este cae, puede provocar un fallo en toda la red, este firewall se enfoca en la seguridad perimetral, es decir, la protección de la red desde el borde externo, este dispositivo no detecta todas las amenazas y vulnerabilidades, sin embargo, esto no es suficiente para proteger completamente la red.

En España, se realizó un trabajo "Detección automática de Spam utilizando Regresión Logística Bayesiana". El sistema que aquí se propone es capaz de detectar automáticamente los correos spam, además compara el modelo con otros dos, con en fin de comprobar la eficiencia y efectividad de los algoritmos, usando la colección de datos de correos SPAMBASE [4]

A nivel de Latinoamérica, en Argentina se realizó un trabajo de "Detección de anomalías en tráfico de red de Sistemas de Control Industrial soportada en algoritmos de machine learning", en el cual se propone la aplicación de algoritmos de ML de clasificación binaria para la construcción de un modelo de detección de anomalías en sistemas ciber físicos industriales en el cual se reflejen condiciones normales y no normales de las comunicaciones entre los dispositivos interconectados a dicha red [5].

En la Península de Santa Elena se desarrolló un proyecto que consiste en desarrollar un agente para la detección de spam en un servidor zimbra, usando técnicas de machine laearning, el cual compara varios modelos usando el mismo Dataset en el cual se almacenan los correos analizados en una base de datos para su posterior. [6].

Una vez descrito los trabajos antes mencionados y explicada la problemática que se presenta en la institución, se desea implementar un algoritmo usando modelos de aprendizaje automático supervisados y no supervisados para poder detectar eventos anómalos maliciosos que se puedan generar en la red mediante un flujo de red preprocesados, que provengan de fuentes heterogéneas.

1.2. Descripción del Proyecto

En la Facultad de Sistemas y Telecomunicaciones, el tráfico de correo electrónico que se genera es muy grande, lo que puede dar lugar a la recepción de un gran número de correos spam. El trabajo consiste en el desarrollo de un sistema de detección de correos spam mediante algoritmos de Deep Learning. Este sistema se entrenará utilizando un conjunto de datos de correos electrónicos etiquetados como spam y no spam.

La falta de medidas de seguridad ante correos electrónicos que contengan spam, archivos o direcciones maliciosas en la que cualquier persona pueda ser victima de este tipo de ataques, esto conlleva a ser un riesgo potencial para la información no solo del usuario si no de la institución en general.

Metodología OMSTD

Para el presente sistema se implementará la metodología OMSTD (Metodología abierta para desarrolladores de herramientas de seguridad) es una metodología para las buenas prácticas en Python para el desarrollo de herramienta de seguridad [32] La metodología está pensada para trabajar en Python, pero se puede extender las mismas ideas para otros lenguajes de programación [32].

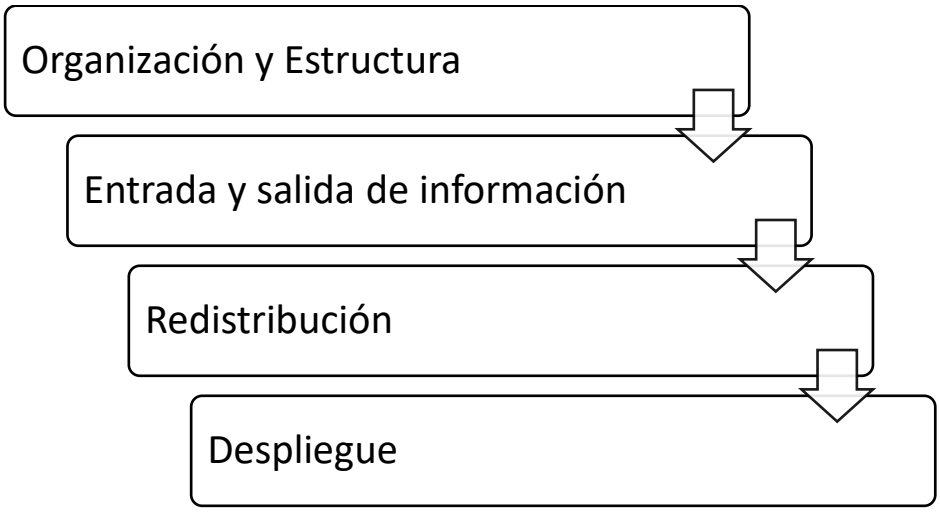


Fig. 1 Modelo OMSTD

1.2.1 Herramientas

Conjunto de datos

- **Dataset:** representa un conjunto completo de datos, incluyendo las tablas que contienen, ordenan y restringen los datos. [7]

Entorno de desarrollo integrado

- **Visual Studio Code:** es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C ++, C #, Java, Python, PHP, Go, .NET). [8]

Validaciones directorio

- **Os:** El módulo os proporciona funciones para interactuar con el sistema operativo. Se utilizará para realizar operaciones relacionadas con archivos y directorios, como la navegación por el sistema de archivos y la manipulación de rutas de archivos [11].

Manipulación de datos tabulares

- **Csv:** El módulo csv de Python proporciona funcionalidades para leer y escribir archivos CSV (Comma-Separated Values), que son comúnmente utilizados para almacenar datos tabulares. Se utilizará para el manejo de datos en formato CSV en el sistema de detección de tráfico anómalo [12].

Análisis y manipulación de datos

- **Pandas:** La librería panda proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento. Se utilizará para la manipulación, limpieza y transformación de datos en el sistema de detección de tráfico anómalo [13].
- **Numpy:** La librería numpy proporciona un soporte eficiente para la manipulación de arreglos multidimensionales y operaciones matemáticas en Python. Será

utilizada para realizar cálculos numéricos en el sistema de detección de tráfico anómalo [14].

Aprendizaje automático y redes neuronales

- **Sklearn**: es una librería de machine learning en Python que proporciona una amplia gama algoritmos de aprendizaje automático supervisados y no supervisados. Ofrece herramientas para el preprocesamiento de datos, selección de características, reducción de dimensionalidad, entrenamiento de modelos y evaluación de rendimiento [15].
    - **Model_selection:** El módulo model_selection de la librería scikit-learn proporciona herramientas para dividir conjuntos de datos en subconjuntos de entrenamiento y prueba. Se utilizará para realizar la división de datos para entrenamiento y evaluación de modelos [16].
    - **Preprocessing:** El módulo preprocessing de scikit-learn proporciona herramientas para preprocesar los datos antes de entrenar un modelo. Esto incluye la normalización de características, codificación de variables categóricas, entre otros [17].
    - **Feature_selection:** El módulo feature_selection de scikit-learn proporciona técnicas para seleccionar las características más relevantes de un conjunto de datos. Ayuda a reducir la dimensionalidad y mejorar la eficiencia del modelo [18].
    - **Decomposition:** El módulo decomposition de scikit-learn proporciona técnicas de reducción de dimensionalidad, como Análisis de Componentes Principales (PCA), para extraer características más significativas [19].
    - **Metrics:** El módulo metrics de scikit-learn proporciona métricas para evaluar el rendimiento de los modelos de aprendizaje automático. Esto incluye métricas como precisión, recall, F1-score, entre otras [20].
- **Keras:** Es una librería de alto nivel para construir y entrenar redes neuronales en Python. Proporciona una interfaz intuitiva y fácil de usar para crear modelos de aprendizaje profundo. Keras permite la construcción de redes neuronales desde

cero o utilizando modelos predefinidos y ofrece una amplia gama de capas y funciones de activación [21].

- **Layers:** El módulo layers de Keras proporciona una amplia gama de capas para construir redes neuronales. Incluye capas como Dense, Conv2D, LSTM, entre otras, que se utilizan para definir la arquitectura del modelo de detección de tráfico anómalo [22].
- **Models:** El módulo models de Keras proporciona herramientas para definir y compilar modelos de aprendizaje automático. Se utilizará para construir el modelo de detección de tráfico anómalo y configurar su proceso de entrenamiento [23].

Visualización de datos

- **Matplotlib:** Es una librería de visualización de datos en Python que proporciona una amplia variedad de gráficos y visualizaciones. Permite crear gráficos estáticos, gráficos interactivos, gráficos en 3D y mucho más [24].
  - **Pyplot:** El módulo pyplot de la librería matplotlib proporciona funciones para crear visualizaciones y gráficos. Se utilizará para visualizar los resultados y realizar análisis exploratorio de los datos en el sistema de detección de tráfico anómalo [25].

Este proyecto contribuirá a la línea de investigación de Tecnología y Sistemas de la Información (TSI) en las organizaciones y en la sociedad, debido a que el presente proyecto se relaciona con temas de seguridad de las Tecnologías de la información (TI), virtualización y seguridad de la infraestructura de la información que permita generar información indispensable para la toma de decisiones. [26]

1.3. Objetivos del Proyecto

Objetivo general

Desarrollar un sistema inteligente mediante el uso de código Python para la detección de tráfico de correo spam en la Facultad de Sistemas y Telecomunicaciones

Objetivos específicos

- Analizar los modelos Deep Learning para la detección de tráfico de correos electrónicos.
- Utilizar algoritmos para el análisis de correos electrónicos.
- Programar un sistema inteligente que permita detectar correos spam.

1.4. Justificación del Proyecto

El IDS monitorea las violaciones e intrusiones en una red o sistema al monitorear los registros del sistema y determinar si alguna actividad parece anómala o desviada del comportamiento normal especificado para un dispositivo o red [27]. Que no pueden ser identificados por un firewall tradicional. Esto es vital para lograr una alta protección contra acciones que comprometan la disponibilidad, integridad o confidencialidad de los sistemas informáticos. [28]

En Ecuador, de acuerdo con el Ministerio de Telecomunicaciones y de la Sociedad de la Información (MINTEL) mediante su Centro de Respuesta a Incidentes Informáticos Ecucert, se notificaron 7.292 ataques en los cuatro primeros meses del 2022, mientras que en 2021 hubo 15.847 alertas, lo que demuestra que desde el primer cuatrimestre del año ya se ha superado el número de incidentes del año pasado. [29]

Una vez realizado el análisis se problemáticas que se presentan en la red de la institución, se propone la implementación de un sistema inteligente basado en modelos supervisados y no supervisados. Este sistema permitirá la detección correos spam mediante el reconocimiento de patrones de comportamiento, brindando una mayor capacidad de protección y seguridad.

Para el desarrollo de un plan de acción se tomará en cuenta la documentación sobre las vulnerabilidades encontradas en el análisis para de esta manera poder ir agregando las respectivas configuraciones y cambios para que el sistema sea capaz de contener esta amenaza previo a la ejecución de éste, dentro de la red. De la misma manera se tomará en cuenta las medidas de ciberseguridad con los que cuenta la institución ante un posible ataque.

El tema planteado está alineado con los objetivos del Plan de Creación de Oportunidades la cual se va a describir a continuación:

Eje social [30]

Objetivo 5. Proteger a las familias, garantizar sus derechos y servicios, erradicar la pobreza y promover la inclusión social. [30]

Política 5.4

A4. Fortalecer la conectividad y el acceso a las TIC como una vía para mejorar el acceso a otros servicios [30]

Objetivo 7. Potenciar las capacidades de la ciudadanía y promover una educación innovadora, inclusiva y de calidad en todos los niveles [30]

Política 7.2. Promover la modernización y eficiencia del modelo educativo por medio de la innovación y el uso de herramientas tecnológicas. [30]

1.5. Alcance del Proyecto

La implementación del sistema inteligente basado en Deep Learning permitirá detectar los correos spam que se presentan en la red de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

El propósito del proyecto es realizar un algoritmo que detecte tráfico de correos spam en la red mediante el uso de un sistema inteligente para la Facultad de Sistemas y Telecomunicaciones, con el fin de tener una medida de seguridad ante posibles ataques que la institución pueda recibir. Mediante métodos de comprobación de un listado de datos anómalos se alertará este tipo de tráfico. Para esto se van a tener en cuenta las siguientes fases:

En la organización y estructura, se llevará a cabo la estructura del sistema, también se establecerán las funciones y los parámetros para poder analizar de mejor manera el tráfico de correos spam. Se identificarán y comprenderán los principales activos de la red y su importancia para la organización. Este proceso ayudará a obtener una visión integral de la organización y sentará las bases para las siguientes etapas del proyecto.

Durante la fase entrada y salida de información, el sistema tomará un dataset para su entrenamiento, esto permitirá que los algoritmos pueden predecir si un correo es spam o no, de acuerdo con los parámetros establecidos previos a su entrenamiento.

Para la fase de redistribución, se desarrollará un plan detallado para el desarrollo de cada componente del sistema, como la extracción de características, entrenamiento de modelos, y la predicción. Esto implica seleccionar las tecnologías y algoritmos adecuados para la detección de correos spam, como técnicas de aprendizaje profundo y análisis de comportamiento. Se definirá un marco de evaluación y métricas para medir su efectividad. Todo esto se realizará con el objetivo de desarrollar un sistema robusto y eficaz para la detección de spam en los correos.

En la fase final del proyecto, se llevará a cabo el despliegue del sistema inteligente para la detección de correos spam. Esto incluirá la preparación de los conjuntos de datos necesarios para entrenar y probar el sistema. Se desarrollará e implementará el sistema utilizando técnicas de aprendizaje profundo y análisis de comportamiento. En función de los resultados obtenidos, se realizarán ajustes y mejoras al sistema para lograr resultados óptimos. Finalmente, se documentará detalladamente el proceso de prueba, incluyendo las configuraciones utilizadas, los resultados obtenidos y las lecciones aprendidas durante el desarrollo del proyecto.

CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

2.1. Marco Teórico

Modelo de defensa en profundidad para la protección contra ciberataques

En lo que respecta a seguridad de la información en las redes corporativas, existe un concepto de mucha utilidad para aquellos que participen en las áreas involucradas en la misma, denominado defensa en profundidad (también conocido como Defense in Depth). Se trata de un modelo que pretende aplicar controles en seguridad para proteger los datos en diferentes capas, tal como muestra la siguiente imagen: [39]

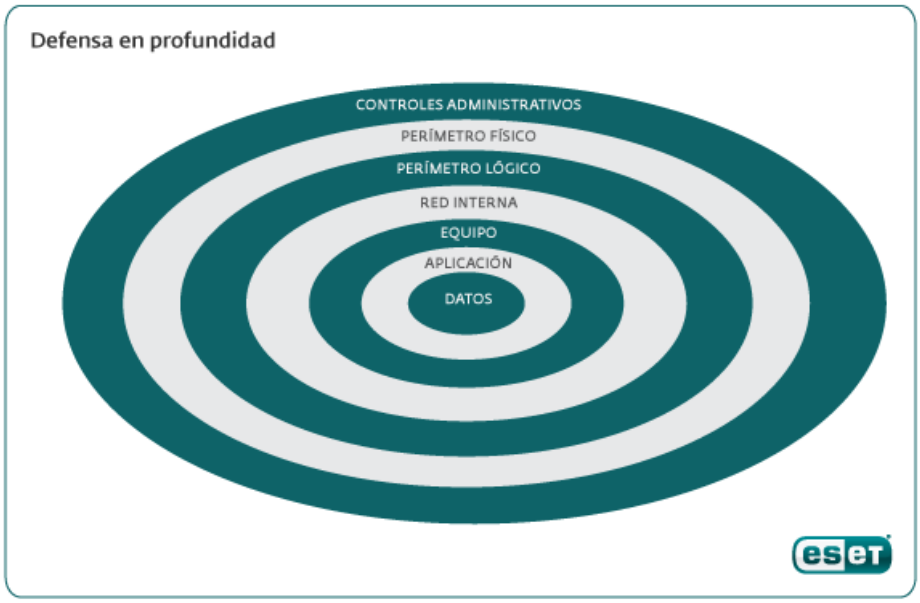


Fig. 2 Defensa en profundidad

El modelo de defensa en profundidad es una teoría de seguridad informática ampliamente aceptada que propone implementar múltiples capas de seguridad para proteger los sistemas y los datos de los ciberataques. Este enfoque se basa en el supuesto de que se requiere más de una medida de seguridad para una protección completa [40].

Nohota Cantor en su artículo "Defensa en profundidad para la protección contra las amenazas persistentes avanzadas", indica que en los últimos años los atacantes informáticos utilizan con más frecuencia las nuevas tecnologías. Lo que da paso al concepto en seguridad de "Amenazas Persistentes Avanzadas (APT)" [40].

El modelo se basa en la hipótesis de que las violaciones de seguridad pueden detectarse monitoreando los registros de auditoría de un sistema para detectar patrones anormales de uso del sistema [41]. El modelo incluye perfiles para representar el comportamiento de los sujetos con respecto a objetos en términos de métricas y modelos estadísticos, y reglas para adquirir conocimiento sobre este comportamiento a partir de registros de auditoría y para detectar un comportamiento anómalo [41].

La profundidad de la defensa informática estará dada por cada una de las zonas en las que dividamos los sistemas de nuestra organización. Las puertas de acceso y los caminos entre ellas lo proporcionan los diferentes elementos de red (switchs, puntos de acceso

inalámbricos, routes, firewalls, etc.) y la interconexión entre ellos. Estos nodos de red delimitarán y segmentarán las diferentes áreas en las que deseemos instalar los servidores y hosts. Las zonas mínimas que debemos contemplar son [42]:

- Redes externas.
- DMZs (Zonas desmilitarizadas).
- MZs (Zonas Militarizadas).
- Core (Zona de máxima seguridad).

Técnica de reputación de URL en la detección de anomalías en el tráfico de FTP

Según Animesh Patcha y Parque Jung Min en su artículo "Una visión general de las técnicas de detección de anomalías: soluciones existentes y últimas tendencias tecnológicas", indica que el uso de esta técnica aporta una capa adicional de seguridad, al implementar estas en conjunto con otras técnicas de detección, se puede mejorar la identificación y clasificación de posibles amenazas [43].

Para equilibrar el rendimiento y la eficiencia del almacenamiento, los sistemas de archivos en clúster modernos suelen almacenar primero los datos con replicación, seguidos de la codificación de los datos replicados con codificación de borrado [44].

Una técnica constituye el enfoque conceptual para extraer la información de los datos, y, en general es implementada por varios algoritmos. Cada algoritmo representa, en la práctica, la manera de desarrollar una determinada técnica paso a paso, de forma que es preciso un entendimiento de alto nivel de los algoritmos para saber cuál es la técnica más apropiada para cada problema [45].

Riesgos de seguridad en las organizaciones

La mayoría de los dispositivos tecnológicos utilizados por organizaciones en todo el mundo presentan vulnerabilidades. Según CYBSEC Security, muchas de estas debilidades pueden estar presentes en el producto desde su diseño, posiblemente debido a la omisión de requisitos mínimos de seguridad de la información que todo nuevo software debe cumplir. Estos aspectos deben ser tenidos en cuenta por los analistas de sistemas desde la etapa de ingeniería de requisitos.[46].

2.2. Metodología del Proyecto

Para la elaboración del proyecto se necesitará seleccionar una correcta metodología, que permita la gestión desde que se empieza como el progreso de las diferentes fases de acuerdo con los objetivos ya establecidos. De tal forma que se lleva un control para resolver problemas que se puedan presentar en el proceso de desarrollo. La metodología que se va a emplear es OSMTD que es una metodología para el desarrollo del script.

Bloques

La forma de bloques que se utilizara para elaborar los scripts tiene los siguientes subtemas [32].

- **Organización y estructura (ST):** Se emplea una serie de normas para definir la estructura que va a tener el código, se establece secciones para cada tipo de funciones y parámetros. Se define una estructura de carpeta que contendrán parte o documentos específicos del algoritmo.
- **Entrada y salida de Información (IO):** El sistema permitirá el ingreso de un archivo .CSV, el cual contendrá los archivos de correos electrónicos para su análisis.
- **Redistribución:** Se desarrollarán scripts individuales para cada componente del sistema de detección de spam, incluyendo la extracción de características, el entrenamiento del modelo y la predicción.
- **Despliegue:** El sistema de detección de spam se implementará en un entorno de prueba. Se realizarán pruebas y se analizarán los resultados para garantizar la eficacia del sistema.

2.2.1. Metodología de Investigación

Debido a que existen vacíos en cuanto a la información y lo relevante de los datos incluyendo la obtención de éstos, se escoge utilizar para esta investigación la metodología exploratoria [31]. Tomando de referencia trabajos que tenga similitud o relación a la implementación de un sistema inteligente que detecte spam en los correos electrónicos.

Al referirse a trabajos relacionados con la implementación de un sistema inteligente para detectar spam en correos electrónicos, se puede obtener una visión general de las funciones y procesos en el campo de la ciberseguridad, especialmente en la protección de datos. Esto permite familiarizarse con los temas clave que se abordan para profundizar la comprensión de los requerimientos y establecer los criterios más precisos.

Con el fin de cumplir con los objetivos establecidos anteriormente, en esta parte de la documentación se usa la metodología de tipo diagnóstica. [31] En este tipo de metodología, su fin es poder llevar un análisis de situaciones puntuales sobre la seguridad que mantiene la institución, esto nos permitirá tomar en cuenta con mayor profundidad la situación.

Con la propuesta se busca mejorar la seguridad de la infraestructura de TI que tiene la institución reduciendo el tiempo en la que el personal de TI identifica los correos spam en la red de la institución. Para la recopilación, procesamiento y análisis de la información se basará en el análisis de los modelos a entrenar.

2.2.2. Técnicas e instrumentos de recolección de datos

Entrevistas: En esta parte de la recolección de información, se realiza un diálogo de manera de entrevista al director de Tecnologías y Sistemas de Información de la institución. (véase anexo 1) Donde se nos proporcionaran datos más importantes con el fin de establecer los requerimientos puntuales con respecto a las funciones que desempeñan sus sistemas y las medidas de seguridad que están empleando.

El tiempo de detección de anomalías que se presentan en la red no es estimado, ni tiene una estimación, ya que no se cuenta con algún sistema o aplicativo que ayude a identificar este tipo de tráfico de manera interna, el único método con el que se cuenta es mediante el proveedor de servicio de internet, sin embargo, esto puede tomar horas, o inclusive semanas antes de poder identificar alguna anomalía.

2.2.3. Metodología de desarrollo

Metodología OSMTD

Organización y estructura (ST): Se define la estructura que va a tener el código del sistema de detección de spam. Se establecen secciones para cada tipo de funciones y

parámetros, como la extracción de características, el entrenamiento del modelo y la predicción. Se define una estructura de carpeta que contendrá partes o documentos específicos del algoritmo de Deep Learning.

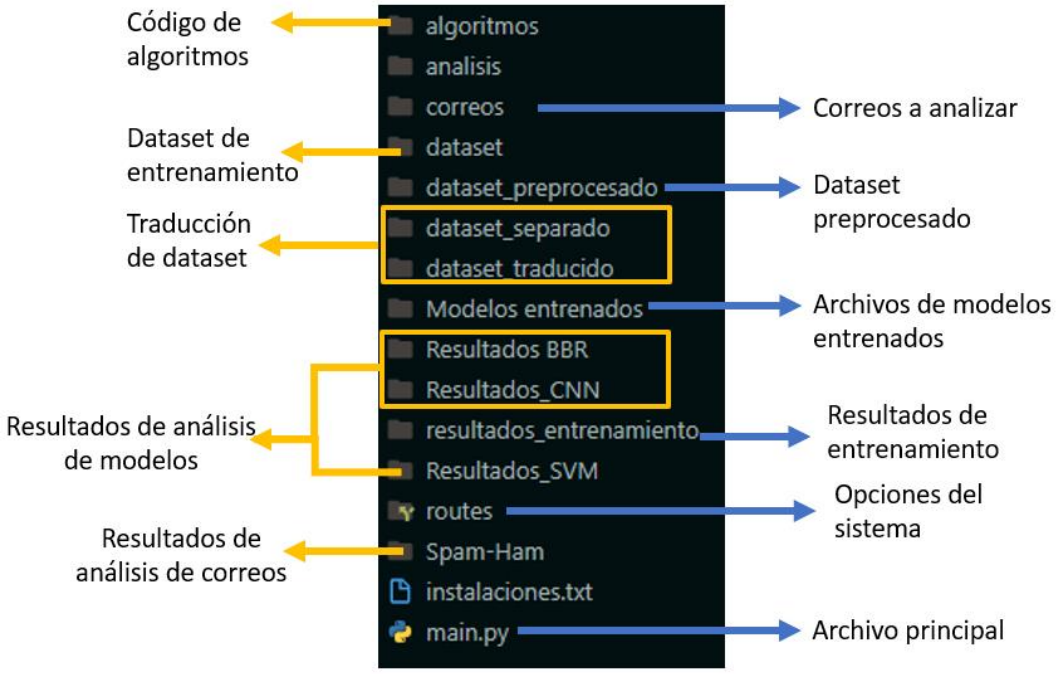


Fig. 3 Estructura del sistema

Entrada y salida de Información (IO): El sistema permitirá el ingreso de un archivo .CSV, el cual contendrá los datos de correos electrónicos para su análisis. La salida será una clasificación de cada correo electrónico como spam o no spam. El archivo de entrada para el entrenamiento del sistema es un CSV, en este caso un Dataset el cual cuenta con etiquetas de spam y de texto.

Text	Class
Supply Quality China's EXCLUSIVE dimensions at Unbeatable Price.Dear Sir, We are pleas	1
over. SidLet me know. Thx.	0
Dear Friend,Greetings to you.I wish to accost you with a request that would be of immense b	1
MR. CHEUNG PUIHANG SENG BANK LTD.DES VOEUX RD. BRANCH,CENTRAL HONG K	1
Not a surprising assessment from Embassy.	0
Monica -Huma Abedin <Huma@clintonemail.com>Tuesday June 29 2010 6:01 AM'hanleymr	0
Pls print.H <hrod17@clintonemail.com>Thursday October 8 2009 8:01 PM'JilotyLC@state.g	0
Dear Tom--H <hrod17@clintonemail.com>Friday December 11 2009 5:41 PMCould we sche	0
Greetings from barrister Robert Williams=2CDear friend=2C I know that my letter will come t	1
FYI. Thanks again for signing the book ---- and I do hope you get royalties from Mongolia! Al	0
Pls putRELEASE IN PARTB6H <hrod17@clintonemail.com>Friday July 31 2009 3:25 PM'Jil	0
Not sure about SRAP. I'll forward it on. Hope the trip is going well.Jm	0
SOLICITING FOR A BUSINESS VENTURE AND PARTNESHIP.DEAR SIR,I AM THE SON C	1
Dear Sir/Madam,Compliments of the season. It is indeed my pleasure to write to you thislette	1
Dear Bill and TomI'm sorry that I cannot be on the Hill today as we had long planned but very	0

Fig. 4 Datos del dataset Fraud Email

Redistribución: Se desarrollarán múltiples scripts para cada modelo del sistema de detección de spam. Se considerará su compatibilidad con el sistema operativo. Para poder realizar el entrenamiento de los modelos se debe procesar el Dataset a usar, este proceso incluye eliminación de datos duplicados, ya que esto puedo tener consecuencias al momento de ser utilizado.

La eliminación de características del conjunto de datos también es un proceso que ayuda al modelo, sin embargo, al usar un Dataset que solo cuenta con las etiquetas de “Text” y “Class”, tal como se observa en la figura 4, no es necesario realizar la eliminación, con Dataset que cuentan con más etiquetas que las antes mencionadas si se necesitará realizar este paso. La eliminación de datos nulos es otro paso que se debe realizar en todo Dataset a usar, esto permite que las etiquetas cuenten con el mismo número de datos en cada una.

```python
# Eliminar duplicados basándose en la columna 'Text'
df = df_original.drop_duplicates(subset='Text', keep='first')

# Eliminar columnas no especificadas (en este caso, ninguna)
df = df.drop([], axis=1)

# Eliminar filas con valores nulos
df = df.dropna()
```

Fig. 5 Código para eliminación de datos duplicados y nulos

Los Dataset en su mayoría se encuentran en inglés, por ello es necesario realizar la traducción del mismo, al ser un archivo que almacena un gran número de datos la traducción de éste conlleva mucho tiempo, por esto se dividió el Dataset en partes iguales para de esta manera poder traducir parte por parte.

```python
# Función para dividir un archivo CSV en partes
def dividir_csv(input_file):
    # Cargar el archivo CSV en un DataFrame
    ruta = os.getcwd()
    df = pd.read_csv(input_file)
    carpeta = 'dataset_separado'
    chunk_size = 1000  # Número de filas por parte

    # Dividir el DataFrame en partes iguales
    total_rows = len(df)
    num_chunks = total_rows // chunk_size
    if total_rows % chunk_size != 0:
        num_chunks += 1

    print("Dividiendo archivo en partes.")
    for i in range(num_chunks):
        start_idx = i * chunk_size
        end_idx = (i + 1) * chunk_size
        chunk_df = df[start_idx:end_idx]

        # Guardar cada parte en un nuevo archivo CSV
        output_folder = os.path.join(ruta, carpeta)
        os.makedirs(output_folder, exist_ok=True)

        chunk_filename = os.path.join(output_folder, f'fraud_{i + 1}.csv')
        chunk_df.to_csv(chunk_filename, index=False)
        print(f'Fraud {i + 1} guardada en {carpeta}')
```

Fig. 6 Función para dividir el archivo en partes iguales

Una vez ejecutado el código se obtendrá los siguientes archivos:

Fig. 7 Separación de dataset en partes iguales

Para poder reducir el tiempo de traducción se crea una función para poder ejecutar la traducción usando los procesadores lógicos del computador, esto reduce el tiempo de traducción notablemente. Antes de este proceso se comprueba el número de procesadores logios del computador.

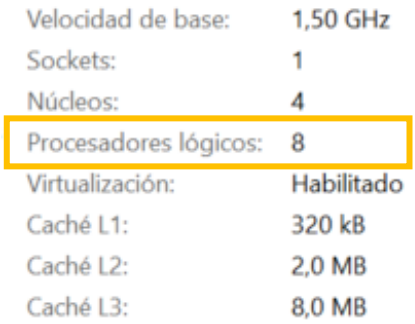


Fig. 8 Número de procesadores lógicos disponibles

Para poder realizar la traducción del archivo se procedió a crear una función en donde se pueda usar automáticamente la cantidad máxima de procesadores lógicos disponibles, también se muestra una barra de progreso que va aumentando conforme vaya traduciendo las partes.

```python
# Función para traducir archivos en una carpeta usando hilos de manera concurrente
def traducir_archivos_en_carpeta(carpeta_entrada, carpeta_salida):
    if not os.path.exists(carpeta_salida):
        os.makedirs(carpeta_salida)

    archivos_csv = [archivo for archivo in os.listdir(carpeta_entrada) if archivo.endswith('.csv')]

    hilos_utilizados = os.cpu_count()
    print(f'Número de hilos de procesamiento lógico disponibles: {hilos_utilizados}')

    # Usa tqdm para agregar una barra de progreso
    with tqdm(total=len(archivos_csv), desc='Procesando archivos') as pbar:
        def actualizar_progreso(*_):
            pbar.update()

        with concurrent.futures.ThreadPoolExecutor(max_workers=hilos_utilizados) as executor:
            for archivo in archivos_csv:
                archivo_entrada = os.path.join(carpeta_entrada, archivo)
                # Agrega la carpeta 'dataset_traducido' al path de salida
                carpeta_salida_traducido = os.path.join(carpeta_salida)
                executor.submit(traducir_archivo, archivo_entrada, carpeta_salida_traducido).add_done_callback(actualizar_progreso)
```

Fig. 9 Función de traducción de cada parte del dataset usando procesadores lógicos disponibles

A continuación, se observa una vista previa del proceso y el tiempo que tarda en realizar la traducción, también se visualiza la ruta donde se guardan los archivos traducidos.

```
Número de hilos de procesamiento lógico disponibles: 8
Procesando archivos:   0%|                                                    | 0/11 [00:00<?, ?it/s]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_11.c
sv
Procesando archivos:   9%|▉                                                   | 1/11 [02:04<20:42, 124.29s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_1.cs
v
Procesando archivos:  18%|█▊                                                  | 2/11 [07:34<36:46, 245.16s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_4.cs
v
Procesando archivos:  27%|██▋                                                 | 3/11 [07:49<18:42, 140.34s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_3.cs
v
Procesando archivos:  36%|███▋                                                | 4/11 [07:58<10:18, 88.40s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_10.c
sv
Procesando archivos:  45%|████▌                                               | 5/11 [08:00<05:42, 57.14s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_5.cs
v
Procesando archivos:  55%|█████▍                                              | 6/11 [08:04<03:15, 39.16s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_2.cs
v
Procesando archivos:  64%|██████▎                                             | 7/11 [08:10<01:53, 28.43s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_6.cs
v
Procesando archivos:  73%|███████▎                                            | 8/11 [10:33<03:14, 64.97s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_7.cs
v
Procesando archivos:  82%|████████▏                                           | 9/11 [14:54<04:12, 126.11s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_8.cs
v
Procesando archivos:  91%|█████████                                          | 10/11 [25:48<04:49, 289.14s/it]
Archivo traducido y guardado como C:\Users\Administrator\Desktop\CORREOS\dataset_traducido\fraud_9.cs
v
Procesando archivos: 100%|██████████| 11/11 [25:58<00:00, 141.68s/it]
Archivo final guardado como fraud_esp.csv

Proceso completo en 26 minutos y 4.94 segundos.
```

Fig. 10 Proceso de traducción de cada parte del Dataset

Al finalizar el proceso de traducción se guardan los archivos en una carpeta, después de esto, se debe realizar la combinación de los archivos traducidos para poder entrenar los modelos de manera correcta con un conjunto de datos consolidado.

```python
# Función para unir archivos CSV traducidos en uno solo
def unir_archivos_csv():
    ruta = os.getcwd()
    carpeta = "dataset_traducido"
    carpeta_entrada = os.path.join(ruta, carpeta)
    archivo_salida = 'fraud_esp.csv'
    ruta_salida = carpeta_entrada

    # Obtener la lista de archivos en la carpeta de entrada, ordenados alfabéticamente
    archivos_partes = sorted(
        [archivo for archivo in os.listdir(carpeta_entrada) if archivo.endswith('.csv')],
        key=lambda x: int(''.join(filter(str.isdigit, x.split('_')[1])))
    )

    # Crear un DataFrame vacío para almacenar los datos
    df_final = pd.DataFrame()

    # Iterar sobre cada archivo y cargarlo en el DataFrame final
    for archivo_parte in archivos_partes:
        ruta_archivo_parte = os.path.join(carpeta_entrada, archivo_parte)
        df_parte = pd.read_csv(ruta_archivo_parte)
        df_final = pd.concat([df_final, df_parte], ignore_index=True)

    # Guardar el DataFrame final en un archivo CSV
    ruta_salida_completa = os.path.join(ruta_salida, archivo_salida)
    df_final.to_csv(ruta_salida_completa, index=False)
    print(f'Archivo final guardado como {archivo_salida}')
```

Fig. 11 Función para unir los archivos traducidos en un solo

Al finalizar este proceso obtendremos un archivo combinado junto a las partes del mismo. Tal como se observa a continuación:

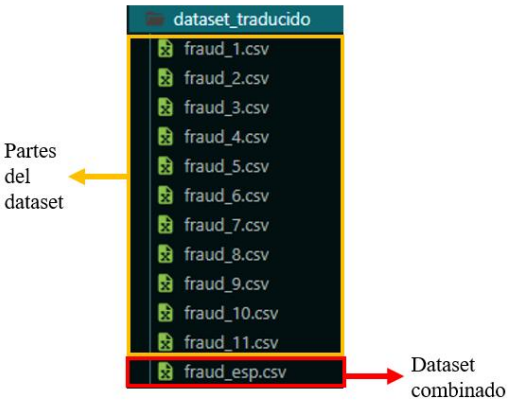


Fig. 12 Resultado de la combinación de archivos

Text	Class
Calidad de suministro Las dimensiones exclusivas de China a un precio inmejorable.Produc	1
encima.Sidlet me sabe.Gracias.	0
Querido amigo, saludos. Deseo acostarlo con una solicitud que sea de inmenso beneficio pa	1
SEÑOR.Cheung Puihang Seng Bank Ltd.Des Voeux Rd.Rama, Central Hong Kong, Hong K	1
No es una evaluación sorprendente de la embajada.	0
Monica -Huma Abedin <huma@clintonemail.com> Martes 29 de junio de 2010 6:01 am'hanl	0
Pis print.h <hrod17@clintonemail.com> Jueves 8 de octubre de 2009 8:01 pm'jilotylc@state.	0
Estimado Tom-h <hrod17@clintonemail.com> Viernes 11 de diciembre de 2009 5:41 PM ¿p	0
Saludos desde el abogado Robert Williams = 2CDear Friend = 2C Sé que mi carta vendrá a	1
FYI.¡Gracias de nuevo por firmar el libro ---- y espero que obtengas regalías de Mongolia!Ta	0
Pls putrelease en partb6h <hrod17@clintonemail.com> Viernes 31 de julio de 2009 3:25 pm'	0
No estoy seguro de srap.Lo reenviaré.Espero que el viaje vaya bien.jm	0

Fig. 13 Resultado de traducción de Dataset

Para poder usar el algoritmo con mayor precisión se analizaron los resultados del entrenamiento de cada uno, obteniendo como resultado los siguientes cuadros:

El primer modelo entrenado fue el algoritmo de Regresión Logística Bayesiana (BBR), para esto se utilizó la librería tensorFlow la cual se usa para construir y entrenar modelos tanto de machine learning como Deep learning, también se usó la librería Scikit-Learn que sirve para el aprendizaje automático en Python.

Para el modelo se usa el optimizador “adam” y la metrica elegida para evaluar el rendimiento es la precisión o “accuracy”, para el entrenamiento se establecen los parámetros de “epochs=3”, esto se refiere que se analiza el conjunto de datos 3 veces, se utilizan lotes en 32 muestras en cada actualización con “batch_size=32” y se usa el 20% de los datos de entrenamiento para evaluar en rendimiento durante el entrenamiento con “validation_split=0,2”, tal como se observa en el anexo 2 el cual contiene el código del entrenamiento.

El modelo también utiliza el embeddings preentrenado Universal Sentence Encoder (USE), el cual se usa para reducir el tiempo y recursos necesarios para poder entrenar un modelo, también se usó la tokenización, usando un total 2050 datos contenidos en el dataset fraud_email.csv, en proceso de entrenamiento tuvo una duración de 2 minutos y con una precisión total de 0.9551.

	Precisión	Recall	F1- score	Support
HAM	0.94	0.99	0.96	1181
SPAM	0.98	0.91	0.95	869
Accuracy			0.96	2050

Tabla 1 Resultados del entrenamiento del algoritmo BBR

El siguiente modelo entrenado fue el algoritmo de Máquina de Vectores de Apoyo (SVM), el cual no usa tokenización, en el cual se configuraron los siguientes parámetros, para regularización se usó "C" en cual controla la penalización por error en el entrenamiento, el campo "coeficiente de kernel" se estableció en "gamma", el cual controla el alcance de influencia de un solo ejemplo, en el anexo 3 se puede ver la codificación del modelo.

Usando un total 2050 datos contenidos en el dataset fraud_email.csv, el proceso de entrenamiento tuvo una duración de 20 minutos dando una precisión total de 0.9863 y como resultado los siguientes datos:

	Precisión	Recall	F1- score	Support
HAM	0.98	1.00	0.99	1181
SPAM	1.00	0.97	0.98	869
Accuracy			0.99	2050

Tabla 2 Resultados de entrenamiento del algoritmo SVM

Despliegue: Una vez entrenado los modelos, y comparando los resultados se opta por usar el modelo de Máquina de Vectores de Apoyo (SVM), con el cual se llevarán a cabo pruebas y se analizarán los resultados. Esto permitirá ajustar y mejorar el sistema.

Para poder realizar en análisis del sistema primero de debe cargar el modelo seleccionado, a continuación, se visualizan las líneas de código para poder realizar este proceso:

```python
# Cargar modelos y vectorizador previamente entrenados
vectorizer_path = "Modelos entrenados/vectorizer_SVM.joblib"
modelo_svm_path = "Modelos entrenados/modelo_SVM.joblib"

vectorizer = load(vectorizer_path)
modelo_svm = load(modelo_svm_path)
```

Fig. 14 Función para cargar el modelo entrenado

Luego se crea una función que nos permita listar los archivos de una carpeta en especifico para poder elegir el que se desea analizar.

```python
# Listar archivos en una carpeta y permitir al usuario elegir uno
def listar_y_elegir_archivo(carpeta):
    archivos_en_carpeta = [f for f in os.listdir(carpeta) if os.path.isfile(os.path.join(carpeta, f))]
    print('<<<<<<<----------------------------------------------------->>>>>>>')
    if not archivos_en_carpeta:
        print("No hay archivos en la carpeta.")
        return None

    print("Archivos disponibles:")
    for i, archivo in enumerate(archivos_en_carpeta, start=1):
        print(f"{i}. {archivo}")

    while True:
        try:
            seleccion = int(input("Seleccione el número del archivo a analizar (o 0 para salir): "))
            if seleccion == 0:
                return None
            elif 1 <= seleccion <= len(archivos_en_carpeta):
                return os.path.join(carpeta, archivos_en_carpeta[seleccion - 1])
            else:
                print("Número inválido. Por favor, elija un número válido.")
        except ValueError:
            print("Entrada inválida. Por favor, ingrese un número.")
```

Fig. 15 Función para listar los archivos de una carpeta

Cabe recalcar que el sistema solo admite archivos .eml el cual es el formato de un correo electrónico, una vez se elije el archivo, se procede a extraer la información del correo electrónico con el fin de guardarla para su posterior análisis.

```python
# Extraer información relevante de un correo a partir de su contenido en bytes
def extraer_informacion_correo(correo_bytes):
    try:
        # Parsear el correo a partir de bytes
        msg = mailparser.parse_from_bytes(correo_bytes)

        # Extraer información relevante del correo
        informacion_correo = {
            'id': None,  # Identificador único del correo
            'date': msg.headers.get('Date', 'Fecha no encontrada'),
            'from': msg.from_[0][1],  # Remitente
            'to': [x[1] for x in msg.to],  # Destinatarios
            'id_group_mail': msg.headers.get('Message-ID', 'Id no encontrado'),  # ID del grupo de correos
            'subject': msg.headers.get('Subject', 'Asunto no encontrado'),  # Asunto del correo
            'body': msg.text_plain[0] if msg.text_plain else 'No hay mensaje',  # Cuerpo del mensaje
            'attachments': [file.get('filename') for file in msg.attachments],  # Archivos adjuntos
            'es_spam': None  # Etiqueta de spam, se llenará después del análisis
        }

        return informacion_correo
    except Exception as e:
        print(f'Ha ocurrido un error en la función extraer_informacion_correo: {str(e)}')
        return None
```

Fig. 16 Función para la extracción de los campos del correo electrónico

Para poder identificar cada correo analizado se crea una función para poder asignarle un id único, a continuación, se presenta la función.

```python
# Generar un ID único para cada correo
def generar_id_unico():
    return str(uuid.uuid4())
```

Fig. 17 Función para crear un id único para cada correo

Luego se procede a analizar el correo, para esto creamos una función que revisa por parámetros el cuerpo del correo, y el modelo entrenado, para así, poder predecir si el correo es spam o no.

```python
# Analizar si un correo es spam utilizando un modelo SVM previamente entrenado
def analizar_spam(correo, vectorizer, modelo_svm):
    try:
        # Obtener el cuerpo del mensaje del correo
        cuerpo_del_mensaje = correo.get('body', '')

        # Realizar predicción sobre si es spam o no
        es_spam = modelo_svm.predict(vectorizer.transform([cuerpo_del_mensaje]))[0]
        correo['es_spam'] = bool(es_spam)
    except Exception as e:
        print(f'Error al analizar el correo: {str(e)}')
        correo['es_spam'] = None

    return correo['es_spam']
```

Fig. 18 Función de análisis de correo

Por ultimo se guardan los resultados en un archivo JSON, para su posterior visualización.

```python
# Guardar resultados en un archivo JSON
def guardar_resultados_en_json(correo, carpeta_destino):
    correo_id = correo['id']
    if correo_id is None:
        print("No se puede guardar el resultado sin un ID asignado.")
        return

    nombre_archivo = f'resultado_correo.json'
    ruta_json = os.path.join(carpeta_destino, nombre_archivo)

    try:
        # Intentar cargar resultados anteriores
        with open(ruta_json, 'r') as json_file:
            resultados_antiguos = json.load(json_file)
    except (FileNotFoundError, json.decoder.JSONDecodeError):
        # Si el archivo no existe o no se puede decodificar como JSON, empezar con una lista vacía
        resultados_antiguos = []

    # Agregar el nuevo correo a la lista existente
    resultados_antiguos.append(correo)

    # Guardar la lista actualizada en el archivo JSON
    with open(ruta_json, 'w') as json_file:
        json.dump(resultados_antiguos, json_file, indent=4)
```

Fig. 19 Función para guardar el análisis del correo

CAPÍTULO 3. PROPUESTA

3.1. Requerimientos

3.1.1. Requerimientos Funcionales

R01. El sistema debe ser capaz de analizar el contenido de los correos electrónicos en busca de posibles amenazas o contenido malicioso.

R02. El sistema debe detectar y bloquear correos electrónicos sospechosos o potencialmente dañinos.

R03. El sistema debe detectar y bloquear correos electrónicos con enlaces maliciosos o URLs sospechosas.

R04. El sistema debe ser capaz de identificar patrones de spam y filtrar correos electrónicos no deseados.

R05. El sistema debe analizar los metadatos de los correos electrónicos para identificar posibles anomalías o comportamientos sospechosos.

R06. El sistema debe generar alertas y notificaciones en tiempo real cuando se detecte tráfico malicioso de correos electrónicos.

R07. El sistema debe ser capaz de generar informes detallados sobre el tráfico de correos electrónicos y las amenazas detectadas.

R08. El sistema debe ser escalable y capaz de gestionar grandes volúmenes de tráfico de correos electrónicos.

R09. El sistema debe tener un rendimiento óptimo para no afectar negativamente el ancho de banda de la red.

R010. El sistema debe ser capaz de integrarse con soluciones de seguridad existentes, como firewalls y sistemas de prevención de intrusiones.

R011. El sistema debe contar con mecanismos de actualización automática para mantenerse al día con las últimas amenazas y técnicas de ataque.

R012. El sistema debe tener un panel de administración intuitivo y fácil de usar para la configuración y supervisión.

R013. El sistema debe contar con registros de entrenamientos detallados para el seguimiento y la revisión de actividades.

R014. El sistema debe ser capaz de identificar y bloquear correos electrónicos que contengan información confidencial o datos sensibles.

R015. El sistema debe ser capaz de realizar análisis de reputación de remitentes para identificar correos electrónicos provenientes de fuentes no confiables.

R016. El sistema debe contar con mecanismos de respaldo y recuperación de datos para garantizar la disponibilidad y la integridad de la información.

R017. El sistema debe ser capaz de detectar y bloquear correos electrónicos con contenido inapropiado o no permitido según las políticas de la Facultad.

R018. El sistema debe tener la capacidad de realizar análisis de comportamiento para detectar actividades anómalas en los correos electrónicos.

R019. El sistema debe permitir la gestión centralizada de las reglas de detección de tráfico malicioso de correos electrónicos.

3.1.2. Requerimientos no Funcionales

R01. El sistema debe ser fácil de administrar y mantener. Debe incluir herramientas de gestión que permitan a los administradores realizar tareas de mantenimiento de manera eficiente.

R02. El sistema debe ser capaz de manejar grandes volúmenes de tráfico de correos electrónicos sin degradar su rendimiento.

R03. El sistema debe ser capaz de adaptarse y actualizarse constantemente para hacer frente a las nuevas amenazas y técnicas utilizadas por los atacantes en el tráfico malicioso de correos electrónicos.

3.2. Componentes de la Propuesta

A continuación, se describen los componentes de hardware y software para la realización del proyecto:

Hardware

Cantidad	**Herramientas**	**Requisitos**
1	Laptop HP	500 Gb SSD 12 Gb RAM 8 procesadores lógicos

Tabla 3 Componentes de harware

Software

Herramientas
Python
Visual Studio Code
Librerias de python

Tabla 4 Componentes de software

3.2.1. Arquitectura del Sistema

El siguiente grafico representa la arquitectura del sistema, y como los algoritmos detectan si un correo es spam.

Fig. 20 Arquitectura del sistema

3.2.2. Diagramas de casos de uso

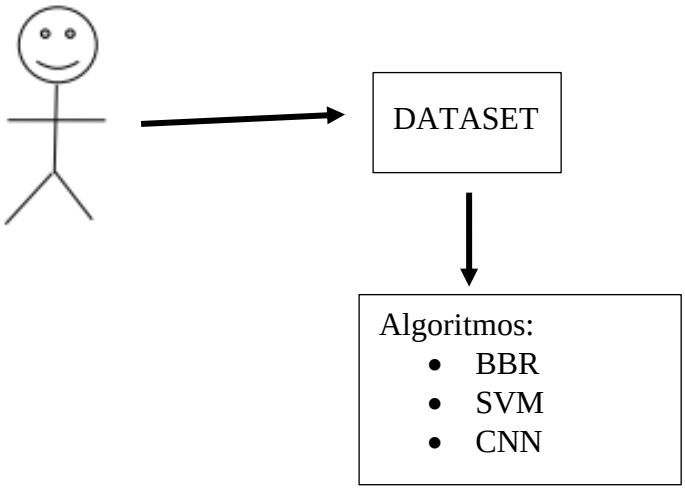
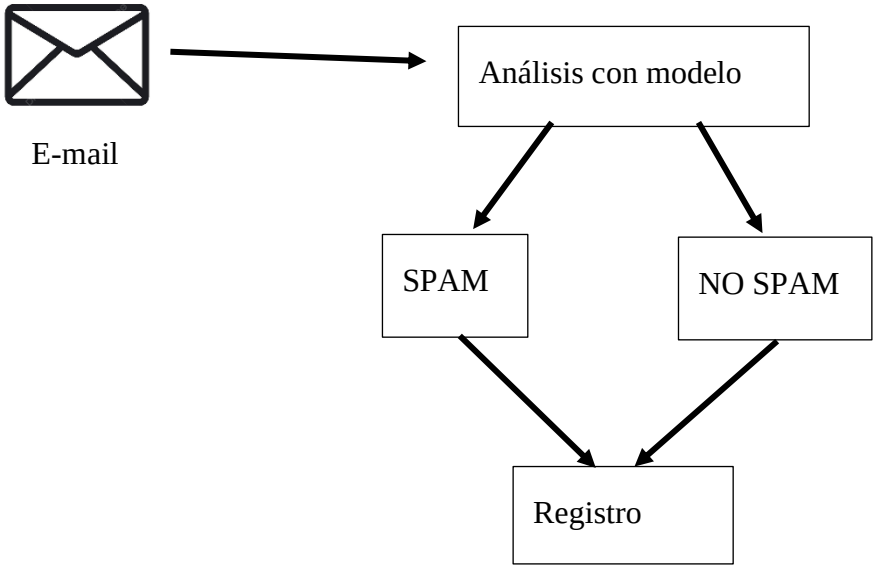
<table>
<tr><td colspan="2">Caso de uso: Entrenamiento de algoritmos</td></tr>
<tr><td>Actor</td><td>Administrador</td></tr>
<tr><td>Propósito</td><td>Enseñar al algoritmo a identificar las características que distinguen los correos electrónicos no deseados de los correos electrónicos legítimos.</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Pasos realizados:
• Selección de dataset para el entrenamiento
• Elección de algoritmos a usar
• Elegido el dataset y los algoritmos, se los entrena con el mismo conjunto de datos</td></tr>
</table>

Tabla 5 Proceso de entrenamiento de modelos

Caso de uso: Entrenamiento de algoritmos	
Actor	Administrador
Propósito	Analizar correos electrónicos.



Pasos realizados:

- Seleccionar el correo electrónico
- Análisis con modelo elegido
- Guardar los datos del correo analizado

Tabla 6 Proceso de análisis

Caso de uso: Visualización de resultados de entrenamiento	

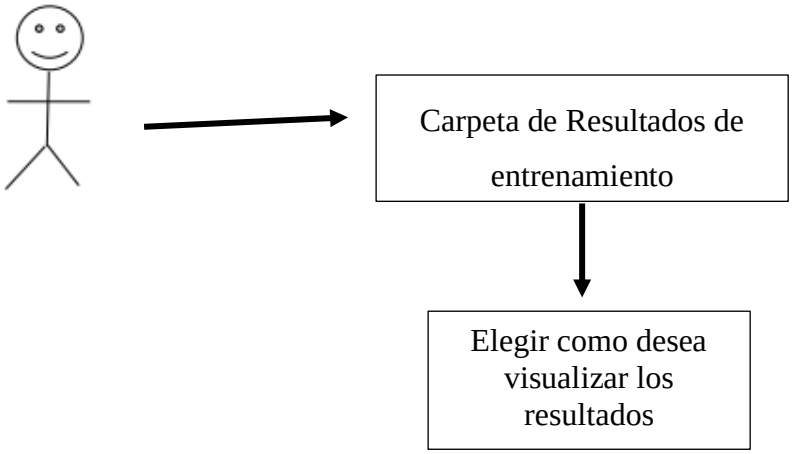
Actor	Administrador
Propósito	Visualización de los resultados obtenidos del entrenamiento de los algoritmos.
	
Pasos realizados: • El resultado de los entrenamientos se guarda en una sola carpeta. • Se deberá elegir el archivo a mostrar • Elegir la forma de visualización	

Tabla 7 Proceso de visualización de resultados

3.2.3. Modelado de Datos

Los datos se guardarán en un archivo JSON para su posterior visualización, los datos a guardar serán fecha, remitente, destinatario, asunto, cuerpo del correo y una etiqueta de si es o no spam, es cual se asignará después de su análisis, la estructura en que se manejaran los datos es la siguiente:

```
"id": 1,
"date": "Fri, 01 Dec 2023 11:30:41 +0000",
"from": "marketing@contents.com",
"to": [
    "dalemberg.rosales@gmail.com"
],
"id_group_mail": "<getresponse.xserial.s56br48HSVeVdMp4V0/Hw/WbmFKPJWq2JnT0MXW3i
+zENMPVHVv3pC3KPOCG6Gm6Y1C4G3zVPLd4nU3Ak8cfKn7orjarOPGkyJDXNiZ1uGJEKoUYU7+HoxAi72llWqru9V06iqVlJrTVGLBigCMz+7SYeQ==@gr-mail6.
com>",
"subject": "Cyber Monday - Maximiza tus contenidos con herramientas de texto",
"body": "Descubre c\u00f3mo nuestras herramientas pueden mejorar tus contenidos. \nIf you don't read this message correctly,
click here. [https://app.getresponse.com/view.html?x=a62b&m=BX40RO&mc=lM&s=BlkD1Gd&u=McSn4&z=EFy2MMZ&]\n\n[https://us-ms.
gr-cdn.com/getresponse-McSn4/photos/9718ebc8-a3e5-41ce-b226-ad444b963f5b.gif] [https://app.getresponse.com/click.html?x=a62b&
lc=hIExfV&mc=lM&s=BlkD1Gd&u=McSn4&z=EVim9IS&]\n\n\n\u00a1Hola!\n\n\nEste Cyber Monday, \u00a1descubre c\u00f3mo nuestras
herramientas de texto pueden transformar tus contenidos!\n\n\n * Reformula los contenidos existentes para lograr un mayor
impacto.\n   \n * Sintetiza textos claros y comprensibles.\n   \n * Elimina errores con el corrector gramatical.\n   \n *
Simplifica conceptos complejos.\n   \n * Parafrasea para que tu contenido sea \u00fanico y fresco.\n   \n * Extrae lo mejor
de cualquier URL.\n   \n   \n\nHaga clic aqu\u00ed [https://app.getresponse.com/click.html?x=a62b&lc=hIEx8N&mc=lM&s=BlkD1Gd&
u=McSn4&z=EB7ihiT&]para no te pierdas la oferta exclusiva: el 50 % de descuento de por vidacon el c\u00f3digo CM50,
\u00a1v\u00e1lido hasta el 7 de diciembre!\n\n\n\n\nEmma de Contents.com\n\n\n\n\n\nfacebook [https://us-as.gr-cdn.com/images/
common/templates/messages/v2/social/facebook9.png] [https://app.getresponse.com/click.html?x=a62b&lc=hIExU2&mc=lM&s=BlkD1Gd&
u=McSn4&z=EVvq9OQ&]\n\ninstagram [https://us-as.gr-cdn.com/images/common/templates/messages/v2/social/instagram9.png]
[https://app.getresponse.com/click.html?x=a62b&lc=hIExkh&mc=lM&s=BlkD1Gd&u=McSn4&z=ESwfMZH&]\n\nlinkedin [https://us-as.
gr-cdn.com/images/common/templates/messages/v2/social/linkedin9.png] [https://app.getresponse.com/click.html?x=a62b&lc=hIExYY&
mc=lM&s=BlkD1Gd&u=McSn4&z=EItXwC1&]\n\ntwitter [https://us-as.gr-cdn.com/images/common/templates/messages/v2/social/twitter9.
png] [https://app.getresponse.com/click.html?x=a62b&lc=hIExxR&mc=lM&s=BlkD1Gd&u=McSn4&z=Et7sz0b&]\nCopyright \u00a9 2023
Contents SPA, All rights reserved.\nPrivacy Policy [https://app.getresponse.com/click.html?x=a62b&lc=hIExTg&mc=lM&s=BlkD1Gd&
u=McSn4&z=EEkAQso&]|Terms & Conditions [https://app.getresponse.com/click.html?x=a62b&lc=hIExpL&mc=lM&s=BlkD1Gd&u=McSn4&
z=EFRFhat&]\n\n\n\nYou may unsubscribe [https://app.getresponse.com/unsubscribe.html?x=a62b&m=BX40RO&mc=lM&s=BlkD1Gd&u=McSn4&
z=EILkIn&pt=unsubscribe] or change your contact details [https://app.getresponse.com/change_details.html?x=a62b&m=BX40RO&
s=BlkD1Gd&u=McSn4&z=E94XVDS&pt=change_details] at any time.\n",
"attachments": [],
"es_spam": false
```


Fig. 21 Estructura de datos

3.3. Diseño de Interfaces

El sistema tendrá una opción de menú de ingreso, mediante el cual se va a poder elegir las opciones de limpieza de dataset, traducción de dataset, entrenamiento de algoritmos, visualización de resultados y análisis de correos.

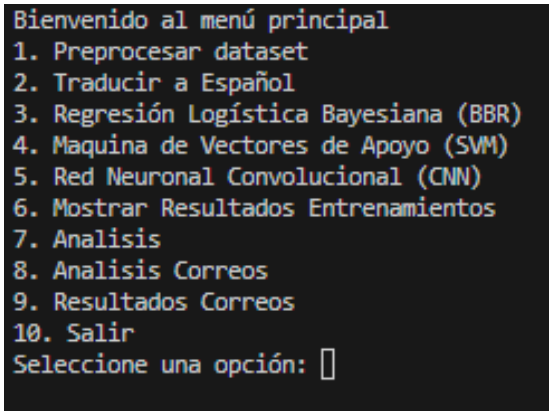
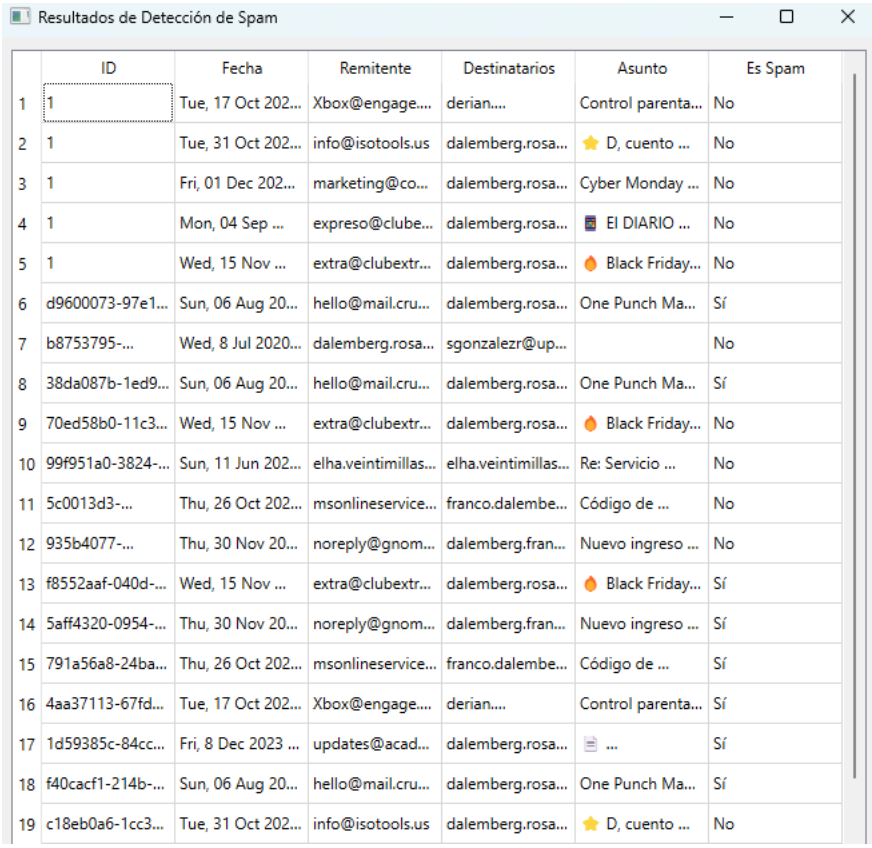


Fig. 22 Interfaz de menú principal

Gráfico de detalles de correos almacenados después de su respectivo análisis



	ID	Fecha	Remitente	Destinatarios	Asunto	Es Spam
1	1	Tue, 17 Oct 202...	Xbox@engage....	derian....	Control parenta...	No
2	1	Tue, 31 Oct 202...	info@isotools.us	dalemberg.rosa...	D, cuento ...	No
3	1	Fri, 01 Dec 202...	marketing@co...	dalemberg.rosa...	Cyber Monday ...	No
4	1	Mon, 04 Sep ...	expreso@clube...	dalemberg.rosa...	El DIARIO ...	No
5	1	Wed, 15 Nov ...	extra@clubextr...	dalemberg.rosa...	Black Friday...	No
6	d9600073-97e1...	Sun, 06 Aug 20...	hello@mail.cru...	dalemberg.rosa...	One Punch Ma...	Sí
7	b8753795-...	Wed, 8 Jul 2020...	dalemberg.rosa...	sgonzalezr@up...		No
8	38da087b-1ed9...	Sun, 06 Aug 20...	hello@mail.cru...	dalemberg.rosa...	One Punch Ma...	Sí
9	70ed58b0-11c3...	Wed, 15 Nov ...	extra@clubextr...	dalemberg.rosa...	Black Friday...	No
10	99f951a0-3824-...	Sun, 11 Jun 202...	elha.veintimillas...	elha.veintimillas...	Re: Servicio ...	No
11	5c0013d3-...	Thu, 26 Oct 202...	msonlineservice...	franco.dalembe...	Código de ...	No
12	935b4077-...	Thu, 30 Nov 20...	noreply@gnom...	dalemberg.fran...	Nuevo ingreso ...	No
13	f8552aaf-040d-...	Wed, 15 Nov ...	extra@clubextr...	dalemberg.rosa...	Black Friday...	Sí
14	5aff4320-0954-...	Thu, 30 Nov 20...	noreply@gnom...	dalemberg.fran...	Nuevo ingreso ...	Sí
15	791a56a8-24ba...	Thu, 26 Oct 202...	msonlineservice...	franco.dalembe...	Código de ...	Sí
16	4aa37113-67fd...	Tue, 17 Oct 202...	Xbox@engage....	derian....	Control parenta...	Sí
17	1d59385c-84cc...	Fri, 8 Dec 2023 ...	updates@acad...	dalemberg.rosa...	...	Sí
18	f40cacf1-214b-...	Sun, 06 Aug 20...	hello@mail.cru...	dalemberg.rosa...	One Punch Ma...	Sí
19	c18eb0a6-1cc3...	Tue, 31 Oct 202...	info@isotools.us	dalemberg.rosa...	D, cuento ...	No

Fig. 23 Guardado de correos analizados

Gráfico de datos almacenados de todos los análisis

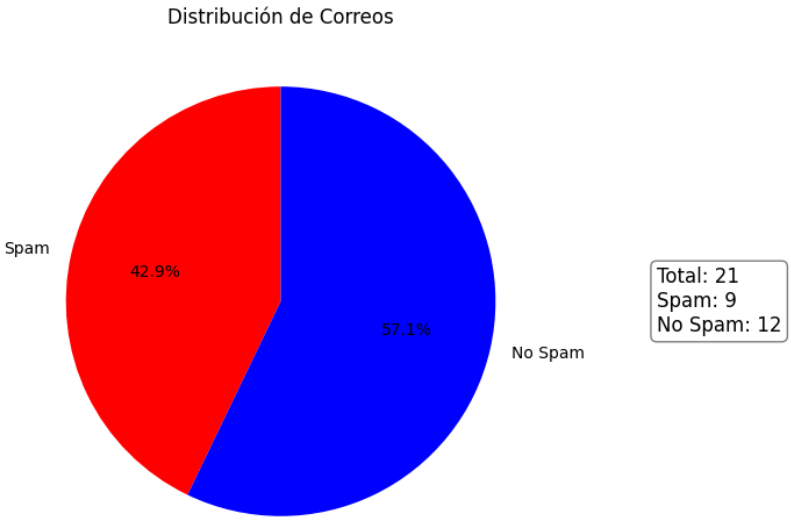


Fig. 24 Presentación de resultados totales

A continuación, se muestra un gráfico estadístico sobre las palabras claves analizadas en todos los correos:

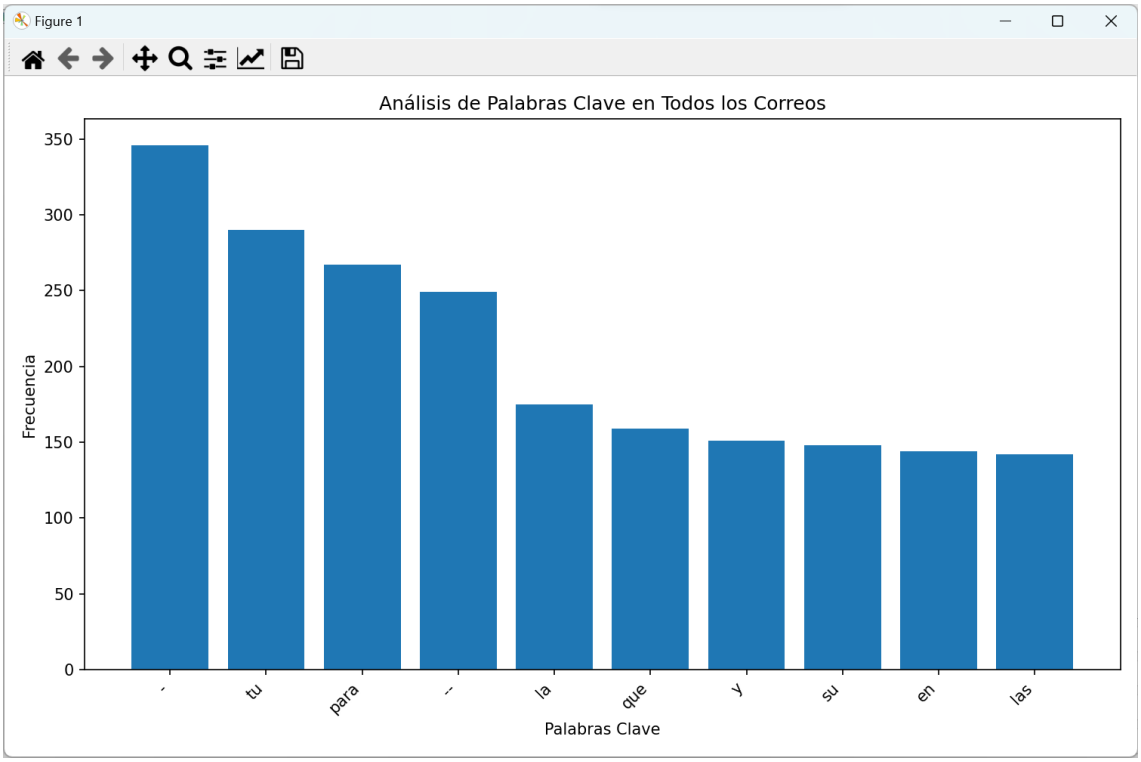


Fig. 25 Gráfico sobre análisis de palabras claves en todos los correos

También se muestra un gráfico sobre la frecuencia de palabras en los correos detectados como spam y los que no:

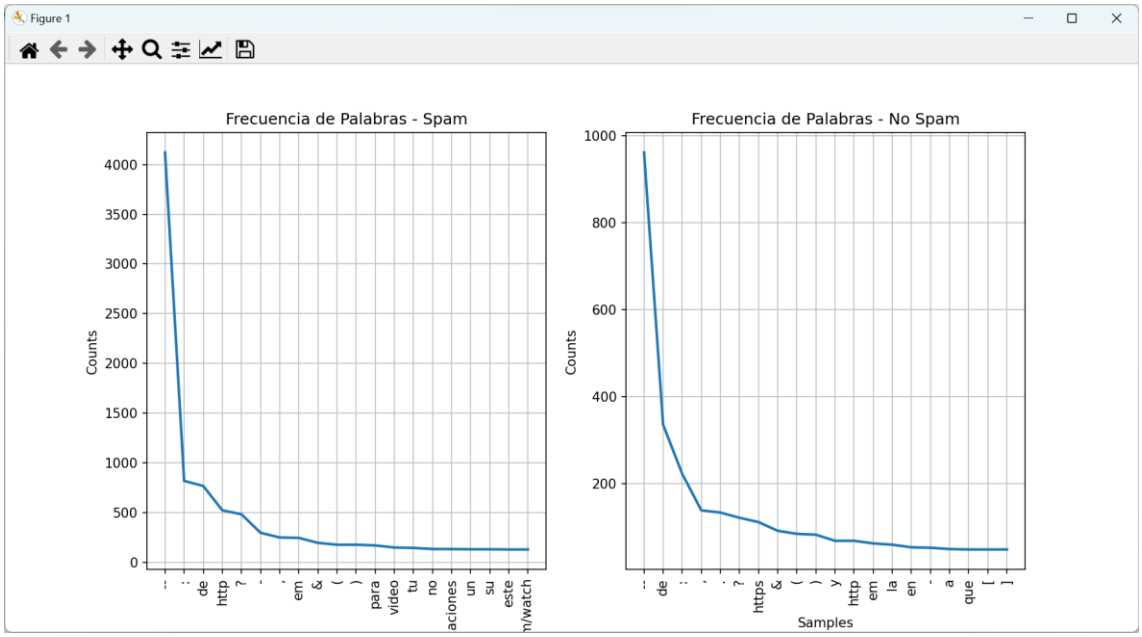


Fig. 26 Gráfico de frecuencia de palabras en los correos

Se genera un gráfico para la visualización de detección de outliers, el cual ayuda a entender como “piensa” el modelo a la hora de decidir si un correo es spam o no, mientras más alto o bajo el valor más seguro esta el modelo.

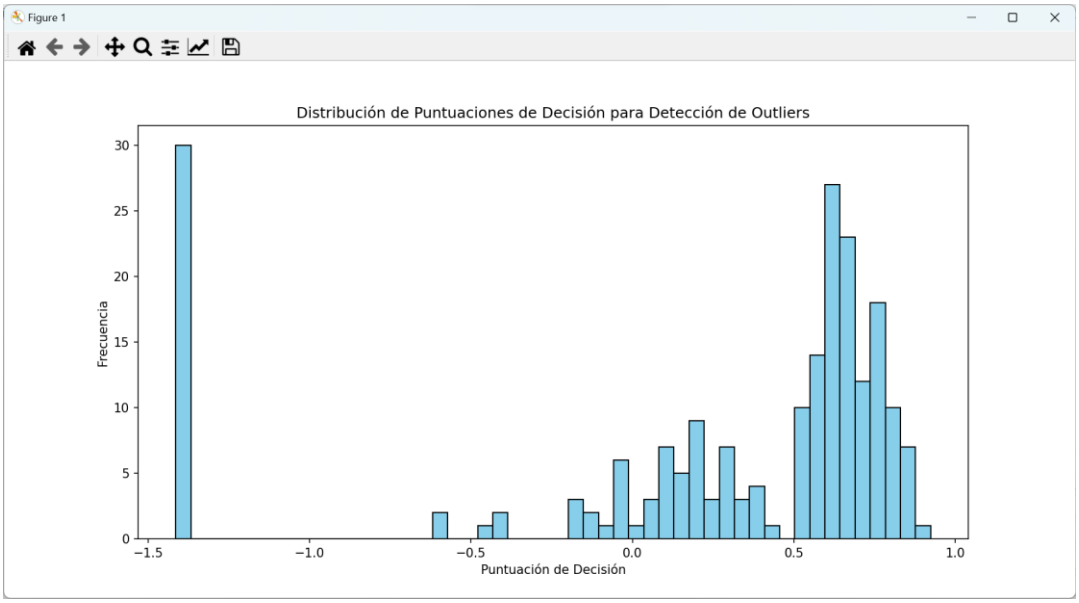


Fig. 27 Gráfico de visualización de outliers

Menú de opciones para poder visualizar el resultado del entrenamiento de los modelos

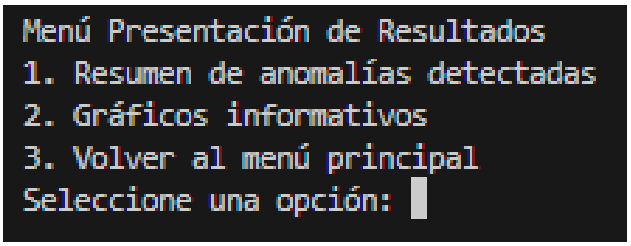


Fig. 28 Menú de presentación de resultados de modelos entrenados

3.4. Pruebas

Se realizaron varias pruebas para poder corroborar que el sistema detecta sí un correo es spam o no.

	ID	Fecha	Remitente	Destinatarios	Asunto	Es Spam
5	1	Wed, 15 Nov ...	extra@clubextr...	dalemberg.rosa...	Black Friday...	No
6	d9600073-97e1...	Sun, 06 Aug 20...	hello@mail.cru...	dalemberg.rosa...	One Punch Ma...	Sí
7	b8753795-...	Wed, 8 Jul 2020...	dalemberg.rosa...	sgonzalezr@up...		No
8	38da087b-1ed9...	Sun, 06 Aug 20...	hello@mail.cru...	dalemberg.rosa...	One Punch Ma...	Sí
9	70ed58b0-11c3...	Wed, 15 Nov ...	extra@clubextr...	dalemberg.rosa...	Black Friday...	No
10	99f951a0-3824-...	Sun, 11 Jun 202...	elha.veintimillas...	elha.veintimillas...	Re: Servicio ...	No
11	5c0013d3-...	Thu, 26 Oct 202...	msonlineservice...	franco.dalembe...	Código de ...	No
12	935b4077-...	Thu, 30 Nov 20...	noreply@gnom...	dalemberg.fran...	Nuevo ingreso ...	No

Fig. 29 Resultado de análisis de correos

En la figura 26 se observa el campo “Es Spam”, este campo se asigna automáticamente según sistema, aquí es donde se asigna la variable “Si” o “No”, esto dependerá del análisis del contenido del correo.

El sistema también permite el análisis de los correos directamente desde la bandeja de entrada, sin embargo, solo funciona con el dominio de Gmail, al intentar usar el dominio de Microsoft, se presentan varios errores, como error en la compatibilidad de las versiones necesarias para poder realizar esta operación. Para poder analizar los correos desde el dominio Gmail se necesita activar la opción de Acceso IMAP, el cual nos permitirá conectarnos al correo.

CONCLUSIONES

- Al existir muchos algoritmos de Deep Learning, se complica la elección del modelo a usar para la detección de spam.
- Mediante la comparación de los resultados de entrenamiento se logró determinar cual modelo tiene mayor precisión.
- Para poder validar que el modelo entrenado elegido estuviese bien se realizaron pruebas con el otro modelo (BBR), el cual daba resultados erróneos a la hora de realizar el análisis, con esto con verifica que se optó por el modelo correcto.

RECOMENDACIONES

- Se recomienda que entrenar cualquier modelo ya sea de machine learning o Deep learning, siempre se debe preprocesar el conjunto de datos con el que se va a trabajar, para de esta manera, obtener un modelo sin fallas y muy certero al momento de realizar un análisis.
- Se recomienda entrenar el modelo con regularidad para que aprenda nuevas tendencias para la detección de spam.
- Se recomienda implementar en un entorno controlado para poder realizar más pruebas a todo tipo de correos electrónicos.

REFERENCIAS

[1] I. Fosić, D. Žagar, K. Grgić y V. Križanović, «Detección de anomalías en el tráfico de red NetFlow mediante algoritmos supervisados de aprendizaje automático,» *Revista de Integración de la Información Industrial*, vol. 33, 2023.

[2] K. Fotiadou, T.-H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou y T. Zahariadis, «Information,» *Detección de anomalías de tráfico de red a través de Deep Learning,* vol. 15, nº 5, 2021.

[3] U. E. P. d. S. Elena, «Universidad Estatal Peninsula de Santa Elena,» 26 Julio 2018. [En línea]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=10&Itemid=188. [Último acceso: 15 Noviembre 2022].

[4] A. R. Pedroviejo, Mayo 2021. [En línea]. Available: https://repositorio.uam.es/bitstream/handle/10486/700030/ramos_pedroviejo_alba_tfg.pdf?sequence=1&isAllowed=y. [Último acceso: 14 Noviembre 2022].

[5] M. A. T. Muñoz, 10 Febrero 2023. [En línea]. Available: https://repository.urosario.edu.co/server/api/core/bitstreams/59f6d889-25c8-4ddd-9037-6300453ea369/content. [Último acceso: 25 Mayo 2023].

[6] J. D. G. RUIDIAZ, 2022. [En línea]. Available: https://repositorio.upse.edu.ec/bitstream/46000/8681/1/UPSE-TTI-2022-0039.pdf. [Último acceso: 13 Noviembre 2022].

[7] L. Microsoft, «Microsoft,» 21 Febrero 2023. [En línea]. Available: https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/dataset-datatable-dataview/. [Último acceso: 5 Junio 2023].

[8] V. S. Code, «Visual Studio Code,» [En línea]. Available: https://code.visualstudio.com/docs/supporting/faq. [Último acceso: 23 Noviembre 2022].

[9] «PyShark documentation,» [En línea]. Available: https://kiminewt.github.io/pyshark/. [Último acceso: 6 Agosto 2023].

[10] «Biblioteca de manipulación IPv4 / IPv6,» [En línea]. Available: https://docs.python.org/es/3/library/ipaddress.html. [Último acceso: 6 Agosto 2023].

[11] «Documentación oficial de Python para el módulo os,» [En línea]. Available: https://docs.python.org/es/3/library/os.html. [Último acceso: 6 Agosto 2023].

[12] «Documentación oficial de Python para el módulo csv,» [En línea]. Available: https://docs.python.org/es/3/library/csv.html. [Último acceso: 6 Agosto 2023].

[13] « Documentación oficial de pandas,» [En línea]. Available: https://pandas.pydata.org/docs/. [Último acceso: 6 Agosto 2023].

[14] «Documentación oficial de numpy,» [En línea]. Available: https://numpy.org/doc/stable/. [Último acceso: 6 Agosto 2023].

[15] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/documentation.html. [Último acceso: 6 Agosto 2023].

[16] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/model_selection.html. [Último acceso: 6 Agosto 2023].

[17] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing. [Último acceso: 6 Agosto 2023].

[18] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/modules/feature_selection.html. [Último acceso: 6 Agosto 2023].

[19] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition. [Último acceso: 6 Agosto 2023].

[20] «Documentación oficial de scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Último acceso: 6 Agosto 2023].

[21] «Documentación oficial de Keras,» [En línea]. Available: https://keras.io/api/. [Último acceso: 6 Agosto 2023].

[22] «Documentación oficial de Keras,» [En línea]. Available: https://keras.io/api/layers/. [Último acceso: 6 Agosto 2023].

[23] «Documentación oficial de Keras,» [En línea]. Available: https://keras.io/api/keras_nlp/models/. [Último acceso: 6 Agosto 2023].

[24] «Documentación oficial de matplotlib,» [En línea]. Available: https://matplotlib.org/stable/users/index.html. [Último acceso: 6 Agosto 2023].

[25] «Documentación oficial de matplotlib,» [En línea]. Available: https://matplotlib.org/stable/users/explain/api_interfaces.html. [Último acceso: 6 Agosto 2023].

[26] Universidad Estatal Penindula de Santa Elena, Santa Elena - La Libertad, Resolución RCF-FST-SO-09 No. 03-2021.

[27] C. P. S. T. Ltd., «GEEKFLARE,» 31 Agosto 2021. [En línea]. Available: https://geekflare.com/es/ids-vs-ips-network-security-solutions/. [Último acceso: 18 Junio 2023].

[28] A. Khraisat, I. Gondal, P. Vamplew y J. Kamruzzaman, «Estudio de los sistemas de detección de intrusos: técnicas, conjuntos de datos y desafíos,» *Cyberseguridad,* vol. 2, p. 20, 17 Julio 2019.

[29] Prensa.ec, «Prensa.ec,» 22 Junio 2022. [En línea]. Available: https://prensa.ec/2022/06/22/ecuador-es-uno-de-los-paises-mas-vulnerables-para-los-ciberdelincuentes/. [Último acceso: 19 Junio 2023].

[30] S. N. d. Planificación, Noviembre 2021. [En línea]. Available: https://www.protrade.ec/wp-content/uploads/2022/06/PND-Plan-de-Creaci%C3%B3n-de-Oportunidades-2021-2025-.pdf. [Último acceso: 20 Junio 2023].

[31] R. H. Sampieri, C. F. Collado y P. B. Lucio, Metodología de la Investigación Sexta Edición, México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V., 2014.

[32] J. M. O. Candel, Hacking ético con herramientas Python, Madrid: Ra-MA, 2019.

[33] UPSE. [En línea]. Available: https://www.upse.edu.ec/secretariageneral/images/archivospdfsecretaria/5.%20INSTRUCTIVOS/09%20EXPEDIR%20EL%20INSTRUCTIVO%20DE%20POLITICAS%20DE%20GESTIONINSTITUCIONAL%20DE.pdf. [Último acceso: 23 Septiembre 2023].

[34] «UPSE,» 14 Diciembre 2021. [En línea]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=12&Ite. [Último acceso: 23 Septiembre 2023].

[35] Google, «Google Maps,» [En línea]. Available: https://maps.app.goo.gl/Mp2RNfRqQzPxqxer7. [Último acceso: 30 Septiembre 2023].

[36] A. López, Seguridad Informatica, Editex, 2010.

[37] J. P. Sifre, «IDS de red para la detección de ataques sobre SSH y FTP,» Junio 2020. [En línea]. Available: https://rua.ua.es/dspace/bitstream/10045/107579/1/IDS_de_red_para_la_deteccion_de_ataques_sobre_SSH_y_FTP_Perez_Sifre_Jose.pdf. [Último acceso: 23 Septiembre 2023].

[38] H. H. C. Huamán, A. Han y L. S. García, Junio 2020. [En línea]. [Último acceso: 23 Septiembre 2023].

[39] S. Bortnik, «We Live Security,» ESET, 24 Mayo 2010. [En línea]. Available: https://www.welivesecurity.com/la-es/2010/05/24/defensa-en-profundidad/. [Último acceso: 30 Septiembre 2023].

[40] C. O. N. Milena, «Repositorio Institucional Universidad Piloto de Colombia,» 30 Abril 2019. [En línea]. Available: http://repository.unipiloto.edu.co/handle/20.500.12277/5974. [Último acceso: 23 Septiembre 2023].

[41] D. Denning, «An Intrusion-Detection Model,» *IEEE Transactions on Software Engineering,* vol. 13, nº 2, pp. 222 - 232, Febrero 1987.

[42] A. C. Estrada, Ciberseguridad "Una estrategia informático/militar", Madrid: DarFe, 2017.

[43] A. Patcha y J.-M. Park, «An overview of anomaly detection techniques: Existing solutions and latest technological trends,» *Science Direct,* vol. 51, pp. 3448 - 3470, 2007.

[44] «Enabling Efficient and Reliable Transition from Replication to Erasure Coding for Clustered File Systems,» *IEEE Transactions on Parallel and Distributed Systems,* vol. 28, nº 9, pp. 2500 - 2513, 2017.

[45] C. S. M. Yolanda y N. B. R. Amparo, «DSpace Escuela Superior Politécnica de Chimborazo,» 12 Noviembre 2014. [En línea]. Available: http://dspace.espoch.edu.ec/handle/123456789/3545#. [Último acceso: 30 Semtiembre 2023].

[46] L. F. Fuentes, «MALWARE, UNA AMENAZA DE INTERNET,» *Universidad Nacional Autónoma de México,* vol. 9, nº 4, 2008.

ANEXOS

Anexo 1 Entrevista

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES TECNOLOGÍAS DE LA INFORMACIÓN	
Entrevista realizada al Ing. Fabricio Ramos, director de Tecnologías y Sistemas de Información	
Objetivo: Conocer el funcionamiento de la institución	
1. ¿Cuántas personas trabajan en el departamento de TI?	
18 personas actualmente trabajan en el departamento.	
2. ¿Cuáles son las principales responsabilidades del departamento de TI en términos de seguridad de la información y protección de datos?	
• Existen dos personas encargadas del manejo de la confidencialidad con respecto a los servidores. • Actualmente no cuentan con personas que se encarguen del manejo del análisis de datos.	
3. ¿Cuáles son los principales sistemas y tecnologías que utiliza la institución en su infraestructura de TI?	
• SGA. • Servidores HP.	• Equipo de seguridad FORTINET. • Equipo de comunicación CISCO.

4. ¿Cuáles son las medidas de seguridad implementadas actualmente en la red de la empresa y cómo se mantienen actualizadas?
• Configuración de reglas por medio del FORTINET. • Políticas de acceso a la red de forma lógica.
5. ¿Cómo se gestiona y responde a incidentes de seguridad en la red?
• Por medio del proveedor de red, el cual comunica de los incidentes. • CSIR, el cual se revisa mediante los servidores para poder las medidas adecuadas ante los incidentes que se presenten.
6. ¿Qué herramientas y tecnologías se utilizan para monitorear la seguridad de la red?
• FORTINET • Monitoreo PRTG • Aplicación PRTG

Anexo 2 Entrenamiento de algoritmo BBR

```python
def analisis_bbr():
    directorio = os.getcwd()
    carpeta = "dataset_traducido"
    archivo = "fraud_esp.csv"
    ruta_completa = os.path.join(directorio, carpeta, archivo)

    # Cargar tu conjunto de datos, por ejemplo, un archivo CSV con columnas 'texto' y 'spam'
    data = pd.read_csv(ruta_completa, encoding='ANSI')

    # Preprocesamiento de datos
    X_text = data['Text'].values
    y = data['Class'].values

    # Convertir etiquetas a valores numéricos (0 o 1)
    le = LabelEncoder()
    y = le.fit_transform(y)

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(X_text, y, test_size=0.2, random_state=42)

    # Modelo con embeddings preentrenados (Universal Sentence Encoder)
    hub_layer = hub.KerasLayer("https://tfhub.dev/google/universal-sentence-encoder/4", input_shape=[], dtype=tf.string, trainable=False)

    model = tf.keras.Sequential([
        hub_layer,
        Dense(16, activation='relu'),
        Dense(1, activation='sigmoid', kernel_regularizer='l2')
    ])

    # Guardar el modelo entrenado con los siguientes parámetros:
    # optimizer='adam': Algoritmo de optimización utilizado durante el entrenamiento
    # loss='binary_crossentropy': Función de pérdida utilizada para evaluar el rendimiento del modelo
    # metrics=['accuracy']: Métricas a evaluar durante el entrenamiento, en este caso, precisión
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

    # Imprimir el número total de datos a analizar
    total_datos_a_analizar = len(X_test)
    print(f"Número total de datos a analizar: {total_datos_a_analizar}")
```

```python
# Registrar el tiempo de inicio
start_time = time.time()

# Entrenar el modelo con los siguientes parámetros:
# X_train: Conjunto de datos de entrenamiento
# y_train: Etiquetas correspondientes al conjunto de entrenamiento
# epochs=1: Número de veces que el modelo recorrerá el conjunto de entrenamiento completo
# batch_size=32: Número de muestras utilizadas en cada actualización de los pesos del modelo
# validation_split=0.2: Proporción del conjunto de entrenamiento que se utilizará como conjunto de validación
model.fit(X_train, y_train, epochs=3, batch_size=32, validation_split=0.2)

# Guardar el modelo entrenado en la carpeta "MODELOS_ENTRENADOS"
modelos_entrenados_folder = "Modelos entrenados"
os.makedirs(modelos_entrenados_folder, exist_ok=True)
model.save(os.path.join(modelos_entrenados_folder, "modelo_BBR.h5"))

# Evaluar el modelo con el conjunto de prueba y obtener la precisión con el siguiente parámetro:
# X_test: Conjunto de datos de prueba
# y_test: Etiquetas correspondientes al conjunto de prueba
accuracy = model.evaluate(X_test, y_test)[1]
print(f'Precisión en el conjunto de prueba: {accuracy}')

# Predecir las etiquetas en el conjunto de prueba
y_pred = model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)  # Convertir las probabilidades a etiquetas binarias
```

Anexo 3 Entrenamiento de algoritmo SVM

```python
def analisis_svm():
    directorio = os.getcwd()
    carpeta = "dataset_traducido"
    archivo = "fraud_esp.csv"
    ruta_completa = os.path.join(directorio, carpeta, archivo)

    # Cargar tu conjunto de datos
    data = pd.read_csv(ruta_completa, encoding='ANSI')

    # Manejar valores NaN en el campo 'Message'
    data['Text'].fillna('', inplace=True)

    # Preprocesamiento de datos
    X = data['Text'].values
    y = data['Class'].values

    # Asegurarse de que hay al menos dos clases únicas
    clases_unicas = np.unique(y)
    if len(clases_unicas) < 2:
        raise ValueError("El conjunto de datos debe tener al menos dos clases únicas.")

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Crear un vectorizador TF-IDF (frecuencia de término - inversa de la frecuencia del documento)
    vectorizer = TfidfVectorizer(sublinear_tf=True, encoding='utf-8', decode_error='ignore', stop_words='english')

    # Medir el tiempo de entrenamiento
    start_time = time.time()
    X_train_vectorized = vectorizer.fit_transform(X_train)
    end_time = time.time()
    training_time = end_time - start_time

    # Medir el tiempo de predicción
    start_time = time.time()
    X_test_vectorized = vectorizer.transform(X_test)
    end_time = time.time()
    prediction_time = end_time - start_time
```

```python
# Crear el modelo SVM
modelo_svm = SVC()

# Configurar la búsqueda de cuadrícula
parametros_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['rbf']}
grid_search = GridSearchCV(modelo_svm, parametros_grid, refit=True, verbose=3, cv=3)
# param_grid: Diccionario con las combinaciones de parámetros a probar.
#   'C': Parámetro de regularización, controla la penalización por error en el entrenamiento.
#   'gamma': Coeficiente del kernel, controla el alcance de influencia de un solo ejemplo de entrenamiento.
#   'kernel': Tipo de kernel a utilizar.
# modelo_svm: Modelo se utilizará.
# refit=True: Ajustará el modelo al conjunto de entrenamiento completo.
# verbose=3: Proporciona información sobre el progreso de la búsqueda.
# cv=3: Estrategia de validación cruzada con 3 divisiones.

# Realizar la búsqueda de cuadrícula en los datos de entrenamiento
grid_search.fit(X_train_vectorized, y_train)

# Obtener el mejor modelo después de la búsqueda de cuadrícula
mejor_modelo = grid_search.best_estimator_

# Realizar predicciones en el conjunto de prueba
predicciones = mejor_modelo.predict(X_test_vectorized)

# Evaluar el rendimiento del modelo
precision = accuracy_score(y_test, predicciones)
informe_clasificacion = classification_report(y_test, predicciones)

# Imprimir resultados
print(f"Mejores parámetros después de búsqueda de cuadrícula: {grid_search.best_params_}")
print(f"Precisión del modelo: {precision}")
print("Informe de clasificación:\n", informe_clasificacion)
# Imprimir tiempos
print(f"Tiempo de entrenamiento: {format_time(training_time)}")
print(f"Tiempo de predicción: {format_time(prediction_time)}")
```