

PROYECTO DE TITULACIÓN

CARRERA DE SOFTWARE

TEMA: IMPLEMENTAR UN SISTEMA DE DETECCIÓN Y PREVENCIÓN DE INTRUSIONES ENFOCADO A ATAQUES DE INGENIERÍA SOCIAL DISEÑADO ESPECÍFICAMENTE PARA SU APLICACIÓN EN ENTORNOS DE TRANSACCIONES EN LÍNEA, A TRAVÉS DEL USO DE INTELIGENCIA DE AMENAZAS.

AUTORAS: NACIMBA CAIZA, MELANY VANESA
SANTAMARÍA ZAMBRANO, LISSETTE HELENA

TUTOR: ING. NAVAS MOYA, MILTON PATRICIO. MGTR

Latacunga, 16 de Agosto del 2024

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Reporte de Verificación de Contenido



CERTIFICADO DE ANÁLISIS

Trabajo de Integracion Melany_Lisseth

8% Textos sospechosos

Similitudes
1% similitudes entre comillas
< 1% entre las fuentes mencionadas
4% Idiomas no reconocidos
< 1% Textos potencialmente generados por la IA

Nombre del documento: Trabajo de Integracion Melany_Lisseth.docx
ID del documento: 827b0c4b0a8e1224fe8f0f6241c1bdd6a5228a9b2
Tamaño del documento original: 2,89 MB
Autores: []

Depositante: JOSÉ LUIS CARRILLO
Fecha de depósito: 15/8/2024
Tipo de carga: interface
Fecha de fin de análisis: 15/8/2024

Número de palabras: 17.282
Número de caracteres: 115.053

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario · El documento proviene de otro grupo	< 1%		
2	repositorio.uisek.edu.ec · 1 fuente similar	< 1%		
3	repositorio.uta.edu.ec | Repositorio Universidad Técnica de Ambato: Modelo de M...	< 1%		
4	Documento de otro usuario · El documento proviene de otro grupo	< 1%		
5	scielo.sld.cu | Principales mecanismos para el enfrentamiento al phishing en las r...	< 1%		

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repository.universidadean.edu.co	< 1%		
2	repositorio.espe.edu.ec | DEPARTAMENTO DE ELECTRICA Y ELECTRONICA	< 1%		
3	www.redalyc.org | Principales mecanismos para el enfrentamiento al phishing en ...	< 1%		
4	kupdf.net | Mineria de datos modelos y algoritmos.pdf - Free Download PDF	< 1%		
5	Tesis_Castillo_Chuquitarco-Antiplagio-Compilation_15-02-2023_.docx | Tes... · El documento proviene de mi biblioteca de referencias	< 1%		

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://burjcdigital.urjc.es/handle/10115/38000
2 https://sparning.com/blog/cyberattacks-2021-phishing-ransomware-data-breach-statistics/
3 https://docta.ucm.es/entities/publication/f9bbdc95-25c1-4467-94ef-53bca8e3f245
4 https://repositorio.uniandes.edu.co/server/api/core/bitstreams/de635c47-f374-41dc-b4f0-ac5dc0b2cc97/content
5 https://data.europa.eu/doi/10.2824/622757

Ing. Navas Moya, Milton Patricio. Mgtr.

C. C.: 0502029275

Departamento de Ciencias de la Computación

Carrera de Software

Certificación

Certifico que el proyecto de titulación: "**Implementar un sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas**" fue realizado por las señoritas **Santamaría Zambrano, Lissette Helena y Nacimba Caiza, Melany Vanesa;** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 16 de agosto del 2024

Ing. Navas Moya, Milton Patricio. Mgtr.

C. C.: 0502029275

Departamento de Ciencias de la Computación

Carrera de Software

Responsabilidad de Autoría

Nosotras, **Santamaría Zambrano, Lissette Helena y Nacimba Caiza, Melany Vanesa**, con cédulas de ciudadanía n° 2300472608 y 1726595422, declaramos que el contenido, ideas y criterios del proyecto de titulación: "**Implementar un sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas**" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 16 de agosto del 2024

Santamaría Zambrano, Lissette Helena

C.C.: 2300472608

Nacimba Caiza, Melany Vanesa

C.C.: 1726595422

Departamento de Ciencias de la Computación

Carrera de Software

Autorización de Publicación

Nosotras **Santamaría Zambrano, Lissette Helena y Nacimba Caiza, Melany Vanesa**, con cédulas de ciudadanía n° 2300472608 y 1726595422, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el proyecto de titulación: "**Implementar un sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 16 de agosto del 2024

Santamaría Zambrano, Lissette Helena

C.C.: 2300472608

Nacimba Caiza, Melany Vanesa

C.C.: 1726595422

Dedicatoria

El presente trabajo me gustaría dedicárselo a todas aquellas personas que han sido un pilar en mi vida, cuya compañía y apoyo fueron fundamentales para llegar hasta aquí. En especial a mis padres: a mi madre, Erlinda Zambrano, por su sabiduría y amor incondicional que me han iluminado siempre que sentía perder el camino y de quién he obtenido la fuerza para seguir adelante. A la memoria de mi padre, Walter Santamaría, a quien agradezco profundamente los años que compartimos, sus enseñanzas y valores que me han hecho ser lo que soy, y el infinito amor que siempre me brindó. A mis hermanos, Karen y Walter Santamaría, por su apoyo incondicional y sus palabras de aliento que reconfortan el alma.

También quisiera dedicárselo a mis compañeros de estudio, con quienes he compartido mis días universitarios y seguramente será un preciado recuerdo que me acompañará a lo largo de mi vida.

Por último, también se lo dedico a mis profesores, a quienes agradezco de todo corazón el conocimiento impartido, así como su guía y apoyo que sin duda han repercutido en formación académica y profesional.

Lissette Helena Santamaría Zambrano

Ecuador, agosto 2024

Dedicatoria

Dedicado con profundo amor a las personas más importantes en mi vida, a mis padres Soledad y Marco quienes han sido mi fuente de inspiración y mi motor para seguir adelante. Ustedes han sido un claro ejemplo de lucha y constancia, les agradezco por todo el esfuerzo que hicieron por mí, a pesar de estar lejos de casa nunca me sentí sola.

A mi otra mitad, mi confidente y compañera de aventuras mi hermana Mishell gracias por siempre ser la primera en creer en mí, por ser la que me empuja a salir de mi zona de confort y a ser mejor cada día, por ti cada día me esfuerzo en ser mejor persona, gracias por siempre estar a mi lado cuando siento que no puedo más por escucharme y siempre estar pendiente de mí. Este logro es de los cuatro porque sin su apoyo no hubiera conseguido llegar a la meta los amo con todo mi corazón.

A mis abuelitos que sé que desde el cielo me siguieron bendiciendo y me guiaron para que el día de hoy puedo escribir estas líneas, ustedes han dejado una huella imborrable en mi corazón y a pesar de su partida siempre estarán presentes en cada uno de mis logros. A Luna Tony porque gracias a ellos el estrés de la universidad fue más llevadero. Y finalmente pero no menos importante a mi Dori tu presencia en mi vida es un regalo invaluable, ser una luz en mis días grises.

Nacimba Caiza, Melany Vanesa

Ecuador, agosto 2024

Agradecimiento

Antes que nada, quiero dar gracias Dios por ser mi fuerza espiritual. Su gracia divina me ha cuidado y protegido cuando más indefensa me he sentido y se ha convertido en mi fortaleza en los momentos de dificultad.

También agradezco infinitamente a mi entrañable núcleo familiar, con quienes he podido superar los buenos y los malos momentos; y en quienes encuentro paz y tranquilidad. Su confianza y transigencia me han motivado a seguir adelante.

A mis amigos, por su sincera amistad, cuya compañía ha hecho menos solitario este camino. Quiero agradecer especialmente a Evelyn Serpa, quien aún en la distancia, ha estado presente en cada momento, brindándome su apoyo y cariño de manera constante. A Alexander Quiñonez, una persona muy cálida, cuya amabilidad y predisposición a ayudar nunca han faltado, siempre dispuesto a dar una mano amiga.

Al Ing. Patricio Navas, mi tutor de tesis, por su orientación a lo largo del desarrollo de este trabajo. Gracias por su compromiso y su guía constante.

Por último, quiero expresar un sincero agradecimiento a todas y cada una de las personas, que han formado parte de mi vida y han ayudado en mi crecimiento personal y académico.

Lissette Helena Santamaría Zambrano

Ecuador, agosto 2024

Agradecimiento

En primer lugar, agradezco a Dios por haberme permitido culminar con éxito mi carrera universitaria. A mis padres por el esfuerzo que hacen día a día para darnos lo mejor a mi hermana y a mí. No encuentro palabras para expresarles todo el agradecimiento que tengo hacia ustedes por todo lo que han hecho por mí.

A mi hermana por toda la ayuda que me a brindado, por siempre estar dispuesta a darme su mano cuando lo necesito.

A mí mejor amigo quien ha estado a mi lado de principio a fin, gracias por esta linda amistad y por hacer que mis días lejos de mi familia no sean tan solitarios, por tu apoyo incondicional y tus consejos.

Al Ing. José Luis Carrillo por permitirme se parte de este proyecto de investigación por su apoyo incondicional. A mi tutor de tesis el Ing. Patricio Navas por sus consejos a lo largo de este trabajo. Y a todos mis profesores por todo el conocimiento que me han brindado en el aula de clases contribuyendo de alguna manera a este trabajo.

Nacimba Caiza, Melany Vanesa

Ecuador, agosto 2024

Índice de contenido

Carátula ..... **I**

Reporte de Verificación de Contenido ..... **II**

Certificación ..... **III**

Responsabilidad de Autoría ..... **IV**

Autorización de Publicación ..... **V**

Dedicatoria ..... **VI**

Dedicatoria ..... **VII**

Agradecimiento ..... **VIII**

Agradecimiento ..... **IX**

Índice de contenido ..... **X**

Índice de figuras ..... **XIV**

Índice de tablas ..... **XV**

Resumen **16**

Abstract **17**

1. Introducción **18**

Antecedentes **18**

Planteamiento del problema **19**

Justificación **20**

Objetivos **21**

Objetivo General **21**

Objetivos Específicos **21**

Aproximación de la solución **22**

Estado del Arte **22**

2. Marco Teórico **27**

Sistema de transacciones en línea **27**

Ingeniería Social **27**

Phishing **28**

Sistema de Detección y Prevención de Intrusos **30**

Sistema de detección de intrusos **30**

Sistema de prevención de intrusos **31**

Machine Learning **31**

Inteligencia de Amenazas **33**

Metodología Scrum **35**

Herramientas **36**

Medidas de Evaluación **37**

3. Metodología **40**

Metodología Scrum **40**

Definición de Roles **40**

Release Planning **41**

Captura de Requisitos (Historias de Usuario) **41**

Prioridad de las Historias de Usuario **42**

Product Backlog **43**

Lista de Tareas **43**

Diseño **44**

Diagrama de la Arquitectura Lógica **45**

Definición de las tecnologías a usar **46**

Capa de Presentación **46**

Capa de Aplicación **47**

Capa de Servicios **47**

Capa de Datos **47**

Capa de Red **48**

Capa de Infraestructura **48**

Sprint 01: Entrenamiento del modelo machine learning **51**

Historia de Usuario detallada **51**

Sprint Backlog 01 ..... **52**

Resultados del Sprint 01 ..... **53**

Sprint 02: Integración con las APIs de inteligencia de amenazas **59**

Historia de Usuario detallada **59**

Sprint Backlog 02 ..... **60**

Resultados del Sprint 02 ..... **61**

Sprint 3: Dashboard administrativo **65**

Historia de Usuario Detallada **65**

Sprint Backlog 03 ..... **66**

Resultados del Sprint 3 ..... **68**

Sprint 04: Implementación de Alertas y/o notificaciones **71**

Historia de Usuario detallada **71**

Sprint Backlog 04 ..... **72**

Resultados del Sprint 04 ..... **73**

Validación del Sistema **74**

4. Resultados **80**

Análisis de resultados **92**

5. Conclusiones y Recomendaciones **94**

Conclusiones **94**

Recomendaciones **96**

6. Referencias bibliográficas **97**

7. Anexos **106**

Índice de figuras

Figura 1 *Diagrama de la metodología propuesta* 35

Figura 2 *Diagrama de la arquitectura lógica del sistema* 46

Figura 3 *Diagrama de la arquitectura lógica del sistema con las tecnologías a usar* 49

Figura 4 *Diagrama de Arquitectura Física* 50

Figura 5 *Preprocesamiento de texto* 54

Figura 6 *Vectorización del texto* 55

Figura 7 *Extracción de características de URL* 56

Figura 8 *Repositorio de Kaggle* 57

Figura 9 *Ejemplo de información proporcionada por VirusTotal* 61

Figura 10 *Ejemplo de búsqueda de una IP en la API AbuseIPDB* 62

Figura 11 *API de reputación de URL de Apivoid* 63

Figura 12 *Integración de la API Virus Total* 64

Figura 12 *Integración de la API Virus Total (continuación)* 65

Figura 13 *Dashboard del proyecto* 68

Figura 14 *Interfaz del Administrador* 69

Figura 15 *Información detallada del correo* 70

Figura 16 *Botón reporte de amenazas* 71

Figura 17 *Configuración websocket* 74

Figura 18 *Configuración del Servidor DNS en Windows Server* 75

Figura 19 *Configuración de Postfix* 76

Figura 20 *Script de filtrado y logs del servidor de correo* 77

Figura 21 *Emisión de Alerta en el dashboard* 77

Figura 22 *Proceso de comunicación entre los servidores* 78

Figura 23 *Correo de phishing enviado a la carpeta de spam* 78

Índice de tablas

Tabla 1 *Modelos de Machine Learning* 32

Tabla 2 *Matriz de Confusión* 38

Tabla 3 *Fórmulas de las métricas de evaluación* 39

Tabla 4 *Asignación de Roles en Scrum* 40

Tabla 5 *Historias de usuario* 41

Tabla 6 *Prioridad de las historias de usuario* 42

Tabla 7 *Product Backlog del proyecto* 43

Tabla 8 *Lista de Tareas del proyecto* 43

Tabla 9 *Estimación de las tareas del producto backlog* 44

Tabla 10 *Historia de usuario detallada para el entrenamiento del modelo* 51

Tabla 11 *Sprint Backlog 01* 52

Tabla 12 *Resultados de evaluación de los modelos machine learning para URL* 58

Tabla 13 *Resultados de evaluación de los modelos machine learning para correo* 58

Tabla 14 *Historia de usuario detalladas de las APIs de inteligencia de amenazas* 59

Tabla 15 *Sprint Backlog 02* 60

Tabla 16 *Historia de usuario para el dashboard administrativo* 66

Tabla 17 *Sprint Backlog 03* 67

Tabla 18 *Historia de usuario detalladas de alertas sobre ataques de phishing* 72

Tabla 19 *Sprint Backlog 04* 72

Tabla 20 *Resultado de pruebas Phishing en URL* 80

Tabla 21 *Matriz de Confusión de la URL* 85

Tabla 22 *Métricas de evaluación calculadas del modelo Random Forest* 86

Tabla 23 *Matriz de Confusión cuerpo del correo* 87

Tabla 24 *Métricas de evaluación en Random Forest para el cuerpo del correo* 87

Tabla 25 *Resultado de las Apis de Inteligencia de Amenazas* 89

Resumen

El presente proyecto aborda la implementación de un sistema de detección y prevención de intrusos (IDPS) diseñado específicamente para enfrentar ataques de ingeniería social en entornos de transacciones en línea. Este sistema integra inteligencia de amenazas para identificar de manera precisa patrones maliciosos en IPs, dominios, URLs y cuerpos de correos electrónicos. Tras evaluar diversos modelos de machine learning, se seleccionó Random Forest por su destacada precisión y rendimiento en la detección de ataques de phishing. La combinación de técnicas avanzadas de machine learning con inteligencia de amenazas permite al sistema ofrecer una detección de phishing con alta precisión y una velocidad de procesamiento optimizada. Esta integración no solo mejora la exactitud en la identificación de amenazas, sino que también asegura una respuesta ágil y eficiente ante posibles ataques. El sistema se distingue por su interfaz intuitiva y fácil de usar, que facilita la visualización y gestión de alertas en tiempo real. Los usuarios reciben notificaciones instantáneas sobre posibles amenazas, lo que permite una respuesta rápida y eficiente. Las alertas se presentan en un dashboard claro y accesible, donde también tienen la opción de reportarlas y almacenarlas en una base de datos para su análisis y revisión futura. Este enfoque integral no solo mejora la capacidad de detección de ataques, sino que también optimiza la respuesta frente a amenazas emergentes, fortaleciendo así la seguridad en transacciones en línea.

Palabras clave: Inteligencia de amenazas, ingeniería social, phishing, machine learning, análisis de datos.

Abstract

This project addresses the implementation of an intrusion detection and prevention system (IDPS) specifically designed to address social engineering attacks in online transaction environments. This system integrates threat intelligence to accurately identify malicious patterns in IPs, domains, URLs and email bodies. After evaluating several machine learning models, Random Forest was selected for its outstanding accuracy and performance in detecting phishing attacks. The combination of advanced machine learning techniques with threat intelligence enables the system to provide highly accurate phishing detection with optimised processing speed. This integration not only improves the accuracy of threat identification, but also ensures a fast and efficient response to potential attacks. The system is distinguished by its intuitive and easy-to-use interface, which facilitates real-time viewing and management of alerts. Users receive instant notifications of potential threats, enabling a fast and efficient response. Alerts are presented in a clear and accessible dashboard, where they also have the option to report and store them in a database for future analysis and review. This comprehensive approach not only improves the ability to detect attacks, but also optimises the response to emerging threats, thus strengthening the security of online transactions.

Keywords: Threat intelligence, social engineering, phishing, machine learning, data analytics.

1. Introducción

Antecedentes

En una era digital donde el internet cuenta con millones de usuarios en todo mundo y la tecnología ha transformando por completo la forma en la que las personas se comunican, interactúan, trabajan y viven.[1]

Las transacciones en línea se han catapultado como parte esencial en la vida cotidiana, miles de personas a diario realizan pagos de facturas, compras o transacciones de todo tipo. Provocando un aumentando desmesurado en el comercio electrónico. Sin embargo, este progreso ha venido acompañado de un crecimiento exponencial de los ciberataques los cuales están en constante evolución con el único objetivo de vulnerar el sistema y acceder a la privacidad del usuario o empresa.[2]

Los ataques más comunes en las transacciones en línea son: phishing, el malware y los ataques de fuerza bruta. Las cuales se encargan de vulnerar la privacidad del usuario provocando daños financieros, robos de identidad e incluso un decremento de usuarios en el caso del comercio electrónico. [3]

De acuerdo a la investigación realizada por Nadezhda Demidoca de Kaspersky Lab, el Phishing se considera una de las amenazas más importantes a nivel mundial, registrando ataques a sitios web desde el 2017.[4]

Hoy en día es considerado el ataque cibernético más común, sencillo y eficaz para el robo de información. A través de técnicas de ingeniería social, el Phishing se ha catapultado como una de las técnicas más usadas por los ciberdelincuentes.[5]

Según [6], se arrojó que cerca del 90% de robo de datos en 2021 fueron mediante ataques de Phishing. En el mismo año IBM dio a conocer en su informe Cost of a Data Breach Report que este tipo de ataque es considerado uno de los más costosos, llegando a causar grandes pérdidas económicas para los usuarios o empresas.[6]

En este ámbito, la seguridad informática es una pieza fundamental en la protección de la información privada y financiera de usuarios e instituciones. [5] Para mitigar estos ataques se han creado los sistemas de detección y prevención de intrusiones o (IDS/IPS en inglés) los cuales funcionan como un muro de protección indispensable para identificar y mitigar posibles amenazas que causen daño al sistema.[7] Sin embargo, estos sistemas tradicionales a pesar de su efectividad, llegan a ser deficientes contra ataques de phishing.[8]

A día de hoy, se han realizado diferentes estudios e investigaciones de como detectar, prevenir y/o mitigar los ataques de Phishing, dando como resultado que la clave está en el uso de modelos y/o algoritmos de Machine y técnicas de Procesamiento de Lenguaje Natural, incluso combinar estas técnicas. [9]

Planteamiento del problema

En la actualidad estos sistemas de detección y prevención de intrusos se han visto vulnerados, ya que se enfrentan a nuevas y sofisticadas amenazas que ponen en riesgo su eficiencia, la principal limitante de los sistemas tradicionales es su dependencia de firmas y patrones de ataque predefinidos lo cual retrasa la detección e incluso puede omitir nuevas amenazas ya que no se alinea a los patrones predefinidos. Por otra parte, los atacantes se mantienen en constante evolución desarrollando nuevas técnicas de ataque que les permitan vulnerar el sistema. Un claro ejemplo de ello son los ataques de ingeniería social como el phishing llegando a ser extremadamente personalizado y dirigido, lo cual entorpece la detección de amenazas ya que estos sistemas se basan en reglas estáticas.[7]

Los sistemas IDS e IPS son susceptibles a provocar falsos positivos lo cual provoca una sobrecarga en los administradores de seguridad con alertas irrelevantes, generando un retraso en la detección de amenazas. En un entorno de transacciones en línea que se utiliza las 24 horas la rapidez y precisión son vitales. De lo contrario un retraso en la detección de una amenaza pueda dar como resultado grandes pérdidas económicas[8]

Por esta razón, surge la necesidad de incorporar en los sistemas IDS e IPS la inteligencia de amenaza, ya que permite que los sistemas se actualicen en tiempo real con la última información que se tenga de nuevas técnicas de ataque. Al integrar la inteligencia de amenazas en los sistemas tradicionales reduce la probabilidad de generar falsos positivos además de mejorar la detección de ataques sofisticados que al no contar con la inteligencia de amenazas pasaban desapercibidos. [8]

Justificación

La creciente amenaza en la economía digital con respecto a los ataques de ingeniería social, como el pretexting, el baiting, el farming, el vishing y el phishing representan un grave problema para los sistemas de seguridad en entornos de transacciones en línea. Estas formas de ataque explotan las debilidades humanas y son difíciles de detectar por los métodos habituales de seguridad. En consecuencia, las soluciones actuales no son tan efectivas para identificar y minimizar estos tipos de ataque.

Los sistemas IPS (Sistemas de Prevención de Intrusiones) e IDS (Sistemas de Detección de Intrusiones) actuales, aunque son eficaces, carecen de la capacidad para adaptarse fácilmente a las nuevas tácticas de ingeniería social que están en constante evolución. Estas soluciones tradicionales presentan problemas para identificar patrones de ataque complejos, puesto que se basan en firmas y reglas predefinidas, dando como resultado altas tasas de falsos positivos.

Para superar estas restricciones, es necesario emplear enfoques más avanzados que ayuden a mejorar la capacidad de respuesta. Por lo tanto, la integración de IDPS e inteligencia de amenazas es una solución prometedora dentro de este contexto. La implementación de inteligencia de amenazas podría ayudar a mejorar los modelos de machine learning para la detección de ataques de ingeniería social, de tal forma que se puedan reducir la cantidad de falsos positivos y garantizando que el sistema esté actualizado frente a nuevas formas de ataque, ofreciendo una defensa robusta y proactiva.

El desarrollo de este sistema enriquece el conocimiento académico y práctico en el campo de la ciberseguridad. También contribuye a llenar los vacíos que existen actualmente en la literatura acerca de temas como la integración de inteligencia de amenazas e IDPS, en lo que se refiere a ataques de ingeniería social.

Con las conclusiones de este trabajo, se espera que sirvan como inspiración para futuras innovaciones en esta línea de investigación. De forma que se pueda mejorar el conocimiento que se tiene respecto a este campo de estudio.

Objetivos

Objetivo General

Implementar un sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social específicamente diseñado para su aplicación en entornos de transacciones en línea, mediante el uso de inteligencia de amenazas.

Objetivos Específicos

- Investigar y analizar las definiciones, conceptos clave y terminología específica sobre sistemas de detección y prevención de intrusos, machine learning, ataques de ingeniería social, y el uso de inteligencia de amenazas en la ciberseguridad.
- Desarrollar prototipos iniciales de un sistema de detección y prevención de intrusos, en un ambiente controlado para ataques de ingeniería social, usando inteligencia de amenazas.
- Optimizar un servidor que gestione datos de correos maliciosos para mejorar la eficiencia del sistema.
- Analizar la eficiencia del sistema desarrollado mediante la aplicación de métricas de evaluación.

Aproximación de la solución

El propósito del presente proyecto es implementar un Sistema de Detección y Prevención de Intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas. Por ello, se propuso el uso de modelos y/o algoritmos de Machine Learning, con el propósito de automatizar el proceso de detección y prevención de ataques de ingeniería social en transacciones en línea. El uso de estos modelos y/o algoritmos de Machine Learning pretende brindar una alta precisión en la detección y prevención de intrusiones. Lo cual permitirá agilizar la de detección y prevención de correos electrónicos fraudulentos ofreciendo así una solución eficiente en la lucha contra los ataques de ingeniería social. En definitiva, este sistema propuesto podría ser una solución viable para afrontar el problema actual de robo de información a través correos electrónicos maliciosos. Al automatizar la detección y prevención de intrusiones, podría reducir significativamente el riesgo de caer en los ataques de ingeniería social y proteger la información personal y confidencial de los usuarios.

Estado del Arte

En esta sección, se lleva a cabo una revisión de diversos artículos, tesis y documentos relevantes a nuestro tema de estudio. El propósito es establecer un marco sólido para nuestra investigación, identificando las tendencias actuales y brechas en la literatura existente. Además, se realiza un análisis comparativo para identificar las principales diferencias entre los estudios previos y nuestro proyecto, resaltando así las contribuciones que nuestra investigación aporta al campo de estudio.

Uno de los estudios más recientes en el campo de la detección de phishing es el de [10], titulado “DEPHIDES: Deep Learning Based Phishing Detection System”. Este artículo presenta el desarrollo de un sistema de detección de ataques de phishing mediante la aplicación de algoritmos de aprendizaje profundo como RNN, BRNN y ATT. Los autores recopilan datos de URLs legítimas y maliciosas para entrenar los modelos y que sean capaces

de identificar sitios web de phishing, destacando la importancia de la calidad de los datos y la eficiencia del sistema en entornos de alto tráfico. DEPHIDES se diferencia de nuestro proyecto en que esta solo se enfoca en la detección de phishing mediante aprendizaje profundo, mientras que nuestro enfoque incluye prevención de ataques usando modelos y/o algoritmos machine learning e inteligencia de amenazas.

Por otro lado, en [11] con el tema “Cyber Threat Intelligence Based Malicious URL Detection Model Using Ensemble Learning”, aborda un aspecto crucial en ciberseguridad que es la detección de URLs maliciosas. Se muestran dos enfoques: la detección basada en características, que analiza de forma automática aspectos de las URLs como longitud y dominio usando técnicas como N-gram y TF-IDF; y la detección basada en listas negras, que usa bases de datos centralizadas y análisis de expertos. Se presenta un modelo llamado CTI-MURLD, que mejora la detección de URLs maliciosas mientras combina características de URLs, información de Whois y datos de inteligencia de amenazas de Google. También maneja técnicas de machine learning como random forest y perceptrón multicapa, para incrementar la precisión. La principal diferencia con nuestro proyecto es que CTI-MURLD se enfoca en la parte de detección de URLs, a diferencia de nuestro proyecto que abarca identificación de IPs, dominios y el contenido de correos electrónicos para una solución más completa.

De igual forma, en [12], denominado “An intelligent cyber security phishing detection system using deep learning techniques”, se centra en la detección de phishing mediante algoritmos de aprendizaje automático para frenar la amenaza de los correos electrónicos de phishing. Para el modelo propuesto, se usaron siete algoritmos diferentes, de los cuales se evaluaron métricas como precisión, recall y F-score en varios conjuntos de datos. Las conclusiones resaltan la eficacia del algoritmo mejorado Boosted Decision Tree para identificar los correos electrónicos con phishing, y enfatiza la importancia de la selección adecuada de las características y señala posibilidades de mejora continua. Aunque tanto el estudio como nuestro proyecto están enfocados en la detección de phishing en correos electrónicos, nuestro

sistema se centra en los correos de transacciones en línea, tratando las particularidades de seguridad en dichos entornos.

Por otra parte en [13] titulado “Machine learning model in the detection of phishing websites”, presenta un modelo para detectar sitios web de phishing mediante machine learning, usando características de URL, código fuente y sitios web de amenaza. Se probo este modelo con 30 características de 11 055 sitios web para entrenar tres modelos: Random Forest, Extra Tree y Decision Tree. Entre estos, al final se seleccionó el modelo Random Forest, debido a que mostro un rendimiento con una precisión del 97,56%, superando al resto de modelos. En contraste, nuestro sistema no solo abarca la detección de URLs de phishing, sino que también incluye medidas de prevención en entornos de transacciones en línea.

En [14] con el tema “Implementation of Intrusion Detection Systems in order to Detect Phishing in the Banking Industry”, se presenta un trabajo enfocado en la implementación de un sistema de detección de intrusos (IDS) para proteger los sistemas de información bancaria frente al phishing. Se propone un sistema PHISHING.IDS + ANFIS (sistema de inferencia neuro-difuso adaptativo) mediante el entorno de programación Matlab. Los resultados muestran mejoras en la detección de fraudes, contribuyendo a la seguridad de la información bancaria. Este estudio se concentra en usar el enfoque neuro-difuso para proteger los sistemas bancarios, mientras que nuestro proyecto emplea inteligencia de amenazas para abordar de manera integral los ataques de ingeniería social.

En el artículo “An Explainable Multimodal Hierarchical Attention Model for Developing Phishing Threat Intelligence” [15], se propone un modelo de atención jerárquica multimodal (MMHAM) para la detección de sitios de phishing que se entrena en tres modalidades principales del contenido del sitio web: URL, información textual y diseño visual. En las pruebas, MMHAM mostró una mejora en la detección de phishing y ha proporcionado un sistema de interpretación jerárquica que permite el desarrollo de inteligencia de amenazas de phishing para informar de sitios de phishing a diferentes niveles. A diferencia de nuestro

enfoque, que se centra en la detección y prevención de ataques de phishing, este estudio está orientado únicamente a la detección de phishing mediante el análisis multimodal.

Asimismo en [16] titulado “Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection”, presenta la plataforma ORISHA, que sirve para el intercambio de información y concienciación, permitiendo la colaboración entre los sistemas de detección de amenazas y otros componentes de concienciación. ORISHA recibe apoyo de una plataforma de Inteligencia de amenazas, la cual facilita la comunicación entre las diferentes capas de detección de varias organizaciones. Los sistemas de detección de amenazas comparten la información, lo que mejora la precisión de las predicciones.

Otro estudio relevante es [17] denominado “Intrusion detection based on phishing detection with machine learning”, el cual presenta un método híbrido de selección conjunta de características (HEFS) para sistemas de detección de phishing basados en machine learning. El método HEFS usa un algoritmo gradiente de función de distribución acumulativa (CDF-g) con el cual crea subconjuntos de características primarias, para luego usarlas en un ensamble de perturbación de datos y crear un subconjunto de características secundarias. Los resultados de HEFS demuestran una mejora en la detección de URL de phishing.

Por otro lado, en [18] con el tema “OFS-NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network”, nos presenta un modelo OFS-NN para detectar sitios web de phishing, donde se combina la selección optima de características con redes neuronales. Para evaluar el impacto de las características sensibles en la detección de phishing, se introduce un nuevo índice y el valor de la validez de las características (FVV). El algoritmo basado en el FVV selecciona las características optimas, solucionando así el problema del sobreentrenamiento de las redes neuronales.

Por último, tenemos el articulo “Artificial Intelligence based Cyber Security Threats Identification in Financial Institutions using Machine Learning Approach” [19], que propone un sistema que use inteligencia artificial para la identificación de ciber amenazas en instituciones

financieras, empleando enfoques de machine learning. Al utilizar tecnologías de IA como lo es el procesamiento del lenguaje natural, algoritmos y sistemas de razonamiento automatizado, los bancos pueden analizar de mejor manera los riesgos y afrontarlos. Los algoritmos de machine learning ayudan a mejorar la identificación de anomalías en los datos que podrían significar una amenaza a la seguridad de las empresas financieras.

Las investigaciones en detección de phishing y ciberamenazas han tenido avances significativos respecto a la integración de técnicas de machine learning y deep learning, como lo son Random Forest, redes neuronales convolucionales y sistemas de inferencia neurofuzzy; que proporcionan altos niveles de precisión en la detección de amenazas.

Una de las lagunas existentes y que se necesita de más desarrollo es la colaboración e intercambio de información entre los sistemas de detección de amenazas y los distintos niveles de las organizaciones. La mayoría de los modelos actuales se centran en el análisis de un conjunto limitado de características, como URL, información textual, entre otros. Por lo que, se requiere de una comprensión más profunda de la interacción entre las distintas modalidades para mejorar la detección.

2. Marco Teórico

En esta sección se presenta la investigación teórica necesaria para abordar el desarrollo de un sistema de detección y prevención de intrusos orientado a mitigar ataques de ingeniería social. Se expondrán los fundamentos y terminología relacionada a temas de inteligencia de amenazas, sistemas de detección y prevención, machine learning, entre otros.

Para respaldar esta investigación, se usaron bases de datos bibliográficas confiables como lo son: SCOPUS, Data Science e IEEE, las mismas que proporcionan acceso a literatura relevante y actualizada en el campo de la ciberseguridad.

Sistema de transacciones en línea

Una transacción en línea es un intercambio de información de dos o más usuarios a través de medios electrónicos conectados a internet.[20]

Estas transacciones permiten la compra, venta, pago de facturas entre otros servicios, desde cualquier parte del mundo sin importar la hora o el lugar del cual se quiera realizar la transacción. Evitando que las personas tengan que acercarse de manera presencial a realizar estas actividades beneficiando a una gran parte de la población que tiene dificultades para movilizarse.[21]

Existen múltiples amenazas como el malware, ransomware, botnets, cryptojacking, phishing los cuales ponen en riesgo la seguridad de las transacciones en línea, una de las que más se destaca en la actualidad es la ingeniería social. [3], [8]

Ingeniería Social

Actualmente la ingeniería social es una de las técnicas más usadas por los ciberdelincuentes. Su habilidad para manipular el comportamiento humano mediante habilidades sociales y técnicas psicológicas, la ha catapultado como una herramienta sencilla y eficaz en el robo de información confidencial [22]. Según [23], la finalidad de un atacante que aplica ingeniería social es el de explotar al eslabón más débil de la organización, el usuario.

Dependiendo de su osadía, el atacante puede utilizar herramientas tecnológicas o incluso los encuentros cara a cara para obtener la información que necesita. Por ello es necesario tener en cuenta que no solo los usuarios son los que se ven expuestos a ataques de ingeniería social, el personal de seguridad informática es igualmente vulnerables.

A nivel global, y en Ecuador no es una excepción, la digitalización acelerada ha traído consigo vulnerabilidades emergentes. Según datos del [24]; Ecuador y otros países de América Latina han experimentado un aumento en los incidentes de ciberseguridad, específicamente en ataques de ingeniería social, lo que demuestra la urgencia de sistemas que puedan proteger nuestra información digital. [25]

Los ataques de ingeniería social se consideran ataques de piratería no técnicos, ya que la mayoría de los ataques se ejecutan sin ningún conocimiento técnico. Es posible que un ingeniero social no tenga ningún conocimiento técnico de piratería informática ni el uso de herramientas sofisticadas para ejecutar un ataque. La manipulación humana se realiza en persona o mediante un medio como un teléfono, correo electrónico o VOIP. [26]

Dentro de la ingeniería social, existen varias tácticas que los atacantes utilizan. Entre las más comunes se encuentran el farming, el baiting, el pretexting y el phishing. es el phishing. El phishing es una técnica particularmente notable por su uso frecuente y efectividad en el campo, utilizada por los estafadores para obtener datos confidenciales de los usuarios alegando que proceden de fuentes conocidas. [27]

Phishing

El phishing en la actualidad es uno de los ataques cibernéticos más comunes a nivel mundial. Este ataque se basa principalmente en el uso técnicas de ingeniería social o métodos técnicos en donde su principal objetivo es atacar al eslabón más débil el usuario, haciéndose pasar por entidades legitimas, logran que el usuario revele información personal como: contraseñas, nombre de usuarios, correo electrónico, entre otros.[28]

Por ejemplo, en un correo electrónico phishing, el remitente puede alentar a los usuarios a proporcionar información privada bajo premisas falsas. Los correos electrónicos de phishing también se han utilizado con frecuencia en instituciones financieras y fraude al consumidor. [27]

Lo cual evidencia los grandes daños que provoca enfrentarse a un ataque de Phishing, que van desde la suplantación de identidad hasta grandes pérdidas económicas para las empresas.[28]

A continuación, se muestran las técnicas de phishing más comunes:

Sitio web falsificado: Este tipo de ataque se basa en que el atacante crea un sitio web falso que parece ser legitimo. El usuario será redireccionado a este sitio web falso en donde a través de anuncios, formularios, entre otros el phisher logra obtendrá información personal del usuario.[28]

Vishing y SMishing: Este tipo de ataque se basa en que phisher a través de llamadas o mensajes se hace pasar por alguien de confianza para engañar al usario. Con la finalidad de obtener contraseñas, pin, o cualquier información que vulnere la privacidad del usuario.

Soshing, phishing en las redes sociales: Este tipo de ataque se basa en que los ciberdelincuentes por medio de las redes sociales vulneran la privacidad usuario a través de secuestro de cuentas, suplantación de identidad, estafas y difusión de malware.[28]

Spear Phishing: Este tipo de ataque se basa en la elaboración de múltiples correos genéricos dirigidos a un grupo en específico pueden ser usuarios o empresas para asegurar el éxito de este ataque el remitente del correo es una persona conocida por la víctima. A diferencia de los otros ataques de phishing este ataque necesita un planificación y preparación minuciosa.[28]

Correo electrónico de phishing: Este tipo de ataque es el más común y se basa en el envió de correos electrónicos desde una fuente poco fiable a diferentes usuarios de manera aleatoria. Este tipo de correos alegan ser personas o entidades bancarias en las que el usuario confía incentivándolo a ingresar ciertos datos personales que vulneran su privacidad. [28]

Tomando en cuenta que el tipo de ataque más común es el correo electrónico de phishing hemos decidido utilizar esta técnica para el desarrollo de nuestro sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas.

Sistema de Detección y Prevención de Intrusos

Los sistemas de detección y prevención de intrusos o (IDPS en inglés) se encarga de monitorear el tráfico de la red para detectar y mitigar cualquier tipo de amenaza que cause daño al sistema.[29]

Sistema de detección de intrusos

Un sistema de detección de intrusiones (IDS en inglés) se encarga de la supervisión del flujo que navega por la red, permitiendo detectar movimientos sospechosos que puedan generar alguna amenaza. Los IDS se clasifican en diferentes grupos dependiendo de sus características como, por ejemplo: tipo de Detección de anomalías y Detección de firmas o reglas, el tipo Ubicación se divide en Host-based IDS y Network-based IDS, El tipo Estructura se divide en Decentralized IDS y Centralized IDS, El tipo Comportamiento se divide en Pasivo y Activo.[30]

En este trabajo vamos a utilizar la detección de anomalías para desarrollar el sistema de detección y prevención de intrusiones enfocado a ataques de ingeniería social diseñado específicamente para su aplicación en entornos de transacciones en línea, a través del uso de inteligencia de amenazas.

Detección de anomalías: En este tipo de detecciones es imprescindible el uso de metodologías estadísticas que permiten la detección de comportamientos anómalos. El sistema básico se fundamenta únicamente en las faltas identificadas mediante el uso de reglas ya establecidas. El sistema intermedio es aquel que, en base a un perfil de actividad, determinado

por el comportamiento y el desempeño de un usuario, usa métricas y estadísticas para evaluar anomalías. Finalmente, el sistema más sofisticado se basa en el aprendizaje automático el cual permite diferenciar un comportamiento común de uno inusual.[30]

Sistema de prevención de intrusos

Los Sistemas de prevención de intrusos (IPS en inglés) son similares a un corta fuego ya que tiene la habilidad de bloquear tráfico, por lo cual se lo puede definir como un software que puede detectar e intervenir de manera proactiva contra tráfico malicioso que muestre una amenaza para un equipo o red. Los IPS tiene una gran ventaja sobre los IDS ya que al poder detectar una intrusión y detenerla a través del uso de normas específicas para ello, logra reducir en gran medida el número de posibles alertas. Otra ventaja es el tiempo de reacción que IPS tiene frente a una posible amenaza. Sin embargo, se debe tener en cuenta que una de las desventajas de este IPS es la capacidad de reacción ya que puede reaccionar a falsos positivos, provocando la denegación de servicios e inoperatividad de los sistemas que ocupen la red.Se clasifican en diferentes grupos según sus características y específicamente en función de su ubicación que se dividen en Host-based IPS, Network-based, Wireless-network-based IPS y Network Behaviour Analysis.[30]

Machine Learning

El aprendizaje automático o machine learning es una rama en evolución de los algoritmos computacionales que están diseñados para emular la inteligencia humana aprendiendo del entorno circundante [31]. En [32], nos habla de que machine learning es la ciencia que hace que los ordenadores “aprendan” a partir de los datos. Es decir, que en lugar de programar toda la solución para cada necesidad, como se suele hacer en la programación tradicional, con machine learning se pueden desarrollar algoritmos que sean capaces de reconocer patrones en varios tipos de datos.

Los modelos y/o algoritmos de aprendizaje automático permiten que los sistemas tomen decisiones de forma autónoma sin ningún apoyo externo. Estas decisiones se toman encontrando patrones subyacentes valiosos dentro de datos complejos.

En el estado del arte se identificaron los modelos y/o algoritmos de machine learning más usados en sistemas de detección y prevención de intrusos enfocados en ataques de phishing. Se analizo la precisión que estos modelos alcanzaron en sus ambientes de prueba.

La Tabla 1 presenta algunos modelos y/o algoritmos de machine learning más destacados para la detección y prevención de amenazas.

Tabla 1

Modelos de Machine Learning

ID	Modelo	Descripción
1	Decision Tree	El aprendizaje mediante árboles de decisión es uno de los algoritmos de aprendizaje con más éxito, debido a sus diversas características atractivas: sencillez, comprensibilidad, ausencia de parámetros y capacidad para manejar datos de tipo mixto. En el aprendizaje por árbol de decisión, se induce un árbol de decisión a partir de un conjunto de instancias de entrenamiento etiquetadas representadas por una tupla de valores de atributo y una etiqueta de clase.[33] Los resultados obtenidos en [34], demostraron que el algoritmo Decision Tree mejora la tasa de verdaderos positivos y aumenta la eficiencia al usar menos características para la clasificación; mejorando la capacidad de los IDS para enfrentar amenazas.
2	Random Forest	Según [35], Random Forest es una versión extendida del árbol de decisiones que puede predecir las instancias futuras con múltiples clasificadores en lugar de un clasificador único para alcanzar la precisión y exactitud de la predicción. Los rendimientos del modelo Random Forest se reconocen y varían con otros modelos de clasificación que producen institucionalización, regularización, conexión, cambio de alta inclinación y resaltan la elección en los modelos de aprendizaje. En [36] se mostró una alta exactitud y precisión del algoritmo random forest,

ID	Modelo	Descripción
		convirtiéndolo en una opción viable para emplearlo en sistemas de detección de intrusos en tiempo real.
3	Support Vector Machines (SVM)	Support Vector Machines (SVM) es un conjunto de métodos relacionados para el aprendizaje supervisado, aplicables tanto a problemas de clasificación como de regresión. Un clasificador SVM crea un hiperplano de margen máximo que se encuentra en un espacio de entrada transformado y divide las clases de ejemplo, mientras maximiza la distancia a los ejemplos divididos limpiamente más cercanos. Los parámetros del hiperplano de solución se derivan de un problema de optimización de programación cuadrática. [37] Este modelo aplicado en [38] logro una precisión del 99.1% en la identificación de ataques y se ha destacado como un algoritmo altamente efectivo en sistemas IDS.

En el campo de la detección y prevención de ataques de pishing, existen diversos modelos y/o algortimos de machine learning que se pueden emplear, desde los más sencillos hasta los más complicados. Es por ello que en la Tabla 1, se ha realizado una selección previa de los modelos más usados para la temática propuesta.

Inteligencia de Amenazas

La inteligencia de amenazas también conocida como Threat Intelligence, es un campo en constante evolución debido al aumento de la sofisticación de las amenazas en línea. El objetivo de la inteligencia de amenazas es proporcionar información oportuna y precisa sobre las amenazas de seguridad actuales y potenciales para ayudar a las organizaciones a tomar medidas preventivas.[39]

Inteligencia de Amenazas implica la recolección, evaluación y aplicación de datos significativos con el fin de reconocer potenciales peligros para la seguridad de una entidad. Este proceso se nutre de diversas fuentes, como registros de seguridad, informes de inteligencia de amenazas, estudios de malware y actividades de hacking. Al

analizar esta información, es posible detectar amenazas emergentes y adoptar medidas preventivas para evitar ataques cibernéticos.[40]

Según [41] basado en [42], el alcance, los objetivos planteados y las partes interesadas, dan como resultado productos desarrollados en inteligencia de amenazas que se pueden clasificar en cuatro categorías: estratégicos, operacionales, técnicos y tácticos como hace mención la agencia de la Unión Europea para la Ciberseguridad.

Inteligencia de amenazas estratégica: Provee a la alta dirección información de alto nivel acerca de riesgos y amenazas cibernéticas, contribuyendo a la toma de decisiones para mitigar impactos potenciales en los activos tecnológicos de la organización. [41], [43]

Inteligencia de amenazas operacional: Está enfocada al personal encargado de defender infraestructuras tecnológicas. La información entregada está relacionada con ataques inminentes, centrándose en los posibles actores de amenaza y las metodologías usadas para llevar a cabo la intrusión.[41], [44]

Inteligencia de amenazas técnica: Proporciona datos como IOCs que suelen corresponder a direcciones IP, urls, hash de archivos y dominios, para ser incorporados en herramientas de seguridad, con la finalidad de monitorear posibles ataques.[41], [43]

Inteligencia de amenazas tácticas: Este tipo de productos está enfocado a conocer las TTP’s utilizadas en los ciberataques, para establecer defensas y generar políticas actualizadas, que correspondan a la realidad de los ataques.[41], [43]

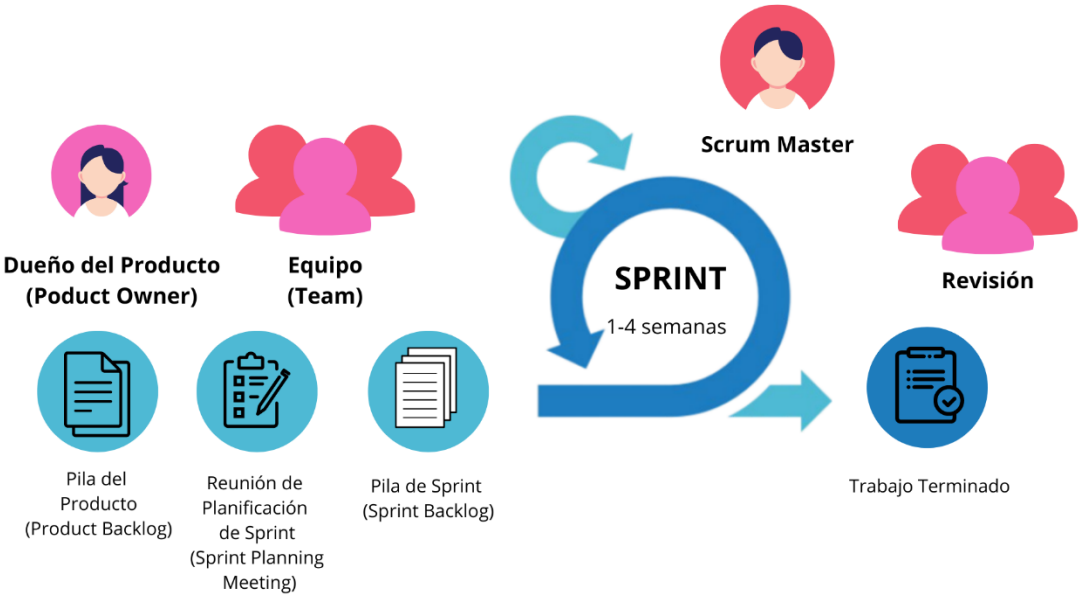
En este trabajo vamos a emplear inteligencia de amenazas técnica a través del uso de IOC para mejorar la detección y prevención de intrusos. Usar IOC nos ayudan a identificar con mayor precisión las características de las intrusiones, permitiendo dar una respuesta adecuada a amenazas potenciales en el sistema. Específicamente, se usaran las APIs de AbuseIPDB, VirusTotal y APIVoid, de acuerdo a [45], [46], [47] podrán proporcionar al sistema datos sobre IPs, dominios, URLs y otros indicadores de compromiso.

Metodología Scrum

Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.[48]

Figura 1

Diagrama de la metodología propuesta



Nota. La información fue tomada de (Por et al., n.d.)

La Figura 1 resume los roles, artefactos y eventos principales en un Sprint de Scrum. Esta metodología se organiza en ciclos denominados Sprints, estas son iteraciones de 1 a 4 semanas. Cada sprint tiene una duración fija, es decir que finaliza en una fecha específica, aunque no se haya terminada el trabajo, y nunca se alargan.[49]

Al iniciar un sprint, un equipo multifuncional selecciona una lista de elementos a priorizar (requisitos del cliente) que se compromete a completar para el final del sprint. Durante el Sprint, los elementos que han sido escogidos no pueden cambiarse. Diariamente el equipo debe reunirse para un informe del progreso, en el cual se actualizan gráficas simples que sirven como guía para el resto del equipo, acerca del trabajo que falta por hacer [49].

Al final de cada Sprint, se revisa el sprint que se desarrolló con los interesados y los miembros del equipo, con el fin de mostrar los avances que se han realizado y lo que se ha construido hasta el momento. De tal forma que se obtienen comentarios y observaciones que pueden incorporarse al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar.[49]

Herramientas

Las herramientas que se utilizaron en la ejecución del sistema se presentan a continuación:

Visual Studio Code, representa un editor de código fuente ligero pero eficaz, desarrollado por Microsoft. Destacado por su vasta variedad de lenguajes y herramientas de programación, lo cual mejora el rendimiento del desarrollador.[50]

Python es un lenguaje de programación utilizado específicamente para el desarrollo de aplicaciones web, desarrollo de software, la ciencia de datos y el machine learning (ML). Se destaca eficiencia y versatilidad con todo tipo de sistema aumentando su velocidad de desarrollo.[51]

Postfix: Es un software libre que se utiliza para facilitar él envió de correos el cual, además, procura ser más rápido, sencillo de manejar y bastante seguro. Fue creado por Wietse Venema en 2016, considerada una variante de Sendmail, aunque es muy diferente internamente. Hoy en día es distribuido bajo la licencia de IBM.[52]

Dovecot: es una opción de servidor de correo electrónico que trae consigo una amplia gama de características que incluyen la fácil interpretación, confiabilidad, alto rendimiento, autenticación flexible, auto optimización y migración, además asegura la protección de la comunicación mediante correos electrónicos, y proporciona opciones de extensión y administración sencilla. [53]

Windows Server: Es una distribución de Microsoft, es un sistema el cual permite que una computadora realice funciones de red como servidor de impresión, controlador de dominio, servidor web y servidor de archivos.[54]

Thunderbird es el cliente o servicio de correo electrónico desarrollado por Mozilla, es un software gratuito diseñado para hacer el correo más fácil se desempeña en diferentes plataformas y posee diferentes alternativas de personalización.[55]

Para el Desarrollo de dashboard administrativo se utilizó React la cual es una librería de JavaScript que permite crear interfaces de usuario interactivas,a partir de piezas individuales llamadas componentes[56]. Taimwind que es un framework de CSS enfocado en el diseño de páginas web. [57] y Typescript el cual es una extensión al lenguaje de programación JavaScript que se distingue por extender su sintaxis en el contexto de los tipos. [58]

Oracle: Se trata de un software de código abierto multiplataforma donde el usuario puede llevar a cabo simultáneamente varios sistemas operativos en una única máquina física gracias lo que permite que el cliente pueda imitar redes complejas a través del desarrollo de códigos en diferentes entornos de esta manera se obtienen mejores resultados y la depuración de aplicaciones es más eficiente. Por ello, al aceptar varios sistemas operativos como Windows, Linux, macOS, entre otros, se convierte en una herramienta fundamental para que el ciclo de desarrollo y la entrega del software sea más ágil.[59]

Medidas de Evaluación

Las métricas de rendimiento juegan un papel muy importante en problemas de clasificación donde se busca discriminar diferentes algoritmos Machine y Deep Learning, con la finalidad de facilitar la elección del mejor algoritmo dependiendo del objetivo de investigación. [60]

Las métricas definidas en este trabajo son: matriz de confusión, precisión, accuracy, recall y F1-score.

La primera métrica se trata de una tabla de frecuencias (Tabla 2) donde las filas pertenecen a la clase predicha y las columnas a la clase real, representando el número de predicciones de cada clase mutuamente excluyentes. [60]

Tabla 2

Matriz de Confusión

		Actual	
		Positivo	**Negativo**
Predicción	**Positivo**	Verdaderos Positivos (TP)	Falsos Positivos (FP)
	Negativo	Falsos Negativos (FN)	Verdaderos Negativos (TN)

Nota. La table muestra los elementos de la matriz de confusión basado en (Yao & Shepperd, 2021)

Donde:

- Verdaderos Positivos (TP): Las instancias correctamente clasificadas como positivas.

  Verdaderos Negativos (TN): Las instancias correctamente clasificadas como negativas.
- Falsos Positivos (FP): Las instancias incorrectamente clasificadas como positivas.

  Falsos Negativos (FN): Las instancias incorrectamente clasificadas como negativas. [61]

La accuracy es la métrica que se ha utilizado ampliamente para discriminar y seleccionar una solución óptima en la construcción de un clasificador optimizado. Sin embargo, su uso conduce el proceso de búsqueda a soluciones subóptimas debido a su limitada capacidad para discriminar valores. [62]

La precisión calcula el porcentaje de verdaderos positivos que se identifican correctamente dividido por el total de predicciones clasificadas como positivas. Es la proporción de unidades propensas a defectos correctamente identificadas de todos los casos clasificados como propensos a defectos. [63]

Recall o sensibilidad es una medida que permite conocer la proporción de casos positivos que fueron correctamente clasificados. En un modelo perfecto el recall es igual a 1 para cada clase. Desde el punto de vista analítico un investigador busca aumentar el recall sin afectar el valor de la accuracy. [64]

La medida F1-score fusiona las métricas accuracy con recall, presentando diferencias en el rendimiento de un clasificador que no son revelados únicamente con la accuracy. Es directamente proporcional al aumento de las dos medidas, por lo tanto, valores altos de F1-score demuestra que el algoritmo de clasificación predice de mejor manera la clase positiva. [65]

En la Tabla 3 se describen las fórmulas que se aplicaran en la evaluación del modelo.

Tabla 3

Fórmulas de las métricas de evaluación

MÉTRICAS DE EVALUACIÓN	FÓRMULA
Accuracy	$accuracy = \frac{VP+VN}{VP+VN+FP+FN}$
Precisión	$precisión = \frac{VP}{VP + FP}$
Recall	$recall = \frac{VP}{VP + FN}$
F1	$F1 = 2 \cdot \frac{precisión \cdot recall}{precisión + recall}$

Nota. La tabla detalla las fórmulas de las métricas de evaluación. Tomado de (Borja-Robalino & Monleon-Getino, 2020)

3. Metodología

En esta sección se presentará la metodología empleada para el desarrollo del sistema de detección y prevención de intrusos, enfocada en ataques de ingeniería social. Se detallará el uso de la metodología Scrum, así como los respectivos artefactos usados en el proceso. Además, se presentará el diseño del sistema que incluye los componentes técnicos y arquitectónicos necesarios para el desarrollo e implementación del entorno.

Metodología Scrum

Para la elección de la metodología a usar en el presente trabajo, se analizaron diferentes alternativas y se determinó emplear la metodología ágil SCRUM. Esta metodología destaca por el uso de buenas prácticas y fomentar el trabajo colaborativo. Además, SCRUM permite realizar entregar parciales del proyecto, que son priorizadas según su valor.

Definición de Roles

En la Tabla 4 se han definido los roles del proyecto, siguiendo lo especificado en [48], asegurando que cada miembro del equipo scrum tenga clara sus responsabilidades.

Tabla 4

Asignación de Roles en Scrum

Definición de Roles	
SCRUM MASTER	Ing. Milton Patricio Navas Moya, Mgtr
PRODUCT OWNER	Ing. José Luis Carrillo Medina, Ph.D.
DEVELOPMENT TEAM	Lissette Helena Santamaría Zambrano Melany Vanesa Nacimba Caiza

A continuación, se presenta una descripción de los roles:

Scrum Master: El Scrum Master es responsable de la eficacia del equipo Scrum y lo hace permitiendo que el equipo mejore sus prácticas dentro del marco de Scrum.[48]

Product Owner: El dueño del producto es el responsable de aumentar el valor del producto que es el resultado del trabajo del equipo Scrum. La forma de hacerlo varía ampliamente entre organizaciones, equipos de Scrum e individuos.[48]

Development Team: El equipo de desarrolladores son las personas que conforman el equipo Scrum y que están comprometidas a crear cualquier aspecto de un incremento utilizable en cada sprint.[48]

Release Planning

La planificación del release se basa en (estimaciones de) el esfuerzo de implementación. En ella se definen los objetivos y los plazos del proyecto. [66]

Visión del proyecto. Desarrollar un sistema de detección y prevención de phishing basado en la integración de modelos y/o algortimos machine learning y APIs de inteligencia de amenazas.

Captura de Requisitos (Historias de Usuario)

Según la definición adaptada en [67] basada en [68], las Historias de Usuario conforman un enfoque ágil, que se concentra en facilitar conversaciones acerca de las necesidades de los clientes. Estas historias son descripciones breves y simples de las funcionalidades del sistema, que son expresadas desde la perspectiva de la persona que desea dicha funcionalidad, usualmente un usuario.

En la Tabla 5 se presentan las HU definidas para este sistema, en la que se especifican los roles, las características y/o funcionalidades deseadas y la razón por la que deben ser implementadas.

Tabla 5

Historias de usuario

ID	Nombre	Rol	Característica / Funcionalidad	Razón / Resultado
001	H.U.01	Como Usuario	Quiero un sistema de detección y prevención de intrusos que por	Para tener un medio que sea capaz de identificar y analizar

ID	Nombre	Rol	Característica / Funcionalidad	Razón / Resultado
			medio del entrenamiento de un modelo de machine learning pueda detectar ataques de phishing.	ataques de phishing, mejorando la seguridad en entornos de transacciones en línea.
002	H.U.02	Como usuario	Quiero que el sistema se apoye en APIs de inteligencia de amenazas para respaldar el modelo.	Para mejor la precisión de la detección de phishing con datos sobre amenazas actualizados.
003	H.U.03	Como usuario	Quiero un dashboard que permita la visualización de la información adquirida sobre ataques de phishing.	Para lograr supervisar y analizar las amenazas, permitiendo una mejor toma de decisiones.
004	H.U.04	Como Usuario	Quiero recibir alertas y/o notificaciones sobre posibles ataques de phishing detectados.	Para poder reaccionar ante las diferentes amenazas en curso y poder tomar las medidas necesarias, con el fin de proteger la integridad del sistema.

Prioridad de las Historias de Usuario

En la Tabla 6, se detalla la priorización de las historias de usuario, según su importancia para el desarrollo del proyecto. Para la asignación de los puntos de Historia se uso el método planning poker, que usa la serie de Fibonacci para votar sobre los puntos de historia.

Tabla 6

Prioridad de las historias de usuario

Prioridad	H.U.	Puntos de Historia	Descripción
Alta	H.U.01	8	implementación los modelos de machine learning, para detectar los ataques de phishing.

Prioridad	H.U.	Puntos de Historia	Descripción
Alta	H.U.02	5	Integración de las APIs de inteligencia de amenazas para mejorar la precisión.
Media	H.U.03	5	Desarrollo de un dashboard que visualice la información de ataques phishing detectados.
Baja	H.U.04	3	Configuración de alertas y/o notificaciones de los ataques de phishing detectados.

Product Backlog

En la Tabla 7 se presentan las historias de usuario que incluye las estimaciones y fechas de inicio y de fin.

Tabla 7

Product Backlog del proyecto

ID	Historia de Usuario	Estimación (días)	Fecha Inicio	Fecha de Fin	N° de Sprint
001	H.U.01	10	27/06/2024	07/07/2024	01
002	H.U.02	9	08/07/2024	16/07/2024	02
003	H.U.03	11	17/07/2024	28/07/2024	03
004	H.U.04	10	29/07/2024	08/08/2024	04

Nota. La tabla proporciona una visión detallada del cronograma del proyecto.

Lista de Tareas

En la lista de tareas se presentan las funcionalidades del sistema, basado en las historias de usuario. En la Tabla 8 se presentan las tareas a realizar.

Tabla 8

Lista de Tareas del proyecto

ID	Nombre	Característica/Funcionalidad	Razón/Resultado
001	L.T.1	Entrenar e implementar los modelos de machine learning, para detectar los ataques de phishing.	Para obtener un sistema que pueda identificar ataques de phishing con mayor exactitud.
002	L.T.2	Integrar y configurar APIs de inteligencia de amenazas.	Para aumentar la precisión del sistema a la hora de detectar amenazas.
003	L.T.3	Desarrollar un dashboard que visualice la información de ataques phishing detectados.	Proveer una interfaz que permita monitorear las amenazas detectadas.
004	L.T.4	Emisión de alertas y/o notificaciones de los ataques de phishing detectados.	Facilitar a los usuarios de la información necesaria para reaccionar ante amenazas del sistema.

En la Tabla 9 se mencionan la estimación en días, la fecha de inicio, fecha de fin y el número de Sprint de las tareas mencionadas en la Tabla 8.

Tabla 9

Estimación de las tareas del producto backlog.

ID	Historia de Usuario	Estimación (días)	Fecha Inicio	Fecha de Fin	N° de Sprint
001	L.T.1	10	27/06/2024	07/07/2024	01
002	L.T.2	9	08/07/2024	16/07/2024	02
003	L.T.3	11	17/07/2024	28/07/2024	03
	L.T.4	10	29/07/2024	08/08/2024	04

Diseño

En esta sección se presentará el diseño de la arquitectura implementada para el desarrollo del sistema de detección y prevención de intrusos, enfocada en ataques de

ingeniería social. Se explicará detalladamente el funcionamiento del sistema para comprobar que se cumpla con el objetivo principal de este proyecto.

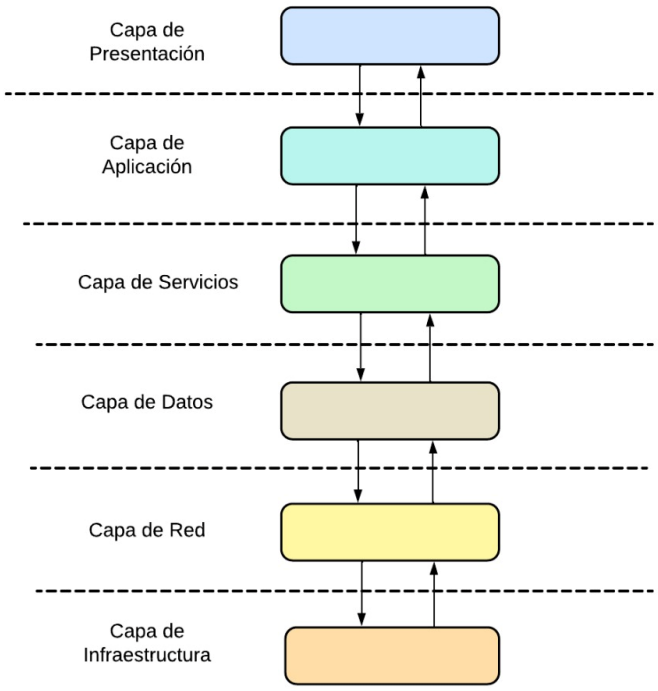
La arquitectura que se empleará en este sistema es la arquitectura multicapa, en comparación con otras arquitecturas, esta es fácil de mantener y escalar, razón por la que se decidió disponer de ella.

Diagrama de la Arquitectura Lógica

La arquitectura en capas permite separar los archivos y componentes del sistema. En este caso para el desarrollo del sistema de detección y prevención de intrusos, enfocada en ataques de ingeniería social, se implementará la arquitectura multicapa como se observa en la Figura 2: Diagrama de la arquitectura lógica del sistema, se tienen seis capas diferentes: Capa de Presentación, Capa de Aplicación, Capa de servicios, Capa de datos, Capa de Red, Capa de Infraestructura. Estas seis capas interactúan entre ellas lo que permite que el sistema funcione correctamente.

Figura 2

Diagrama de la arquitectura lógica del sistema



Definición de las tecnologías a usar

Para la elección de las tecnologías a usar en el sistema de detección y prevención de intrusos, enfocada en ataques de ingeniería social. Se analizaron diferentes alternativas y se determinó emplear las tecnologías que se observa en la Figura 3: Diagrama de la arquitectura lógica del sistema con las tecnologías a usar. La cual muestra la tecnología a emplear en cada capa que se estableció en el modelo de la Figura 2: Diagrama de la arquitectura lógica del sistema. Para comprender de mejor manera la Figura 3, a continuación, se explica brevemente cada capa.

Capa de Presentación

En esta capa se muestran los resultados que se obtuvo de la interacción usuario-sistema a través de un dashboard, el cual analiza una serie de correos electrónicos en tiempo real para detectar phishing en caso de que el correo electrónico contenga un posible ataque de

phishing se envía una alerta al dashboard. Como se puede observar en la Fig 3 el dashboard está desarrollado mediante:

React: Según [56], es una librería de JavaScript que permite crear interfaces de usuario interactivas,a partir de piezas individuales llamadas componentes.

Tailwind: Según [57], es un framework de CSS enfocado en el diseño de páginas web.

TypeScript: Según [58], es una extensión del lenguaje de programación JavaScript que se caracteriza por ampliar su sintaxis en el contexto de los tipos.

Capa de Aplicación

La capa de aplicación a través del servidor Flask gestiona la lógica de negocio del sistema que se encarga de recibir y procesar los correos electrónicos interceptados. Dicho servidor combina modelos de predicción ya entrenados (Random Forest) y supervisa la comunicación con servicios externos Como se observa en la Figura 3 la tecnología principal usada es:

Flask: Es un framework que ofrece diferentes recursos para crear aplicaciones web de manera rápida y eficaz. Su adaptabilidad permite que los desarrolladores estructuren su código de forma libre y puedan añadir las librerías que necesiten [69]

Capa de Servicios

Esta capa comprende los modelos de predicción de phishing, inteligencia de amenazas y sockets. Permite analizar las URLS y cuerpos de los correos electrónicos a través de los modelos de predicción mientras que la inteligencia de amenazas y sockets se encargan de las ips y dominios sospechosos permitiendo una comunicación en tiempo real.

Capa de Datos

En esta capa la base de datos Sqlite nos permite tener un registro detallado de los correos electrónicos evaluados ya que toda la información y predicciones de la inteligencia de amenazas se queda almacenada en la misma.

Capa de Red

Esta capa está constituida por dos servidores el DNS y el SMTP. El primero provee el dominio que será empleado por el servidor SMTP.Y el segundo bloquea y redirecciona los correos electrónicos al servidor flask para su revisión. Las tecnologías más usadas en esta capa son:

Windows Server: Es una distribución de Microsoft, es un sistema el cual permite que una computadora realice funciones de red como servidor de impresión, controlador de dominio, servidor web y servidor de archivos.[54]

Postfix: Es un software libre que se utiliza para facilitar él envió de correos el cual, además, procura ser más rápido, sencillo de manejar y bastante seguro. Fue creado por Wietse Venema en 2016, considerada una variante de Sendmail, aunque es muy diferente internamente. Hoy en día es distribuido bajo la licencia de IBM.[52]

Dovecot: Es una opción de servidor de correo electrónico que trae consigo una amplia gama de características que incluyen la fácil interpretación, confiabilidad, alto rendimiento, autenticación flexible, auto optimización y migración, además asegura la protección de la comunicación mediante correos electrónicos, y proporciona opciones de extensión y administración sencilla. [53]

Capa de Infraestructura

En esta capa es fundamental el uso de máquinas virtuales para obtener una distribución equilibrada de las cargas de trabajo, se empelan tres maquina virtuales en las que se ejecuta los servidores DNS, SMTP Y Flask lo que garantiza una fácil gestión del sistema

Figura 3

Diagrama de la arquitectura lógica del sistema que incluye las tecnologías que se usarán

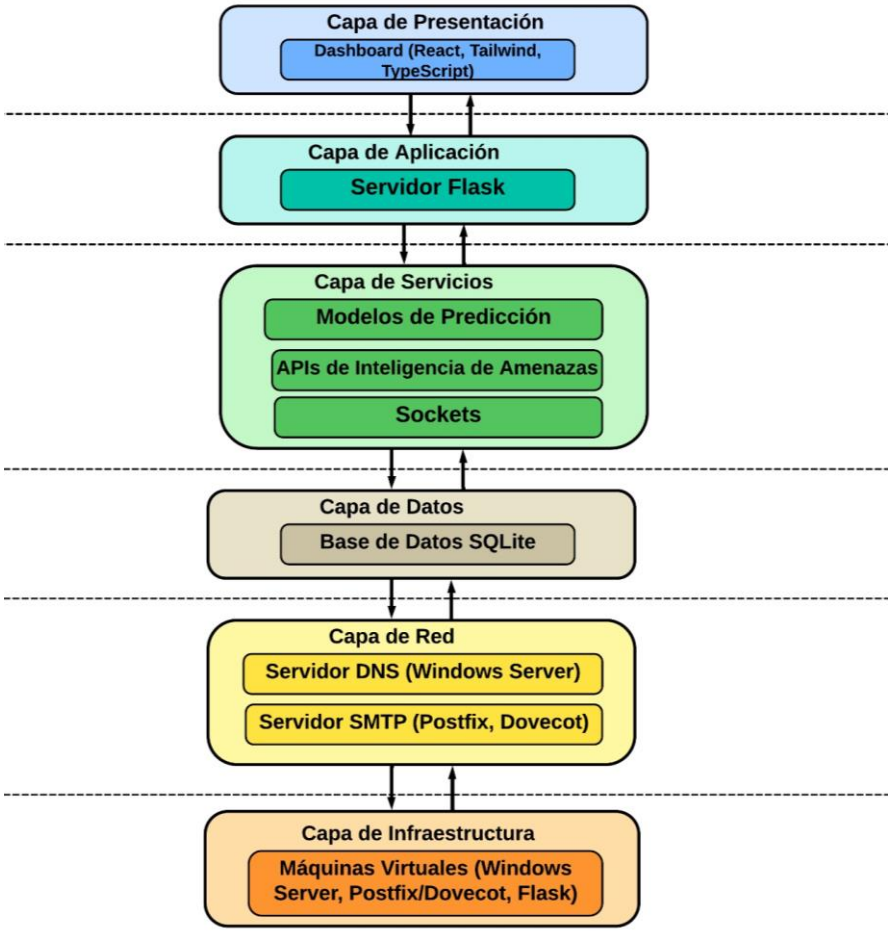
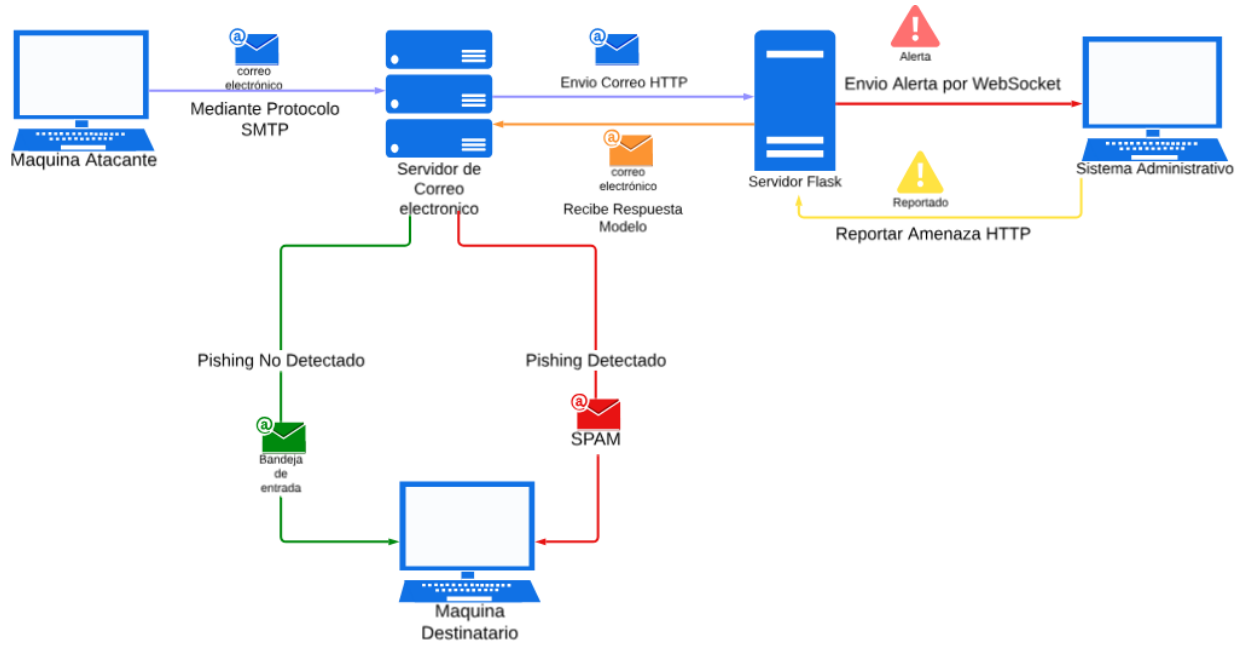


Figura 4

Diagrama de Arquitectura Física



En la Figura 4, se muestra el proceso que se realiza cuando se envía un correo electrónico. La arquitectura que se desarrolló sirve para que los usuarios se protejan ante ataques de phishing.

Cuando el correo es enviado, viaja al servidor SMTP por medio del protocolo IMAP, y una vez que llega es reenviado al servidor Flask, por medio del protocolo HTTP.

Al llegar el correo al servidor Flask este utiliza el modelo de predicción random forest y Apis de inteligencia de amenazas para analizar el contenido del correo electrónico, una vez que el modelo termina el análisis. En caso de detectarse un ataque de phishing el servidor flask envía automáticamente una alerta al sistema administrativo, al llegar esta alerta al sistema administrativo el administrador puede reportar esta amenaza. Una vez reportada la amenaza se envía nuevamente esta información al servidor Flask, el cual se encarga se almacenar esta información en una base de datos local. Al llegar la respuesta del modelo al servidor SMTP este

se encarga de enviar el correo electrónico de acuerdo a su contenido: si es legítimo el correo electrónico ingresa a la bandeja de entrada de la máquina del destinatario. Si por el contrario el correo electrónico es ilegitimo es decir contiene phishing lo envía a la carpeta de spam. La gran ventaja de este sistema es que va a prendiendo de estas detecciones y en caso de que se envié nuevamente este correo el servidor automáticamente lo identificara como spam sin necesidad de repetir nuevamente todo el proceso. Lo que permite una detección más rápida y eficaz al momento de identificar este tipo de correos.

Sprint 01: Entrenamiento del modelo machine learning

En esta sección se presenta el desarrollo del primer sprint, el cual consiste en la selección de los modelos de machine learning y su posterior entrenamiento para que sea capaz de detectar ataques de phishing en entornos de transacciones en línea.

Historia de Usuario detallada

A continuación, se presenta la H.U.01 de la Tabla 5 con su respectiva tarea L.T.1 propuesta en la Tabla 8. En la tabla 10 se explica con más detalle la H.U.01 y se añaden los criterios de validación.

Tabla 10

Historia de usuario detallada para el entrenamiento del modelo.

Historia de Usuario			
Nombre: H.U.01		**Usuario**: Usuario de internet	
Nombre de historia: Selección y entrenamiento del modelo machine learning			
Prioridad del Negocio:	Alta	**Riesgo en desarrollo**:	Moderado
Puntos estimados (días):	10	**Interacción asignada:**	1
Programador(es) responsable(s):	Lissette Santamaría y Melany Nacimba		
Descripción: Como usuario, quiero un sistema de detección y prevención de intrusos que por medio del entrenamiento de un modelo de machine learning pueda detectar ataques de phishing.			
Validación (Criterios de aceptación)			

Historia de Usuario
• El sistema debe entrenar el modelo de machine learning con datos de URLs maliciosas y los cuerpos de correos electrónicos. • El modelo debe detectar ataques de phishing bajo los umbrales establecidos. • El sistema debe ser capaz de mejorar la precisión y efectividad conforme se le agregan datos adicionales.

Sprint Backlog 01

En el Sprint Backlog 01 se presentan las actividades realizadas durante el primer sprint. En la Tabla 11 se muestran algunos detalles como las fechas de inicio y de fin, responsables y las tareas.

Tabla 11

Sprint Backlog 01

Sprint 01					
Fecha Inicio		**Fecha Fin**			**Jornada**
27/06/2024		07/07/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
01	Investigación inicial y selección del modelo de machine learning.	8	27/06/2024	27/06/2024	Lissette Santamaría y Melany Nacimba.
01	Revisión y selección de datasets de phishing y no phishing.	6	28/06/2024	28/06/2024	Lissette Santamaría y Melany Nacimba.
01	Configuración del ambiente de desarrollo.	8	28/06/2024	29/06/2024	Lissette Santamaría y Melany Nacimba.
01	Limpieza y preprocesamiento de datos.	10	29/06/2024	30/06/2024	Lissette Santamaría y Melany Nacimba.

Sprint 01					
Fecha Inicio		**Fecha Fin**			**Jornada**
27/06/2024		07/07/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
01	Extracción de características de URLs y cuerpo del correo.	12	01/07/2024	02/07/2024	Lissette Santamaría y Melany Nacimba.
01	Implementación de los modelos machine learning seleccionados	12	02/07/2024	03/07/2024	Lissette Santamaría y Melany Nacimba.
01	Entrenamiento del modelo con los conjuntos de datos preprocesados.	12	04/07/2024	05/07/2024	Lissette Santamaría y Melany Nacimba.
01	Ajuste de hiperparámetros para la optimización del modelo.	9	05/07/2024	06/07/2024	Lissette Santamaría y Melany Nacimba.
01	Pruebas unitarias del modelo entrenado.	11	06/07/2024	07/07/2024	Lissette Santamaría y Melany Nacimba.

Resultados del Sprint 01

A continuación, se presentan los resultados obtenidos durante el desarrollo del primer sprint. El primer paso en este proceso fue el preprocesamiento del cuerpo de los correos electrónicos. Para ello, se usó la librería BeautifulSoup con el fin de remover el HTML que proviene del cuerpo del correo (Figura 5), el cual posteriormente fue dividido en tokens. Dichos tokens representan palabras individuales del texto y son la base para su posterior análisis.

Figura 5

Preprocesamiento de texto

```python
import re
from bs4 import BeautifulSoup
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk

# Descargar recursos necesarios para NLTK
nltk.download("punkt")
nltk.download("stopwords")


class TextProcessor:
    def __init__(self):
        self.stop_words = set(stopwords.words("english"))

    def preprocess_text(self, text):
        text = self.remove_html(text)
        text = self.remove_punctuation_and_special_chars(text)
        text = self.convert_to_lowercase(text)
        tokens = self.tokenize(text)
        tokens = self.remove_stop_words(tokens)
        return tokens

    def remove_html(self, text):
        soup = BeautifulSoup(text, "html.parser")
        return soup.get_text()

    def remove_punctuation_and_special_chars(self, text):
        return re.sub(r"\W", " ", text)

    def convert_to_lowercase(self, text):
        return text.lower()

    def tokenize(self, text):
        return word_tokenize(text)

    def remove_stop_words(self, tokens):
        return [word for word in tokens if word not in self.stop_words]
```

Con los tokens obtenidos se procedió a vectorizar el texto. Se usó TfidfVectorizer para transformarlos en una representación numérica. TfidfVectorizer se emplea para convertir el texto en una matriz de características numéricas basadas en la frecuencia de los términos y la inversa de la frecuencia de documentos, dando como resultado que el modelo machine learning pueda procesar el texto de manera efectiva.

Figura 6

Vectorización del texto

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectKBest, f_classif
import pandas as pd

class BodyExtractionFeature:
    def __init__(self, num_features=500):
        self.vectorizer = TfidfVectorizer()
        self.num_features = num_features
        self.selector = None

    def fit_transform(self, texts, labels):
        X = self.vectorizer.fit_transform(texts)
        self.selector = SelectKBest(f_classif, k=self.num_features)
        X_new = self.selector.fit_transform(X, labels)
        feature_names = self.vectorizer.get_feature_names_out()
        selected_feature_names = [feature_names[i] for i in self.selector.get_support(indices=True)]
        tfidf_df = pd.DataFrame(X_new.toarray(), columns=selected_feature_names)
        return tfidf_df, self.vectorizer, self.selector

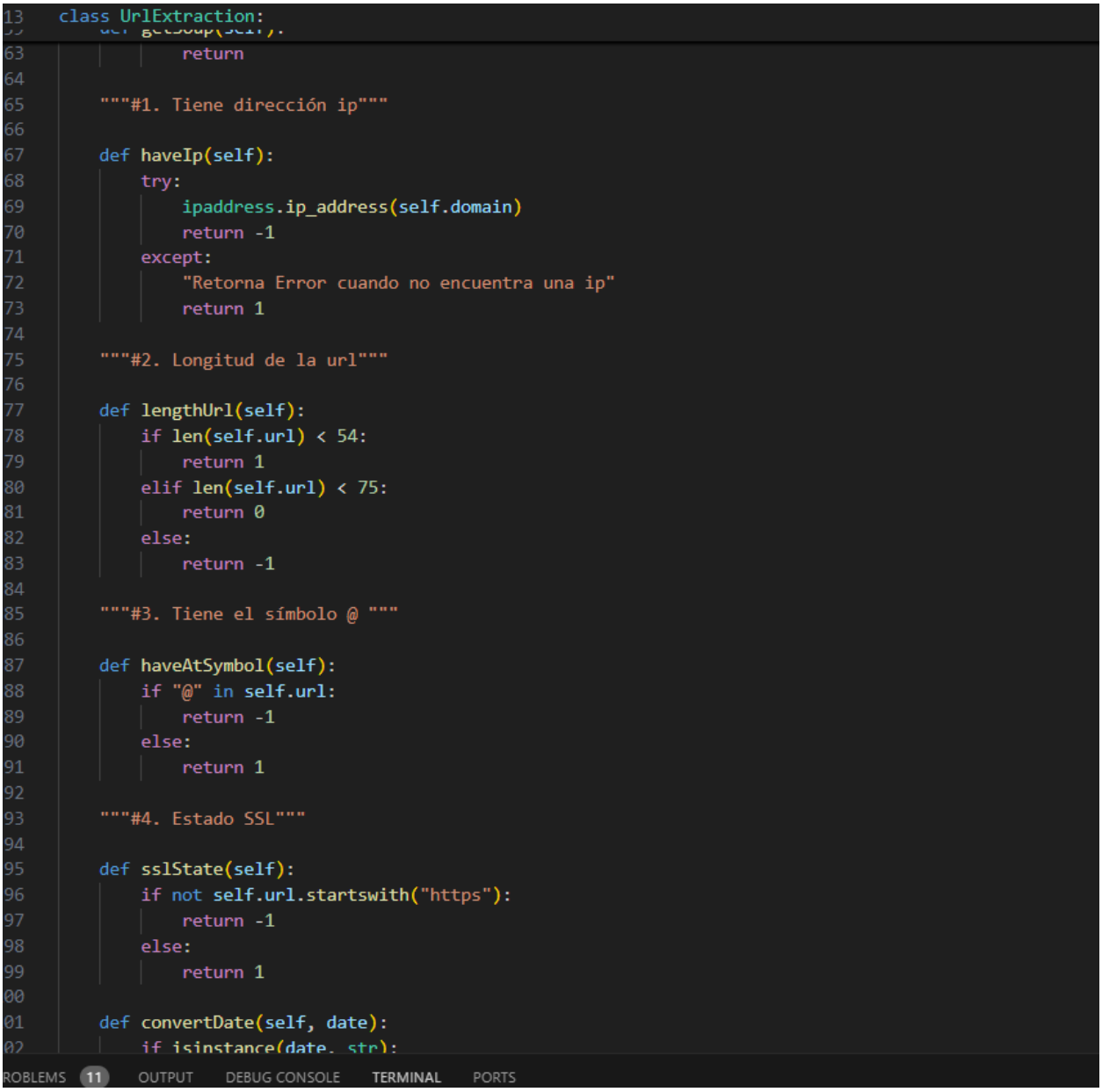
    def transform(self, texts):
        X = self.vectorizer.transform(texts)
        X_new = self.selector.transform(X)
        feature_names = self.vectorizer.get_feature_names_out()
        selected_feature_names = [feature_names[i] for i in self.selector.get_support(indices=True)]
        tfidf_df = pd.DataFrame(X_new.toarray(), columns=selected_feature_names)
        return tfidf_df
```

Paralelamente al preprocesamiento del cuerpo del correo, se realizó la extracción de características de las URLs. Las características de URLs que se usaron, fueron basadas en [70], cuyo artículo realiza una revisión de la literatura para identificar las característica más relevantes; las cuales incluyen la longitud de la URL, la presencia de patrones o palabras clave, la complejidad de la estructura, entre otros.

Las URLs se representaron mediante un conjunto de características cuantitativas, que se integraron en un formato adecuado para el modelo. El proceso convirtió las URLs en vectores de características numéricas, similar al proceso que se realizó con el cuerpo del correo.

Figura 7

Extracción de características de URL



```
class UrlExtraction:
        return

    """#1. Tiene dirección ip"""

    def haveIp(self):
        try:
            ipaddress.ip_address(self.domain)
            return -1
        except:
            "Retorna Error cuando no encuentra una ip"
            return 1

    """#2. Longitud de la url"""

    def lengthUrl(self):
        if len(self.url) < 54:
            return 1
        elif len(self.url) < 75:
            return 0
        else:
            return -1

    """#3. Tiene el símbolo @ """

    def haveAtSymbol(self):
        if "@" in self.url:
            return -1
        else:
            return 1

    """#4. Estado SSL"""

    def sslState(self):
        if not self.url.startswith("https"):
            return -1
        else:
            return 1

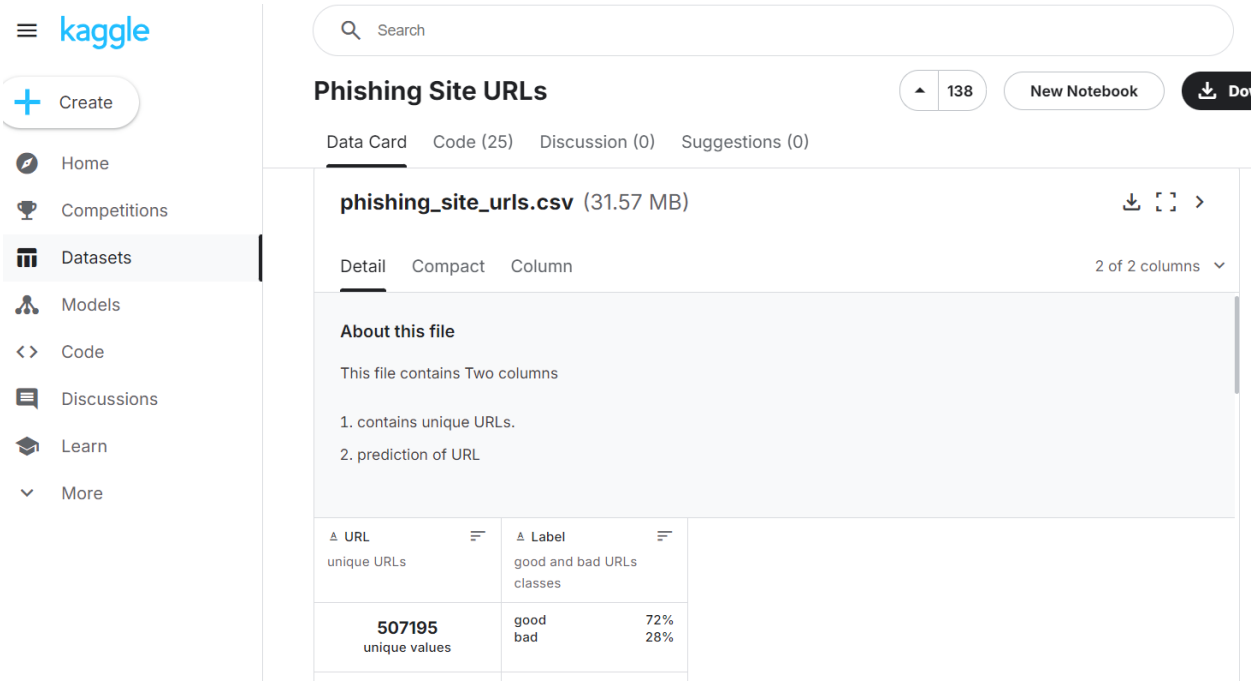
    def convertDate(self, date):
        if isinstance(date, str):
```

ROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Lo siguiente por hacer, fue preparar el conjunto de datos para entrenar los modelos de machine learning. Los datasets seleccionados se obtuvieron del repositorio de KAGGLE.

Figura 8

Repositorio de Kaggle



De lo analizado en el estado del arte, se identificó tres modelos y/o algoritmos de machine learning para aplicarlos en el sistema, estos son: Support Vector Machine (SVM), Decision Tree y Random Forest. La elección de estos modelos se basa en su capacidad de manejar datos textuales y eficacia en la clasificación binaria.

El conjunto de datos total que se utilizó fue de 60 000 URLs y 18 000 cuerpos de correo. Con el fin de evaluar los modelos machine learning, se divididieron en un 70% para entrenamiento y un 30% pruebas. Los modelos fueron evaluados usando métricas de rendimiento como accuracy, precisión, recall y F1.

En la Tabla 12 se muestran los resultados de las métricas de evaluación, obtenidas luego de probar los diferentes modelos con los datos de URL.

Tabla 12

Resultados de evaluación de los modelos y/o algoritmos machine learning para URL

Modelo y/o algoritmo	Recall	Precision	Accuracy	F1 Score
Decision Tree	96.36%	96.43%	96.36%	96.38%
Random Forest	96.39%	96.47%	96.39%	96.40%
Support Vector Machine	95.34%	95.48%	95.34%	95.37%

Los tres modelos que se evaluaron, presentaron un alto rendimiento y los valores resultantes fueron muy similares. No obstante, en la evaluación de las URLs, el modelo Random Forest tuvo mejores porcentajes en comparación con los otros modelos, puesto que el F1 score, que señala un equilibrio entre la precisión y el recall, dio un total del 96.40%,

En la Tabla 13 se muestran los resultados de las métricas de evaluación, obtenidas luego de probar los diferentes modelos con los datos de cuerpo del correo.

Tabla 13

Resultados de evaluación de los modelos y/o algoritmos machine learning para el cuerpo

Modelo y/o algoritmo	Recall	Precision	Accuracy	F1 Score
Decision Tree	92.28%	92.34%	92.28%	92.30%
Random Forest	97.03%	97.04%	97.03%	97.03%
Support Vector Machine	97.54%	97.62%	97.54%	97.55%

Para la evaluación del cuerpo del correo, SVM sin duda presenta porcentajes de métrica ligeramente más altos en comparación con los otros modelos. Por otro lado, es necesario destacar que todos los modelos presentan altos niveles en precisión y recall. SVM alcanzó un F1 score de 97.55% que refleja su capacidad en este conjunto de datos.

Aunque en la evaluación del cuerpo del correo, SVM alcanzó métricas ligeramente más elevadas que Random Forest y Decision Tree, se optó por no seleccionarla debido al tiempo de procesamiento. Tanto Random Forest como Decision Tree tardaron un promedio de 30 a 45 segundos en completar la clasificación, mientras que SVM se demoró aproximadamente unos 6 minutos en hacerlo.

Dado que el sistema maneja un gran volumen de datos en tiempo real, se tomó en cuenta los tiempos de ejecución, por lo que naturalmente el modelo seleccionado para el desarrollo de este sistema fue Random Forest, debido a su equilibrio entre rendimiento y tiempos de procesamiento.

Sprint 02: Integración con las APIs de inteligencia de amenazas

Durante el segundo sprint se realizó una investigación previa acera de inteligencia de amenazas y las APIs disponibles en el mercado. Se seleccionaron algunas de ellas y se integró con el sistema, enriqueciendo con información adicional al sistema.

Historia de Usuario detallada

En la Tabla 14 se presenta con más detalle la H.U.02 de la Tabla 5 que le corresponde la tarea L.T.2 de la Tabla 8.

Tabla 14

Historia de usuario detalladas de las APIs de inteligencia de amenazas

Historia de Usuario			
Nombre: H.U.02		**Usuario:** Usuario de internet	
Nombre de historia: Integración con las APIs de Inteligencia de amenazas			
Prioridad del Negocio:	Alta	**Riesgo en desarrollo:**	Moderado
Puntos estimados (días):	9	**Interacción asignada:**	2
Programador(es) responsable(s):	Lissette Santamaría y Melany Nacimba		
Descripción: Como usuario, quiero que el sistema se apoye en APIs de inteligencia de amenazas para respaldar el modelo.			
Validación (Criterios de aceptación) • El sistema debe ser capaz de integrarse con APIs de inteligencia de amenazas que faciliten información actualizada sobre amenazas. • Las APIs deben enriquecer el modelo de machine learning con datos relevantes para mejorar la precisión de la detección.			

Sprint Backlog 02

En el Sprint Backlog 02 se muestran las tareas realizadas a lo largo del segundo sprint. En la Tabla 15 se detallan las fechas de inicio y de fin, así como las tareas y los responsables de las mismas.

Tabla 15

Sprint Backlog 02

Sprint 02					
Fecha Inicio		**Fecha Fin**			**Jornada**
08/07/2024		16/07/2024			
H.U.	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
02	Investigación y selección de APIs de inteligencia de amenazas.	8	08/07/2024	08/07/2024	Lissette Santamaría y Melany Nacimba.
02	Estudio de las capacidades y limitaciones de las APIs seleccionadas.	8	09/07/2024	09/07/2024	Lissette Santamaría y Melany Nacimba.
02	Integración del sistema con las APIs seleccionadas.	18	10/07/2024	12/07/2024	Lissette Santamaría y Melany Nacimba.
02	Optimización del sistema basado en la información de las APIs.	16	12/07/2024	13/07/2024	Lissette Santamaría y Melany Nacimba.
02	Ejecución de pruebas unitarias para corroborar la correcta integración de las APIs.	24	14/07/2024	16/07/2024	Lissette Santamaría y Melany Nacimba.

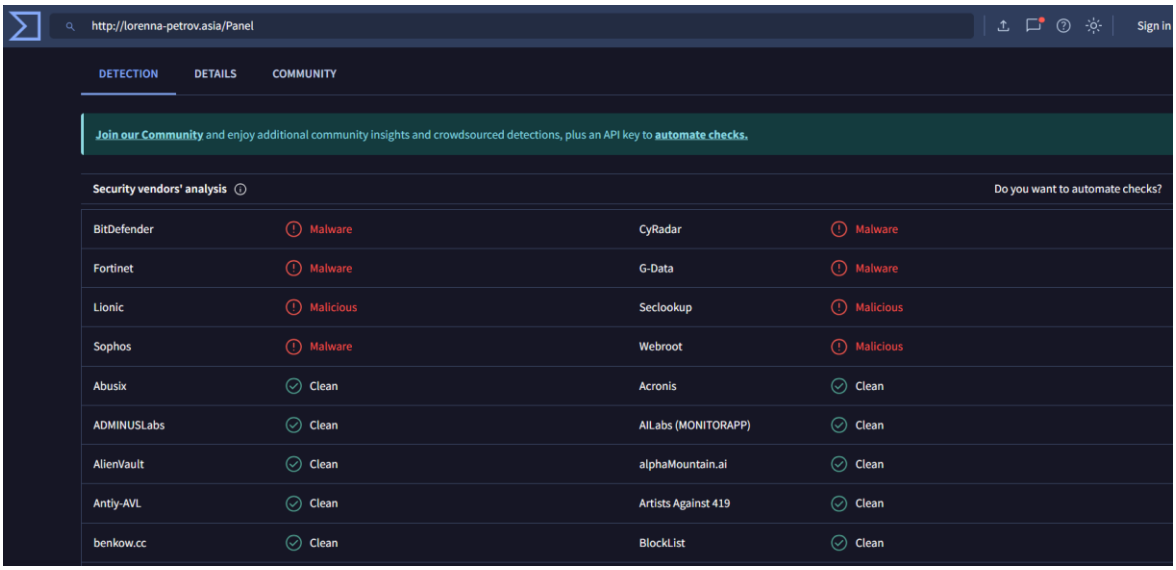
Resultados del Sprint 02

Para el segundo sprint, se realizó la integración de las APIs de inteligencia de amenazas. Se comenzó con la investigación y selección de las APIs, en el marco teórico se especificó la selección de las mismas, entre las cuales se eligieron AbuseIPDB, VirusTotal y APIVoid para el análisis de IPs, dominios y respaldar el modelo de machine learning.

Virus total proporciona información de URLs maliciosas, como reputación de dominios, URL e IP, lo que permite identificar comportamientos sospechosos.

Figura 9

Ejemplo de información proporcionada por VirusTotal



Por otro lado, la API AbuseIPDBS, ayuda a identificar IPs maliciosas por medio de la recopilación y análisis de informes de porcentaje de abuso.

Figura 10

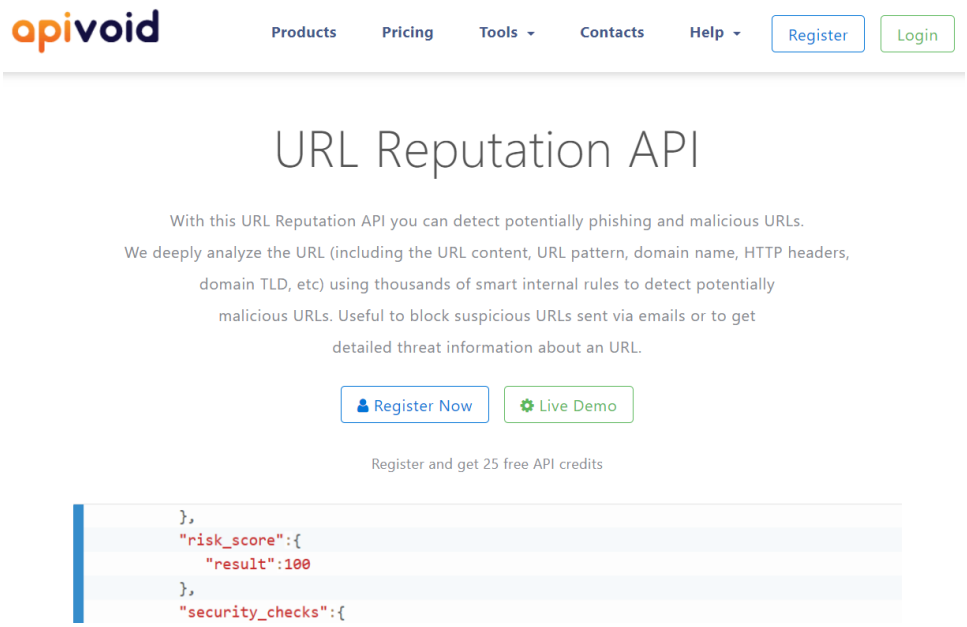
Ejemplo de búsqueda de una IP en la API AbuseIPDB



Al igual que la VirusTotal, la API de Apivoid también proporciona información de IPs, dominios y URLs lo que ayuda a la detección de amenazas.

Figura 11

API de reputación de URL de Apivoid



Una vez culminada la selección, se procedió a integrarlo con el sistema. En este proceso se realizó la conexión del sistema con las APIs para obtener la información relevante sobre IPs, dominios y URLs que estuviesen involucrados en actividades maliciosas.

Figura 12

Integración de la API Virus Total

```python
import datetime
import os
import base64
import requests

from dotenv import load_dotenv

# Cargar variables del archivo .env
load_dotenv()

class VirusTotalService:
    def __init__(self):
        self.api_key = os.getenv('VIRUSTOTAL_API_KEY')
        self.base_url = 'https://www.virustotal.com/api/v3'

    def get_url_id(self, url):
        """
        Convierte una URL en su representación base64 para usarla como ID en VirusTotal.
        """
        url_bytes = url.encode('utf-8')
        url_base64 = base64.urlsafe_b64encode(url_bytes).decode('utf-8').strip('=')
        return url_base64

    def scan_url(self, url):
        """
        Envía una URL a VirusTotal para su análisis.
        """
        headers = {
            'x-apikey': self.api_key
        }
        response = requests.post(f'{self.base_url}/urls', headers=headers, data={'url': url})
        return response.json()
```

```python
def get_url_analysis(self, url_id):
    """
    Obtiene el informe de análisis de una URL usando su ID.
    """
    headers = {
        'x-apikey': self.api_key
    }
    response = requests.get(f'{self.base_url}/urls/{url_id}', headers=headers)
    return response.json()

def parse_analysis_data(self, analysis_result):
    if(analysis_result.get('error')):
        return {
            'malicius': 0,
            'suspicious': 0,
            'harmless': 0,
            'undetected': 0,
            'last_analysis_date': 'N/A',
            'categories': {},
            'http_response_code': '404',
            'reputation': 'N/A',
            'times_submitted': 'N/A',

        }
```

Figura 13

Integración de la API Virus Total (continuación)

```
    attributes = analysis_result['data']['attributes']
    last_analysis_stats = attributes['last_analysis_stats']
    print(last_analysis_stats)
    last_analysis_date = datetime.datetime.fromtimestamp(attributes['last_analysis_date']).strftime('%Y-%m-%d %H:%M:%S')
    http_response_code = attributes.get('last_http_response_code', 'N/A')
    reputation = attributes.get('reputation', 'N/A')
    times_submitted = attributes.get('times_submitted', 'N/A')
    categories = attributes.get('categories', {})
    analysis_summary = {
        'malicious': last_analysis_stats['malicious'],
        'suspicious': last_analysis_stats['suspicious'],
        'harmless': last_analysis_stats['harmless'],
        'undetected': last_analysis_stats['undetected'],
        'last_analysis_date': last_analysis_date,
        'http_response_code': http_response_code,
        'reputation': reputation,
        'times_submitted': times_submitted,
        'categories': categories,
    }
    print("hola")
    return analysis_summary
```

La integración con las APIs ayudó con la optimización del sistema al enriquecer los datos disponibles para la detección de phishing. Adicionalmente se ajustó los parámetros y se mejoró la precisión del sistema, aportando un enfoque más completo a la detección de amenazas.

Sprint 3: Dashboard administrativo

A continuación, se detalla el proceso de implementación del dashboard administrativo el cual permite visualizar correos electrónicos detectados como phishing. A través del uso de Apis de inteligencia de amenazas se analizan los correos y se notifica de posibles amenazas.

Historia de Usuario Detallada

La historia de usuario H.U.03 de la Tabla 5 y tarea L.T.3 de la Tabla 8, hacen mención a la implementación del dashboard administrativo el cual permite visualizar la detección de amenazas en correos electrónicos y reportarlas. En la tabla 16 se puede observar detalladamente la historia de usuario del Sprint 3 con sus respectivos criterios de aceptación.

Tabla 16

Historia de usuario para el dashboard administrativo

Historia de Usuario			
Nombre: H.U.03		**Usuario:** Usuario de internet	
Nombre de historia: Dashboard Administrativo			
Prioridad del Negocio:	Alta	**Riesgo en desarrollo:**	Medio
Puntos estimados (días):	15	**Interacción asignada:**	2
Programador(es) responsable(s):	Lissette Santamaría y Melany Nacimba		

Descripción:

Como usuario, Quiero un dashboard que permita la visualización de la información adquirida sobre ataques de phishing.

Validación (Criterios de aceptación)

- El sistema generara una tabla con: la ip de origen, día de envío y un botón de detalles de los correos que se detectaron como phishing.
- Al darle clic al botón de detalles se visualizará una nueva interfaz con el análisis de las Apis de inteligencia de amenazas las cuales se centran en: la ip, url y dominio del correo, Además de un informe minucioso del mismo.
- El usuario podrá reportar inmediatamente las amenazas. Una vez que se haga el reporte esta información se enviará al servidor para que guarde este informe en una base de datos local lo cual servirá para futuras detecciones.
- Los administradores cuentan con una interfaz sencilla, que facilita el control y gestión de posibles amenazas de forma eficaz.

Sprint Backlog 03

En el Sprint Backlog 03 se especifica las tareas que se llevaron a cabo dentro del segundo Sprint. En la Tabla 17 se puede observar que se detalla: las tareas, fecha de inicio y fin, los responsable y finalmente el estado de la historia de usuario.

Tabla 17

Sprint Backlog 03

Sprint 03					
Fecha Inicio		**Fecha Fin**			**Jornada**
14/07/2024		28/07/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
03	Maquetación de la interfaz del dashboard y planificación de la arquitectura en React.	8	14/07/2024	14/07/2024	Lissette Santamaría y Melany Nacimba
03	Creación de una tabla que contenga: la ip de origen, día de envío y un botón de detalles de los correos que se detectaron como phishing.	12	15/07/2024	16/07/2024	Lissette Santamaría y Melany Nacimba
03	Implementación de la interfaz que mostrará las características del correo como: la IP, dirección de correo del (remitente -destinario) y tabs de análisis.	16	17/07/2024	20/07/2024	Lissette Santamaría y Melany Nacimba
03	Desarrollo del tab IP en el cual se mostrará el resultado del análisis de la IP a través del uso de Apis de inteligencia de amenazas.	10	21/07/2024	22/07/2024	Lissette Santamaría y Melany Nacimba
03	Desarrollo del tab url en el cual se mostrará el resultado del análisis de la url a través del uso de Apis de inteligencia de amenazas.	10	23/07/2024	24/07/2024	Lissette Santamaría y Melany Nacimba
03	Desarrollo del tab dominio en el cual se mostrará el resultado del análisis del dominio a través del uso de Apis de inteligencia de amenazas.	10	25/07/2024	26/07/2024	Lissette Santamaría y Melany Nacimba

Sprint 03					
Fecha Inicio		**Fecha Fin**			**Jornada**
14/07/2024		28/07/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
03	Desarrollo del botón (reporte de amenazas), este reporte se envía al servidor, el cual se encarga de guardar esta información en una base de datos local.	12	27/07/2024	28/07/2024	Lissette Santamaría y Melany Nacimba
03	Pruebas de integración y validación del dashboard para garantizar un adecuado rendimiento del sistema.	8	28/07/2024	28/07/2024	Lissette Santamaría y Melany Nacimba

Resultados del Sprint 3

En el Sprint 3 se finalizó con la creación del dashboard administrativo, el cual facilita a los usuarios la detección de fraude en correos electrónicos.

Figura 14

Dashboard del proyecto



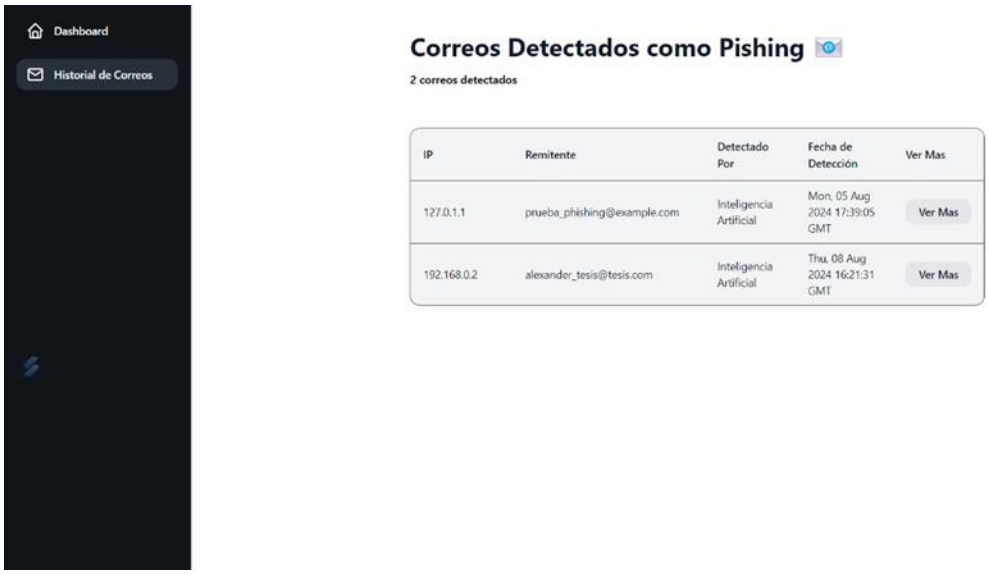


Para la implementación del dashboard en el frontend se usó: TypeScript, React, y Tailwind CSS, mientras que para el backend: El framework Flask en Python.

Como se puede observar en la Figura 13, se desarrolló una interfaz intuitiva con el objetivo de que el administrador no tenga problemas con el uso del sistema. Como se puede observar al ingresar a la interfaz del administrador, en la parte izquierda tenemos el Dashboard e Historial de Correos. Para la implementación de esta interfaz las herramientas que se usaron son: TypeScript, React, Phython y Tailwind CSS.

Figura 15

Interfaz del Administrador



En el dashboard se creó una interfaz, la cual está desarrollada para que el usuario observe toda la información detallada de los correos electrónicos maliciosos, incluyendo la IP de la máquina desde dónde fue enviado y las direcciones de correo tanto del remitente como del destinatario. Además, cuenta con 3 tabs de análisis: Tab de análisis de IP , Tab de análisis de URL y Tab de Análisis de Dominio; el primer Tab indica la información que se obtuvo del backend sobre la IP, en dónde se incluye fecha de análisis, número de motores que la catalogaron como sana, sospechosa, no detectada o maliciosa, y la categoría de amenaza a la que corresponde, esto gracias a la utilización de APIs de inteligencia de amenaza; el siguiente Tab indica los resultados de los URL examinados, con algunos detalles parecidos al de la IP

examinada; el último Tab expone el análisis del dominio del emisor en dónde se incluye su clasificación y los resultados que se obtuvo de los motores de detección.

Figura 16

Información detallada del correo



Reporte de Amenazas: Para garantizar la seguridad ante futuras amenazas de phishing se añadió un botón de "Reportar amenaza" el cual al ser ejecutado por el usuario emite la información examinada de la IP, url y del dominio hacia el backend para su posterior almacenamiento en la base de datos SQ Lite, esto permite que amenazas con características similares sean detectadas automáticamente sin tener que pasar nuevamente por los filtros de inteligencia de amenaza.

Figura 17

Botón reporte de amenazas



Se llevaron a cabo minuciosas pruebas de integración y validación entre el dashboard y el servidor asegurando un ciclo de trabajo eficiente. Teniendo mayor enfoque en realizar pruebas de interacción entre el frontend y backend garantizando que el análisis de las Apis de inteligencia de amenazas llegue al backend de forma clara y concisa.

Sprint 04: Implementación de Alertas y/o notificaciones

En este sprint se presenta la implementación de un sistema de alertas y/o notificaciones que se genera luego de que se detectan ataques de phishing, estas se visualizan en la interfaz gráfica de usuario.

Historia de Usuario detallada

A continuación, se detalla historia de usuario que hace referencia a la H.U.04 de la Tabla 5 y la tarea L.T.04 de la Tabla 8del product backlog. En la Tabla 18 se describe la implementación del sistema de alertas y sus criterios de aceptación correspondientes.

Tabla 18

Historia de usuario detalladas de alertar y/o notificaciones sobre ataques de phishing.

Historia de Usuario			
Nombre: H.U.04		**Usuario:** Usuario de internet	
Nombre de historia: Implementación de alertas sobre ataques de phishing.			
Prioridad del Negocio:	Alta	**Riesgo en desarrollo:**	Moderado
Puntos estimados (días):	10	**Interacción asignada:**	4
Programador(es) responsable(s):	Lissette Santamaría y Melany Nacimba		
Descripción: Como usuario, quiero recibir alertas y/o notificaciones sobre posibles ataques de phishing detectados.			
Validación (Criterios de aceptación) • El sistema debe enviar alertas cuando se detecten ataques de phishing. • El módulo de notificaciones en el backend debe ser funcional y permitir la gestión de alertas. • La interfaz de usuario debe permitir visualizar y gestionar las alertas.			

Sprint Backlog 04

En la Tabla 19 se detallan las actividades realizadas a lo largo del cuarto sprint que pertenecen al Sprint Backlog 04.

Tabla 19

Sprint Backlog 04

Sprint 04					
Fecha Inicio		**Fecha Fin**			**Jornada**
29/07/2024		08/08/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
04	Diseño del sistema de alertas.	8	29/07/2024	29/07/2024	Lissette Santamaría y Melany Nacimba.
04	Desarrollo del módulo de notificaciones en el backend.	14	30/07/2024	31/07/2024	Lissette Santamaría y Melany Nacimba.

Sprint 04					
Fecha Inicio		**Fecha Fin**			**Jornada**
29/07/2024		08/08/2024			8 horas diarias
H.U	**Tareas realizadas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsables**
04	Integración con Socket.IO	12	31/07/2024	02/08/2024	Lissette Santamaría y Melany Nacimba.
04	Configuración de WebSocket para comunicación bidireccional.	10	02/08/2024	03/08/2024	Lissette Santamaría y Melany Nacimba.
04	Desarrollo de la funcionalidad del sistema de notificaciones.	12	03/08/2024	04/08/2024	Lissette Santamaría y Melany Nacimba.
04	Implementación de la interfaz de Usuario para la emisión de alertas.	16	05/08/2024	06/08/2024	Lissette Santamaría y Melany Nacimba.
04	Pruebas unitarias del sistema de notificaciones.	16	07/08/2024	08/08/2024	Lissette Santamaría y Melany Nacimba.

Resultados del Sprint 04

Para el cuarto sprint, en el dashboard se desarrolló un sistema de alertas y notificaciones. El cual mediante el protocolo WebSocket y la librería SocketIO permite enviar de forma automática una alerta a la máquina del administrador. El uso de este protocolo se debe a que permite una comunicación bidireccional entre el backend y frontend lo cual facilita que el sistema al detectar una amenaza envié automáticamente una alerta mientras que la librería se usó ya que permite una gestión eficiente de las conexiones.En la Figura 18 se puede observar la configuración del protocolo WebSocket y la librería Socket.IO lo cual nos permite garantizar que el servidor y cliente tengan una comunicación en tiempo real. De esta manera las alertas se enviarán automáticamente del dashboard a la maquina administradora cuando el sistema detecte correos maliciosos.

Figura 18

Configuración websocket



```
    socketio.init_app(self.app)
    self.register_blueprints()

def register_blueprints(self):
    self.app.register_blueprint(controller, url_prefix="/api/v1")

def run(self):
    socketio.run(self.app, port=5000, debug=True, host='0.0.0.0')
```

```
# Emitir evento de alerta si se detecta phishing
socketio.emit('phishing_alert', {'data': email_data})
```

Para el dashboard administrativo se utilizó la librería Sonner.js. La cual permite que una vez que la alerta llegue al dashboard administrativo se pueda visualizar en la parte superior de la pantalla, tiene una duración de ocho segundos en la pantalla. El administrador debe cerrar manualmente la notificación de la alerta si ya no quiere verla en la pantalla, si por el contrario necesita más información sobre la alerta al darle clic se le desplegara información detallada sobre el análisis del correo como: ip, url y dominio.

Validación del Sistema

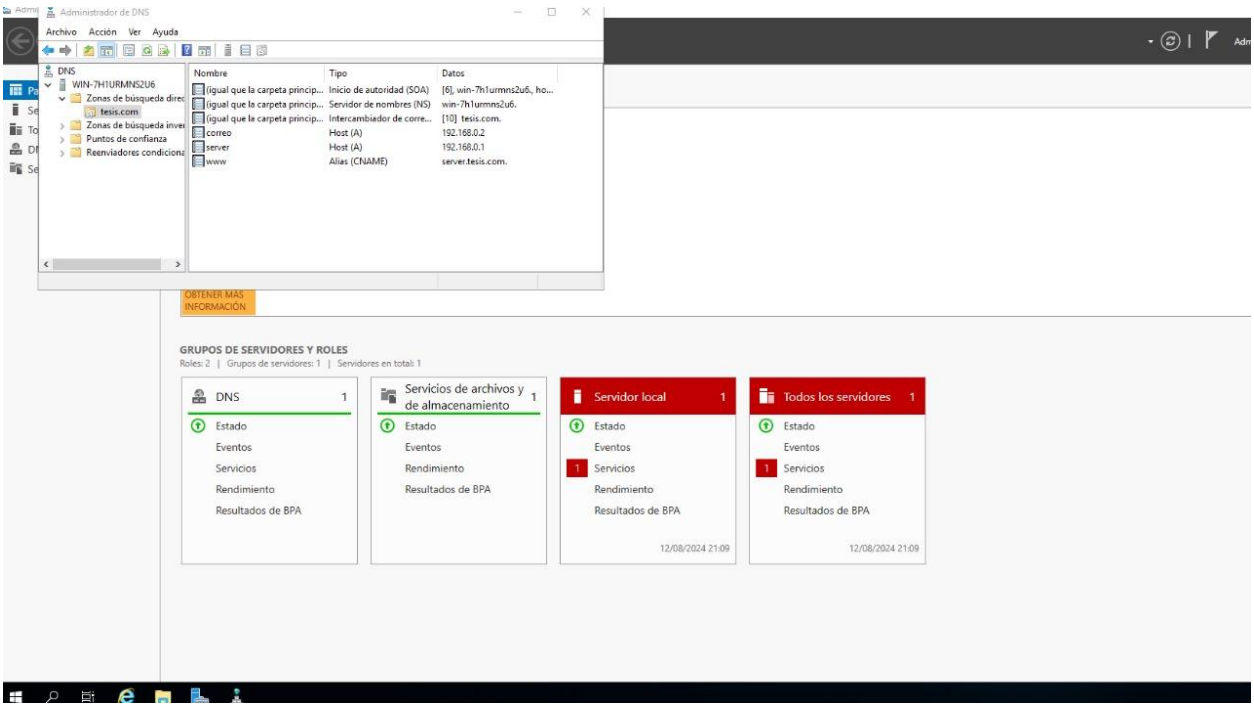
Para validar la funcionalidad del sistema desarrollado, se realizaron las respectivas pruebas en un entorno simulado. Esto hizo posible que el proceso fuese lo más semejante posible a un entorno real.

Para dichas pruebas se empezó con la configuración del servidor de correo electrónico y DNS con la ayuda de máquinas virtuales, en este caso se usó VirtualBox (Figura 18). En la primera máquina virtual, se instaló Windows Server 2019, el cual actuó como servidor DNS, gestionando la parte de las direcciones IP y los nombres de dominio usados en las pruebas. Adicionalmente, se realizó la creación de registros DNS que apunten al servidor de correo

electrónico, de forma que todo el tráfico del correo se dirija correctamente a través del sistema de análisis.

Figura 19

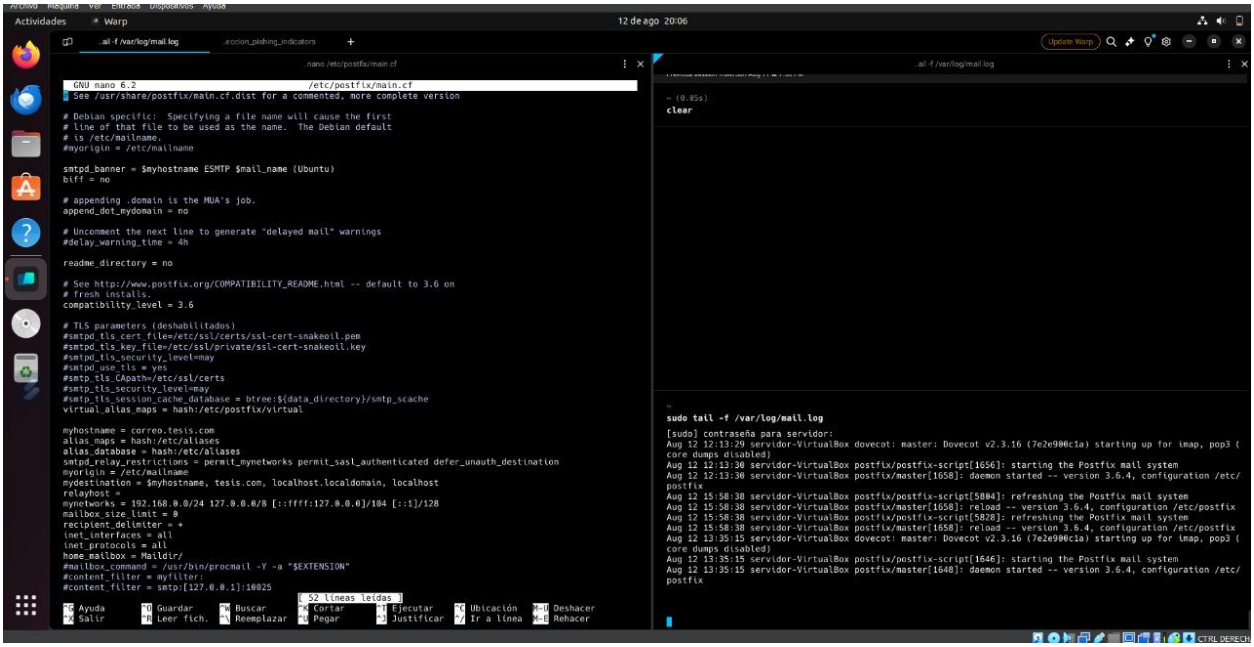
Configuración del Servidor DNS en Windows Server



En la segunda máquina virtual, se instaló Ubuntu, y se configuro Postfix como servidor de correo electrónico. Postfix se configuró para que pudiese recibir los correos enviados (Figura 19), pasándolos por un filtro a través de Python, para que analice y determine si el contenido del correo es Phishing. Este filtro usa el modelo de machine learning Random Forest previamente entrenado y APIs de inteligencia de amenazas para evaluar la seguridad de los correos.

Figura 20

Configuración de Postfix



Una vez finalizada la configuración del entorno de prueba, se procedió a probar el sistema, usando un Dataset diseñado para simular ataques de phishing, los cuales también contenían correos legítimos. Este conjunto de datos cubre lo que son técnicas de phishing, que incluyen URL, IP y dominios maliciosos, así como cuerpos de correo con lenguaje manipulado.

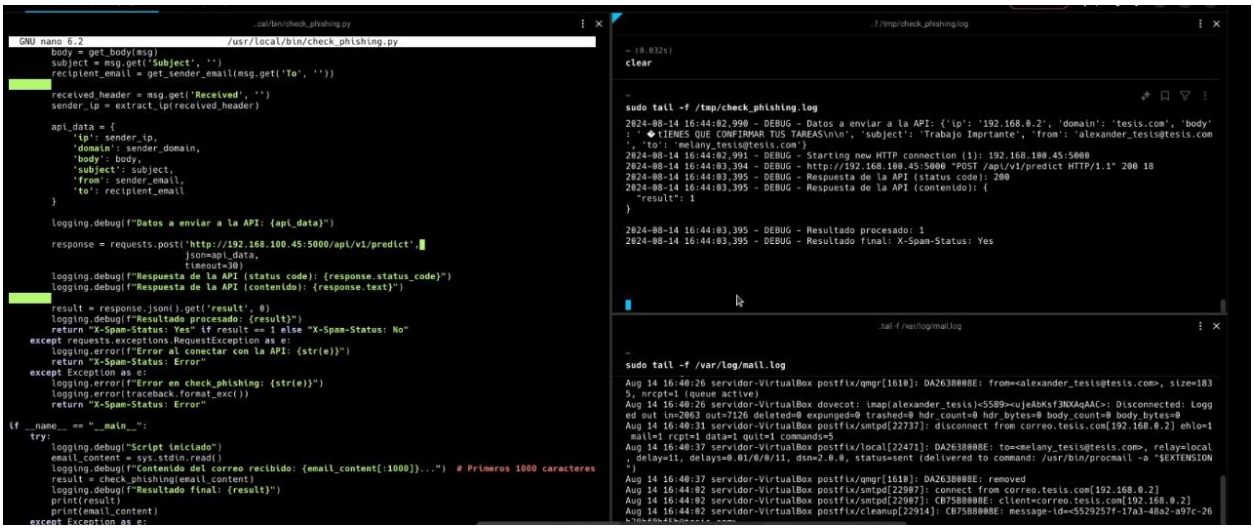
El proceso de las pruebas empieza en la primera máquina virtual (Windows Server), en la que se enviaron correos electrónicos que imitaban un ataque de phishing desde una computadora externa. Los correos pasaron a través del servidor de correo electrónico configurado previamente en la segunda máquina virtual (Ubuntu). El sistema examinó los correos, verificando los IOCs de URL, IPs, dominio y cuerpo del correo.

El sistema analiza cada correo y determina si se trata de un posible ataque de phishing o no. En el caso de que se detecte como phishing, inmediatamente se emite una alerta en el dashboard, mediante el uso del protocolo WebSocket. Las alertas enviadas ayudan al administrador a visualizar de mejor manera las amenazas detectadas.

En la Figura 20 se presenta el script de filtrado que se usó durante la prueba, junto con los logs del servidor de correo, el cual expone el proceso de envío del correo al servidor Flask para su respectivo análisis.

Figura 21

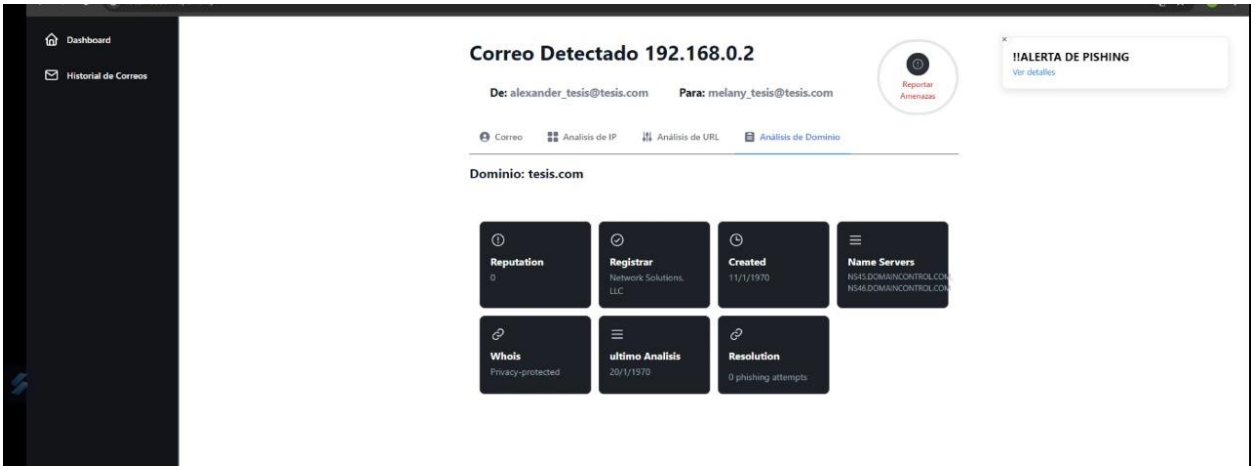
Script de filtrado y logs del servidor de correo



Para los casos en los que el sistema clasifica un correo electrónico como phishing, se emite una alerta en el dashboard, como se puede apreciar en la Figura 21.

Figura 22

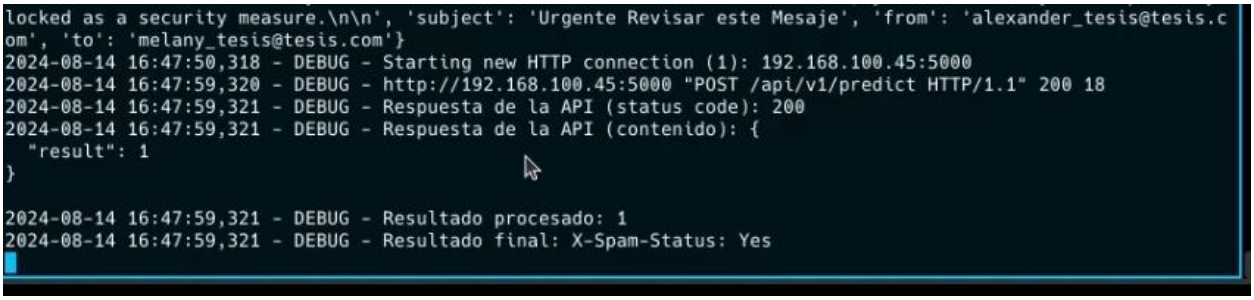
Emisión de Alerta en el dashboard



Luego, el servidor Flask responde a la solicitud del servidor de correo, señalando que el mensaje debe clasificarse como spam. En la Figura 22, se puede visualizar la interacción entre ambos servidores.

Figura 23

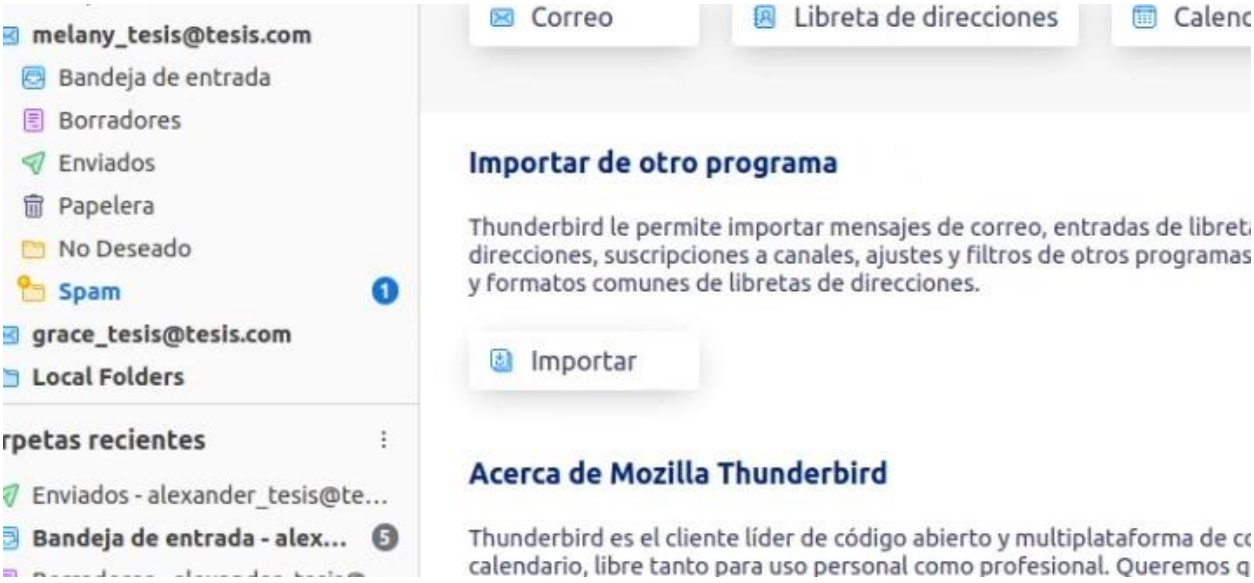
Proceso de comunicación entre los servidores



Por último, el correo que fue marcado como phishing no llega a la bandeja de entrada del destinatario, sino que se envía a la carpeta de spam. De tal forma que el usuario no se expone a contenidos maliciosos. (Figura 23)

Figura 24

Correo de phishing enviado a la carpeta de spam



Para los casos en los que el correo se identificaba como legítimo, se enviaba directamente a la bandeja de entrada, sin la emisión de alertas.

Con las pruebas realizadas se verificó la efectividad del sistema, al momento de detectar y prevenir ataques de phishing y también su capacidad, al integrarse sin mayores complicaciones en un entorno de simulación real.

4. Resultados

En esta sección se presentarán los resultados obtenidos de las diferentes pruebas que se llevaron a cabo, para evaluar el sistema. También se expondrán los diferentes análisis que se realizaron en base ellos, llegando a una conclusión final acerca del funcionamiento del programa.

Se enviaron 25 correos electrónicos, para el proceso de evaluación en cuanto al modelo de detección de phishing en URLs. Los presentes correos, cuentan con una combinación de correos legítimos y correos phishing. Y el propósito de la prueba, es analizar el desempeño en un modelo de entorno real, donde los correos pueden contener una o varias URLs en las que deben ser evaluadas.

De los 25 correos enviados, se extrajeron un total de 87 URLs, las mismas que se muestran en la Tabla 20.

Tabla 20

Resultado de pruebas Phishing en URL

Sitio Web	**Ord.**	**Pruebas Sitio Web Legítimo**	**Pruebas Sitio Web Legítimo**
		Resultado esperado	**Predicción**
Zara	1	Legítimo	Phishing
H&M	2	Legítimo	Legítimo
AliExpress	3	Legítimo	Phishing
Temu	4	Legítimo	Phishing
Tiendamia	5	Legítimo	Phishing
Alibaba	6	Legítimo	Phishing

Sitio Web	Ord.	Pruebas Sitio Web Legítimo	Pruebas Sitio Web Legítimo
		Resultado esperado	Predicción
Almacenes La Ganga	7	Legítimo	Phishing
Adidas	8	Legítimo	Legítimo
Banco del Pacífico	9	Legítimo	Phishing
Tipti	10	Legítimo	Phishing
Novicompu	11	Legítimo	Legítimo
CACPECO	12	Legítimo	Legítimo
Banco Guayaquil	13	Legítimo	Phishing
Banco Bolivariano	14	Legítimo	Phishing
Produbanco	15	Legítimo	Phishing
PedidosYa	16	Legítimo	Phishing
Bershka	17	Legítimo	Phishing
WordPress CloudClusters	18	Phishing	Phishing
Jungle Jam Play	19	Legítimo	Legítimo
Derby Arena	20	Legítimo	Legítimo
Cabanatuan City Gov	21	Legítimo	Legítimo
Luxlogics	22	Phishing	Phishing
Juguetón	23	Legítimo	Phishing
Firebase App	24	Phishing	Phishing
Banco BCR Personas	25	Phishing	Phishing

Sitio Web	Ord.	Pruebas Sitio Web Legítimo	Pruebas Sitio Web Legítimo
		Resultado esperado	Predicción
DNS Failover	26	Phishing	Phishing
Swap Protocol	27	Phishing	Phishing
Mail Dinshaak	28	Phishing	Phishing
De Prati	29	Legítimo	Phishing
Live Alireza Dalqak	30	Phishing	Phishing
Salam Bereson	31	Phishing	Phishing
v.ht	32	Phishing	Phishing
Optima Italia	33	Legítimo	Legítimo
CalState SecureVDR	34	Phishing	Legítimo
QA Cafe	35	Legítimo	Legítimo
Seewes	36	Legítimo	Legítimo
Selleck Chem	37	Legítimo	Legítimo
Firebase Page On The Way	38	Phishing	Phishing
Link und Link	39	Phishing	Phishing
Shiina Testing	40	Phishing	Phishing
Zapp Telecom	41	Phishing	Phishing
Allegiance Landscaping	42	Legítimo	Legítimo
Banccadisti	43	Phishing	Phishing
Lucy Books Online Club	44	Phishing	Phishing

Sitio Web	Ord.	Pruebas Sitio Web Legítimo	Pruebas Sitio Web Legítimo
		Resultado esperado	Predicción
Baseball Pilgrimages	45	Legítimo	Legítimo
Godaddy Sites	46	Phishing	Legítimo
Automobile 000a	47	Phishing	Phishing
Western Uranium	48	Legítimo	Legítimo
IPFS DWeb Link	49	Phishing	Phishing
Lenzs Claim	50	Phishing	Phishing
Fhewkhwjehwekjfhwehfwe	51	Phishing	Phishing
Hvjaoahgv Site	52	Phishing	Phishing
Jpostdo	53	Phishing	Phishing
WinAppUpdater	54	Phishing	Phishing
Fleek Storage	55	Phishing	Phishing
SIAM	56	Legítimo	Phishing
Teresita Producciones Peru	57	Phishing	Phishing
Hindu Devotional Blog	58	Legítimo	Legítimo
AOL Weebly Site	59	Phishing	Phishing
Miami Skin Dr	60	Legítimo	Legítimo
24.lc	61	Phishing	Phishing
Funeral Business Solution	62	Phishing	Phishing
Bellsouth Verification	63	Phishing	Phishing

Sitio Web	Ord.	Pruebas Sitio Web Legítimo	Pruebas Sitio Web Legítimo
		Resultado esperado	Predicción
Teterin Segurados	64	Phishing	Phishing
Green Math Workers Dev	65	Phishing	Phishing
Satellite BBS News	66	Phishing	Phishing
Palghat	67	Legítimo	Legítimo
YNOT	68	Legítimo	Legítimo
Legba Firebase App	69	Phishing	Phishing
Strefa Gracza	70	Phishing	Phishing
Mynembercard Point	71	Phishing	Phishing
Sarpetrol Globe Portal	72	Phishing	Phishing
Warehouse Load	73	Phishing	Phishing
Aeroweb FR	74	Legítimo	Legítimo
Direct SMTB	75	Phishing	Phishing
Fortmatic TVT	76	Phishing	Phishing
Square Site	77	Phishing	Phishing
VN	78	Phishing	Phishing
Floor Cube	79	Legítimo	Legítimo
Cordele Dispatch	80	Legítimo	Legítimo
Fleek Storage Beta	81	Phishing	Phishing
Outlook Livecom	82	Phishing	Phishing

Sitio Web	Ord.	Pruebas Sitio Web Legítimo	Pruebas Sitio Web Legítimo
		Resultado esperado	Predicción
Google Sign-In	83	Phishing	Phishing
Stiller Ffox Site	84	Phishing	Phishing
Thirty Eighth Street	85	Legítimo	Legítimo
CloudWP Credit Agricole	86	Phishing	Phishing
Alacarta	87	Legítimo	Legítimo

Nota. Esta tabla muestra los sitios web que se utilizaron para la detección de phishing

En primer lugar, el modelo se probó de forma individual para cada una de las URLs presentadas en la Tabla 21. Con las predicciones realizadas por el modelo, es posible estructurar una matriz de confusión, la misma que se muestra en la Tabla 20.

Tabla 21

Matriz de Confusión de la URL

	POSITIVO	NEGATIVO
POSITIVO	46	2
NEGATIVO	16	23

Nota. Esta muestra los resultados de la matriz de confusión

En la matriz de la Tabla 21, se muestra que 46 URLs fueron correctamente identificadas como phishing (verdaderos positivos), mientras que 23 fueron correctamente catalogadas como legítimas (verdaderos negativos). Por otro lado, 2 de las URLs fueron clasificadas incorrectamente como phishing (falsos positivos), y 16 se esperaban que fuesen phishing, pero al final se clasificaron como legítimas (falsos negativos).

A partir de los resultados de la Tabla 21, se calcularon las métricas de evaluación que se detallan en la Tabla 22.

Tabla 22

Métricas de evaluación calculadas del modelo Random Forest.

MÉTRICA DE EVALUACIÓN	RESULTADO
Accuracy	79.31%
Precision	95.83%
Recall	74.19%
F1 score	83.72%

Nota. Esta tabla muestra los resultados del modelo Random Forest

Los resultados de la Tabla 22 muestran que para la métrica de accuracy, la detección de phishing es del 79.31%, este valor revela que el modelo tiene un buen desempeño. Sin embargo, el accuracy no siempre refleja completamente la efectividad del modelo, puesto que se ve afectado por el balance entre correos phishing y no phishing en el conjunto de datos.

La métrica de precisión dio un resultado del 95.83%, lo que significa el modelo efectivamente tiene la capacidad para identificar URLs maliciosas.

Para la métrica de recall, se obtuvo el 74.19%. El valor es adecuado, puesto que es importante no pasar desapercibida alguna URL maliciosa.

Por último, para la métrica de F1 score se consiguió un 83.72%, lo que demuestra que el modelo es estable, en relación a la detección de ataques y la disminución de falsos positivos.

En los porcentajes obtenidos de las métricas, se logró obtener un rendimiento sólido en nuestro modelo de detección de phishing, por lo que, se convierte en una herramienta útil para detectar amenazas potenciales. Aunque, se presenta posibilidades de mejora en la reducción de falsos positivos, los resultados demuestran la efectividad del modelo.

Los mismos 25 correos electrónicos que se probaron anteriormente para las URLs, también fueron evaluados usando el modelo de detección de phishing que se aplicó al cuerpo del correo. Al contrario de las URLs, las predicciones basadas en el contenido del cuerpo del correo presentan un mayor desafío debido a la complejidad del lenguaje natural, por lo que representa un problema al momento de entrenar adecuadamente el modelo.

En la Tabla 23 se presenta la matriz de confusión con respecto a los resultados obtenidos del cuerpo del correo.

Tabla 23

Matriz de Confusión cuerpo del correo

	POSITIVO	**NEGATIVO**
POSITIVO	56	0
NEGATIVO	4	40

Nota. Resultado de la matriz de confusión para el cuerpo del correo

Los resultados de la Tabla 23, muestran que 56 de los correos enviados fueron identificados correctamente como phishing (verdaderos positivos), mientras que 40 fueron catalogados como legítimos (verdaderos negativos). Por otro lado, 0 correos legítimos se clasificaron erróneamente como phishing (falsos positivos) y 4 correos que son de phishing no se detectaron, por lo que se identificaron incorrectamente como legítimos (falsos negativos).

En la Tabla 24 se muestran los resultados de las métricas de evaluación que se obtuvieron a partir de la Tabla 23.

Tabla 24

Métricas de evaluación calculadas en Random Forest para el cuerpo del correo

MÉTRICA DE EVALUACIÓN	**RESULTADO**
Accuracy	96.00%
Precisión	100.00%

MÉTRICA DE EVALUACIÓN	RESULTADO
Recall	93.33%
F1	96.55%

Nota. Resultado de las métricas de evaluación calculadas en Random Forest para el cuerpo del correo

Los porcentajes de la Tabla 24, muestran que accuracy obtuvo un 96% de aciertos en las predicciones. El valor obtenido, en comparación con la predicción de las URLs, sin duda es más bajo; lo que indica la dificultad de trabajar con lenguaje natural, donde la complejidad del texto puede conducir al modelo a hacer predicciones incorrectas.

La métrica de precisión es la que evalúa la proporción de correos que fueron detectados como phishing de los que realmente lo son, esta fue del 100%, lo que demuestra que, el modelo tiene cierta capacidad para detectar correos de phishing.

En recall se obtuvo un 93.33%, lo que indica que el modelo ciertamente es eficaz en la detección de correos de phishing,

Por último, la métrica de F1 score alcanzó un 96.55%, lo que significa que el modelo muestra un rendimiento decente; sin embargo, la predicción basada en el cuerpo del correo es menos confiable dado los problemas que conlleva el procesamiento del lenguaje natural.

Los resultados que se obtuvieron corroboran que la predicción en los cuerpos del correo sin duda es más desafiante que las predicciones en URLs. Pero, a pesar de ello el modelo tiene una capacidad razonable para la detección de correos de phishing.

Por otro lado, tenemos en la Tabla 25 los resultados de la implementación de APIs de inteligencia de amenazas para la detección de sitios web legítimos y de phishing.

Tabla 25

Resultado de las Apis de Inteligencia de Amenazas

Sitio Web	Ord.	Apis de inteligencia de amenazas	Predicción
Temu	1	Legítimo	Legítimo
AliExpress	2	Legítimo	Legítimo
Zara	3	Legítimo	Legítimo
Adidas	4	Legítimo	Legítimo
H&M	5	Legítimo	Legítimo
Tiendamia	6	Legítimo	Legítimo
Alibaba	7	Legítimo	Legítimo
Almacenes La Ganga	8	Legítimo	Legítimo
PedidosYa	9	Legítimo	Legítimo
Produbanco	10	Legítimo	Legítimo
Banco del Pacífico	11	Legítimo	Legítimo
Banco Bolivariano	12	Legítimo	Legítimo
Banco Guayaquil	13	Legítimo	Legítimo
CACPECO	14	Legítimo	Legítimo
Novicompu	15	Legítimo	Legítimo
Tipti	16	Legítimo	Legítimo
Supermaxi	17	Legítimo	Legítimo
Juguetón	18	Legítimo	Legítimo
De Prati	19	Legítimo	Legítimo
Bershka	20	Legítimo	Legítimo
Cabanatuan City Gov	21	Legítimo	Legítimo
Derby Arena	22	Legítimo	Legítimo
CloudClusters	23	Phishing	Legítimo
Jungle Jam Play	24	Legítimo	Legítimo
Live Alireza Dalqak	25	Legítimo	Legítimo
Previsite	26	Legítimo	Legítimo
DNS Failover	27	Legítimo	Legítimo

Sitio Web	Ord.	Apis de inteligencia de amenazas	Predicción
Firebase App	28	Phishing	Phishing
Salam Bereson	29	Legítimo	Legítimo
Swap Protocol	30	Phishing	Phishing
Mail Dinshaak	31	Phishing	Legítimo
CalState SecureVDR	32	Phishing	Legítimo
v.ht	33	Legítimo	Legítimo
Optima Italia	34	Legítimo	Legítimo
QA Cafe	35	Legítimo	Legítimo
Seewes	36	Phishing	Legítimo
Selleck Chem	37	Legítimo	Legítimo
Firebase App Page	38	Phishing	Legítimo
Link und Link	39	Phishing	Phishing
Zapp Telecom	40	Phishing	Legítimo
Fraunhofer IVV	41	Legítimo	Legítimo
Allegiance Landscaping	42	Legítimo	Legítimo
Automobile 000a	43	Legítimo	Legítimo
Baseball Pilgrimages	44	Legítimo	Legítimo
Lucy Books Online Club	45	Legítimo	Legítimo
Banccadisti	46	Phishing	Legítimo
Western Uranium	47	Legítimo	Legítimo
Godaddy Sites	48	Phishing	Legítimo
IPFS DWeb Link	49	Phishing	Legítimo
Lenzs Claim	50	Phishing	Legítimo
Fhewkhwjehwekjfhwehfwe	51	Legítimo	Legítimo
Teresita Producciones Peru	52	Phishing	Legítimo
Gippsland Times	53	Legítimo	Legítimo
Jpostdo	54	Phishing	Phishing
Fleek Storage	55	Phishing	Legítimo
WinAppUpdater	56	Phishing	Legítimo
Hindu Devotional Blog	57	Legítimo	Legítimo

Sitio Web	Ord.	Apis de inteligencia de amenazas	Predicción
SIAM	58	Legítimo	Legítimo
AOL Weebly Site	59	Phishing	Legítimo
Miami Skin Dr	60	Legítimo	Legítimo
24.lc	61	Phishing	Phishing
Funeral Business Solution	62	Phishing	Legítimo
Bellsouth Verification	63	Phishing	Legítimo
Teterin Segurados	64	Phishing	Legítimo
Green Math Workers Dev	65	Phishing	Legítimo
YNOT	66	Legítimo	Legítimo
Palghat	67	Legítimo	Legítimo
Satellite BBS	68	Legítimo	Legítimo
Legba Firebase App	69	Phishing	Legítimo
Warehouse Load	70	Phishing	Legítimo
Ask Tog	71	Legítimo	Legítimo
Google Sign-In	72	Phishing	Legítimo
Sarpetrol Globe Portal	73	Phishing	Legítimo
Aeroweb FR	74	Legítimo	Legítimo
Mynembercard Point	75	Phishing	Legítimo
VN	76	Legítimo	Legítimo
Direct SMTB	77	Phishing	Legítimo
Square Site	78	Phishing	Legítimo
Fortmatic TVT	79	Phishing	Legítimo
Floor Cube	80	Legítimo	Legítimo
Cordele Dispatch	81	Legítimo	Legítimo
Fleek Storage Beta	82	Legítimo	Legítimo
Outlook Live	83	Phishing	Legítimo
Thirty Eighth Street	84	Legítimo	Legítimo
Stiller Ffox Site	85	Phishing	Legítimo
CloudWP Credit Agricole	86	Phishing	Legítimo
Alacarta	87	Legítimo	Legítimo

Nota. Esta tabla muestra las Apis que se utilizaron para la detección de phishing

Las APIs de inteligencia de amenazas utilizadas muestran un buen rendimiento general en la detección de sitios legítimos, pero presentan limitaciones notables en la identificación precisa de sitios de phishing. Las discrepancias entre la predicción de las APIs y la realidad sugieren la necesidad de mejoras en los algoritmos o la integración de más datos y análisis contextuales. Es crucial abordar estos errores para aumentar la confianza en las herramientas de detección y proteger a los usuarios de manera más efectiva contra ciberamenazas.

Análisis de resultados

Para la validación de los resultados que se obtuvieron del modelo de machine learning Random Forest, se procedió a dar paso al proceso de pruebas en un entorno virtual, donde se simula la realidad del envío y recepción de correos electrónicos, los cuales son tanto legítimos como de phishing. En condiciones controladas, se evaluó el rendimiento del modelo donde se trató de replicar las complejidades y variabilidades de un entorno real.

Para las pruebas, se consideraron dos aspectos importantes: las predicciones basadas en las URLs y las predicciones basadas en el cuerpo del correo. En primer lugar, se analizaron los resultados de la evaluación de URLs, las mismas que mostraron un buen rendimiento del modelo, los resultados obtenidos fueron los siguientes: acurracy del 79.31%, precisión del 95.83%, recall del 74.19% y F1 score del 83.72%. Los resultados demuestran que el modelo ciertamente es efectivo al momento de detectar URLs con phishing. Según [71] un modelo de detección de phishing que alcanza un F1 socre superior al 75% es considerado apto para aplicaciones en entornos reales, de tal forma que se valida la eficacia de nuestro modelo con respecto a los resultados que se obtuvieron.

En segundo lugar, se analizó las predicciones basadas en el cuerpo del correo, lo cual presentaron desafíos debido a la complejidad del procesamiento del lenguaje natural (PLN). A pesar de ello, el modelo logró un accuracy del 96%, precisión del 100%, recall del 93.33%, y F1 score del 96.55%. Aunque los resultados son buenos, se presentaron varias dificultades en

comparación con la predicción de URLs, esto se debe a la naturaleza no estructurada del lenguaje, que es más difícil de modelar. Según [72] el entrenar modelos de Machine Learning para procesar lenguaje natural es considerablemente más complejo y tiene una tendencia a presentar un mayor tasa de error, por lo que se justifica los resultados obtenidos de la predicción.

Las pruebas realizadas en diversos escenarios, incluyendo correos electrónicos con múltiples URLs y cuerpos de texto con diferentes niveles de complejidad lingüística, permitieron evaluar el rendimiento tanto del modelo de detección de phishing como de las APIs de inteligencia de amenazas en un entorno realista. Los resultados obtenidos demuestran la efectividad del modelo, especialmente en la detección de URLs maliciosas, donde logró un desempeño sólido. Sin embargo, las predicciones basadas en el análisis del cuerpo del texto siguen presentando desafíos, debido a la complejidad inherente al procesamiento del lenguaje natural.

En cuanto a las APIs de inteligencia de amenazas utilizadas, los resultados arrojaron un alto nivel de concordancia con las predicciones del modelo para la mayoría de los sitios legítimos y de phishing. No obstante, hubo algunas discrepancias en las que el modelo clasificó ciertos sitios como legítimos mientras que las APIs los identificaron como phishing, y viceversa. Estas diferencias resaltan la importancia de seguir mejorando la integración y el ajuste fino entre el modelo y las APIs para lograr una detección aún más precisa. En general, la combinación de estos enfoques refuerza la capacidad del sistema para identificar amenazas en un entorno digital en constante cambio, aunque queda claro que hay áreas que requieren mayor atención y optimización.

5. Conclusiones y Recomendaciones

Conclusiones

A continuación, se presentan las conclusiones a las que se llegaron una vez culminado este trabajo:

- En la investigación realizada sobre técnicas de detección de phishing e inteligencia de amenazas, se seleccionó y adoptó tres modelos de machine learning adecuados para el análisis de cuerpos de correos y URLs: Random Forest, SVM y Decision Tree. Esta investigación previa, permitió una fluida integración de los modelos con las APIs de inteligencia de amenazas.
- La metodología Scrum demostró ser una metodología ágil y eficaz para desarrollar sistemas de detección y prevención de intrusos en ciberseguridad. Permite a los equipos adaptarse rápidamente a cambios, priorizar tareas clave y mejorar la colaboración. Con roles bien definidos, se optimiza la comunicación y la eficiencia, acelerando el proceso y garantizando un producto final de alta calidad que responde a las amenazas en evolución.
- En los modelos y/o algoritmos de machine learning, Random Forest presenta una superioridad según la métrica F1 Score, donde presenta un mayor porcentaje en la evaluación en las predicciones de URLs. Aunque, en la evaluación del cuerpo del correo el modelo SVM presenta mayor porcentaje en la métrica, se sigue prefiriendo el modelo Random Forest ya que el tiempo de respuesta es significativamente más rápida, lo que ayuda a optimizar el sistema.
- El uso de inteligencia de amenazas técnica, que proporciona datos cruciales como IPs, URLs, hashes de archivos y dominios maliciosos, enriquece las herramientas de seguridad al ofrecer indicadores de compromiso (IOCs) para el monitoreo de posibles ataques. Sin embargo, al integrar estos datos con modelos de machine learning, se ha

evidenciado la necesidad de ajustes adicionales para mejorar la precisión en la clasificación de amenazas.

- El dashboard permite al administrador reportar amenazas, lo que alimenta una base de datos que almacena información sobre correos maliciosos en un servidor Flask. Este servidor utiliza los datos previamente guardados para comparar características de nuevos correos, evitando la necesidad de analizar la información repetidamente, lo que reduce el tiempo de análisis y a su vez mejora la eficiencia del sistema IDPS en la detección y prevención de ataques de phishing.
- En un ambiente controlado para las pruebas, se implementó las tecnologías Postfix, Devecot y Thunderbird para desarrollar un sistema que sea capaz de analizar e interceptar correos electrónicos; con la incorporación del modelo de machine learning e inteligencia de amenazas se facilitó la identificación de ataques de phishing basado en la información proporcionada sobre IPs, dominios y URLs reportadas. Demostrando la fiabilidad del sistema en un entorno de transacciones en línea, protegiendo a los usuarios de posibles robos de información.

Recomendaciones

Una vez concluido el desarrollo del proyecto a continuación, se presentan algunas recomendaciones a tomar en cuenta para nuevos proyectos.

- Al iniciar la investigación se debe tomar en cuenta palabras claves que puedan enriquecer a la cadena de búsqueda y así obtener artículos relevantes que sirvan como base para abordar el tema.
- El uso de fuentes confiables Web of Science, IEEE y SCOPUS son indispensables a la hora de realizar la búsqueda de artículos científicos, ya que nos permite filtrar la información y asi garantizar una búsqueda eficaz
- Para el desarrollo de este tipo de sistemas se debe tener en cuenta las características físicas del computador, ya que al no contar con las características necesarias esto podría retrasar el entrenamiento del modelo.
- Retroalimentar el sistema constantemente con datasets actualizados los cuales incorporen nuevas tácticas y estrategias de phishing para enriquecer el entrenamiento del modelo y así pueda ser más eficaz al momento de detectar una amenaza.
- Actualizar las APIs de inteligencia de amenazas, para abastecer a un mayor número de indicadores de compromiso, lo cuales permitan optimizar el análisis de las IPs, urls y dominios de un correo electrónico.

6. Referencias bibliográficas

[1] L. Tang y Q. H. Mahmoud, «A Survey of Machine Learning-Based Solutions for Phishing Website Detection», *Mach. Learn. Knowl. Extr.*, vol. 3, n.º 3, Art. n.º 3, ago. 2021, doi: 10.3390/make3030034.

[2] D. S. Fonseca, W. R. Pérez, y M. L. M. Faurés, «PASARELA DE PAGOS PARA LA SEGURIDAD DE TRANSACCIONES BANCARIAS EN LINEA».

[3] A. Martín Polo, «GAMIFICACIÓN COMO TÉCNICA DE PREVENCIÓN DE PHISING.», jul. 2024, Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://burjcdigital.urjc.es/handle/10115/38000

[4] J. Basantes y F. Patricia, «Modelo de Machine Learning para mitigar los fraudes informáticos de phishing basados en la ingeniería social en la Facultad de Ingeniería en Sistemas Electrónica e Industrial», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Tecnologías de la Información, 2023. Accedido: 31 de julio de 2024. [En línea]. Disponible en: https://repositorio.uta.edu.ec/jspui/handle/123456789/38430

[5] A. Hernández Dominguez, W. Baluja García, A. Hernández Dominguez, y W. Baluja García, «Principales mecanismos para el enfrentamiento al phishing en las redes de datos», Accedido: 31 de julio de 2024. [En línea]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000500413

[6] S. C. Apps, «Cyberattacks 2021: Statistics From the Last Year», Spanning. Accedido: 31 de julio de 2024. [En línea]. Disponible en: https://spanning.com/blog/cyberattacks-2021-phishing-ransomware-data-breach-statistics/

[7] O. Quivaqui y A. Igor, «Programa de seguridad de información ante ciber ataques de ingeniería social para empleados de una compañía de telecomunicaciones de Lima», jul. 2022, Accedido: 2 de agosto de 2024. [En línea]. Disponible en: https://tesis.pucp.edu.pe/repositorio//handle/20.500.12404/22798

[8] J. D. P. Gómez, J. I. S. Herrera, y L. F. B. Amaya, «INCIDENTES DE SEGURIDAD EN LAS PASARELAS DE PAGO».

[9] M. Sameen, K. Han, y S. O. Hwang, «PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System», *IEEE Access*, vol. 8, pp. 83425-83443, 2020, doi: 10.1109/ACCESS.2020.2991403.

[10] O. K. Sahingoz, E. BUBEr, y E. Kugu, «DEPHIDES: Deep Learning Based Phishing Detection System», *IEEE Access*, vol. 12, pp. 8052-8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[11] M. Alsaedi, F. A. Ghaleb, F. Saeed, J. Ahmad, y M. Alasli, «Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning», *Sensors*, vol. 22, n.º 9, Art. n.º 9, ene. 2022, doi: 10.3390/s22093373.

[12] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, y E. A. Elsoud, «An intelligent cyber security phishing detection system using deep learning techniques», *Clust. Comput.*, vol. 25, n.º 6, pp. 3819-3828, dic. 2022, doi: 10.1007/s10586-022-03604-4.

[13] J. V. Cubas y G. L. E. Maquen-Niño, «Machine learning model in the detection of phishing websites», *RISTI - Rev. Iber. Sist. E Tecnol. Inf.*, vol. 2022, n.º E52, pp. 161-173, 2022.

[14] A. Sahifa y M. Grami, «Implementation of Intrusion Detection Systems in order to Detect Phishing in the Banking Industry», vol. 12, n.º 2, 2020.

[15] Y. Chai, Y. Zhou, W. Li, y Y. Jiang, «An Explainable Multi-Modal Hierarchical Attention Model for Developing Phishing Threat Intelligence», *IEEE Trans. Dependable Secure Comput.*, vol. 19, n.º 2, pp. 790-803, mar. 2022, doi: 10.1109/TDSC.2021.3119323.

[16] M. Guarascio, N. Cassavia, F. S. Pisani, y G. Manco, «Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection», *Future Gener. Comput. Syst.*, vol. 135, pp. 30-43, oct. 2022, doi: 10.1016/j.future.2022.04.028.

[17] R. Jayaraj, A. Pushpalatha, K. Sangeetha, T. Kamaleshwar, S. Udhaya Shree, y D. Damodaran, «Intrusion detection based on phishing detection with machine learning», *Meas. Sens.*, vol. 31, p. 101003, feb. 2024, doi: 10.1016/j.measen.2023.101003.

[18] E. Zhu, Y. Chen, C. Ye, X. Li, y F. Liu, «OFS-NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network», *IEEE Access*, vol. 7, pp. 73271-73284, 2019, doi: 10.1109/ACCESS.2019.2920655.

[19] D. Kumar y K. P. Kumar, «Artificial Intelligence based Cyber Security Threats Identification in Financial Institutions Using Machine Learning Approach», en *2023 2nd International Conference for Innovation in Technology (INOCON)*, mar. 2023, pp. 1-6. doi: 10.1109/INOCON57975.2023.10100967.

[20] M. Tene y H. Edmundo, «Desarrollo de un sistema E-commerce B2C para gestionar ofertas en productos y servicios de la casa musical Macro Show de Riobamba», bachelorThesis, Riobamba, Universidad Nacional de Chimborazo, 2023. Accedido: 2 de agosto de 2024. [En línea]. Disponible en: http://dspace.unach.edu.ec/handle/51000/11054

[21] H. H. Coronado Huamán, A. Han, y L. Sanz García, «Detección automática de sitios web fraudulentos», Accedido: 31 de julio de 2024. [En línea]. Disponible en: https://docta.ucm.es/entities/publication/ff6bdc95-25c1-44d7-94af-55bce8e3f245

[22] M. R. Peñafiel Suárez, «Ingeniería social en una institución de educación superior aplicando técnicas computacionales y no computacionales», bachelorThesis, La Libertad: Universidad Estatal Península de Santa Elena. 2022, 2022. Accedido: 9 de agosto de 2024. [En línea]. Disponible en: https://repositorio.upse.edu.ec/handle/46000/8683

[23] C. E. L. Grande y R. S. Guadrón, «Ingeniería Social: El Ataque Silencioso», 2015.

[24] I. A. D. Bank, «Informe anual del Banco Interamericano de Desarrollo 2020: Reseña del año», *IDB Publ.*, mar. 2021, doi: 10.18235/0003118.

[25] C. S. G. Ibarra, C. A. N. Tapia, A. M. I. Tene, R. J. E. Toapanta, y G. S. E. Ormaza, «Análisis de los Ataques de Ingeniería Social en Ecuador», *Cienc. Lat. Rev. Científica Multidiscip.*, vol. 8, n.º 1, Art. n.º 1, feb. 2024, doi: 10.37811/cl_rcm.v8i1.9777.

[26] C. Sekhar Bhusal, «Systematic Review on Social Engineering: Hacking by Manipulating Humans», *J. Inf. Secur.*, vol. 12, n.º 01, pp. 104-114, 2021, doi: 10.4236/jis.2021.121005.

[27] S. Kumar Birthriya y A. K. Jain, «A Comprehensive Survey of Phishing Email Detection and Protection Techniques», *Inf. Secur. J. Glob. Perspect.*, vol. 31, n.º 4, pp. 411-440, jul. 2022, doi: 10.1080/19393555.2021.1959678.

[28] Z. Alkhalil, C. Hewage, L. Nawaf, y I. Khan, «Fronteras | Ataques de phishing: un estudio exhaustivo reciente y una nueva anatomía», doi: 10.3389/fcomp.2021.563060.

[29] Antonio Hernández Dominguez y A. H. Dominguez, «Sistema para la detección de ataques PHISHING utilizando correo electrónico.», vol. 17, n.º 2, pp. 60-70, ene. 2019.

[30] N. Odeh, D. Eleyan, y A. Eleyan, «A SURVEY OF SOCIAL ENGINEERING ATTACKS: DETECTION AND PREVENTION TOOLS», sep. 2021.

[31] I. El Naqa y M. J. Murphy, «What Is Machine Learning?», en *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li, y M. J. Murphy, Eds., Cham: Springer International Publishing, 2015, pp. 3-11. doi: 10.1007/978-3-319-18305-3_1.

[32] J. Bobadilla, *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. Ediciones de la U, 2021.

[33] J. Su, «A Fast Decision Tree Learning Algorithm».

[34] K. Saraswat, M. Devi, D. Professor, y A. Guleria, «Decision Tree Based Algorithm for Intrusion Detection», *Int. J. Adv. Netw. Appl.*, vol. 07, pp. 2828-2834, ene. 2016.

[35] A. B. Shaik y S. Srinivasan, «A Brief Survey on Random Forest Ensembles in Classification Model», en *International Conference on Innovative Computing and*

Communications, S. Bhattacharyya, A. E. Hassanien, D. Gupta, A. Khanna, y I. Pan, Eds., Singapore: Springer, 2019, pp. 253-260. doi: 10.1007/978-981-13-2354-6_27.

[36] «Detección de Ataques de Intrusión con Algoritmos de Machine Learning». Accedido: 8 de agosto de 2024. [En línea]. Disponible en: https://repositorio.uniandes.edu.co/server/api/core/bitstreams/de635c47-f374-41dc-b4f0-ac5dc0b2cc97/content

[37] A. Shmilovici, «Support Vector Machines», en *Data Mining and Knowledge Discovery Handbook*, O. Maimon y L. Rokach, Eds., Boston, MA: Springer US, 2010, pp. 231-247. doi: 10.1007/978-0-387-09823-4_12.

[38] D. F. Huertas Morales y H. A. Ortiz Camargo, «Detección de ataques Probe en el dataset NSL KDD, mediante el uso del modelo de clasificación de máquinas de soporte vectorial (SVM)», abr. 2023, Accedido: 8 de agosto de 2024. [En línea]. Disponible en: http://hdl.handle.net/11349/31174

[39] F. Enrique y C. Estrella, «Implementación de una plataforma de inteligencia de amenazas.».

[40] L. R. Quevedo, «Tecnologías de defensa frente a inteligencia de amenazas y ciberataques», *InnDev*, vol. 3, n.º 1, Art. n.º 1, abr. 2024, doi: 10.69583/inndev.v3n1.2024.94.

[41] «VargasCejer2024.pdf». Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://repository.universidadean.edu.co/bitstream/handle/10882/13563/VargasCejer2024.pdf?sequence=2&isAllowed=y

[42] Agencija Europske unije za kibersigurnost (Tijelo ili agencija EU-a), *ENISA threat landscape report 2018: 15 top cyber threats and trends*. Ured za publikacije Europske unije, 2019. Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://data.europa.eu/doi/10.2824/622757

[43] W. Tounsi y H. Rais, «A survey on technical threat intelligence in the age of sophisticated cyber attacks», *Comput. Secur.*, vol. 72, pp. 212-233, ene. 2018, doi: 10.1016/j.cose.2017.09.001.

[44] C. Chica Vargas y D. F. Pinto Prada, «Definición de un modelo de estrategia de inteligencia de amenazas cibernéticas en entidades gubernamentales de Colombia como mecanismo de anticipación de riesgos cibernéticos», Master Thesis, Maestría en Gerencia de Sistemas de Información y Proyectos Tecnológicos Virtual, 2024. Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://repository.universidadean.edu.co/handle/10882/13563

[45] A. Salem, S. Banescu, y A. Pretschner, «Maat: Automatically Analyzing VirusTotal for Accurate Labeling and Effective Malware Detection», *ACM Trans Priv Secur*, vol. 24, n.º 4, p. 25:1-25:35, jul. 2021, doi: 10.1145/3465361.

[46] «abuseipdb.com Opiniones | el sitio web es seguro o falso», abuseipdb.com. Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://www.scamadviser.com/es/comprobar-sitio-web/abuseipdb.com

[47] M. Salva, «Desplegar la herramienta “Bro IDS” y su posterior explotación para el análisis de actividades sospechosas en la red».

[48] K. S. & J. Sutherland, «La Guía Definitiva de Scrum: Las Reglas del Juego», nov. 2020, Accedido: 8 de agosto de 2024. [En línea]. Disponible en: https://repositorio.uvm.edu.ve/handle/123456789/59

[49] «INFORMACIÓN BÁSICA DE SCRUM». Accedido: 8 de agosto de 2024. [En línea]. Disponible en: https://goodagile.com/scrumprimer/scrumprimer_es.pdf

[50] «Visual Studio: IDE y Editor de código para desarrolladores de software y Teams», Visual Studio. Accedido: 12 de agosto de 2024. [En línea]. Disponible en: https://visualstudio.microsoft.com/es/

[51] «¿Qué es Python? - Explicación del lenguaje Python - AWS», Amazon Web Services, Inc. Accedido: 12 de agosto de 2024. [En línea]. Disponible en: https://aws.amazon.com/es/what-is/python/

[52] «La página de inicio de Postfix». Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://www.postfix.org/

[53] «Manual de Dovecot — Documentación de Dovecot». Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://doc.dovecot.org/

[54] «¿Qué puede hacer con Windows Server 2019 (+ nuevas funciones)?» Accedido: 12 de agosto de 2024. [En línea]. Disponible en: https://softtrader.es/que-puede-hacer-con-windows-server-2019-nuevas-funciones/

[55] «Qué es y qué ventajas tiene Thunderbird, el servicio de correo que se ve en el laptop de Mark Zuckerberg», *BBC News Mundo*. Accedido: 12 de agosto de 2024. [En línea]. Disponible en: https://www.bbc.com/mundo/noticias-36609622

[56] «Empezando – React». Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://es.legacy.reactjs.org/docs/getting-started.html

[57] «Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.» Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://tailwindcss.com/

[58] A. Casero, «¿Qué es TypeScript? | KeepCoding Bootcamps». Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://keepcoding.io/blog/typescript/#%C2%BFQue_es_TypeScript

[59] «Oracle VM VirtualBox». Accedido: 12 de agosto de 2024. [En línea]. Disponible en: https://www.oracle.com/es/virtualization/virtualbox/

[60] R. Borja-Robalino, A. Monleon-Getino, y J. Rodellar, «Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning», *RISTI - Rev. Iber. Sist. E Tecnol. Inf.*, vol. 30, pp. 184-196, jun. 2020.

[61] «T-ESPEL-SOF-0126.pdf». Accedido: 11 de agosto de 2024. [En línea]. Disponible en: https://repositorio.espe.edu.ec/bitstream/21000/37728/1/T-ESPEL-SOF-0126.pdf

[62] M. Hossin, M. N. Sulaiman, A. Mustapha, N. Mustapha, y R. W. Rahmat, «A hybrid evaluation metric for optimizing classifier», en *2011 3rd Conference on Data Mining and Optimization (DMO)*, Putrajaya, Malaysia: IEEE, jun. 2011, pp. 165-170. doi: 10.1109/DMO.2011.5976522.

[63] J. Yao y M. Shepperd, «The impact of using biased performance metrics on software defect prediction research», *Inf. Softw. Technol.*, vol. 139, p. 106664, nov. 2021, doi: 10.1016/j.infsof.2021.106664.

[64] W. Drzewiecki, «Thorough statistical comparison of machine learning regression models and their ensembles for sub-pixel imperviousness and imperviousness change mapping», *Geod. Cartogr.*, vol. 66, dic. 2017, doi: 10.1515/geocart-2017-0012.

[65] M. Bekkar y D. H. K. Djemaa, «Evaluation Measures for Models Assessment over Imbalanced Data Sets», 2013.

[66] «SOFTWARE RELEASE PLANNING | Handbook of Software Engineering and Knowledge Engineering». Accedido: 15 de agosto de 2024. [En línea]. Disponible en: https://www.worldscientific.com/doi/abs/10.1142/9789812775245_0013

[67] «Anteproyecto_Requerimientos_en_Metodologías_Agiles.pdf». Accedido: 8 de agosto de 2024. [En línea]. Disponible en: https://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/pub/file/Tesis/Anteproyecto_Requerimientos_en_Metodolog%C3%ADas_Agiles.pdf

[68] M. Cohn, «User Stories and User Story Examples by Mike Cohn», Mountain Goat Software. Accedido: 8 de agosto de 2024. [En línea]. Disponible en: https://www.mountaingoatsoftware.com/agile/user-stories

[69] M. Kovtun, «Scalable Honeypot Monitoring and Analytics».

[70] M. E. Castillo Veloz y K. J. Chuquitarco Velasco, «Sistema de detección de intrusos en sitios web, usando modelos y,o algoritmos de Machine Learning : caso práctico Phishing Google Chrome.», bachelorThesis, Universidad de las Fuerzas Armadas ESPE. ESPEL. Carrera de Ingeniería en Software., 2023. Accedido: 10 de agosto de 2024. [En línea]. Disponible en: http://repositorio.espe.edu.ec/jspui/handle/21000/35751

[71] J. C. W. Zhou y Y. Cao, «Phishing Detection via Machine Learning: A Survey», *IEEE Access*, vol. vol.8, n.º pp. 37358-37371, 2020.

[72] X. Yin, H. Wang, y J. Li, «Challenges and Advances in Natural Language Processing for Phishing Detection», *IEEE Trans. Knowl. Data Eng.*, vol. 31, n.º 7, pp. 1348-1362, 2023.

7. Anexos