Sistemas de detección de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing.

Caisa Llano, Dennis Sebastián y Guevara Jiménez, Adrian Fernando

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Trabajo de integración curricular, previo a la obtención del título de Ingeniero de Software

Ing. Corral Diaz, Maria Alexandra, Msc.

21 de agosto del 2023

Latacunga

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Reporte de verificación de contenidos

Copyleaks
Plagiarism report

Sistemas de detección de intrusos en...

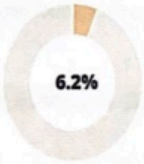
Scan details

Scan time: August 24th, 2023 at 16:56 UTC

Total Pages: 101

Total Words: 25121

Plagiarism Detection



Types of plagiarism		Words
Identical	0.3%	71
Minor Changes	0.1%	22
Paraphrased	5%	1256
Omitted Words	13%	3277

AI Content Detection

Text coverage
- AI text
- Human text

.....

Ing. Corral Diaz, María Alexandra, Msc

C. C.: 0501970487

ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de integración curricular: "**Sistemas de detección de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing**" fue realizado por los señores **Caisa Llano, Dennis Sebastián** y **Guevara Jiménez, Adrian Fernando**, 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 analizada en su totalidad por la herramienta de verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 21 de agosto de 2023

.....

Ing. Corral Diaz, María Alexandra, Msc

C. C.: 0501970487

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Auditoria

Nosotros, **Caisa Llano, Dennis Sebastián y Guevara Jiménez, Adrian Fernando**, con cédulas de ciudadanía N° ° 0550049688 y 1750781955, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: "**Sistemas de detección de intrusos en sitios web, usando Indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing**", es 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, 21 de agosto de 2023

Caisa Llano, Dennis Sebastián

C .C: 0550049688

Guevara Jiménez, Adrian Fernando

C. C: 1750781955

ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Caisa Llano, Dennis Sebastián** y **Guevara Jiménez, Adrian Fernando**, con cédulas de ciudadanía N° 0550049688 y 1750781955, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: "**Sistemas de detección de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing**", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 21 de agosto de 2023

.....

Caisa Llano, Dennis Sebastián

C. C.: 0550049688

.....

Guevara Jiménez, Adrian Fernando

C. C.: 1750781955

Dedicatoria

Deseo dedicar este logro a Dios, quien me ha otorgado el regalo más significativo y puro: la guía de mis padres, Byron Caisa y Maria Llano junto a mi hermano Saúl, ellos son las personas más importantes en mi vida, quienes me han inculcado valores que han moldeado la persona que soy. Sus enseñanzas, consejos y amor han sido mi constante fuente de motivación para enfrentar desafíos y perseverar en la búsqueda de mis sueños y metas.

Cada gota de esfuerzo y dedicación en este camino está dedicada a ellos, como un tributo a su incansable apoyo. A través de este logro, quiero honrar su legado y el amor inquebrantable que me han brindado. Mi corazón está lleno de amor y gratitud hacia ellos, pues han sido el faro que ha iluminado mi camino.

Por y para ellos, con amor y agradecimiento infinitos.

Dennis Sebastián Caisa Llano

Ecuador, agosto 2023

Agradecimiento

Quiero expresar mi profundo agradecimiento primero a Dios por la guía espiritual que siempre me rodeó, a mi tutora Ingeniera Alexandra Corral por su orientación constante, apoyo invaluable y paciencia a lo largo de este proceso de investigación. Agradezco sinceramente a mi familia y amigos por su amor, aliento y apoyo inquebrantable durante toda mi trayectoria académica. Mi gratitud se dirige a mis profesores y mentores, quienes han contribuido significativamente a mi formación y crecimiento intelectual. Deseo agradecer a la Universidad de las Fuerzas Armadas ESPE por proporcionar los recursos y el entorno para llevar a cabo este proceso de mi área académica.

Quiero agradecer a mis compañeros de investigación, cuya colaboración y voluntad de compartir sus experiencias han enriquecido este trabajo de manera invaluable. Agradezco al Ingeniero José Carrillo por sus valiosas sugerencias y comentarios durante las discusiones y revisiones críticas. Y no puedo pasar por alto la ayuda proporcionada por Adrián quien compartió sus conocimientos y brindó apoyo en momentos cruciales.

Mi más profundo agradecimiento va hacia mis padres y hermano, cuyo amor, comprensión y paciencia me han motivado a seguir adelante en cada paso de este viaje. A nada y nadie lo puedo dejar a un lado y dedico este logro a todas esas personas que sin importar las circunstancias estuvieron a mi lado día tras día a lo largo de esta travesía académica

Dennis Sebastián Caisa Llano

Ecuador, agosto 2023

Dedicatoria

Quiero dedicar este significativo logro a todos aquellos que son especiales en mi vida. Sin su constante apoyo, este gran paso en mi viaje de aprendizaje y crecimiento no habría sido posible. En particular, deseo expresar mi profundo agradecimiento a mi familia. Mi madre, Cesilia, ha sido mi pilar fundamental en los momentos más desafiantes, brindándome su amor incondicional. Mi padre, Ángel, ha compartido su tiempo y apoyo. A mis queridas hermanas, Lizeth y Johanna, les agradezco por su amor inmenso y sus valiosas lecciones, las cuales me han guiado con sabiduría en este recorrido.

Es con el calor de sus palabras y el aliento de su apoyo que encuentro la fuerza para seguir avanzando en mi camino. Gracias por ser parte de este capítulo en mi vida y por ser el fundamento que me impulsa a continuar.

Adrian Fernando Guevara Jiménez

Ecuador, agosto 2023

Agradecimiento

Quiero expresar el más sincero agradecimiento a Dios por ser mi guía en este gran camino de aprendizaje y enseñanzas, por cada día iluminarme y ayudarme en cada paso de mi vida.

Agradecer también a mi familia por siempre brindarme el más sincero apoyo y amor que me ayudaron a formarme como persona y ser humano para poder llegar a cumplir unos de mis sueños y metas.

Agradezco también a mis profesores quienes fueron mi guía de enseñanza fortaleciendo cada día más mi conocimiento y aprendizajes dentro de mi proceso de estudio.

A mis amigos que gracias a su gran amistad hemos conseguido dar este nuevo paso en nuestras vidas, ayudándome a que mi instancia en la universidad sea de las mejores experiencias y enseñanzas. Cada riza, llanto significo mucho. Agradezco por tan buenos amigo que llegaron a mi vida.

Adrian Fernando Guevara Jiménez

Ecuador, agosto 2023

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

En la actualidad el internet se ha convertido en una de las herramientas indispensables para tareas cotidianas, pero por desgracia vivimos en una sociedad en la que algunas personas buscan posibilidades de causar daño a los cibernautas, siendo el caso de los phisher, quienes engañan maliciosamente a sus víctimas con el fin de robarles sus datos personales para acceder a diversas plataformas, incluidas las bancarias y personales. Los phishers estafan a sus víctimas creando sitios web falsos (con phishing), asemejándose a los auténticos, estos sitios web utilizan un formulario que permite a las víctimas introducir sus datos, para luego ser robados. Del mismo modo existen los anti-phisher que usan modelos y/o algoritmos de Machine Learning en el campo de la ciberseguridad para obtener mejores resultados en la detección de los ataques, tal es el caso implementado en la resolución de este proyecto, conjunto a los Sistemas de Detección de Intrusos (IDS) incorporando un extra de seguridad con Indicadores de Compromiso (IoC) adaptándonos a la metodología ágil Scrum, llegando a realizar pruebas en la aplicación para su revisión y validación trabajando tanto en un entorno de entrenamiento como en un entorno real/simulado las herramientas Zphisher y NexPhisher como comparativa de los casos que se pueden prevenir con la extensión Phishing Impact de los cuales los resultados obtenido fueron los deseados.

Palabras clave: Ciberseguridad, Aprendizaje Automático, Detección, Amenazas, Aplicación.

Abstract

Nowadays the Internet has become one of the indispensable tools for daily tasks, but unfortunately, we live in a society in which some people look for possibilities to cause harm to cybernauts, being the case of phishers, who maliciously deceive their victims to steal their personal data to access various platforms, including banking and personal ones. Phishers scam their victims by creating fake (phishing) websites, resembling the real thing, these websites use a form that allows victims to enter their data, which is then stolen. Similarly there are anti-phisher that use Machine Learning models and/or algorithms in the field of cybersecurity to obtain better results in the detection of attacks, such is the case implemented in the resolution of this project, together with the Intrusion Detection Systems (IDS) incorporating an extra security with Indicators of Compromise (IoC) adapting to the agile methodology Scrum, We tested the application for review and validation, working both in a training environment and in a real/simulated environment using the Zphisher and NexPhisher tools as a comparison of the cases that can be prevented with the Phishing Impact extension, of which the results obtained were the desired ones.

Key words: Cybersecurity, Machine Learning, Detection, Threats, Application.

Capítulo I

Introducción

Propósito y contextualización del tema

En la actualidad, Internet se ha convertido en una parte indispensable de la vida de las personas. Es difícil imaginar un mundo sin esta poderosa herramienta. Ha transformado significativamente la forma en que vivimos y trabajamos, especialmente desde el inicio de la pandemia a finales de 2019. Muchas industrias se han visto obligadas a migrar hacia servicios en línea para garantizar su supervivencia en esta nueva realidad (Tang Y Mahmoud, 2021). Como resultado, los usuarios de Internet dejan una gran cantidad de información confidencial en línea, incluyendo datos de inicio de sesión, contraseñas, información de tarjetas de crédito y preguntas de seguridad. Esto convierte a las industrias y a los usuarios en objetivos principales para los ciberdelincuentes (Tang Y Mahmoud, 2021).

En términos generales, Internet es una infraestructura sin control y con deficiencias en cuanto a seguridad y el control de la información, lo cual conlleva un conjunto amplio de vulnerabilidades para los usuarios, los sistemas y la infraestructura física. Estas amenazas pueden ser aprovechadas para causar daños financieros, robo de identidad e incluso la pérdida de usuarios, como es el caso del comercio (Tang Y Mahmoud, 2021), entre otras consecuencias preocupantes.

Se han identificado diversas amenazas en Internet que utilizan métodos y formas de ataques para comprometer la seguridad de la red. Estos problemas de ciberseguridad incluyen el Phishing, Malware, Inyección de SQL, explotación de vulnerabilidades día cero, ataques de denegación de servicio (DoS), ataques de intermediario (man-in-the-middle), y la categorización de tunelización de DNS. Según la investigadora de seguridad Nadezhda Demidoca de Kaspersky Lab, el Phishing se considera una de las amenazas más significativas a nivel mundial, habiendo registrado ataques de este tipo desde el año

2017 (Jaramillo Basantes, 2023). Actualmente, el Phishing se destaca como uno de los ataques cibernéticos más simples pero efectivos para robar información personal y corporativa. Mediante técnicas de ingeniería social y subterfugios técnicos, el Phishing busca obtener datos de identidad personal y credenciales (Domínguez Y García, 2021).

Según el informe de CISCO sobre las tendencias en amenazas de seguridad, se destaca que alrededor del 90% de las brechas de datos ocurridas en el año 2021 fueron resultado de ataques de Phishing (Apps, 2022). Además, una investigación llevada a cabo por Ponem Institute y analizada por IBM Security en 2022 reveló que los ataques de Phishing son la causa más común de vulnerabilidades, siendo también los más costosos con un promedio del 16% de los costos de violación de datos, que ascienden a aproximadamente 4.91 millones de dólares (Coste de la vulneración de datos 2022 - España | IBM, s. f.). Estos hallazgos destacan que, tanto a nivel individual como en el ámbito organizacional, sufrir un ataque de Phishing puede resultar en pérdidas económicas significativas.

En esta coyuntura, la seguridad informática juega un papel fundamental en la protección de la información privada y financiera tanto de los usuarios como de las empresas (Cabrera, 2017). En la actualidad, se han llevado a cabo numerosos estudios e investigaciones orientados a detectar, prevenir y mitigar los ataques de phishing. Estas investigaciones han concluido que una de las soluciones más frecuentes radica en la utilización de modelos y técnicas de Machine Learning, Deep Learning e incluso combinaciones de ambas (Domínguez Y García, 2021).

Sin embargo, a medida que avanza la tecnología, los sitios web que se dedican al phishing han evolucionado con el tiempo, lo que exige el desarrollo de sistemas de detección capaces de identificar patrones distintivos para diferenciar entre sitios web legítimos y aquellos que contienen phishing (Coronado, 2020). Este proceso requiere un esfuerzo constante y consume mucho tiempo si se realiza

manualmente, por lo tanto, la mejor opción es implementar sistemas de detección automáticos que utilicen técnicas de Inteligencia Artificial, en particular el aprendizaje automático (Machine Learning).

El propósito del proyecto es desarrollar un Sistema de detección de intrusos (IDS) enfocado a la detección de sitios web con Phishing, como una extensión de Google Chrome, a través de motores de búsqueda, utilizando indicadores de compromiso (IOCs) y modelos y/o algoritmos de Machine Learning (ML), con el propósito de automatizar el proceso de identificación de sitios web con Phishing y garantizar una alta precisión en la detección, se busca implementar sistemas que utilicen técnicas de Inteligencia Artificial, específicamente Machine Learning. Esto permitirá agilizar la identificación de sitios web fraudulentos y distinguirlos de aquellos legítimos, ofreciendo así una solución eficiente y confiable en la lucha contra el Phishing. En conclusión, esta extensión propuesta podría ser una solución viable para abordar la problemática actual de robo de información a través de sitios web con Phishing. Al automatizar la detección y proporcionar una alta precisión en la identificación de sitios fraudulentos, se podría reducir significativamente el riesgo de caer en trampas de Phishing y proteger la información personal y confidencial de los usuarios en línea.

Justificación

En los últimos años, el Internet ha experimentado un crecimiento notable gracias a su transformación en el desarrollo de actividades relacionadas con el trabajo, la educación y otras tareas comunes en la vida diaria. Esta evolución ha permitido que cada vez más personas dependan de Internet para realizar sus labores, acceder a recursos educativos y llevar a cabo diversas actividades en su rutina diaria. Esta tendencia ha sido impulsada por los avances tecnológicos, la conectividad global y la creciente adopción de dispositivos digitales, lo que ha llevado a una mayor integración del Internet en diferentes aspectos de la vida moderna (Delgado, 2022). El intercambio de información en línea conlleva vulnerabilidades y riesgos, como la pérdida de datos privados, el fraude de datos, los daños monetarios,

los robos de cuentas e incluso la posibilidad de que los hackers suplanten nuestra identidad una vez que obtienen nuestros datos privados. Estas amenazas son cada vez más relevantes en un mundo digital donde la seguridad de la información es primordial (Li et al, 2021). Así es, el uso de Internet sin ninguna medida de seguridad expone a una amplia gama de amenazas. Estas amenazas son aprovechadas por individuos malintencionados, comúnmente conocidos como hackers, quienes no solo causan pérdidas de información, sino que también generan daños económicos dirigidos tanto a individuos como a empresas. Estas pérdidas económicas pueden ser el resultado de robos de información confidencial, fraudes financieros, ataques de ransomware, entre otros delitos cibernéticos perpetrados con el objetivo de obtener beneficios económicos ilegales. Por tanto, es crucial tomar medidas de seguridad adecuadas para protegerse contra estas amenazas y salvaguardar tanto la información personal como los activos financieros (Dominguez Y García, 2021).

Así es, un Sistema de Detección de Intrusos (IDS) para la detección de phishing en sitios web sería de suma importancia para salvaguardar la seguridad de los datos de los usuarios mientras navegan por Internet. Esto permite tomar medidas proactivas para proteger la información y los datos de los usuarios. En este trabajo, se propuso el desarrollo de IDS automáticas que aprovechen las ventajas de obtener Indicadores de Compromiso (IOC) a través de una extensión para el navegador Google Chrome. Esta extensión sería de fácil instalación y estaría al alcance de los usuarios, permitiéndoles fortalecer la seguridad de su navegación web. Además, implementando modelos y/o algoritmos de Machine Learnig eficientes de extracción de características de IOC a partir de una URL y así permitir que la extensión del navegador actúa como una capa adicional de seguridad, proporcionando una detección proactiva y alertas oportunas a los usuarios cuando se detecten indicios de phishing. De esta manera, los usuarios podrán tomar medidas preventivas para proteger sus datos y evitar caer en trampas de phishing mientras navegan por Internet utilizando su navegador Chrome.

Objetivos

Objetivo general

Desarrollar un sistema de detección de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso Práctico Phishing Google Chrome

Objetivos específicos

- Entender la vanguardia en cuanto a los indicadores de compromiso y su utilidad en la detección de intrusos en sitios web mediante el uso de técnicas de phishing en el navegador Google Chrome.
- Poner en marcha un sistema de detección de intrusos en páginas web mediante el desarrollo de una extensión para Google Chrome, empleando métodos de aprendizaje automático.
- Evaluar los resultados obtenidos, analizar y corregir posibles errores identificados en los indicadores de compromiso del sistema de detección de intrusos.

Metodología

El presente proyecto tiene como propósito fundamental el desarrollar un sistema capaz de detectar intrusos y reconocer sitios web con técnicas de phishing, con el fin de lograr los objetivos establecidos. Para ello, se implementarán tres fases en el proceso:

La fase I parte del análisis de la literatura científica y selección de indicadores de compromiso y algoritmos de Machine Learning. Se utilizan métodos teóricos como el método histórico-lógico y análisis-síntesis para formular el marco teórico del proyecto. Se estudian las características de los sitios web que permiten determinar si un sitio web tiene o no phishing, particularmente a través de la URL. Se investigan y seleccionan los indicadores de compromiso y algoritmos de Machine Learning (ML) más

utilizados y con alta precisión en la detección de phishing en sitios web. Se utilizan bases de datos bibliográficas (pueden ser públicos y privados) como SCOPUS y plataformas de Data Science como KAGGLE y MISP para recopilar información relevante.

La fase II procede en el desarrollo del sistema de detección de intrusos y la extensión de Google Chrome, seleccionando conjuntos de datos que contengan las características seleccionadas en la fase anterior. Luego se crean diferentes escenarios con variaciones en la cantidad de características, sitios web legítimos y sitios web con phishing para continuar con la implementación de algoritmos de Machine Learning seleccionados anteriormente. Para luego realizar pruebas en cada escenario utilizando los modelos de Machine Learning implementados y se ajustan según los resultados obtenidos. Además, se crea un conjunto de datos propio a partir de un repositorio de datos que contenga URL legítimas y URLs con phishing, utilizando un código de extracción de características desarrollado previamente. Finalizando esta fase con el entrenamiento y prueba del modelo de predicción utilizando un simulador de sitios web con phishing junto a la implementación y despliegue de una extensión de Google Chrome capaz de realizar peticiones al modelo de predicción desarrollado y cargado en la nube.

La fase III aspira la evaluación del sistema de detección de intrusos (IDS) corroborando los resultados obtenidos y se analiza la efectividad del sistema en la detección de sitios web con phishing para poner fin con el desplegado hacia el público.

Capítulo II

Marco Teórico

En este capítulo, se realiza un análisis exhaustivo de los sistemas de detección de intrusos (IDS), las características utilizadas para la verificación, los Indicadores de compromiso (IOCs) y los modelos y/o algoritmos de Machine Learning, métricas de precisión, elementos vitales en el proceso de construcción de sistemas de detección de phishing que permitan determinar si un sitio web contiene o no phishing.

Para el desenvolvimiento del tema se lleva a cabo el proceso de revisión sistemática de la literatura, que consiste en una secuencia definida y estricta de pasos metodológicos, desarrollado de acuerdo con un protocolo.

Para iniciar, se genera la cadena de búsqueda que incluye los términos relevantes relacionados con el objetivo de estudio planteado. Consecuentemente se lleva a cabo el procesamiento de la cadena de búsqueda en SCOPUS otorgándonos la documentación para proceder con la revisión y selección de artículos destacados para la investigación de la cual se basó en 3 criterios principales i) Los artículos deben tener una conexión directa con el tema de investigación, ii) El número de artículos citados debe ser igual o superior a 7, iii) los artículos deben encontrarse en un rango de cinco años en este caso se tomó desde el 2017 hasta 2022. De acuerdo con estos criterios se encontraron 10 artículos, la información mencionada se representa en Anexos 2: Artículos para la revisión de la literatura. Las plataformas utilizadas para la búsqueda del dataset fueron: Data Science KAGGLE, Malware Information Sharing Platform (MISP) y MendeleyData.

Las palabras clave y sinónimos que se utilizaron durante la ejecución de la revisión son:

Cybersecurity: Cyber Safety, Computer Security

Machine Learning: Deep Learning, Self-Learning, Neural Networks,

Detection: identification, recognize,

Threats: vulnerabilities y

Application: app

Se obtuvo un total de 5 dataset, como se aprecia en el apartado de Anexo 3: Los conjuntos de datos que fueron seleccionados a través de una revisión bibliográfica.

Para finalizar el capítulo se extiende información sobre las extensiones de Google Chrome con el objetivo de abordar el tema de investigación objeto de estudio.

Sistema de detección de intrusos (IDS)

Un Sistema de Detección de Intrusos (IDS) considerado como un sistema de software o hardware utilizado para automatizar el proceso de monitoreo de eventos e identificación de actividades maliciosas en un sistema informático o red y mantener la seguridad de los usuarios (Stefan, 2002).

Hay dos categorías principales de IDS: El primero se refiere a un Sistema de Detección de Intrusos que se fundamenta en firmas (SIDS), usa una base de datos de firmas de ataques conocidos, mediante la comparación de patrones la cual genera una alarma cuando el tráfico de red coincide con una regla (Muñoz, 2019), el segundo se refiere a un Sistema de detección de intrusos que se fundamenta en anomalías (AIDS), analiza el flujo de la red para identificar el comportamiento de las anomalías y al igual que SIDS puede trabajarse con base de datos resultando extremadamente valioso al detectar ataques que aún no se han identificado (Becerril, 2018).

Los Sistemas de Detección de Intrusos (IDS) es capaz de identificar cualquier tipo de ataque para lo cual necesita una o más entradas conocidas como comprobación y, en particular para este proyecto, son las características de los sitios web que pueden extraerse de una URL y que están almacenadas en una base de datos (es, 2008).

En la actualidad, uno de los métodos más frecuentes de aprendizaje automático (ML) en los Sistemas de Detección de Intrusos (IDS) es ampliamente adoptado como uno de los enfoques más populares para hacer frente a los ataques de seguridad en las redes informáticas de hoy en día. Esta adopción se debe a la necesidad de hacer frente a la evolución constante del comportamiento de los ataques (Pérez, n.d.).

La implementación de técnicas de aprendizaje automático en los IDS tiene como objetivo mejorar la precisión en la detección de ataques. (WU, 2021). Los Sistemas de Detección de Intrusos pueden detectar diversos tipos de ataques, los cuales pueden tener diversas características, Siendo uno de los ataques más conocidos el malware, por ejemplo: Denegación de servicios (DoS), Adware, Spoofing, Spyware, etc., o también capaz de detectar ataques de ingeniería social, por ejemplo: Phishing, Smishing, Pretexting, entre otros (Mus, 2020).

El phishing se ha convertido en el método de ataque más utilizado en la actualidad por los ciberdelincuentes ya que es una técnica sencilla de ejecutar y alta efectividad. Este tipo de ataque precisa como el arte de simular sitios web lícitos con el fin de obtener datos que se encuentren dentro de la misma con el objetivo de llevar a cabo acciones de robo o fraude (Sonmez, 2018).

Indicadores de Compromiso

Los indicadores de compromiso (IOC) son medidas o métricas que se implementan para evaluar las brechas de seguridad y medir la capacidad del software para prevenir ataques de phishing, así como

para evaluar su impacto en la seguridad de los usuarios (Verma, 2018). Dentro del área de estudio de los indicadores de compromiso (IOC) se identifican los siguientes:

- Detección de amenazas: Encargados de evaluar la capacidad de un sistema de seguridad para detectar diferentes tipos de amenazas y ataques. Estos indicadores pueden incluir la tasa de detección de malware, la identificación de intentos de intrusión o la detección de actividades sospechosas en la red (Alhazmi, 2005).
- Falsos positivos y falsos negativos: Evalúan la precisión del sistema en la detección y clasificación de amenazas. Los falsos positivos ocurren cuando se identifica erróneamente una actividad benigna como maliciosa, mientras que los falsos negativos suceden cuando se pasa por alto una actividad maliciosa. Ambos aspectos son importantes para evaluar la efectividad del sistema de seguridad (*Indicadores de compromiso (IOC)*, 2022).
- Vulnerabilidades y parches: Los IOC también pueden evaluar la gestión de vulnerabilidades y la aplicación de parches. Estos indicadores pueden incluir la identificación y clasificación de vulnerabilidades conocidas, el tiempo promedio para aplicar parches de seguridad y la frecuencia de actualizaciones de seguridad (Alhazmi, 2007).

La tabla 1 proporciona un resumen de algunos de los indicadores de compromiso comúnmente observados en las URLs, haciendo referencia a la iniciativa en la que se pueden visualizar los autores a través de la codificación que se presenta en el Anexo 1. Estos indicadores de compromiso (IOC) se utilizan para evaluar la confiabilidad de los sitios web antes de que los usuarios interactúen con ellos.

Tabla 1

Indicadores de compromiso con respecto a una URL

Ord.	Recursos de comprobación	Descripción	Iniciativa
1	MD5	Genera una representación única de un mensaje arbitrario al procesar una entrada, creando así una "huella digital" o resumen del mensaje con una longitud de 128 bits. Cada URL se convierte en un identificador único mediante la generación de un MD5.	S07, S08
2	SHA1	Se utiliza un cálculo hash que puede ser algo demorado, pero proporciona una mayor seguridad, examinando las URL con el fin de crear claves únicas.	S09, S10
3	YARA	La detección de malware implica la identificación de archivos que satisfacen ciertas condiciones, junto con la implementación de un conjunto de reglas estratégicas para generar descripciones de familias de malware, basadas en patrones de texto o de código binario.	S11
4	SHA256	El algoritmo de hashing utilizado tiene la función de transformar cualquier texto de longitud variable en un texto de longitud fija de 256 bits. Su principal objetivo es detectar ataques de phishing dentro del ámbito de autenticación de un único dominio.	S12, S13
5	Domain	Un término utilizado para designar un ámbito de independencia administrativa, autoridad o control. Frecuentemente se emplea	S14, S15

Ord.	Recursos de comprobación	Descripción	Iniciativa
		para reconocer los servicios ofrecidos en Internet y verificar su legitimidad, evitando así el riesgo de phishing	
6	HostName	Se lleva a cabo mediante la utilización de algoritmos con concurrencias en las URL para comparar los nombres de host, junto con la exploración de que el nombre de dominio coincida con el propietario del nombre.	S16, S17
7	IPDSt	El reconocimiento de URL distintas implica identificar si el contenido de un sitio web es original.	S18
8	IPSrc	La vigilancia de la dirección IP de la computadora se realiza para determinar si el sitio web tiene encriptación o utiliza nombres de dominio en lugar de direcciones IP, con el objetivo de evitar el phishing en el sitio web.	S19, S20

Nota. Características necesarias evaluadas dentro de los parametros en las URLs

Características para detección de intrusos - Phishing

De acuerdo con el estudio realizado en el ámbito de la literatura, se recopilaron un total de 62 atributos, los cuales se registraron en el Anexo 3: Tabla de características para la detección de Phishing. Es importante destacar que se encontraron numerosos atributos sumamente específicos, por lo tanto, se generalizaron como se puede apreciar en el Anexo 2: Conjuntos de datos seleccionados en la revisión de la literatura llevada a cabo. En última instancia, se eligieron un total de 30 atributos, basándose en su frecuencia de aparición, la cual debe ser igual o superior al valor medio de la frecuencia total. Los atributos seleccionados se presentan en orden descendente según su frecuencia como se indica en la

tabla 2. Además, se proporciona una breve descripción de cada recurso de verificación (atributo) y el número de artículo en el cual se presenta dicho atributo, en relación con la iniciativa en la que se pueden visualizar los autores a treves de la codificación mostrada en el Anexo 1

Tabla 2

Características de sitios web a partir de la URL

Ord.	Recursos de Comprobación	Descripción	Iniciativa
1	Longitud de la URL	Los sitios web de phishing pretenden ocultar el nombre del dominio, por lo que usan URL largas. La longitud de la URL de un sitio web legítimo es menor a los 54 caracteres, si se excede hasta los 75 caracteres tiene una alta probabilidad de considerarse phishing, y si es igual o supera los 75 caracteres es considerado phishing en su totalidad.	S01, S02, S03, S04, S05, S06, S24
2	URL de anclaje	Consiste en contar la cantidad de veces que las etiquetas <a> con enlaces del sitio web dirigen a un dominio diferente de este. Si la cantidad excede el valor de 31 % es considerada como sospechosa de phishing.	S01, S02, S03, S04, S05, S06, S24
3	URL anormal	Se revisa si la URL contiene hostname y que esta coincida con el dominio en la URL, en caso de no poseer estas 2 características, entonces es considerado como phishing	S01, S02, S03, S04, S05, S06, S24
4	Request URL	Examina si los objetos externos contenidos en una página web se cargan desde otro dominio. Si la dirección de la URL	S01, S02, S03, S04,

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		se encuentra fuera del dominio, entonces es considerado como phishing.	S05, S06, S24
5	Edad del dominio	Un sitio web puede ser valorado por la duración de su dominio. Si el tiempo desde su creación es menor a 6 meses tiene una alta probabilidad de ser un sitio web con phishing.	S01, S02, S03, S04, S05, S06, S24
6	Presencia de subdominio	Un sitio web de phishing contiene más de 2 subdominios en su URL. Para identificarlo, se debe observar la cantidad de puntos existentes en el dominio, ya que si es mayor a 2 se considera phishing.	S01, S02, S03, S04, S05, S06, S24
7	Duración del registro del dominio	Esta información se obtiene con el registro whois; si el número de años que ha sido registrado el dominio de un sitio web es menor o igual a 1 año es considerado como phishing.	S01, S02, S03, S04, S05, S06, S24
8	Token HTTPS	Se trata de la utilización del protocolo TLS/SSL juntamente con HTTP seguro.	S01, S02, S03, S04, S05, S06, S24
9	Presencia de dirección IP	En lugar de utilizar un nombre de dominio en una URL, el uso de una dirección IP aumenta el riesgo de que el sitio web sea objeto de ataques de phishing.	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
10	Rango de la página	El cálculo del rango de un sitio web se realiza contando tanto los enlaces salientes como los entrantes presentes en él, lo cual refleja su importancia. Si este valor es inferior a 0.2, existe una alta sospecha de que se trate de un sitio de phishing.	S01, S02, S03, S04 S05, S06, S24
11	Presencia del símbolo @	La presencia del símbolo (@) en una URL hace que el navegador web ignore todo lo que está antes de dicho símbolo, lo cual incrementa la probabilidad de ser redirigido a sitios web con phishing.	S01, S02, S03, S04, S05, S06, S24
12	Estado SSL	Se hace hincapié en determinar si un sitio web tiene un certificado SSL, ya que los sitios web de phishing generalmente no encriptan los datos enviados y, por lo tanto, no tienen este certificado. Para identificarlo, se verifica si la URL comienza con "HTTPS" y si sus proveedores son confiables.	S01, S02, S03, S04, S05, S06, S24
13	Informe estadístico	Los informes estadísticos proporcionan información sobre sitios web legítimos y aquellos con phishing, así como otros datos estadísticos relevantes.	S01, S02, S03, S04, S05, S06, S24
14	Enlaces que apuntan a la página	La validez de un sitio web se puede determinar mediante el cálculo de la cantidad de enlaces que apuntan hacia	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		dicho sitio. Para ser considerado legítimo, se requiere que al menos existan 2 enlaces que lo respalden.	
15	Redirección de doble barra	Si la posición de (//) en una URL es mayor a 7, se considera que el sitio web es un posible caso de phishing. Esto se debe a que, en los sitios web legítimos, generalmente se utiliza solo una vez la redirección de doble barra.	S01, S02, S03, S04, S05, S06, S24
16	Prefijo / Sufijo	En los sitios web de phishing, se utiliza el símbolo (-) para añadir prefijos y sufijos a las URL, mientras que los sitios web legítimos generalmente no hacen uso de este símbolo. El empleo de prefijos y sufijos en URL con el símbolo (-) es una característica comúnmente asociada a los intentos de phishing.	S01, S02, S03, S04, S05, S06, S24
17	Enlaces en etiquetas	Durante el análisis de un sitio web, se examinan todas las etiquetas existentes y hacia dónde se redirigen. Si una de las etiquetas redirige a un sitio web presente en la lista negra, se considera que el sitio web está involucrado en actividades de phishing.	S01, S02, S03, S04, S05, S06, S24
18	Deshabilitar clic derecho	En la actualidad, es común que los sitios web legítimos desactiven la opción de clic derecho para evitar que los usuarios realicen cambios en el código fuente del sitio.	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		Esta medida de seguridad tiene como objetivo proteger la integridad y la propiedad del contenido del sitio web.	
19	Uso de ventana emergente	Las ventanas emergentes en los sitios web legítimos suelen aparecer con un propósito específico y desaparecen con un clic. Sin embargo, en el caso de sitios web con phishing, se utilizan ventanas emergentes para solicitar información al usuario de manera sospechosa o maliciosa. La presencia de ventanas emergentes que solicitan información personal es una señal de advertencia de un posible sitio web de phishing.	S01, S02, S03, S04, S05, S06, S24
20	Favicon	El favicon es el ícono utilizado para identificar fácilmente un sitio web. Si el favicon de un sitio web es diferente al dominio que se muestra en la URL, esto indica una alta probabilidad de presencia de phishing. Es decir, si el favicon no coincide con el dominio legítimo del sitio web, puede ser un indicio de que se trata de un sitio web fraudulento.	S01, S02, S03, S04, S05, S06, S24
21	IFrame	Las etiquetas iframe se utilizan para incrustar contenido de otro sitio web dentro de una página web. Sin embargo, estas etiquetas también pueden ser utilizadas de manera maliciosa para engañar a los usuarios. Es decir, se pueden emplear para redirigir a los usuarios a sitios web	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		fraudulentos o para realizar ataques de phishing, Por lo tanto, la presencia de etiquetas iframe en un sitio web puede ser una señal de advertencia de posibles intenciones maliciosas.	
22	Registro DNS	El registro DNS de un sitio web contiene información crucial y relevante. Sin embargo, los sitios web con phishing tienden a ocultar este registro para evitar su detección.	S01, S02, S03, S04, S05, S06, S24
23	Índice de Google	Los sitios web con phishing tienden a tener una vida útil corta y, como resultado, no son indexados por los motores de búsqueda como Google.	S01, S02, S03, S04, S05, S06, S24
24	Puerto utilizado	Los puertos 8080 y 443 se consideran puertos confiables y ampliamente utilizados para servicios web seguros. Si un sitio web utiliza un puerto diferente a estos, existe una alta probabilidad de que sea un sitio de phishing. Los sitios web legítimos suelen utilizar los puertos estándar como el 8080 y el 443 para garantizar conexiones seguras y confiables.	S01, S02, S03, S04, S05, S06, S24
25	SFH (Controlador de Formulario de servidor)	En la gestión de formularios, se examina si el botón "submit" devuelve un mensaje vacío o no produce ninguna respuesta al completarse. Este análisis se utiliza	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		para identificar posibles sitios web de phishing. Si el botón "submit" no muestra un mensaje o no genera ninguna respuesta en etiquetas a un sitio web, puede indicar la presencia de phishing.	
26	Recuento de dirección del sitio web	Este enfoque implica analizar las veces en que las fuentes (por ejemplo, enlaces, imágenes o scripts) redirigen a una sola dirección web o a un sitio web con un dominio diferente al que se muestra en la barra de búsqueda. Esta evaluación se utiliza para identificar posibles casos de phishing. Si las fuentes del sitio web redirigen repetidamente a una única dirección o a un dominio distinto al que aparece en la barra de búsqueda, podría indicar la presencia de phishing.	S01, S02, S03, S04, S05, S06, S24
27	Mouse Over	La función mencionada solía utilizarse para mostrar información sobre el sistema en la parte inferior de la pantalla en sitios web. Sin embargo, en la actualidad, esta función no se utiliza en la mayoría de los sitios web legítimos. Por lo tanto, si un sitio web aún emplea esta función, se considera sospechoso.	S01, S02, S03, S04, S05, S06, S24
28	Trafico web	Un método de evaluación de un sitio web es considerar la cantidad de visitas que recibe diaria, semanal o mensualmente. Generalmente, a medida que este valor es	S01, S02, S03, S04,

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		más alto, el sitio web se considera más confiable y popular. Este indicador puede ser utilizado para evaluar la reputación y la confiabilidad de un sitio web en base a su popularidad y la cantidad de visitantes que atrae.	S05, S06, S24
29	Servicio de Acortamiento	Un servicio de acortamiento de URL es una técnica utilizada para reducir la longitud de una URL y redirigir a la misma página que la dirección original. Sin embargo, la mayoría de los sitios web con phishing hacen uso de este tipo de servicios. Esto se debe a que el acortamiento de URL les permite ocultar la verdadera dirección del sitio web malicioso y dificulta su detección. Por lo tanto, la presencia de una URL acortada puede ser un indicio de posible phishing, ya que es una técnica comúnmente utilizada por los atacantes.	S01, S02, S03, S04, S05, S06, S24
30	Envió de información al correo electrónico	Al analizar un sitio web, se verifica si utiliza algún tipo de servicio "mail () to" dentro del mismo. Si se encuentra la presencia de este servicio, se considera como una señal de posible phishing. El servicio "mail () to" puede ser utilizado para enviar correos electrónicos de manera automática desde el sitio web, lo cual puede ser utilizado con fines maliciosos en ataques de phishing. Por lo tanto, la	S01, S02, S03, S04, S05, S06, S24

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		detección de este servicio puede ayudar a identificar posibles actividades fraudulentas.	

Nota. Características necesarias evaluadas dentro de los parametros en las URLs

Modelos y/o algoritmos de Machine Learning

El Machine Learning (ML) o Aprendizaje Automático siendo una rama de la inteligencia artificial, se centra el desarrollo de algoritmos y técnicas que le permite a los computadores aprender bajo un análisis de un conjunto de datos, Para después realizar toma de decisiones o predicciones de forma autónoma, sin requerir una programación explicita (Heaton, 2018).

El Machine Learning tiene aplicaciones en diversos campos científicos, como la robótica, los videojuegos, el reconocimiento de patrones, visión por computadora, la minería de datos, el procesamiento del lenguaje natural, la medicina, la seguridad informática, entre otros (Heaton, 2018). Los algoritmos de Machine Learning se refieren a un conjunto de instrucciones programáticas que permiten analizar un conjunto de datos con el objetivo de generar un modelo capaz de realizar predicciones o clasificar información (Marsland, 2014).

Los algoritmos y/o modelos de Machine Learning se pueden clasificar en diferentes formas de aprendizaje, como el supervisado y el no supervisado. En el aprendizaje supervisado, se basa en el uso de datos etiquetados para que el modelo aprenda a través de una entrada de datos y salida de datos deseada. Algunos algoritmos comunes en el aprendizaje supervisado incluyen Árboles de Decisión, Support Vector Machines (SVM) y Naive Bayes. Por otro lado, el aprendizaje no supervisado se basa en descubrir patrones en conjuntos de datos no etiquetados y luego clasificarlos. Algunos algoritmos

populares en el aprendizaje no supervisado son KMeans, Clustering Jerárquico y DBSCAN (James, 2013; Marsland, 2014).

Los modelos y/o algoritmos de Machine Leaning pueden ser de dos tipos: Regresión/Clasificación y Agrupación/Reducción. Los modelos de Regresión/Clasificación se enfoca en el aprendizaje supervisado, donde se utilizan directamente los datos de entrada y salida para entrenar y probar el modelo, durante el entrenamiento el modelo busca aprender una función que relacione las variables de entrada como las salidas etiquetadas. Esto permite lograr una mayor eficiencia en los procedimientos empleados durante el entrenamiento y las pruebas (James, 2013).

Por otro lado, los modelos de Agrupación/Reducción se centran en el aprendizaje no supervisado, donde no se proporcionan etiquetas de salida predefinidas. Estos modelos buscan identificar patrones y estructuras ocultas en los datos sin ninguna guía explícita, además, de considerarse como una forma más común de realizar análisis de datos (Shalev-Shwartz Y Ben-David, 2014). En este tipo de modelos, los datos utilizados no poseen una salida o categoría definida que indique a qué grupo pertenecen los datos de entrada. Por lo tanto, es el propio modelo y/o algoritmo el encargado de buscar patrones y agrupar los datos en diferentes categorías.

A partir de la revisión de la literatura (consultar Anexo 1: Artículos seleccionados en la revisión de la literatura), se han identificado los algoritmos y modelos más frecuentemente empleados para la detección de Phishing. Además, se ha llevado a cabo un análisis de la precisión máxima y la incidencia que estos modelos tienen en la detección.

La tabla 3 presenta una descripción detallada los Modelos y/o algoritmos de Machine Learning encontrados y seleccionados durante el proceso. Estos modelos Estos modelos y/o algoritmos se

relacionan con la iniciativa en la que los autores pueden ser visualizados mediante la codificación que se muestra en el Anexo 1.

Tabla 3

Modelos y/o algoritmos de Machine Learning

Ord.	Modelo y/o algoritmo	Descripción	Iniciativa
1	Árbol de Decision	Un tipo de modelo jerárquico se basa en realizar preguntas sobre las características de los datos para tomar decisiones. Un algoritmo ampliamente utilizado para tareas de clasificación y regresión es el árbol de decisión en el aprendizaje supervisado. Este algoritmo se caracteriza por su estructura jerárquica, en la cual los datos se clasifican desde la raíz (nodo raíz) hasta los nodos hoja. Cada nodo en el árbol representa una condición, mientras que los nodos hoja representan las respuestas correspondientes a esas condiciones. El proceso de clasificación/regresión comienza en el nodo raíz y se desplaza a través del árbol hasta llegar a un nodo hoja, donde se realiza la clasificación o regresión final. Según una revisión de la literatura, se ha observado que los árboles de decisión pueden lograr una alta precisión, alcanzando un máximo del 96,60% en la detección de phishing.	S21, S22, S24
2	Random Forest	Un algoritmo que une diversos árboles de decisión para lograr predicciones más precisas es Random Forest. Es uno de los algoritmos de aprendizaje supervisado más populares debido a su	S21, S24

Ord.	Modelo y/o algoritmo	Descripción	Iniciativa
		simplicidad y precisión. Se utiliza para clasificar y predecir datos, y se basa en la construcción de múltiples árboles de decisión para obtener una salida que combina las predicciones de cada árbol. Random Forest se compone de dos etapas: la primera implica la creación de los bosques aleatorios, y la segunda consiste en realizar predicciones utilizando el clasificador de bosques aleatorios. Según estudios anteriores, se ha observado que Random Forest puede alcanzar una alta precisión, con un máximo del 99,33%, en la detección de ataques de phishing.	
3	Redes Neuronales	Los modelos que se basan en la estructura y funcionamiento del cerebro humano consisten en capas de neuronas conectadas entre sí. Debido a su similitud con el cerebro humano, las redes neuronales se han vuelto cada vez más populares entre los investigadores, siendo consideradas un enfoque moderno para el aprendizaje. Estas redes tienen como objetivo enseñar a las computadoras a procesar datos de manera similar al cerebro humano, utilizando nodos interconectados que imitan el comportamiento de las neuronas. A medida que aprenden de los errores, las redes neuronales se mejoran continuamente y se aplican para resolver problemas complejos, con el fin de lograr una mayor precisión. Según la revisión de la literatura, las redes neuronales han logrado una detección de phishing con una precisión máxima del 97%.	S23, S22, S24, S24

Ord.	Modelo y/o algoritmo	Descripción	Iniciativa
4	Support Vector Machines (SVM)	Se utilizan algoritmos para buscar el plano óptimo que puede separar datos en diferentes categorías. La Máquina de Vectores de Soporte (SVM) es un enfoque de Machine Learning supervisado que se puede emplear en clasificación y regresión. Su funcionamiento consiste en crear un plano en el espacio según las características de los datos de entrenamiento. Posteriormente, los datos se categorizan utilizando un separador, lo que permite que el algoritmo pueda recibir nuevas características y predecir a qué categoría pertenecen. Según una revisión de la literatura, se ha observado que, en la detección de phishing, SVM logra una precisión máxima del 96,5%.	S09
5	Mezcla de Gaussianas	El modelo de Mezcla de Gaussianas emerge como una herramienta esencial en el ámbito del Aprendizaje Automático. Este enfoque, se basa en la creación de un marco en el espacio de características, moldeado por los datos de entrenamiento disponibles. Al fusionar múltiples distribuciones Gaussianas, el algoritmo puede adaptarse a la estructura intrínseca de los datos y representar sus diferentes categorías de manera efectiva. En el contexto específico de la detección de patrones y modelado de datos complejos, En el caso concreto de la detección de phishing, la Mezcla de Gaussianas ha demostrado alcanzar tasas de precisión impresionantes, superando consistentemente el umbral del 96,5%.	S25

Ord.	Modelo y/o algoritmo	Descripción	Iniciativa
6	Bayesiano Ingenuos	Los clasificadores bayesianos ingenuos se presentan como una herramienta esencial en el campo del Aprendizaje Automático. Se basan en el principio subyacente de la probabilidad bayesiana y hacen suposiciones simplificadas acerca de la independencia de las características. El funcionamiento de los clasificadores bayesianos ingenuos radica en la construcción de un modelo probabilístico basado en el teorema de Bayes, en el ámbito específico de la detección de phishing, se ha observado que los clasificadores bayesianos ingenuos logran un alto grado de precisión, superando consistentemente el umbral del 96,5%. Estos resultados subrayan la capacidad de estos enfoques para manejar problemas complejos de clasificación y resaltar su utilidad en la categorización de datos en diversos contextos.	S26
7	Redes Bayesianas	Las redes bayesianas se han convertido en poderosas herramientas de aprendizaje automático supervisado tanto para tareas de clasificación como para problemas de regresión. La esencia de su función es construir una cuadrícula en el espacio de características y modelarla de acuerdo con los datos de entrenamiento. En este dominio particular, se ha observado que las redes bayesianas logran una precisión excelente, superando el 96,5% en algunos casos. Este resultado destaca la capacidad de las redes bayesianas para capturar patrones sutiles y relaciones contextuales.	S27

Ord.	Modelo y/o algoritmo	Descripción	Iniciativa
8	Dummin Classifier	El clasificador "Dummin" se destaca como un enfoque interesante y peculiar. Se trata de un clasificador de referencia, utilizado como línea base para comparar y evaluar otros modelos más complejos y sofisticados, opera bajo una estrategia de toma de decisiones extremadamente básica. Su enfoque se basa en asignar todas las instancias de datos a la clase más frecuente en el conjunto de entrenamiento, ignorando por completo las características individuales de los datos. Al comparar el rendimiento de un modelo más complejo con el del clasificador Dummin, los científicos de datos pueden determinar si su enfoque tiene un valor real en la tarea de clasificación o si simplemente está emulando el resultado de una estrategia trivial.	S23, S22, S24, S24

Nota. Algoritmos de Machine Learning necesarios para llevar a cabo la evaluación de la eficiencia de la extensión de Google Chrome, destacando la eficiencia que se obtiene en cada algoritmo.

Metodología Scrum

La metodología ágil Scrum se ha vuelto ampliamente reconocida en la industria del desarrollo de software debido a su enfoque innovador y altamente colaborativo durante el ciclo de vida del desarrollo de software ya que proporciona un marco de trabajo flexible y adaptable tal y como los distintos tipos de metodologías agiles como: Kanban, Dynamic System Development Method (DSDM) o Extreme Programming (XP), Scrum fomenta la transparencia, la comunicación constante y la mejora continua, resultando en una mayor satisfacción dentro del desarrollo de productos software (Mariz, 2010).

Scrum se base en el concepto de iteraciones cortas, conocidas como sprints, que generalmente tienen una duración de dos a cuatro semanas. Durante cada sprint, el equipo se enfoca en un conjunto de objetivos específicos y trabajan de manera colaborativa para lograrlos. Esta estructura de trabajo por iteraciones permite una mayor adaptabilidad a medida que se obtienen nuevos conocimientos y se realizan ajustes en el enfoque del proyecto, además de que se tiene una participación con el cliente o propietario del producto (Kikitamara Y Noviyanti, 2018). A través de reuniones regulares, conocidas como revisiones y retrospectivas, se involucra al cliente en el proceso de desarrollo y se recopilan comentarios y retroalimentación para realizar mejoras continuas (Srivastava, 2017).

Extensiones de Google Chrome

Las extensiones de Google Chrome son programas que se ejecutan en el contexto de un sitio web y tienen como objetivo añadir nuevas funcionalidades para mejorar la experiencia del usuario. Estas extensiones permiten al usuario realizar múltiples tareas simultáneamente gracias a la combinación de características del navegador (Al-Khamis Y Khalafallah, 2015). Desde 2010, Google Chrome se convirtió en uno de los primeros navegadores en implementar funcionalidades de extensión. A partir de su cuarta versión, se permitió la creación de extensiones. Aunque se han desarrollado extensiones para otros navegadores como Opera, Brave, Mozilla Firefox y Microsoft Edge, no siempre son compatibles con Google Chrome debido a que este se basa en el proyecto de código abierto Chromium, el cual es mantenido por diferentes compañías.

Algunas extensiones que han sido probadas y han demostrado buenos resultados en la detección de phishing incluyen PIXM Phishing Protection, My Wot, Retruster Phishing Protection, las cuales pueden ser instaladas en navegadores basados en Chromium. Además, PhishWall es una extensión recomendada para Firefox

El objetivo de este proyecto de investigación es crear una extensión específicamente diseñada para Google Chrome, con el propósito de detectar ataques de phishing en sitios web. Para llevar a cabo esta tarea, se requerirá un conocimiento sólido en tecnologías como HTML, CSS, JavaScript y JSON. Estas habilidades serán fundamentales para desarrollar una extensión efectiva que pueda analizar y evaluar la autenticidad de los sitios web visitados, brindando una capa adicional de protección contra posibles intentos de phishing. El desarrollo de extensiones puede llevarse a cabo desde cualquier sistema operativo.

Las extensiones de Google Chrome suelen tener una mayor aceptación y uso en comparación con las extensiones de otros navegadores. Desde junio de 2022 a junio de 2023, Google Chrome se ha consolidado como el navegador más popular, con una cuota de uso del 62,55%. Esta popularidad y adopción masiva de Google Chrome brinda una mayor oportunidad de llegar a una amplia audiencia al desarrollar y distribuir extensiones para este navegador en particular. (“StatCounter Global Stats - Browser, OS, Search Engine including Mobile ...”)

Herramientas

En el desarrollo del sistema, una de las herramientas principales que se utilizó es Visual Studio Code. Este editor de código, desarrollado por Microsoft, es ampliamente reconocido y versátil. Visual Studio Code ofrece una amplia gama de funciones y extensiones, lo que lo convierte en una poderosa herramienta valorada por su eficiencia y facilidad de uso. Su interfaz intuitiva y personalizable permite a los desarrolladores adaptar el entorno de trabajo a sus preferencias y necesidades específicas (*AWS Toolkit for Visual Studio Code*, n.d.).

PythonAnywhere es un IDE en la nube que brinda a los usuarios la posibilidad de escribir, ejecutar y desplegar aplicaciones Python en línea. Este entorno de desarrollo integrado ofrece diversas características, como la edición de código, entornos de ejecución y acceso a bases de datos, lo que lo

hace adecuado tanto para principiantes como para desarrolladores experimentados (*Host, Run, and Code Python in the Cloud: PythonAnywhere*, n.d.).

JavaScript es un lenguaje de programación relativamente sencillo que se utiliza principalmente para agregar interactividad a las páginas web. Una de sus características distintivas es que los programas en JavaScript, comúnmente conocidos como scripts, se incorporan en las páginas HTML y se ejecutan en el navegador, como Netscape Navigator y Microsoft Explorer. Estos scripts generalmente consisten en funciones que son invocadas desde el propio HTML cuando ocurre algún evento. Esto permite agregar efectos visuales, como cambiar la forma de un botón cuando se pasa el ratón por encima, o abrir una nueva ventana al hacer clic en un enlace (Navarrete, 2006).

Python es ampliamente reconocido como un lenguaje de programación multi-paradigma. Esto significa que admite múltiples enfoques y estilos de programación, como programación orientada a objetos, programación funcional y programación procedural. Además, Python se considera un lenguaje de alto nivel debido a su sintaxis clara y expresiva, lo que lo hace fácil de leer y entender. Una de las ventajas de Python es que proporciona estructuras de datos incorporadas, como listas, diccionarios, conjuntos y tuplas, que simplifican la manipulación y organización de datos. Estas estructuras de datos predefinidas permiten realizar tareas complejas en muy pocas líneas de código, lo que hace que el desarrollo sea más eficiente y legible (Challenger, Y Becerra, 2014).

CSS, por sus siglas en inglés Cascading Style Sheets, es un lenguaje de hojas de estilo utilizado para describir la presentación y el diseño de un documento HTML. (“Cómo puedo ocultar un elemento en una página utilizando CSS”) Es una tecnología fundamental en el desarrollo web y permite controlar la apariencia visual de una página web, incluyendo el diseño, los colores, las fuentes, los márgenes y otros aspectos relacionados con la presentación (Cabrera, 2013).

HTML es un lenguaje basado en texto y es independiente de la plataforma, lo que significa que puede ser interpretado por cualquier navegador web y se puede utilizar en diferentes sistemas operativos. Es el componente clave para la creación de páginas web y se combina con otros lenguajes, como CSS (Cascading Style Sheets) y JavaScript, para mejorar la apariencia y la funcionalidad de los sitios web (Ferrer, García V., Y García R. 2013).

Postman es una herramienta popular para probar y desarrollar API (Application Programming Interface). Se utiliza para facilitar la interacción con APIs, permitiendo enviar y recibir solicitudes HTTP de manera sencilla. Postman ofrece una interfaz gráfica intuitiva que permite crear, enviar y guardar solicitudes HTTP, así como visualizar las respuestas recibidas. Además, proporciona funcionalidades adicionales como la posibilidad de automatizar pruebas, generar documentación de API y colaborar con otros miembros del equipo.

Capítulo III

Implementación del sistema

En el siguiente capítulo se describen los pasos necesarios para poner en marcha el sistema propuesto. El sistema está diseñado para detectar ataques de Phishing mediante la utilización de modelos y/o algoritmos de Machine Learning. La integración de este sistema tuvo lugar dentro de Google Chrome mediante una extensión. En esta parte, se explica la creación de la API Rest, formulada con el objetivo de proporcionar a los usuarios información sobre la legitimidad de un determinado sitio web, discerniendo entre sitios auténticos y sitios de phishing. Además, este capítulo expone la implementación del sistema, su arquitectura, los roles que toma cada una de las personas que forman parte del proyecto, el análisis de requisitos, el desarrollo de la metodología con varias herramientas y gestores, la épica global, la ejecución y la interfaz que se ha emprendido para desarrollar el proyecto.

Implementación del sistema

Para comprender mejor el funcionamiento del sistema de detección propuesto, se describe brevemente su proceso: Inicialmente, la URL del sitio web se transmite a la API. La API se encarga de extraer las características designadas que figuran en la tabla 1 y tabla 2. A continuación, se realiza la clasificación del sitio web, determinando si se trata de un sitio legítimo o con phishing. Este resultado se muestra posteriormente al usuario a través de la extensión personalizada de Google Chrome que se ha creado.

Teniendo en cuenta el siguiente punto que se relaciona con las métricas de evaluación dentro de los modelos de Machine Learning probados cada uno de ellos exhaustivamente para el desarrollo del presente proyecto, teniendo en cuenta (Xin, 2018 Y Castillo, M., K. Chuquitarco., 2023.) se definieron algunas métricas para su correspondiente evaluación:

Accuracy: Se refiere al porcentaje que se obtiene de la clasificación entre positivos y negativos de un grupo de elementos.

Precision: Proporciona la medida en porcentaje de los verdaderos positivos correctos dividido por el número total de predicciones positivas identificadas multiplicado por cien.

Recall: Calcula el número de todos del grupo de elementos que se detectaron correctamente en proporción a todos al otro grupo de elementos que deben detectarse**.**

F1: Cuenta el número de todos los objetos detectados correctamente en el grupo en relación con todos los demás objetos detectados en el grupo.

En la tabla 4, se muestran las fórmulas aplicadas en la evaluación del modelo, mismas que están acordes con (Xin, 2018).

Tabla 4

Fórmulas de Métricas de Evaluación

MÉTRICAS DE EVALUACIÓN	FÓRMULA
Accuracy	$accuracy = \frac{VP + VN}{VP + VN + FP + FN}$
Precision	$precision = \frac{VP}{VP + FP}$
Recall	$recall = \frac{VP}{VP + FN}$
F1	$f1 = 2 * \frac{presicion * recall}{presicion + recall}$

En donde:

- VP (Verdaderos Positivos): Número de sitios web con Phishing calificados correctamente.
- VN (Verdaderos Negativos): Número de sitios web legítimos clasificados correctamente.
- FP (Falsos Positivos): Número de sitios web legítimos clasificados erróneamente como Phishing.
- FN (Falsos Negativos): Número de sitios web con Phishing clasificados erróneamente como legítimos.

Para concluir, se realiza una evaluación del modelo para ello se aplica la matriz de confusión, que como su nombre lo dice, se refiere a una tabla que nos ayuda a la recolección de datos obtenidos de una clasificación permitiéndonos distinguir cada uno de los elementos en clases (Robotham, 2012). Dentro de la tabla 5 se proporciona la matriz de confusión para Phishing Impact, demostrando como es la implementación dentro del aplicativo.

Tabla 5

Matriz de confusión para Hunter Phisher

	POSITIVOS	***NEGATIVOS***
POSITIVOS	*Phishing clasificados correctamente (VP)*	*Legítimos mal clasificados (FP)*
NEGATIVOS	*Phishing mal clasificados (FN)*	*Legítimos clasificados correctamente (VN)*

Arquitectura

Este segmento analiza exhaustivamente el desarrollo del sistema empleado para cumplir el objetivo principal de este proyecto.

Diseño de arquitectura. Según (Bass, Clements y Kazman, 2021) en su libro Software Architecture in practice hace referencia al "Enfoque de comprensiones sólidas sobre los principios y conceptos fundamentales de los diseños de software conociendo un enfoque amplio de cada reseña y proceso que debe seguir el diseño detallado", aportando también que el diseño de una arquitectura es la base fundamental para el desarrollo de un sistema.

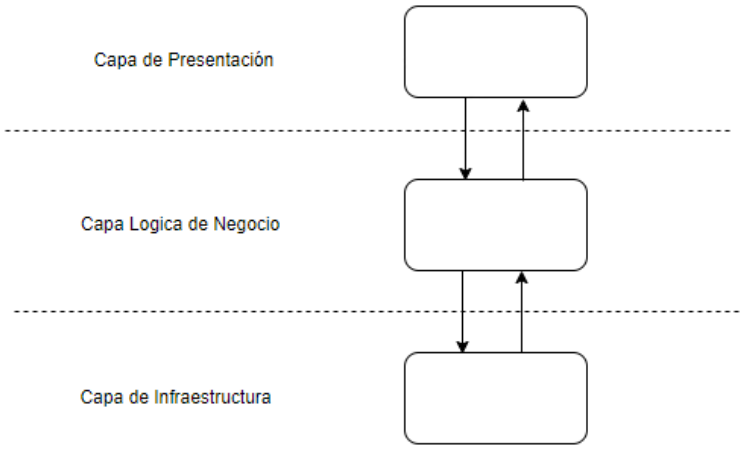
El propósito de esta sección es definir y diseñar la arquitectura de la implementación de software propuesta y la metodología utilizada para desarrollar el sistema propuesto.

Diagrama de Arquitectura Lógica

La gestión de aplicaciones por capas es importante porque permite separar los archivos de la aplicación, lo que facilita el mantenimiento y la reutilización del código. La extensión de Google Chrome se desarrolló utilizando un modelo de tres capas, como se muestra en la figura 1, interacciones entre la capa de presentación, la capa de lógica de negocio y la capa de infraestructura que permite ejecutar la aplicación.

Figura 1

Diagrama de la arquitectura lógica del sistema



Definición de las tecnologías a usar

Las tecnologías de desarrollo de software son programas informáticos que se utilizan para crear distintos tipos de aplicaciones. Existen numerosas tecnologías de desarrollo de software, pero deben elegirse en función del tipo de proyecto que se esté desarrollando, haciendo referencia a la figura 2, muestra las tecnologías utilizadas en cada capa, que forman parte del modelo definido en la figura 1 para comprender mejor la finalidad de cada capa. A continuación, se ofrece una breve descripción de cada tecnología:

Capa de presentación: La capa de presentación es la responsable de la interacción del usuario con la aplicación, se encarga de recoger los datos de la interacción de la aplicación con el usuario que se pasan a otras capas para procesarlos y mostrar los resultados. A continuación, se describen los métodos utilizados por esta capa:

- Html (Lenguaje de Marcas de Hipertexto): Define el significado y el contenido web, a través de componentes que trazan una estructura para la página web (*HTML*, 2023).

- Css (Hojas de estilo en cascada): Consiste en diversas reglas estructuradas, que buscan hacer más atractivo un diseño de un sitio web (Gómez, 2021).
- Js (JavaScript): Es un lenguaje de programación que permite diversas funciones en un sitio web (Robbins, 2012).

Capa lógica del negocio: La capa de lógica de negocio contiene la lógica de la aplicación, que es responsable de recibir las consultas de la capa de presentación y enviarlas a la capa de infraestructura solicitando las predicciones apropiadas y después de recibir los resultados, la capa es responsable de enviar la respuesta a la fuente o capa de presentación para renderizarla al usuario. Las técnicas utilizadas en esta capa son las siguientes:

- Js (JavaScript): Es un lenguaje de programación de alto nivel que puede utilizarse en diversos sistemas operativos para desarrollar una amplia gama de aplicaciones (Robbins, 2012).

Capa de infraestructura: La capa contiene los mecanismos necesarios para interactuar con la interfaz del programa de aplicación y los modelos de aprendizaje automático entrenados propuestos en este proyecto. Básicamente, la capa consta de los siguientes componentes: modelos entrenados y almacenados como archivos SAV, y un servicio que puede ser extendido para predecir el rendimiento de un sitio web a través del navegador Google Chrome. A continuación, se describen las técnicas utilizadas para desarrollar este servicio:

- Python: Se trata de un lenguaje de programación de alto nivel que puede utilizarse en diversos sistemas operativos y desarrollar innumerables aplicaciones (Fernández, 2013).
- Flask: Es un "micro-framework" desarrollado en Python y diseñado para el desarrollo rápido y flexible de aplicaciones web (Grinberg, 2018).

Figura 2

Diagrama de la arquitectura lógica del sistema con las tecnologías a usar

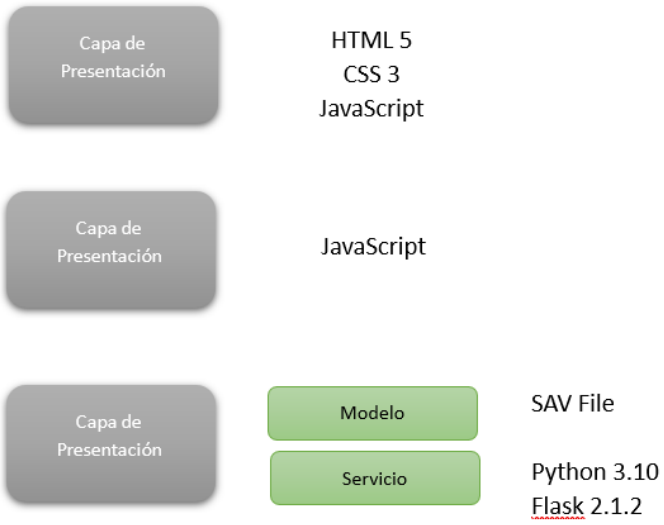


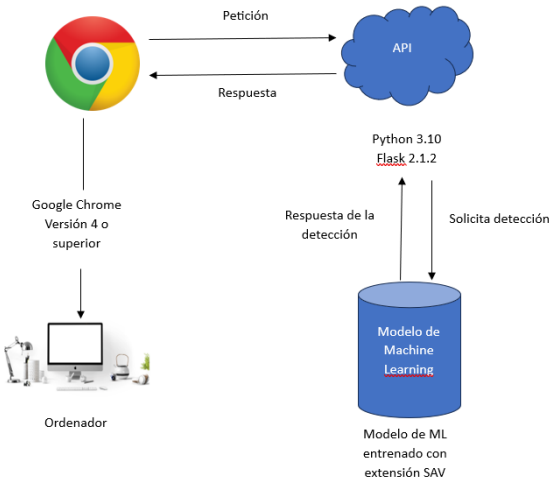
Diagrama de Arquitectura Física

La figura 3 muestra la arquitectura física, la cual constituye el fundamento para el desarrollo de la extensión "Phishing Impact" en Google Chrome. A nivel físico, se establece que cualquier ordenador con una versión del navegador Google Chrome de versión igual o superior a 4 puede acceder al sistema.

Cuando un usuario explora por Internet y hace clic en el complemento, se envía la URL del sitio web actual, llegando a una API. Esta API separa la URL y analiza las características del sitio web. Las características se entrenan como entrada para el Machine Learning, el cual el patrón que devuelve es si el sitio es legítimo o tiene suplantación de identidad para que la API pueda devolver una respuesta al complemento.

Figura 3

Diagrama de la arquitectura física del sistema



Roles

De acuerdo con los lineamientos de la metodología ágil Scrum, se involucran miembros al sistema, como son: el propietario del proyecto o usuario del sistema (Product Owner), el desarrollador encargado de gestionar y controlar el equipo de desarrollo (Scrum Master) y el equipo de desarrollo (Team Developer) (Kurnia, 2018).

La designación de cada uno de estos roles se organizó como se indica en la tabla 6, en donde se encuentra el nombre del rol y del integrante del proyecto que tiene dicho cargo respectivamente.

Tabla 6

Rol de Scrum designados

N°.	Rol	Integrantes	Descripción
01	Scrum Master	Adrian Fernando Guevara Jiménez	Líder de equipo de Scrum
02	Product Owner	Dr. José Luis carrillo Medina	Representante a las partes interesadas
03	Team Development	Dennis Sebastián Caisa Llano Adrian Fernando Guevara Jiménez	Desarrollo y diseño de la aplicación
04	Auditora	Ing. Maria Alexandra Corral Diaz	Auditora de los procesos de desarrollo

De acuerdo con la distribución de roles en la tabla 6, es importante mencionar que, dado que solo hay 2 estudiantes elaborando el proyecto, uno ha sido asignado como Scrum Master y también tiene participación en las actividades dentro del Team Developer, la información se utilizó para obtener requisitos, y posteriormente escribir historias de usuarios. El Scrum Master del proyecto realiza una reunión inicial a la que asisten el Product Owner y el Team Developer.

Análisis de Requisitos

Dentro de esta sección se menciona sobre el análisis de requisitos que es una descripción exhaustiva de los requisitos de software, que abarcan las exigencias del usuario (Wong Durand, 2017). Se utilizó la técnica de la entrevista para así poder obtener información en una reunión que se tuvo con los integrantes del equipo, que fueron el Product Owner y el Team Development.

Alcance

El alcance se centra en definir de manera clara y detallada las características y funcionalidades del sistema, los mismos que abarcan las necesidades y expectativas de los usuarios hasta la especificación de requisitos funcionales y no funcionales que guiaron en el desarrollo del proyecto.

Obtención de Requisitos

Tabla 7

Requisitos Funcionales y no Funcionales

N°.	Requisitos Funcionales	Requisitos No Funcionales
01	El sistema debe ser capaz de detectar proactivamente sitios web con técnicas de phishing mientras los usuarios navegan por Internet utilizando la extensión de Google Chrome.	El sistema debe lograr una alta precisión en la detección de sitios web con phishing, minimizando tanto los falsos positivos como los falsos negativos.
02	El sistema debe ser capaz de extraer características relevantes de las URL y contenido de los sitios web para su posterior análisis.	El sistema debe ser capaz de manejar un número creciente de usuarios y sitios web sin degradar su rendimiento.
03	El sistema debe utilizar algoritmos de Machine Learning para analizar las características extraídas y predecir la probabilidad de que un sitio web sea un caso de phishing.	El sistema debe estar disponible y operativo en todo momento para garantizar la protección continua contra amenazas de phishing.

N°.	Requisitos Funcionales	Requisitos No Funcionales
04	La extensión debe contar con una interfaz de usuario sencilla que permita a los usuarios activar o desactivar la detección de phishing.	El código fuente y los componentes del sistema deben estar bien estructurados y documentados para facilitar su mantenimiento y futuras mejoras.
05	El modelo de predicción desarrollado debe ser integrado en una plataforma en la nube para que la extensión de Google Chrome pueda realizar peticiones y obtener predicciones de forma eficiente.	El sistema debe tener un rendimiento óptimo, evitando el consumo excesivo de recursos de la computadora del usuario.

A partir de la tabla 7, se plantea implementar en una historia de usuario épica, que muestra un contexto de lo que tiene el sistema, el mismo que para validar estos requisitos, se llevo a cabo un exhaustivo procedimiento de pruebas y evaluación en diversos entornos y situaciones. El objetivo es confirmar si el sistema cumple todos los requisitos previos y funciona de acuerdo con las expectativas de los usuarios, el procedimiento de validación desempeñó un papel fundamental para garantizar el calibre y la eficacia del sistema de detección de intrusos.

Desarrollo de Metodología

Herramientas y Gestores

En este apartado se describen tanto las herramientas, como los procesos que se han implementado en los gestores para el desarrollo del proyecto.

- Pythonanywhere: Servidor en la nube que brinda la capacidad de ejecutar líneas de comando ya que no requiere configuraciones dentro del entorno local, se usó esta plataforma para

proporcionar archivos .py que son alojadas en una API desarrollada con el modelo entrando, de tal forma que se pueda verificar su funcionalidad en diferentes entornos, solo que son el hecho de tener conexión a internet y poner la dirección de esta.

- Postman: La utilización de esta herramienta fue con el fin de probar APIs, se usó la conexión con la plataforma antes mencionada “PythonAnywhere”, así mismo Postam facilitó el envío de requerimientos HTTP a la API desarrollada y el análisis de las respuestas, corroborando el funcionamiento deseado.
- Visual Studio Code: La herramienta Visual Studio Code se empleó como entorno de desarrollo gracias a su configuración avanzada para la codificación, depuración y gestión de código, esto ayudó a la identificación de problemas dentro de la codificación, al tiempo contribuye a una perfecta integración con la herramienta de repositorio de código, GitHub.
- Github: GitHub funciona como una plataforma de alojamiento de código y colaboración, desempeñando un papel crucial en el desarrollo y progreso del proyecto. La plataforma facilitó el control de versiones, considerando las modificaciones del código de todos los colaboradores involucrados, contribuyendo en gran medida el trabajo en equipo sin fisuras, mejorando esfuerzos y optimizando tiempos.

Épica

Un Epic se refiere a una historia de usuario de escala significativa, es un término utilizado para describir una historia que no se puede lograr en una sola iteración debido a su esfuerzo sustancial. A diferencia de las historias de usuario normales, las épicas tienen un mayor nivel de complejidad y están ligadas a una notable incertidumbre (Menzinky, 2018).

La tabla 8 exhibe la épica "HU", presentando en detalle la función del usuario final del sistema, las características y/o funcionalidades requeridas, y la razón por la cual deberían ser implementadas.

Tabla 8

Historias de Usuario

ID	Nombre	Rol	Característica/ Funcionalidad	Razón / Resultado
1	H.U. 01	Como Usuario	Deseo una extensión de Google Chrome que pueda detectar y me diga si un sitio web contiene phishing.	Para disponer de un medio para detección si estoy navegando en un sitio web seguro de Internet con Google Chrome.

La tabla 9 presenta la historia de usuario propuesta que se desarrolló a lo largo del proyecto, junto a una estimación de tiempo en días, la fecha de inicio, la fecha de fin y el N° de Sprint al que corresponde cada historia de usuario especificada.

Tabla 9

Product Backlog del Proyecto

Historia de Usuario	Nombre	Estimación (días)	Fecha de Inicio	Fecha Fin	N° de Sprint
1	H.U. 01	64	04/05/2023	17/08/2023	01

Lista de Tareas

Las listas de tareas en un sprint que se encuentran en el Sprint Backlog son características del producto que se divide en tareas más pequeñas a desarrollar (Deemer, P., 2009), la tabla 10 muestra la lista de actividades previstas para el proyecto. Se trata de delimitar las responsabilidades de los desarrolladores, detallar las características o funciones necesarias y explicar los motivos de su incorporación.

Tabla 10

Lista de Tareas

ID	Nombre	Característica/ Funcionalidad	Razón / Resultado
1	L.T 01	La extensión usará el algoritmo y/o modelo más efectivo de Machine Learning para detectar el phishing en sitios web.	El complemento puede hacer predicciones con buena precisión.
2	L.T. 02	Se creará dataset que contenga funciones que puedan identificar los sitios de phishing de los legítimos.	Necesario para entrenar el modelo de Machine Learning
3	L.T. 03	Implementar un modelo capaz de alojar el modelo de Machine Learning, permitiendo la realización de predicciones a través de este servicio.	Obtenga un servicio que se puede utilizar en otras aplicaciones.

La tabla 11 se muestra la lista de tareas propuestas que serán desarrolladas a lo largo del proyecto. Junto a una estimación de tiempo en días, la fecha de inicio, la fecha de fin y el N° de Sprint al que corresponde cada historia de usuario especificada.

Tabla 11

Product Backlog de las listas de Tareas

Lista de Tareas	Nombre	Estimación (días)	Fecha de Inicio	Fecha Fin	N° de Sprint
1	L.T. 01	22	04/05/2023	25/05/2023	01
2	L.T. 02	21	01/06/2023	22/06/2023	02
3	L.T. 03	21	29/06/2023	17/08/2023	03

Implementación de algoritmos y modelos de Machine Learning para sitios web Phishing

La metodología Scrum dicta que la planificación de cada Sprint se realiza cuando se completa el Product Backlog del proyecto (que contiene historias de usuarios y sus asignaciones a Sprints individuales). Los Sprints se organizan según la prioridad de desarrollo, lo que significa crear un Sprint Backlog adecuado (Gonzaga Y Esteban, 2020).

Cabe señalar que para la realización de reuniones dictadas por Scrum se realizan reuniones virtuales a través de la plataforma Discord, aunque también se realizan reuniones presenciales en caso de ser necesario.

Durante el desarrollo de todos los sprints en este proyecto, se empleó un procesador Intel Core i7-13700K, 3.4GHZ de 16 CORE con DDR4 de 32GB de memoria RAM y el sistema operativo Windows 11 como componente de hardware utilizado para ejecutar el código desarrollado.

Sprint 01: Selección del mejor modelo de Machine Learning

La ejecución del Sprint 01, se establece como punto de partida la lista de tarea L.T. 01 que se detalla en la tabla 10, esta lista de tarea establece la necesidad de seleccionar el algoritmo y/o modelo de Machine Learning más adecuado para detectar sitios web con actividad de phishing.

Lista de tareas detalladas. La tabla 12, se encuentra la descripción detallada de la lista de tarea L.T. 01 del sistema de detección de phishing (Phishing Impact). La tabla proporciona información sobre los responsables del desarrollo y establece los criterios de aceptación para seleccionar el modelo y/o algoritmo óptimo para la detección de actividades de phishing en sitios web.

Tabla 12

Historia de usuario para la selección del modelo y/o algoritmo de Machine Learning

	Lista de Tarea
Número: L.T. 01	**Usuario:** Usuario de internet
Nombre de Historia: Selección del mejor modelo/algoritmo para la detección de Phishing en Sitios Web	
Prioridad: Alta	**Riesgo en desarrollo:** Media
Días estimados: 22	**Interacción asignada**: 1
Programadores responsables: Dennis Caisa, Adrian Guevara	
Descripción:	

La extensión utiliza el algoritmo y/o modelo de Machine Learning más eficiente para detectar actividades de phishing en sitios web.

Actividades:

- Los modelos y/o algoritmos de ML desarrollado en la revisión de la literatura con precisión (accuracy) que supera el promedio calculado de los resultados obtenidos.
- Cada modelo seleccionado debe ser validado y los resultados documentados por separado.
- Se elegirá el modelo y/o algoritmo de Machine Learning con el valor de precisión más alto (accuracy).

La tabla 13 detalla las tareas realizadas durante el desarrollo del sprint, junto con los responsables asignados, las fechas planificadas para su ejecución, las estimaciones de tiempo en horas, el esfuerzo real empleado diariamente en horas y el estado actual de cada tarea. Es importante destacar que esta tabla muestra el sprint backlog una vez que ha sido completado.

Tabla 13

Spring Backlog 01

			Sprint 01			
Fecha Inicio:			**Fecha Fin:**		**Jornada**	
04/05/2023			25/05/2023		8 horas diarias	
H.U	**Tareas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsable**	**Estado**
01	Extracción de modelos y/o algoritmos de Machine Learning	24	04/05/2023	09/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Extracción de valores del accuracy	24	10/05/2023	12/05/2023	Dennis Caisa y Adrian Guevara	Finalizado

	respectivamente					
01	Documentación de la Frecuencia de los modelos y/o algoritmos de Machine Learning	24	13/05/2023	15/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Selección de modelos y/o algoritmos de Machine Learning para realización de pruebas	24	16/05/2023	18/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Implementación de modelos y/o algoritmos de Machine Learning seleccionados	16	19/05/2023	20/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Ejecución de pruebas	16	21/05/2023	22/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Documentación de métricas de evaluación resultantes	16	23/05/2023	24/05/2023	Dennis Caisa y Adrian Guevara	Finalizado
01	Selección de modelo y/o algoritmo de Machine Learning con mejor valor de accuracy	8	25/05/2023	25/05/2023	Dennis Caisa y Adrian Guevara	Finalizado

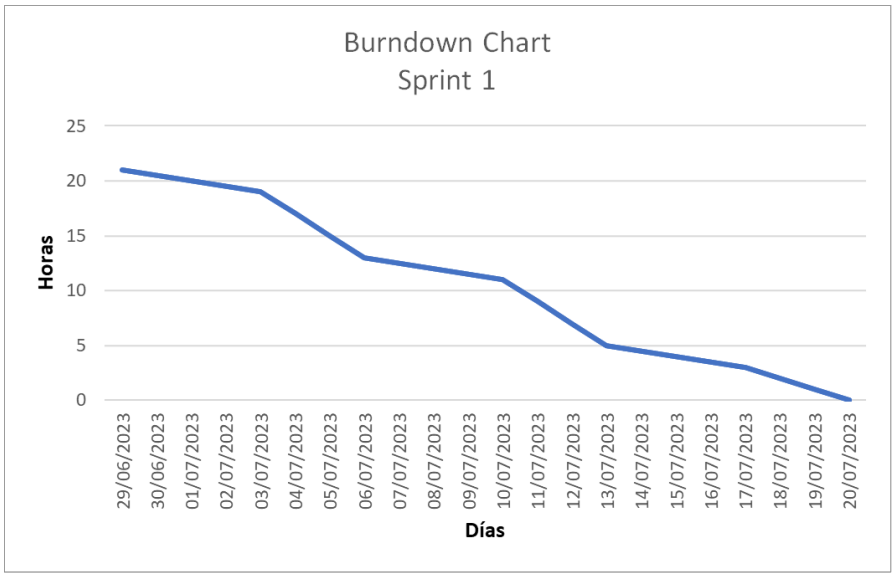
***Burndown chart*.** Burndown chart o también conocido como un gráfico de agotamiento es un radiador de información que representa visualmente una "trayectoria de valores" del sprint/iteración (Dalton J, 2019). Burndown Chart - Sprint 01, representado en la figura 4, ilustra el progreso realizado durante el tiempo previsto para el desarrollo del sprint.

En el eje horizontal (X), se muestran las fechas correspondientes a los días mencionados en la Tabla 11. En este caso, el período de tiempo abarca desde el 04/05/2023 hasta el 25/05/2023. Por otro lado, en el eje vertical (Y), se presenta el número total de horas estimadas al inicio, el valor se obtiene multiplicando el número total de días estimados por las horas trabajadas diariamente.

Para este sprint, se considera un total de 13 días con actividades de 1-2 horas de trabajo por día, lo que resulta en 34 horas. Este valor máximo representa el punto de partida en el eje Y, a medida que avanzan los días, el valor de las horas debe disminuir progresivamente con el objetivo de alcanzar cero y completar el sprint exitosamente.

Figura 4

Burndown Chart - Sprint 01



***Resultado del Sprint*.** En esta sección se presenta un resumen del proceso realizado y los resultados más destacados obtenidos durante la ejecución del sprint y al finalizarlo.

Una vez identificados los modelos y/o algoritmos de Machine Learning que mostraron los mejores valores de precisión en la detección de phishing en sitios web, se procedió a implementarlos y evaluarlos utilizando el conjunto de datos del estudio "Phishing Website Detection by Machine Learning Techniques" (Rashid, 2020). Este conjunto de datos contiene URLs con 40 características extraídas. Se utilizó este conjunto de datos debido a que el objetivo de este sprint era encontrar el mejor modelo y/o algoritmo de Machine Learning. Es importante mencionar que en este sprint aún no se contaba con el conjunto de datos propuesto en el proyecto.

La figura 5 muestra la implementación de los modelos y/o algoritmos de Machine Learning, junto con el código desarrollado y las métricas de evaluación correspondientes: precisión (accuracy), precisión (precision), recall y puntuación F1.

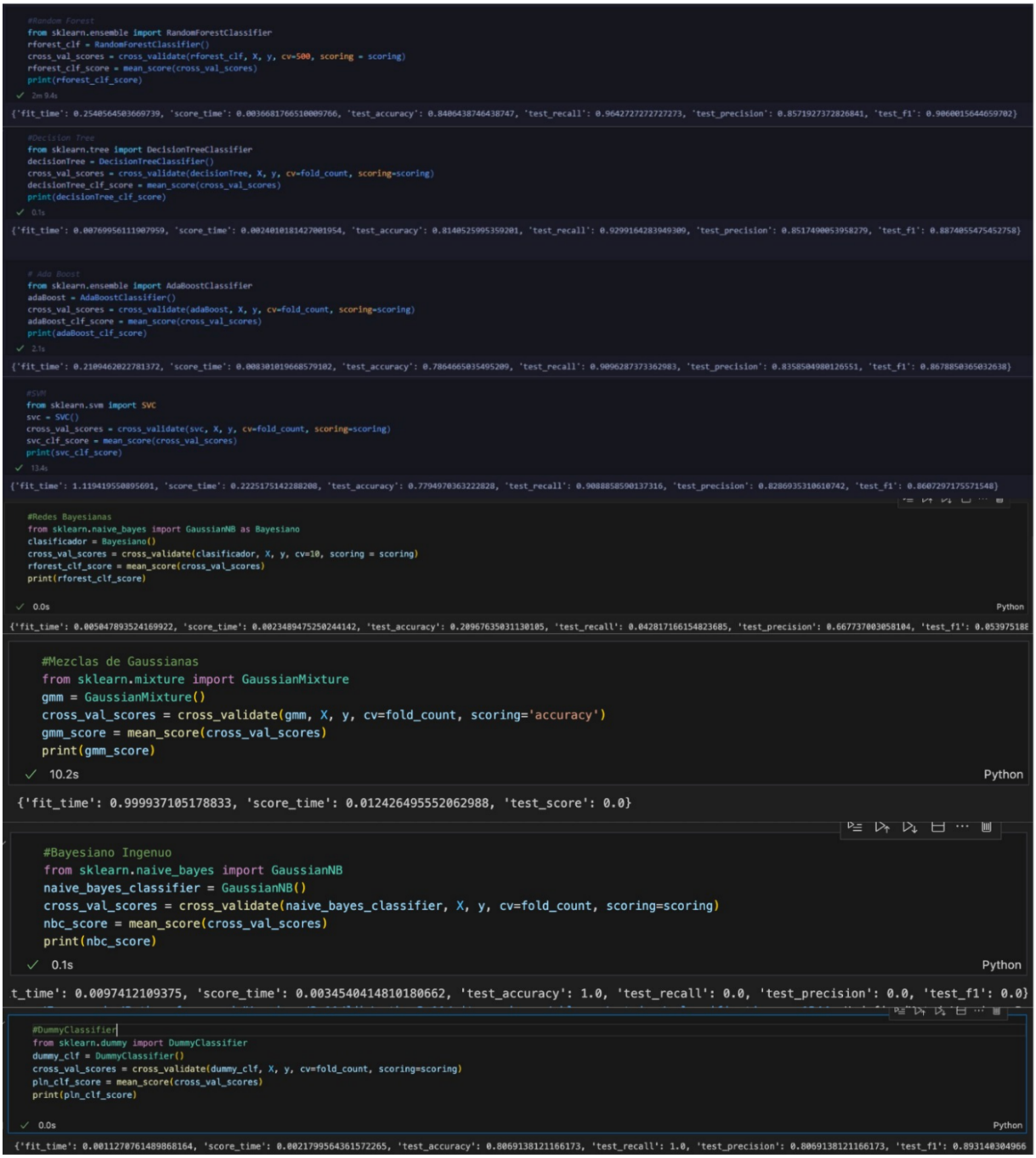
La tabla 14 presenta las métricas de evaluación de los resultados obtenidos después de llevar a cabo las pruebas mencionadas para cada modelo y/o algoritmo seleccionado. Los resultados fueron utilizados para determinar el modelo que se implementó en el desarrollo del sistema, considerando principalmente los valores de precisión (accuracy).

La precision (accuracy) nos proporciona información sobre el porcentaje de elementos clasificados correctamente, tanto positivos como negativos. Se aplicó una heurística que condujo a resultados con una notable mejora en la precisión.

Una heurística conduce a resultados con una notable mejora en la precisión (Ruiz, 2005). Es por eso por lo que al aplicar esta técnica se llegó a la conclusión que al usar todas las características en los modelos es la mejor opción.

Figura 5

Implementación de modelos y/o algoritmos de machine Learning



Después de analizar los resultados presentados en la tabla 14, se decidió seleccionar el modelo y/o algoritmo denominado Random Forest. Este modelo mostró un valor de precisión (accuracy) de 0,8404, lo que representa un 84,06% en términos porcentuales.

Tabla 14

Resultados pruebas modelos y/o algoritmos de machine Learning implementados

Características	**Algoritmos/Modelos**	**Accuracy**	**Recall**	**Precision**	**F1**
40 características	Random Forest	0,8406	0,9642	0,8571	0,9060
	Decision Tree	0.8140	0,9299	0,8517	0,8874
	Ada Boost	0.7864	0,9096	0,8358	0,8678
	SVM	0,7794	0,9088	0,8286	0,8607
	Redes Bayesianas	0,2096	0,0428	0,6677	0,0539
	Mezcla de Gaussianas	0	0	0	0
	Bayesiano Ingenuos	0,2139	0,0330	0,6864	0,0489
	Dummy Classifier	0,8069	1,0	0,8069	0,8931

Cuando se analizaron los resultados de los diversos algoritmos derivados de la investigación, se evidenció que el algoritmo de Random Forest arrojó los resultados más sobresalientes en términos de rendimiento. Este algoritmo demostró ser el más eficaz para su integración en el sistema.

Sprint 02: Creación de DataSet

El presente Sprint se fundamentó en la lista de tarea L.T. 02, tal como se detalla en la tabla 10. En un principio, se abordó el proceso de identificar y elegir las características extraíbles de las URLs.

A continuación, se desarrollaron escenarios para determinar cuáles de estas características contribuyen significativamente en la detección de sitios web con phishing. Por último, se generó un dataset que sería utilizado para entrenar al modelo seleccionado.

Lista de tareas detalladas. La tabla 15 proporciona una descripción exhaustiva de la lista de tarea L.T. 02, que se enfoca en la creación de un dataset para el sistema de detección de phishing (Phishing Impact). En dicha tabla, se identifican claramente los responsables encargados del desarrollo y se establecen los criterios de aceptación para la creación del dataset. Este conjunto de datos contiene las características específicas seleccionadas para la implementación de la propuesta del sistema.

Tabla 15

Historia de usuario para la creación de DataSet

Lista de Tareas	
Número: L.T. 02	**Usuario:** Usuario de internet
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 21	**Interacción asignada:** 1
Programadores responsables: Dennis Caisa, Adrian Guevara	
Descripción: El dataset que incluya características que posibiliten la identificación de sitios web con phishing y diferenciarlos de los legítimos.	
Validación (Criterios de aceptación): • Se seleccionarán características que tengan una frecuencia mayor o igual a la media de la frecuencia total. • Se realizarán pruebas con los modelos y/o algoritmos de Machine Learning en diferentes escenarios, variando la cantidad de características asignadas a cada uno. • Se realizarán pruebas adicionales combinando varios escenarios entre sí. • Se extraerán las características seleccionadas a partir de una URL. • El dataset final se creará utilizando las características de URL de sitios web tanto legítimos como de phishing.	

Sprint Backlog. La tabla 16 se detallan las tareas ejecutadas durante el desarrollo del sprint, indicando los responsables asignados para cada una, las fechas de planificación para la ejecución del sprint, la estimación de tiempo en horas para cada tarea, el esfuerzo real invertido en horas diarias, y el

estado actual de cada tarea. Es importante destacar que esta tabla muestra el sprint backlog ya completado y finalizado.

Tabla 16

Sprint Backlog 02

Sprint 02						
Fecha Inicio: 01/06/2023			**Fecha Fin:** **22/06/2023**		**Jornada** **8 horas diarias**	
H.U	**Tareas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsable**	**Estado**
02	Extracción de características de URL enfocadas en indicadores de compromiso para identificar sitios web con phishing	24	01/06/2023	03/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Conteo de frecuencia de cada característica extraída	16	04/06/2023	05/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Selección de características de indicadores de compromiso para realizar pruebas	8	06/06/2023	07/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Creacion de generación de escenarios con distintas cantidades de características para detectar sitios web con phishing	16	08/06/2023	09/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Implementación de los escenarios de características con los modelos y/o Algoritmos de Machine Learning	16	10/06/2023	11/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Ejecución de pruebas con los diferentes escenarios	8	12/06/2023	13/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Documentación de métricas de evaluación resultantes	8	14/06/2023	15/06/2023	Dennis Caisa y Adrian Guevara	Finalizado

Sprint 02						
Fecha Inicio:			**Fecha Fin:**		**Jornada**	
01/06/2023			**22/06/2023**		**8 horas diarias**	
H.U	**Tareas**	**Horas**	**Fecha Inicio**	**Fecha Fin**	**Responsable**	**Estado**
02	Selección de datasets con phishing y sitios web legítimos	8	16/06/2023	17/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Limpieza y unión de datasets seleccionados	8	18/06/2023	19/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Implementacion de codigo para la extraccion de caracteristicas de una URL	32	20/06/2023	21/06/2023	Dennis Caisa y Adrian Guevara	Finalizado
02	Implementación de código para la creación de dataset	16	21/06/2023	22/06/2023	Dennis Caisa y Adrian Guevara	Finalizado

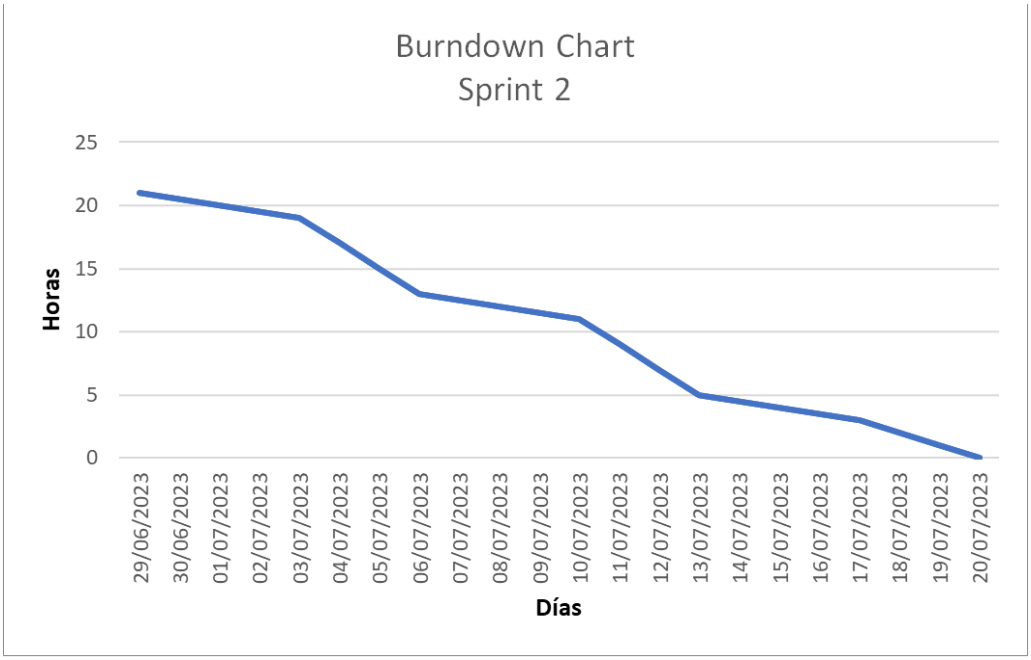
Burndown Chart. La figura 6, ilustra el progreso realizado durante el desarrollo del presente sprint en función del tiempo estimado. En el eje X se representan las fechas de los días específicos, comenzando desde el 01/06/2023 hasta el 22/06/2023. Por otro lado, el eje Y muestra el número total de horas estimadas al inicio del sprint, calculado al multiplicar el total de días estimados por las horas de trabajo diarias.

El sprint tiene una duración de 13 días con actividades de 1-2 horas de trabajo por día, lo que da como resultado un valor máximo de 20 horas. A medida que avanzan los días, el valor de horas en el eje Y debe ir disminuyendo con el objetivo de alcanzar el valor cero, lo que representa la finalización exitosa del sprint al completar todas las tareas planificadas en el tiempo estimado.

El Burndown Chart ofrece una visualización clara del progreso realizado durante el sprint y permite monitorear si el equipo se está ajustando al plan establecido para el proyecto.

Figura 6

Burndown Chart - Sprint 02



Resultados del Sprint. Durante la ejecución del Sprint 02, se llevó a cabo un proceso detallado en el que se trabajaron diversas tareas para el desarrollo del sistema de detección de phishing (Hunter Phisher). Una vez finalizado el sprint, se obtuvieron resultados significativos. Se creó un dataset a partir de dos repositorios que son KAGGLE, MISP y MENDELEYDATA los mismos que contenían las características seleccionadas previamente, el cual sería utilizado para entrenar los modelos y algoritmos de Machine Learning.

Dando continuidad al Sprint, se ha puesto en marcha el código para llevar a cabo pruebas en varios escenarios, diseñados para validar los resultados. En esta etapa, se utilizarán 40 características como recursos de comprobación para cada uno de los escenarios. De estas características, 30 corresponden a las URL y las otras 10 pertenecen a los Indicadores de Compromiso.

En los escenarios, se combinarán las 30 características de URL con 1 Indicador de Compromiso, de modo que se considerarán todas las métricas de evaluación de la tabla 17. En base a esto, se han planteado las combinaciones más relevantes.

Se han desarrollado 6 escenarios con el propósito de evaluar el impacto de diversas características en la detección de phishing en sitios web. Estos escenarios combinaron diferentes grupos de características, tanto las más sobresalientes como las menos relevantes, para determinar cuáles aportan más al proceso de detección.

A continuación, se presentan los ejemplos de cada escenario: En el primer escenario, se combinó las 30 características de URL con los IOCs 1 y 2, en el segundo escenario, se combinó las 30 características de URL con los IOCs 4 y 6, el tercer escenario involucra la combinación de las 30 características de URL con los IOCs 9 y 10, para el cuarto escenario, se combinó las 30 características de URL con los IOCs 1, 2 y 4, en el quinto escenario, se combinó las 30 características de URL con los IOCs 6, 9 y 10, finalmente, el último escenario combinó las 30 características de URL con todos los IOCs anteriores (1, 2, 4, 6, 9, 10), como se muestra en la figura 7.

Durante el Sprint 01, se verificó el rendimiento de las 40 características combinadas, cuyos resultados se encuentran detallados en la tabla 14, es relevante mencionar que estas pruebas se llevaron a cabo utilizando los 4 modelos mencionados en la tabla 3.

La tabla 17 presentan los resultados obtenidos a partir de la ejecución del código, tal como se muestra en la figura 7. Cabe destacar que se aplicaron las métricas de evaluación mencionadas anteriormente para medir el rendimiento de los modelos y/o algoritmos de Machine Learning, las métricas incluyen Accuracy, Precision, Recall y F1. En las pruebas llevadas a cabo, se utilizó un dataset compuesto por la combinación de varias Bases de Datos detalladas en el Anexo 3, el dataset incluía una variedad de URLs, tanto legítimas como con phishing.

Primeramente, se manejó la métrica “Accuracy”, que determina el porcentaje de sitios web correctamente clasificados como legítimos o phishing. Dado su valor informativo, esta métrica fue considerada la más importante en la evaluación. En el primer escenario (que combinó 30 características de URL con los IOCs 1 y 2), el algoritmo Decision Tree alcanzó un accuracy de 0,8134 (84,34%), el segundo escenario (que combinó 30 características de URL con los IOCs 4 y 6), el algoritmo Decision Tree alcanzó un favorable accuracy de 0,8046 (80,46%), por otro lado, el tercer escenario (que combinó 30 características de URL con los IOCs 9 y 10), el algoritmo Random Forest se logró un accuracy de 0,8280 (82,80%).

Una vez examinados los resultados que se obtuvieron en los escenarios 1, 2 y 3, se concluye que la combinación de las 30 características de URL con los IOCs 9 y 10, tienen un aporte mayor en la detección de phishing en páginas web, las combinaciones se realizan con la finalidad de notar si existen aumentos y/o disminuciones en el accuracy de detección.

En la primera combinación, que involucró los escenarios 1 y 2 (que combinó 30 características de URL con los IOCs 1, 2 y 4), el algoritmo Random Forest obtuvo un accuracy con 0,8400 (84,00%), la segunda combinación, que incluyó los escenarios 2 y 3 (que combinó 30 características de URL con los IOCs 6, 9 y 10), el algoritmo random Forest alcanzó un accuracy de 0,8283 (82,83%), la combinación de los tres escenarios (que combinó 30 características de URL con los IOCs 1, 2, 4, 6, 9 y 10), el algoritmo Random Forest sobresalió con un accuracy de 0,8402 (84,02%).

Para la métrica “Precision”, se enfoca en calcular el porcentaje de sitios web correctamente identificados como phishing entre todos los sitios web clasificados como tales. Asimismo, el proceso para evaluar esta métrica fue similar al utilizado para la métrica accuracy.

En el primer escenario (que combinó 30 características de URL con los IOCs 1 y 2), el algoritmo Decision Tree alcanzó una precision de 0,8502 (85,02%), el segundo escenario (que combinó 30

características de URL con los IOCs 4 y 6), el algoritmo Adaboost alcanzó una precision de 0,8444 (84,44%), el tercer escenario (que combinó 30 características de URL con los IOCs 9 y 10), el algoritmo Random Forest se logró una precision de 0,9261 (92,61%).

Analizando los resultados que se obtuvieron en los escenarios 1, 2 y 3, se concluye que la combinación de las 30 características de URL con los IOCs 9 y 10, tienen un aporte mayor en la detección de phishing en páginas web. Por otro lado, las combinaciones se realizan con la finalidad de notar si existen aumentos y/o disminuciones en la precision de detección.

En la primera combinación, que involucró los escenarios 1 y 2 (que combinó 30 características de URL con los IOCs 1, 2 y 4), el algoritmo Random Forest obtuvo una precision de 0,8562 (85,52%), la segunda combinación, que incluyó los escenarios 2 y 3 (que combinó 30 características de URL con los IOCs 6, 9 y 10), el algoritmo random Forest alcanzó una precision de 0,8465 (84,65%), la combinación de los tres escenarios (que combinó 30 características de URL con los IOCs 1, 2, 4, 6, 9 y 10), el algoritmo Random Forest sobresalió con una precision de 0,8500 (85,00%). La métrica precision mejor mientras más características detecta.

La métrica “Recall”, se llevó a cabo el mismo procedimiento que en las métricas de precisión y accuracy para evaluar, la cual establece el porcentaje de sitios web con phishing correctamente identificados con respecto al total de ejemplos de sitios web con phishing en el conjunto de entrenamiento.

El primer escenario (que combinó 30 características de URL con los IOCs 1 y 2), el algoritmo Decision Tree alcanzó un recall de 0,9313 (93,13%), el segundo escenario (que combinó 30 características de URL con los IOCs 4 y 6), el algoritmo Ada Boost alcanzó un favorable recall de 0,9269 (92,69%), el tercer escenario (que combinó 30 características de URL con los IOCs 9 y 10), el algoritmo Random Forest se logró un recall de 0,9607 (96,07%).

Tras examinar los resultados que se obtuvieron en los escenarios 1, 2 y 3, se concluye que la combinación de las 30 características de URL con los IOCs 9 y 10, tienen un aporte mayor en la detección de phishing en páginas web, las combinaciones se realizan con la finalidad de notar si existen aumentos y/o disminuciones en el Recall de detección.

La primera combinación, que involucró los escenarios 1 y 2 (que combinó 30 características de URL con los IOCs 1, 2 y 4), el algoritmo Random Forest obtuvo un recall con 0,9631 (96,31%), la segunda combinación, que incluyó los escenarios 2 y 3 (que combinó 30 características de URL con los IOCs 6, 9 y 10), el algoritmo random Forest alcanzó un recall de 0,9613 (96,13%), la combinación de los tres escenarios (que combinó 30 características de URL con los IOCs 1, 2, 4, 6, 9 y 10), el algoritmo Random Forest sobresalió con un recall de 0,9636 (96,36%). Se analiza que en la métrica recall va mejorando sus números mientras mayor número de características existan.

Para la métrica “F1”, se emplea para unificar las medidas de precisión y recall en un solo valor, lo que facilita la comparación del rendimiento. Se siguió el mismo procedimiento que para las métricas de accuracy, precision y recall.

El primer escenario (que combinó 30 características de URL con los IOCs 1 y 2), el algoritmo Decision Tree alcanzó un F1 de 0,8873 (88,73%), el segundo escenario (que combinó 30 características de URL con los IOCs 4 y 6), el algoritmo Adaboost alcanzó un F1 de 0,8821 (88,21%), el tercer escenario (que combinó 30 características de URL con los IOCs 9 y 10), el algoritmo Random Forest se logró un F1 de 0,8993 (89,93%).

Analizando los resultados que se obtuvieron en los escenarios 1, 2 y 3, se concluye que la combinación de las 30 características de URL con los IOCs 9 y 10, tienen un aporte mayor en la detección de phishing en páginas web, las combinaciones se realizan con la finalidad de notar si existen aumentos y/o disminuciones en el F1 de detección.

En la primera combinación, que involucró los escenarios 1 y 2 (que combinó 30 características de URL con los IOCs 1, 2 y 4), el algoritmo Random Forest obtuvo un F1 de 0,9057 (90.57%), la segunda combinación, que incluyó los escenarios 2 y 3 (que combinó 30 características de URL con los IOCs 6, 9 y 10), el algoritmo random Forest alcanzó un F1 de 0,8996 (89,96%), la combinación de los tres escenarios (que combinó 30 características de URL con los IOCs 1, 2, 4, 6, 9 y 10), el algoritmo Random Forest sobresalió con un F1 de 0,9059 (90,59%). Se observa que el algoritmo F1 mejora considerablemente con mayor número de características.

Figura 7

Pruebas de características con diferentes escenarios

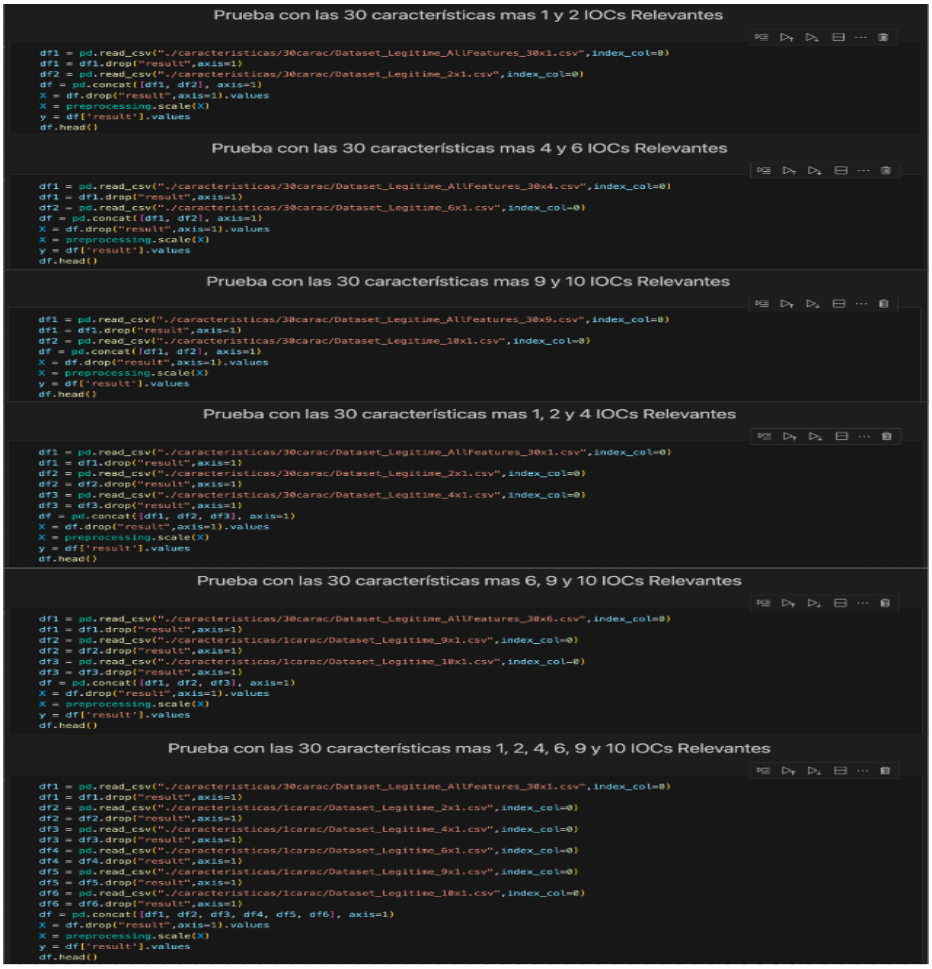


Tabla 17

Resultados pruebas modelos y/o algoritmos de Machine Learning implementados en diferentes escenarios

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
1	30 características vs 1 y 2 IOC	Random Forest	0.8056	0.9226	0.8475	0.8810
		Decision Tree	0.8134	0.9313	0.8502	0.8873
		Ada Boost	0.7982	0.9173	0.8377	0.8727
		SVM	0.7774	0.9056	0.8282	0.8589
		Redes Bayesianas	0.7895	0.9000	0.8447	0.8665
		Mezcla de Gaussianas	0.8095	0.9262	0.8494	0.8842
		Bayesiano Ingenuos	0.7899	0.9116	0.8385	0.8702
		Dummy Classifier	0.7834	0.9071	0.8337	0.8634
2	30 características vs 4 y 6 IOC	Random Forest	0.7974	0.9199	0.8410	0.8762
		Decision Tree	0.8046	0.9269	0.8444	0.8821
		Ada Boost	0.7750	0.9046	0.8283	0.8609
		SVM	0.7774	0.9056	0.8282	0.8589
		Redes Bayesianas	0.7884	0.9067	0.8403	0.8616

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		Mezcla de Gaussianas	0.8022	0.9267	0.8422	0.8806
		Bayesiano Ingenuos	0.7685	0.8972	0.8252	0.8548
		Dummy Classifier	0.7707	0.8913	0.8294	0.8521
3	30 características vs 9 y 10 IOC	Random Forest	0.8280	0.9607	0.8465	0.8993
		Decision Tree	0.7980	0.9245	0.8396	0.8781
		Ada Boost	0.7690	0.8977	0.9261	0.8562
		SVM	0.7725	0.9032	0.8247	0.8565
		Redes Bayesianas	0.7886	0.9156	0.8356	0.8704
		Mezcla de Gaussianas	0.7967	0.9237	0.8384	0.8773
		Bayesiano Ingenuos	0.7700	0.8980	0.8268	0.8568
		Dummy Classifier	0.7655	0.8903	0.8261	0.8496
4	30 características vs 1 - 2 y 4 IOC	Random Forest	0.8400	0.9631	0.8562	0.9057
		Decision Tree	0.8139	0.9298	0.8517	0.8863
		Ada Boost	0.7874	0.9109	0.8361	0.8688
		SVM	0.7779	0.9070	0.8278	0.8592
		Redes Bayesianas	0.8034	0.9186	0.8475	0.8788

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		Mezcla de Gaussianas	0.8007	0.9119	0.8498	0.8761
		Bayesiano Ingenuos	0.7835	0.9040	0.8358	0.8641
		Dummy Classifier	0.7795	0.9052	0.8306	0.8599
5	30 características vs 6 - 9 y 10 IOC	Random Forest	0.8283	0.9613	0.8465	0.8996
		Decision Tree	0.7982	0.9247	0.8396	0.8782
		Ada Boost	0.7690	0.8977	0.8261	0.8562
		SVM	0.7725	0.9032	0.8249	0.8565
		Redes Bayesianas	0.7804	0.8998	0.8363	0.8622
		Mezcla de Gaussianas	0.7913	0.9174	0.8372	0.8726
		Bayesiano Ingenuos	0.7603	0.8874	0.8217	0.8476
		Dummy Classifier	0.7692	0.8988	0.8235	0.8535
6	30 características vs 1 - 2 - 4 - 6 - 9 y 10 IOC	Random Forest	0.8402	0.9636	0.8561	0.9059
		Decision Tree	0.8130	0.9286	0.8514	0.8866
		Ada Boost	0.7874	0.9109	0.8361	0.8688
		SVM	0.7788	0.9081	0.8283	0.8600
		Redes Bayesianas	0.8044	0.9182	0.8487	0.8892

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		Mezcla de Gaussianas	0.8057	0.9186	0.8509	0.8805
		Bayesiano Ingenuos	0.7796	0.8992	0.8341	0.8602
		Dummy Classifier	0.7857	0.9132	0.8328	0.8658

La tabla 18 muestran los modelos y/o algoritmos de ML que destacaron con los mejores resultados con cada una de las métricas aplicadas, el algoritmo Random Forest destaca la métrica accuracy, con mayor porcentaje para la detección precisa en páginas web, según la métrica de Precision, el algoritmo de Decision Tree obtuvo resultados destacados en los dos primeros escenarios, mientras que en los últimos escenarios el algoritmo de Random Forest se posicionó como el ganador.

En cuanto a la métrica Recall, nuevamente el algoritmo Random Forest destacó como el mejor. En relación con la métrica F1, los dos primeros escenarios mostraron un rendimiento sobresaliente por parte del algoritmo Decision Tree, pero no se pudo ignorar que el algoritmo Random Forest también volvió a resaltar.

En conclusión, el algoritmo Random Forest sobresale en la mayoría de los escenarios y métricas evaluadas, mostrando un rendimiento sólido en la detección de páginas web con phishing, los algoritmos Decision Tree y Ada Boost también obtuvieron resultados destacables en ciertas métricas y escenarios específicos en varias ocasiones.

Tabla 18

Ganador de cada escenario

Ord	Escenario / Métrica	Accuracy	Precision	Recall	F1
1	Escenario (30 Características y 1 y 2 IOC)	Decision Tree (81,34%)	Decision Tree (93,13%)	Random Forest (93,13%)	Decision Tree (88,73%)
2	Escenario (30 Características y 4 y 6 IOC)	Decision Tree (80,46%)	Decision Tree (92,69%)	Decision Tree (84,44%)	Decision Tree (88,21%)
3	Escenario (30 Características y 9 y 10 IOC)	Random Forest (82,80%)	Random Forest (96,07%)	Ada Boost (92,61%)	Random Forest (89,93%)
4	Escenario (30 Características y 1 y 2 y 4 IOC)	Random Forest (84,00%)	Random Forest (96,31%)	Random Forest (85,62%)	Random Forest (90,57%)
5	Escenario (30 Características y 9 y 10 y 6 IOC)	Random Forest (82,83%)	Random Forest (96,13%)	Random Forest (84,65%)	Random Forest (88,96%)
6	Escenario (30 Características vs 1 y 2 y 4 y 6 y 10 y 9 IOC)	Random Forest (84,02%)	Random Forest (96,36%)	Random Forest (85,61%)	Random Forest (90,59%)

Creación del dataset

La lista de tarea L.T. 02, se plantea la necesidad de crear un dataset y desarrollar el código para extraer 40 características, tanto de las URLs como de los IOCs, las características arrojarán resultados de 1 y -1, donde 1 indica que es legítimo y -1 que es ilegítimo. La figura 8 presenta el resultado de la compilación del código encargado de extraer todas las características de la URL, donde se muestra la URL enviada (https://ec.ebay.com/), la respuesta del sitio web (<Response [200]>), y un arreglo que contiene las 40 características obtenidas.

Figura *8*

Extracción de características del sitio web eBay

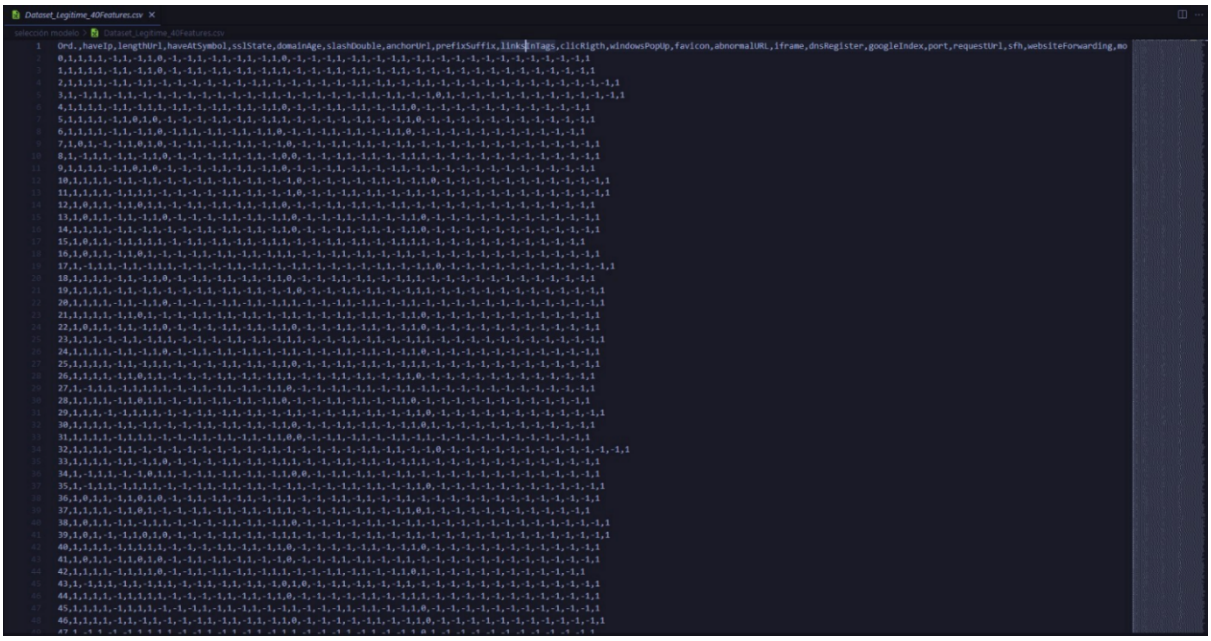
```
La url ingresada es https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&ru=https%3A%2F%2Fec.ebay.com%2F
<Response [200]>
[1, -1, 1, 1, 1, 1, -1, 1, 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, 0, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
 -1, -1, -1, -1]
```

Después de desarrollar, probar y ejecutar el código que extrajo características, se procedió a combinar varios conjuntos de datos que se detallan en el Anexo 3, los conjuntos contienen URLs etiquetadas como legítimas y URLs con phishing, respectivamente, la unión de todas las bases de datos seleccionadas resultó en un total de 65,000 URLs, de las cuales 45,000 eran URLs con phishing y 20,000 eran legítimas. Posteriormente, se realizó una depuración para limpiar los datos, lo que llevó a un total de 13,192 URLs finales, de las cuales 3,987 eran URLs con phishing y 9,205 eran URLs legítimas, las URLs se guardaron en un archivo CSV para su posterior uso.

Una vez almacenado el archivo CSV, se ejecutó el código que es específicamente para la creación del dataset, el código es el encargado de leer el archivo csv, evaluar cada URL y extraer las características correspondientes y realizar una lista que se guarda en otro archivo CSV, esto finalmente genera el dataset que se utilizará para entrenar el modelo seleccionado. La figura 9 presenta una pequeña parte del dataset creado en el actual Sprint, contiene las características extraídas de cada URL, mostrando los resultados de páginas web legítimas (1) o páginas web con phishing (-1).

Figura *9*

DataSet creado



Sprint 03: Creación de la API

El Sprint actual se basó en la lista de tareas L.T. 03, que se detalla en la tabla 10. Se entrena el modelo y/o algoritmo de Machine Learning seleccionando con el dataset que se almacenó, siguiendo con el proceso se guarda el modelo entrenado, y finalizando se crea la API con las diferentes rutas solicitadas para realizar predicciones a través de una URL.

Lista de tareas detalladas. La tabla 19, proporciona la lista de tarea L.T.03, que forma parte del sistema de detección de phishing (Phishing Impact) en detalle, esta historia específica los responsables del desarrollo y los criterios de aceptación que hacen factible la creación de la API.

Tabla 19

Historia de usuario para la creación de la API

Lista de Tareas	
Número: L.T. 03	**Usuario:** Usuario de internet
Nombre historia: Creación de la API	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 20	**Interacción asignada:** 1
Programadores responsables: Dennis Caisa, Adrian Guevara	
Descripción: El modelo de Machine Learning se hallará almacenado en un servidor y conseguirá efectuar predicciones a través de un servicio.	
Validación (Criterios de aceptación): • La API se retendrá en un servidor. • Se pondrán en marcha dos rutas. Primero, ejecutará la predicción donde será obligatorio el atributo URL. Segundo, existirá la ruta raíz donde se presentará un mensaje de bienvenida. • La API tendrá que ejecutar la predicción utilizando el modelo que se halle almacenado en el mismo servidor. • La API tiene que dar como respuesta 1 cuando el sitio web es legítimo, caso contrario tiene que dar -1 si contiene phishing.	

***Sprint Backlog*.** La tabla 20 detalla las tareas ejecutadas durante el desarrollo del sprint, indicando los responsables asignados para cada una, las fechas de planificación para la ejecución del sprint, la estimación de tiempo en horas para cada tarea, el esfuerzo real invertido en horas diarias, y el estado actual de cada tarea. Es importante destacar que esta tabla muestra el sprint backlog ya completado y finalizado.

Tabla *20*

Sprint Backlog 03

Sprint 03						
Fecha Inicio: 29/06/2023			Fecha Fin: 17/08/2023		Jornada 8 horas diarias	
H. U	Tareas	Horas	Fecha Inicio	Fecha Fin	Responsable	Estado
03	Guardar el modelo de Machine Learning entrenado	16	29/06/2023	01/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Preparación del entorno local	16	02/07/2023	06/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Desarrollar API	16	07/07/2023	10/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Pruebas de API	16	11/07/2023	17/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Búsqueda de servidores	16	18/07/2023	19/07/2023	Dennis Caisa y Adrian Guevara	Finalizado

Sprint 03

Fecha Inicio:			Fecha Fin:		Jornada	
29/06/2023			17/08/2023		8 horas diarias	
H. U	Tareas	Horas	Fecha Inicio	Fecha Fin	Responsable	Estado
03	Cotización y selección de un servidor	16	20/07/2023	24/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Contratación de un plan básico del servidor seleccionado	16	25/07/2023	28/07/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Preparación del entorno en el servidor	16	29/07/2023	02/08/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Subida de archivos	8	03/08/2023	05/08/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Despliegue de aplicación	8	06/08/2023	10/08/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	Pruebas de API en el servidor	16	11/08/2023	14/08/2023	Dennis Caisa y Adrian Guevara	Finalizado
03	ConFiguración de CORS	8	15/08/2023	17/08/2023	Dennis Caisa y Adrian Guevara	Finalizado

***Burndown Chart*.** La figura 10 ilustra el progreso realizado durante el desarrollo del presente sprint en función del tiempo estimado. En el eje X se representan las fechas de los días específicos, comenzando desde el 29/07/2023 hasta el 17/08/2023. Por otro lado, el eje Y muestra el número total

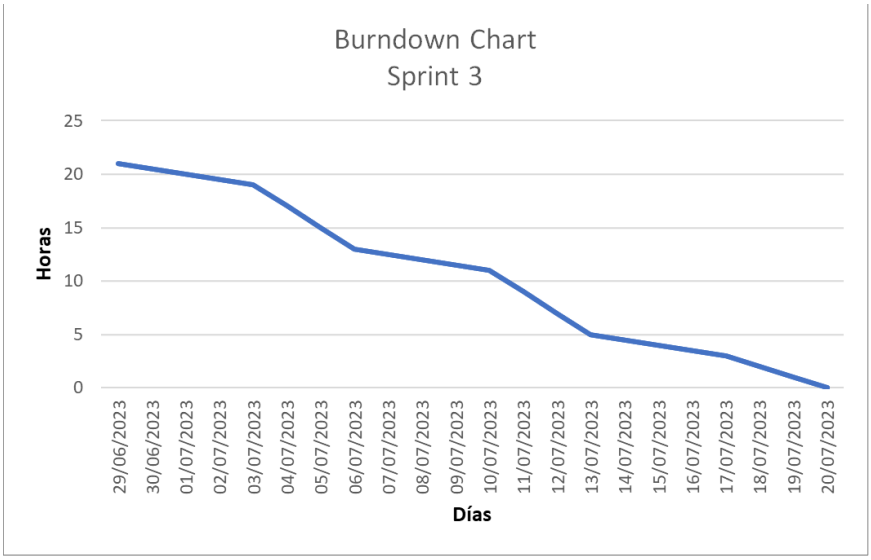
de horas estimadas al inicio del sprint, calculado al multiplicar el total de días estimados por las horas de trabajo diarias.

El sprint tiene una duración de 13 días con actividades de 1-2 horas de trabajo por día, lo que da como resultado un valor máximo de 23 horas. A medida que avanzan los días, el valor de horas en el eje Y debe ir disminuyendo con el objetivo de alcanzar el valor cero, lo que representa la finalización exitosa del sprint al completar todas las tareas planificadas en el tiempo estimado.

El Burndown Chart ofrece una visualización clara del progreso realizado durante el sprint y permite monitorear si el equipo se está ajustando al plan establecido para el proyecto.

F*igura 10*

Burndown Chart - Sprint 03



Resultados del Sprint. Durante la ejecución del Sprint 03, se llevó a cabo un proceso detallado en el que se trabajaron diversas tareas para el desarrollo del sistema de detección de phishing (JKSAD). Una vez finalizado el sprint, se obtuvieron resultados significativos, se entrenó el modelo y/o algoritmo

de ML que se seleccionó al finalizar el Sprint 01 (Random Forest), allí se pudo obtener los valores 0,8416, 0,9642, 0,8571, 0,9060 para las métricas de evaluación de entrenamiento accuracy, recall, precision y f1 respectivamente, la figura 11 presenta que se guardó el modelo ya entrenado.

Figura 11

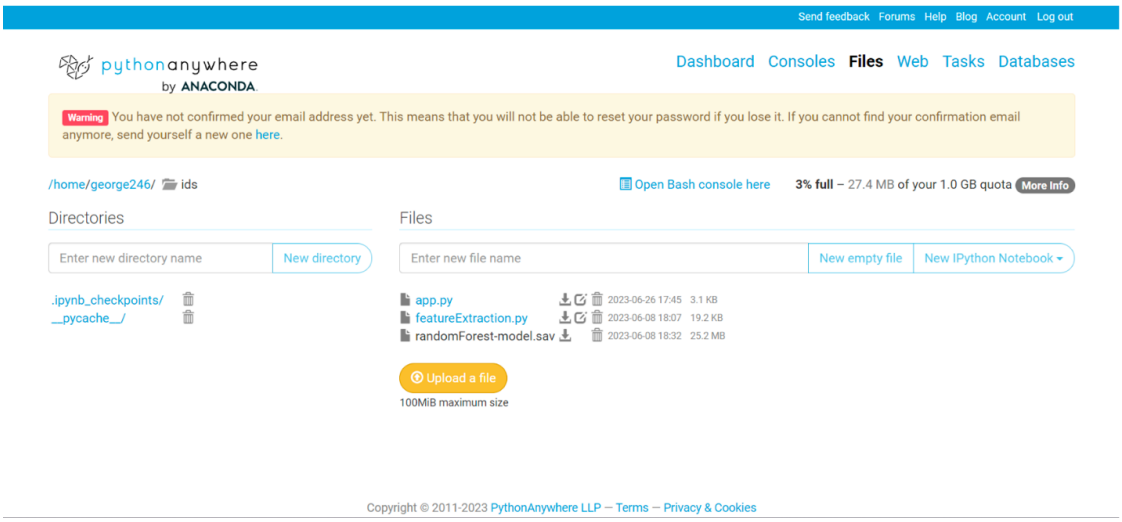
Modelo entrenado y guardado

El lenguaje Python se usó como tecnología cliente, la misma que se planificó en la arquitectura de la aplicación, por tal motivo el servicio de alojamiento web seleccionado para subir la API desarrollada fue PythonAnywhere, que del mismo modo es basado en el lenguaje Python y el framework Flask.

Los archivos importantes para el funcionamiento de la API son: “app.py” el cual abarca rutas y líneas de código necesarias para la predicción, “featureExtraction-DeAd.py” comprende el código usado para la extracción de características de un sitio web, el archivo “randomForest-DA-Legitime.sav” en donde, se almacena el modelo ML seleccionado en el Sprint 01 mismo que se entrenó durante el sprint actual.

Figura 12

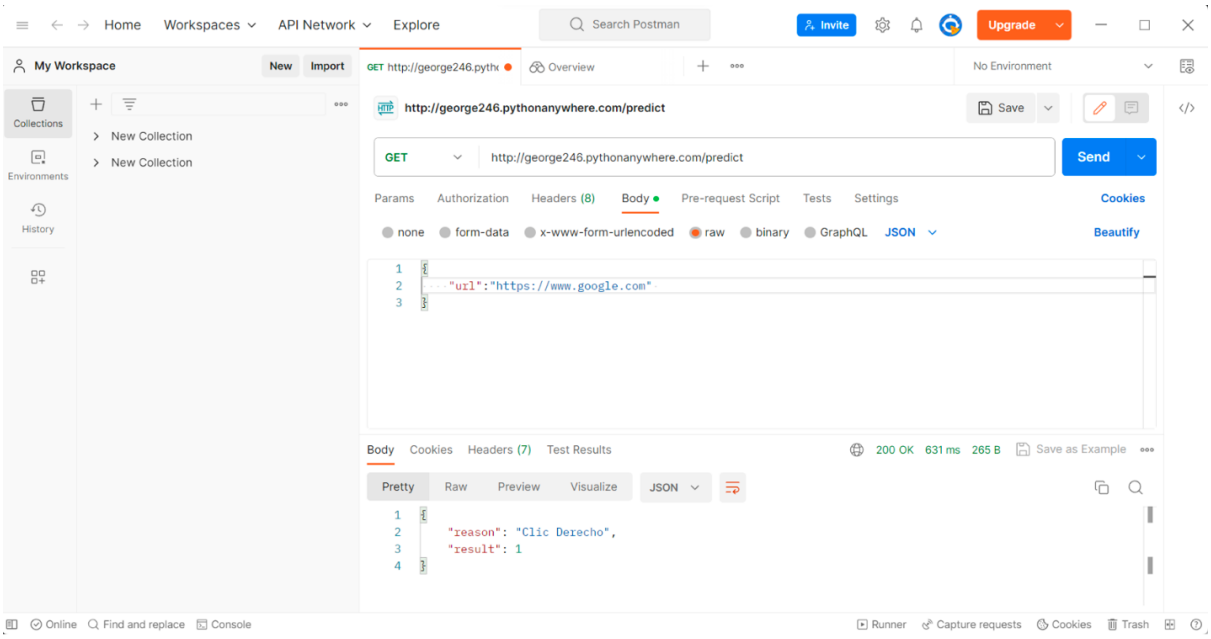
API subido al Servidor

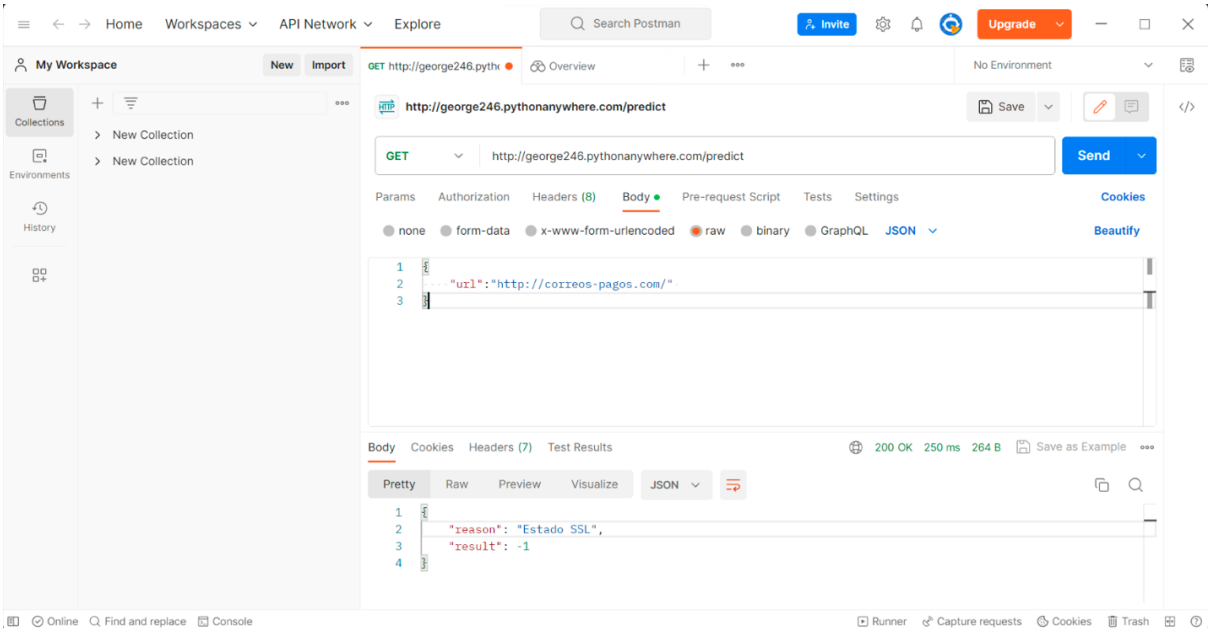


La figura 13 muestran las pruebas validadas de la API almacenadas en el servidor, las pruebas tuvieron como objetivo verificar la funcionalidad de la API, se enviaron dos URLs de prueba: la primera correspondía al sitio web de Google (https://www.google.com), y la API devuelve el valor 1, indicando que se trata de un sitio web legítimo. La segunda URL (https://correo-pagos.com/) era de un sitio web de phishing, y la API devuelve el valor -1, indicando que se trata de un sitio web no legítimo.

Figura 13

Predicción de sitios web utilizando la API desarrollada



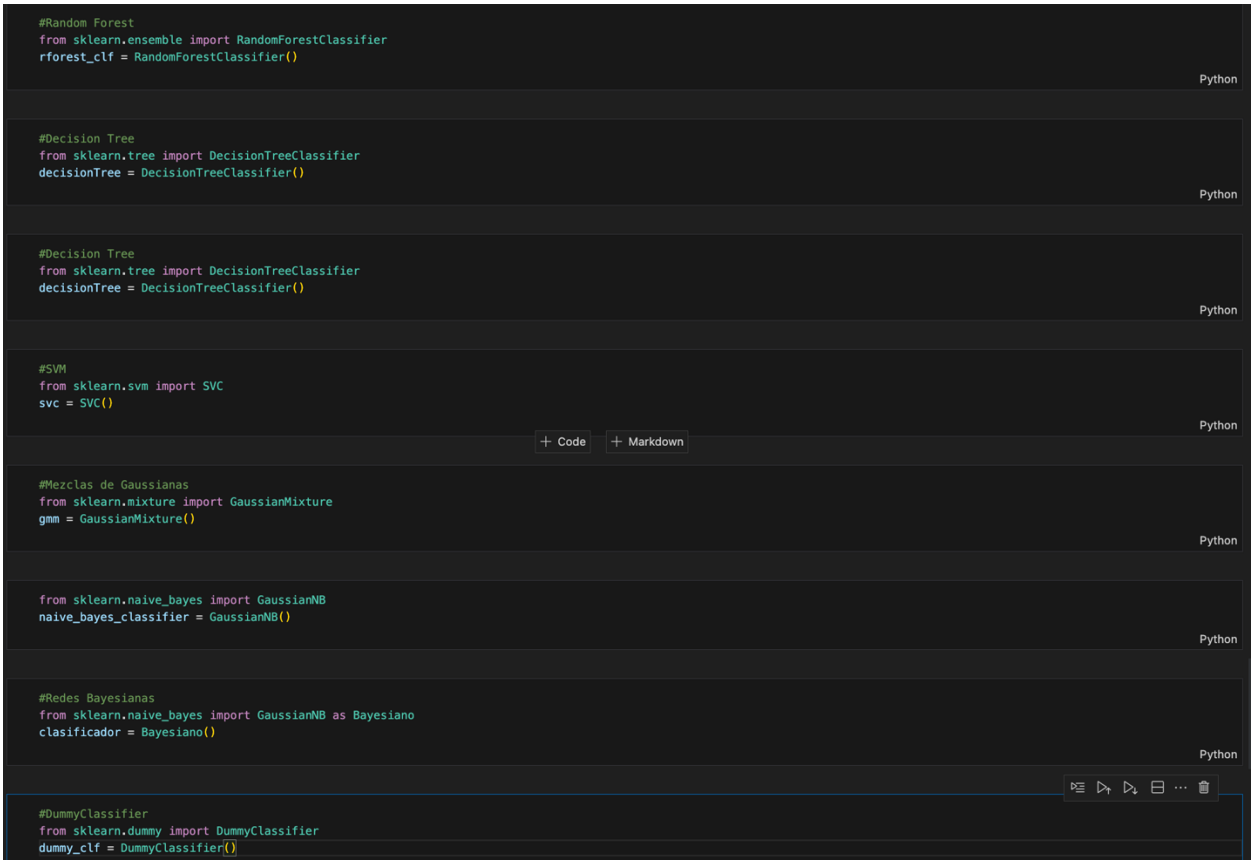


Ejecución

Esta sección se muestra la unión de todos los sprints, con sus respectivas implementaciones, a partir de los 8 algoritmos de Machine Learning se forma las predicciones de sitios con phishing, la figura 14 muestra cada uno de estos.

Figura 14

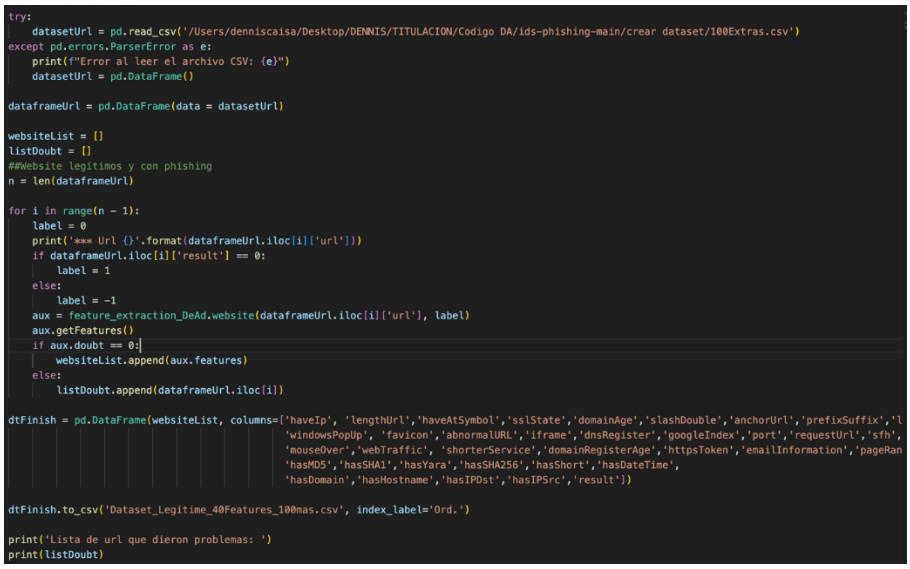
Modelos y/o Algoritmos de Machine Learning



A partir de la selección del modelo, se inicia con la creación del dataset, que es la unión de varias URLs almacenados en Bases de Datos mencionadas en el Anexo 3, la figura 15 contempla el código usado para la extracción de características tanto de las url como de los indicadores de compromiso indicadas en la figura 16.

Figura 15

Algoritmo de 40 características



```
try:
    datasetUrl = pd.read_csv('/Users/denniscaisa/Desktop/DENNIS/TITULACION/Codigo DA/ids-phishing-main/crear dataset/100Extras.csv')
except pd.errors.ParserError as e:
    print(f"Error al leer el archivo CSV: {e}")
    datasetUrl = pd.DataFrame()

dataframeUrl = pd.DataFrame(data = datasetUrl)

websiteList = []
listDoubt = []
##Website legitimos y con phishing
n = len(dataframeUrl)

for i in range(n - 1):
    label = 0
    print('*** Url {}'.format(dataframeUrl.iloc[i]['url']))
    if dataframeUrl.iloc[i]['result'] == 0:
        label = 1
    else:
        label = -1
    aux = feature_extraction_DeAd.website(dataframeUrl.iloc[i]['url'], label)
    aux.getFeatures()
    if aux.doubt == 0:
        websiteList.append(aux.features)
    else:
        listDoubt.append(dataframeUrl.iloc[i])

dtFinish = pd.DataFrame(websiteList, columns=['haveIp', 'lengthUrl','haveAtSymbol','sslState','domainAge','slashDouble','anchorUrl','prefixSuffix','l
                                              'windowsPopUp', 'favicon','abnormalURL','iframe','dnsRegister','googleIndex','port','requestUrl','sfh',
                                              'mouseOver','webTraffic', 'shorterService','domainRegisterAge','httpsToken','emailInformation','pageRan
                                              'hasMD5','hasSHA1','hasYara','hasSHA256','hasShort','hasDateTime',
                                              'hasDomain','hasHostname','hasIPDst','hasIPSrc','result'])

dtFinish.to_csv('Dataset_Legitime_40Features_100mas.csv', index_label='Ord.')

print('Lista de url que dieron problemas: ')
print(listDoubt)
```

Figura 16

URLs del Dataset

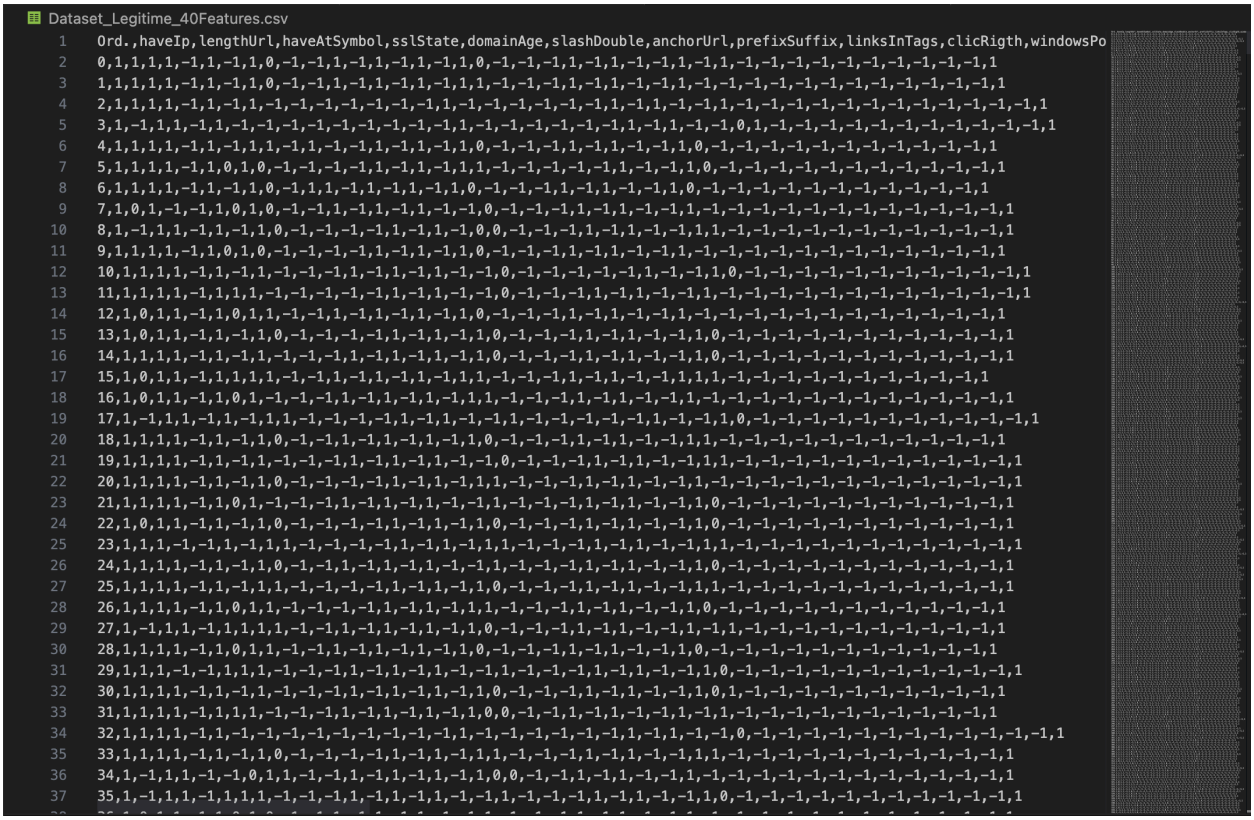
crear dataset > Data-Data-Complete.csv

```
rec_id,url,website,result,created_date
1,serviciosbys.com/paypal.cgi.bin.get-into.herf.secure.dispatch35463256rzr321654641dsf654321874/href/href/href/secure/cen
2,http://mail.printakid.com/www.online.americanexpress.com/index.html,1635709089405696.html,0,10/31/2021 16:53
3,http://ww25.thewhiskeydregs.com/wp-content/themes/widescreen/includes/temp/promocoessmiles/?84784787824HDJNDJDSJSHD//27
4,smilesvoegol.servebbs.org/voegol.php,1635746197718643.html,0,10/31/2021 18:05
5,http://myxxxcollection.com/v1/js/jih321/bpd.com.do/do/l.popular.php,1635707297969905.html,0,7/13/2021 15:44
6,"http://horizonsgallery.com/js/bin/ssl1/_id/www.paypal.com/fr/cgi-bin/webscr/cmd=_registration-run/login.php?cmd=_login
7,http://phlebolog.com.ua/libraries/joomla/results.php,1635702000019683.html,0,10/31/2021 17:02
8,perfectsolutionofall.net/wp-content/themes/twentyten/wiresource/,1635713773762468.html,0,10/31/2021 18:22
9,"http://lingshc.com/old_aol.1.3/?Login=&amp;Lis=10&amp;LigertID=1993745&amp;us=1",1623157557909209.html,0,10/31/2021 18
10,http://anonymeidentity.net/remax./remax.htm,1626170054396892.html,0,10/31/2021 16:71
11,http://dutchweb.gtphost.com/zimbra/exch/owa/uleth/index.html,1613563574363395.html,0,10/31/2021 16:89
12,http://www.avedeoiro.com/site/plugins/chase/,1635703471702368.html,0,2/17/2021 18:02
13,http://asladconcentration.com/paplkuk1/webscrcmd=_home-customer&nav=1/,1613526819401905.html,0,10/31/2021 18:06
14,http://www.regaranch.info/grafika/file/2012/atualizacao/www.itau.com.br/,1635749675039435.html,0,7/13/2021 15:45
15,https://optimistic-pessimism.com/aoluserupdatealert.info.htm,1613579920793711.html,0,7/19/2021 18:51
16,http://com.br/confirmar/,1635707743232302.html,0,10/31/2021 17:03
17,http://com.br/,1609105751340594.html,0,10/31/2021 18:32
18,http://myxxxcollection.com/v1/js/555klisdr/bpd.com.do/do/l.popular.php,1635712197850175.html,0,10/31/2021 18:37
19,http://paypal.com.cgi.bin.webscr.cmd.login.submit.dispatch.5885d80a13c03faee8dcbcd55a50598f04d34b4bf5tt1.mediareso.com
20,https://ebayisapidlld.altervista.org/,1635709325292373.html,0,7/19/2021 18:52
21,http://horizonsgallery.com/js/bin/ssl/_id/www.paypal.com/fr/cgi-bin/webscr/cmd=_registration-run/login.php?cmd=_login-
22,http://www.revitolcream.org/wp-content/plugins/all-in-one-seo-pack/rex/secure-code5/security/login.php,160895699724352
23,http://sontemeda.altervista.org/paypinoko/paypinoko/procesing.php,1607253905758674.html,0,10/31/2021 18:47
24,http://promusic.co/components/interbank.com/,1626991990435478.html,0,7/13/2021 15:47
25,http://remax.com.behinehsazerooyesh.com/remax/index.htm,1635700406057169.html,0,7/19/2021 18:53
26,http://www.aclaydance.com/ncpf.php,1613520940829138.html,0,10/31/2021 17:05
27,http://stthomasedu.ucoz.ua/microsoft.htm,1613531466407313.html,0,10/31/2021 18:52
28,http://paypal.com.us.cgi.bin.webscr.cmd.login.submit.bx3digitallp.com/Review/3a3e3c4dba5e5cc4407a5694adf1aa8e/,1626117
29,http://users11.jabry.com/blopp/Aolupdate.htm,1609088143357945.html,0,7/19/2021 18:52
30,http://www.hoskoyu.net/images/AolMail.htm,1613560311943282.html,0,10/31/2021 17:04
31,http://tad.ly/BUpQLx,1635700796340016.html,0,10/31/2021 18:42
32,http://cardpromocion2012.com.br/vaidevisa/asp/conta/participe/autentica/autentica.php,1609094067719942.html,0,7/13/202
33,http://portalsalinas.com.br/modules/mod_acepolls/spam.php,160894950078672.html,0,7/19/2021 18:52
34,http://trigononline.com/Support.php,1609482535586021.html,0,10/31/2021 17:04
35,http://nextgensmartphones.com//indexer/Trulia/index.htm,1623163531418 71.html,0,10/31/2021 18:42
36,http://lunatic-photography.opidum.de//wp-content/plugins/formulario.php,1613588910992154.html,0,10/31/2021 18:47
```

A partir de la figura 16 se ejecuta el algoritmo de la figura 15, obtener el dataset legítimo que refleja las 30 características de la url y las 10 características de los IOCs, tal como muestra la figura 17.

Figura 17

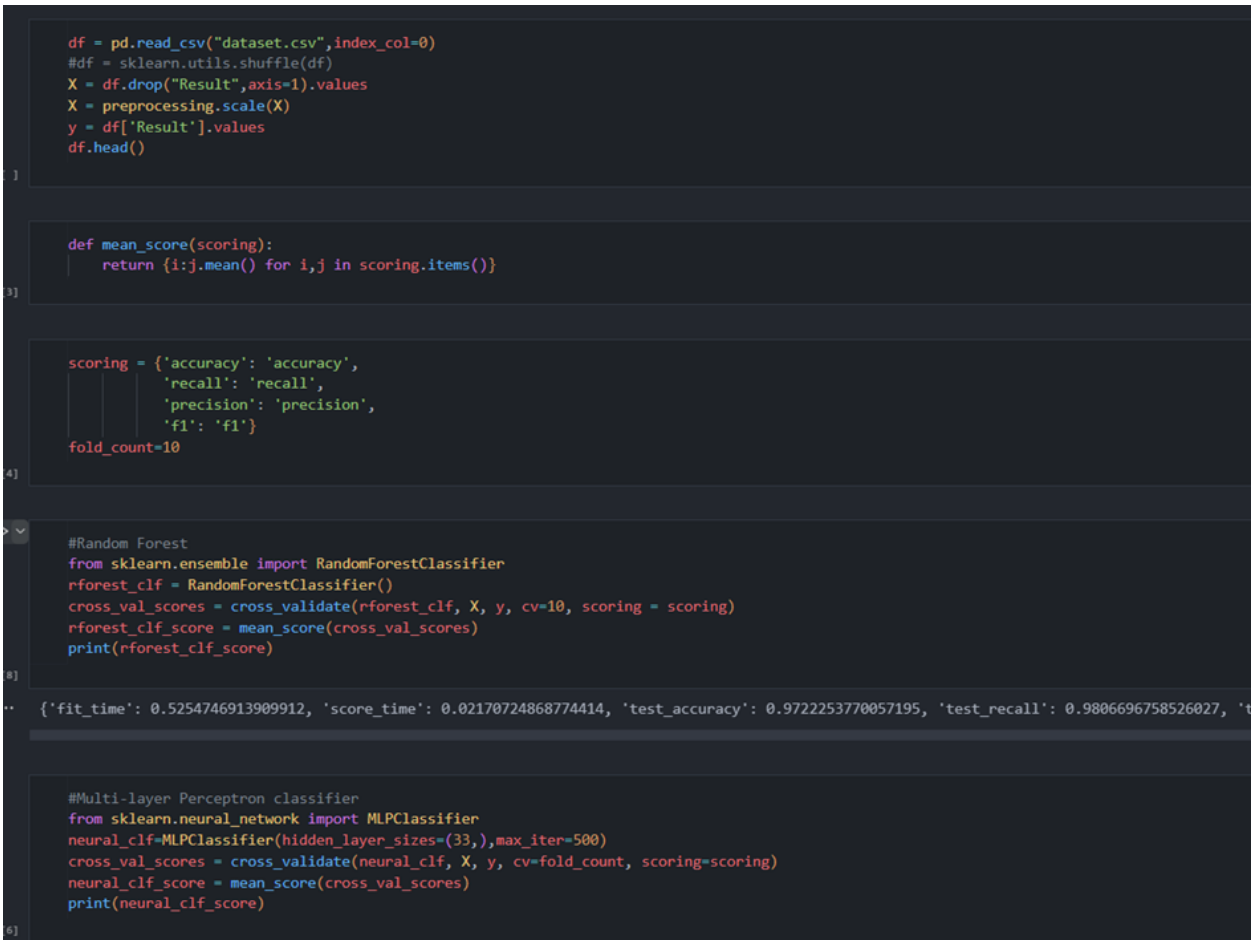
Dataset Creado



El Dataset se envía a todos los modelos y/o algoritmos seleccionados, de tal forma que proporcionen todas las métricas de evaluación que menciona la tabla 4, del mismo modo la figura 18 ilustra el código que se usó para las pruebas.

Figura 18

Código para pruebas de los Modelos



```
df = pd.read_csv("dataset.csv",index_col=0)
#df = sklearn.utils.shuffle(df)
X = df.drop("Result",axis=1).values
X = preprocessing.scale(X)
y = df['Result'].values
df.head()
```

```
def mean_score(scoring):
    return {i:j.mean() for i,j in scoring.items()}
```

```
scoring = {'accuracy': 'accuracy',
           'recall': 'recall',
           'precision': 'precision',
           'f1': 'f1'}
fold_count=10
```

```
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rforest_clf = RandomForestClassifier()
cross_val_scores = cross_validate(rforest_clf, X, y, cv=10, scoring = scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)
```

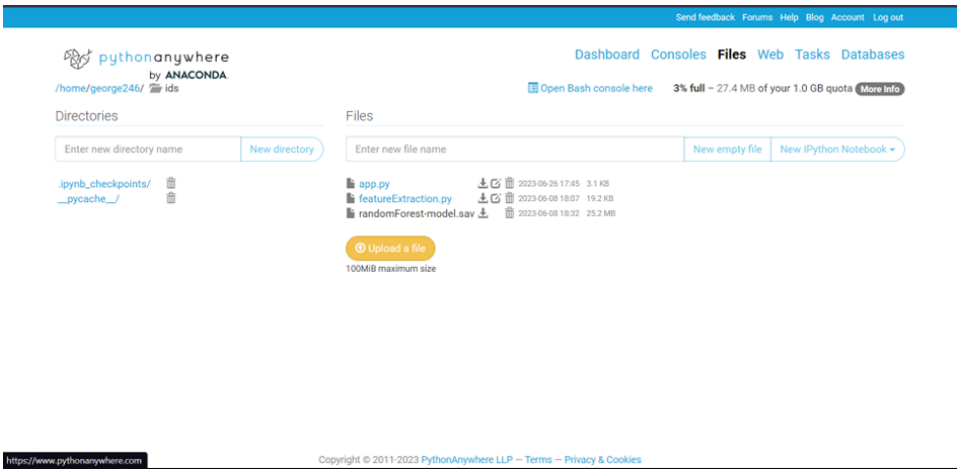
```
{'fit_time': 0.5254746913909912, 'score_time': 0.02170724868774414, 'test_accuracy': 0.9722253770057195, 'test_recall': 0.9806696758526027, 't
```

```
#Multi-layer Perceptron classifier
from sklearn.neural_network import MLPClassifier
neural_clf=MLPClassifier(hidden_layer_sizes=(33,),max_iter=500)
cross_val_scores = cross_validate(neural_clf, X, y, cv=fold_count, scoring=scoring)
neural_clf_score = mean_score(cross_val_scores)
print(neural_clf_score)
```

Con los resultados se escogió el mejor algoritmo que fue Random Forest, se entrenó el dataset para que todos los archivos se envíen a la plataforma PythonAnywhere, la figura 19 evidencia los archivos subidos al servidor.

Figura 19

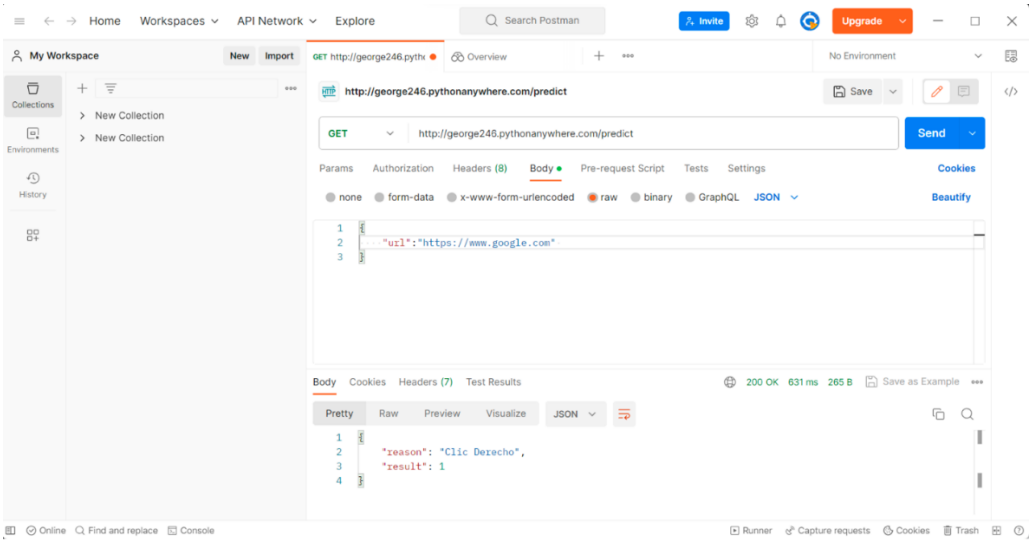
Servidor PythonAnywhere



El servidor es conectado a la plataforma Postman para las pruebas pertinentes, la figura 20 muestra la url enviada, debe devolver una respuesta de valor 1 o -1, los mismo que son legítimos y phishing respectivamente.

Figura 20

Pruebas en Postman



Finalizando, la figura 21 ilustra la extensión de Google Chrome, con la ejecución, mostrando las interfaces de respuesta, ya sea un sitio web legítimo o con phishing.

Figura 21

Interfaz de la Extensión de Google Chrome

Resumen del desarrollo del sistema de detección de sitios web con phishing

Sprint 01: Después de ejecutar el modelo de Machine Learning para la detección de sitios web con phishing o sitios web legítimos, utilizando un manejo de 8 modelos y/o algoritmos de Machine Learning, representados en la tabla 3, con la finalidad de dar garantía de la precisión alta y aceptable para detectar sitios web con phishing, se obtiene como resultado la selección del modelo Random Forest.

Sprint 02: Las 40 características que extraer a partir de una URL, que se probaron en diferentes escenarios y combinados para encontrar mejores valores del accuracy y garantizar una mejor precisión en la predicción de sitios web con phishing, un dataset fue creado con las características extaridas a partir de otro que contenía sólo las URLs de sitios web legítimos y con phishing.

Sprint 03: El modelo de Machine Learning seleccionado fue entrenado utilizando el dataset creado durante el Sprint anterior. Además, se desarrolló una API Rest y esta fue desplegada en un servidor en la nube.

Sprint 04: La extensión de Google Chrome fue desarrollada utilizando las tecnologías HTML, CSS y JavaScript.

Interfaz

En esta sección se presenta el diseño de la interfaz para crear la extensión Google Chrome propuesta, mediante la realización de mockups, o maquetas en español, que son bocetos que pretenden representar la Interfaz de Usuario (UI) probable para la aplicación a desarrollar, mostrando la apariencia general de la UI (Rivero, 2010).

El sistema de detección de phishing realizado con un enfoque del impacto que tiene el phishing en los entornos web y en los ataques que realizan los phishers denominados estafas, se pensó en “Phishing Impact”.

Mockups. Representación ágil de los modelos para representar los requisitos que son entendidos dentro del proceso de los clientes y desarrolladores (Colastra, 2017). Se presenta a continuación de los mockups que se pensaron para la aplicación Phishing Impact.

Mockup 1: Se presenta el Mockup en la figura 14 en la que se analiza el sitio web, para su funcionamiento es necesario dar clic en la extensión para que analice el sitio y hacer la petición al servidor web.

Figura 22

Mockup analizando sitio web

Mockup 2: Se presenta el Mockup en la figura 15 en la que se analiza que el sitio web es legítimo.

Figura 23

Mockup sitio web legítimo

Mockup 3: Se presenta el Mockups en la figura 16 en la que se analiza el sitio web contenga phishing.

Figura 24

Mockup sitio web phishing

Capítulo IV

Validación del sistema

En el presente capitulo se realizaron las pruebas para la validación de la extensión de Google Chrome (Phishing Impact). Con este propósito, se utilizaron dos herramientas, la primera NexPhisher y la segunda Zphisher, diseñadas para el sistema operativo Kali Linux, las cuales permiten crear sitios web con Phishing mediante comandos en el terminal, Estas aplicaciones recopilan la información proporcionada por el usuario y, al final, muestra tanto los datos ingresados como la dirección IP asociada (Alzas Hernandez, 2023; Alshabob, Ahmed Jabeur, Y Alserhani, 2022). Como se muestra en la figura 25 donde se evidencia los ataques disponibles en NexPhisher y en la figura 26 se evidencia los sitios con Phishing que aplicación detecta y los ataques disponibles en Zphisher.

Figura 25

Ataques disponibles NexPhisher

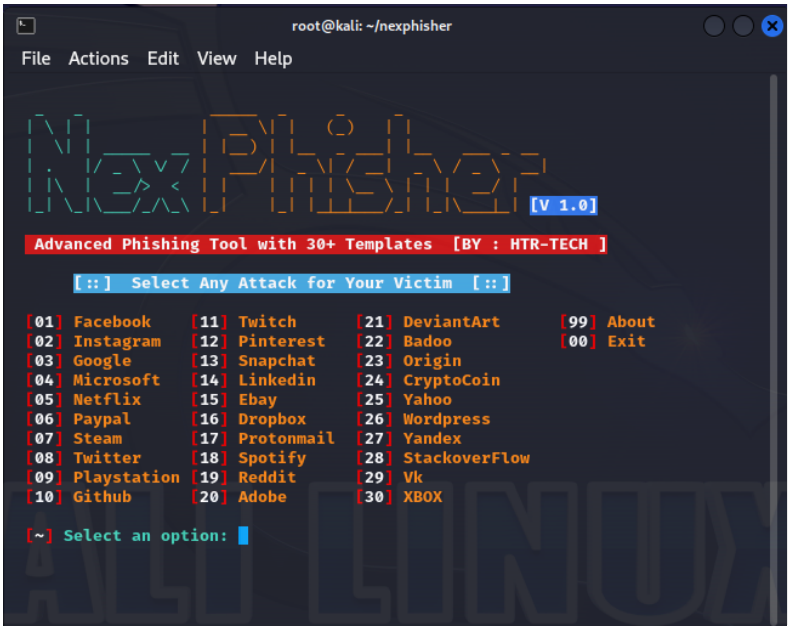
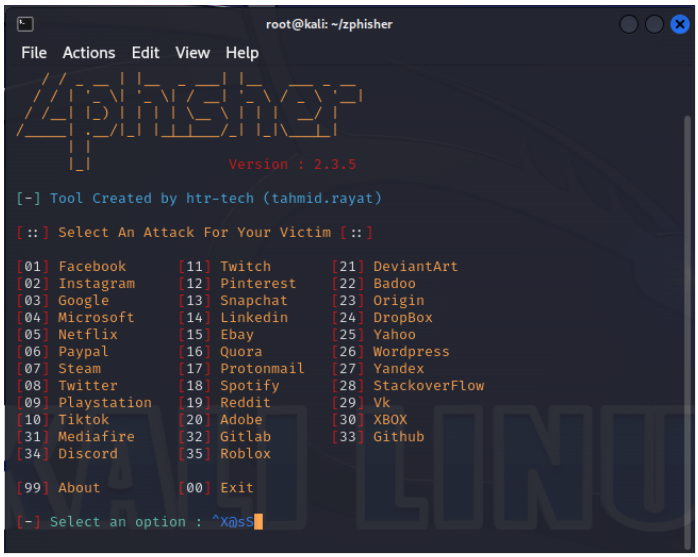


Figura 26

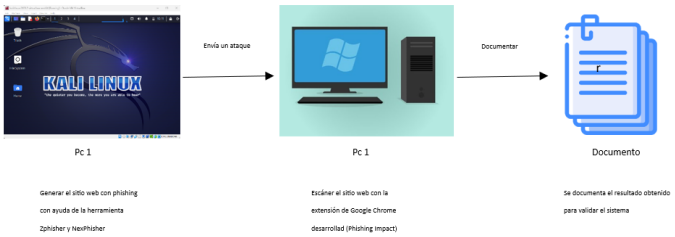
Ataques disponibles ZPhisher



Para la ejecución de las pruebas se requirió de dos computadores, el primer computador se instaló la máquina virtual con el sistema operativo Kali Linux encargado de realizar ataques y sitios web phishing, mientras en el segundo computador se inspecciona con la extensión (Phishing Impact) si el sitio web generado contiene o no phishing. Se presenta la ejecución de las pruebas en la figura 27 mencionadas con anterioridad.

Figura 27

Proceso de ejecución de pruebas



Definición y aplicación de métricas de evaluación

Aplicación de las métricas de evaluación

Para las métricas de evaluación es esencial realizar pruebas con la extensión Phishing Impact con los diferentes modelos de entrenamiento, para el caso se utilizó el primer modelo de Machine Learning entrenado, con las herramientas NexPhisher y Zphisher, resultados que se pueden visualizar en la tabla 22, con las pruebas realizadas desde Phishing Impact con el primer modelo de Machine learning. Esta tabla nos presenta algunos atributos importantes como: nombre del sitio web, nombre de las secciones específicas del sitio web, resultado esperado y la predicción del modelo.

Dado que esta tabla incluye pruebas de sistemas de detección de ataques de phishing, se seleccionó sitios web con características similares en ambas herramientas, lo que dio como resultado 43 sitios web que son legítimos, pero pertenecen a la misma familia de sitios web que los 43 sitios web de phishing generados principalmente por la herramienta Zphisher. Los sitios web seleccionados se muestran en la figura 18 y suman un total de 34, y están disponibles en la herramienta Zphisher, como se muestra en la columna "Sitio web" de la tabla 22, y también incluyen los sitios web mostrados en la figura 17, que suman un total de 72 sitios web, pero para comprender mejor los sitios web introducidos, se categoriza en función de la plataforma o subsistema que impulsa su funcionalidad, como se muestra en la tabla 21.

Tabla 21

Sitios web seleccionados

Categoría	Nombre
Redes Sociales	Facebook, Instagram, X, TikTok, Pinterest, Snapchat, LinkedIn, Quora, DevianART, Badoo, Vk.
Plataforma's SSO (Single Sign-On)	Google, Adobe, Microsoft, Yahoo.
Plataforma de streaming por suscripción	Netflix.
Plataforma de pagos en línea	Paypal.
Plataforma de distribución digital de videojuego	Steam, PlayStation, Origin, XBOX.
Plataforma de streaming de video	Twitch.
Portal web para vender y/o subastar on line	eBay
Correo electrónico	Protonmail, Yandex.
Plataforma de streaming de música	Spotify.
Plataforma social	Reddit, Discord.
Servicio de alojamiento de archivos	DropBox, MediaFire.
Sistema de gestión de sitios web	WordPress.
Blog	StackoverFlow
Gestores de versiona miento	Github, Gitlab

La mayoría de los sitios web prestan atención a la sección "Página de inicio de sesión" porque es la primera interfaz de todo el sistema web donde el usuario ve e introduce sus datos personales y sensibles. Los phishers intentan engañar a la víctima para que introduzca información sensible del usuario, como el nombre de usuario, la contraseña y los datos de la tarjeta de crédito (Sonowal, G. 2022) con el fin de robar la información y cometer fraude y teniendo en cuenta los 74 sitios que se tenía se los distribuyó entre Zphisher y NexPhisher, teniendo así más de una sección del sitio que fue seleccionada para ser completada, tanto las secciones que requerían la participación del usuario como las creadas por Zphisher y NexPhisher.

Para ayudarle a entender cómo se probó el sistema, en esta sección se describe brevemente el proceso de prueba de un sitio (Instagram) en el se prubea fragmentos obtenidos utilizando las herramientas Zphisher y NexPhiser. Utilizando Zphisher, primero se captura fragmentos del sitio de phishing y luego se captura el mismo contenido del sitio legítimo con una URL. A continuación, se verificó cada URL utilizando la extensión Phishing Impact desarrollada a partir del modelo original de aprendizaje automático descrito en la Sección 3: Implementación del sistema.

El resto de los sitios y secciones se probaron de forma similar. Por último, la tabla 21 y la tabla 22 muestran que el modelo sólo puede detectar sitios de phishing porque el sistema trata los sitios legítimos como sitios de phishing.

Tabla 22

Resultados pruebas de Phishing Impact con el primer modelo de Machine Learning

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
Facebook	1	Traditional Login Page	Phishing	Phishing	Legitimo	Legitimo
Instagram	2	Traditional Login Page	Phishing	Legitimo	Legitimo	Legitimo
X	3	X Login Page	Phishing	Legitimo	Legitimo	Legitimo
TikTok	4	TikTok Login Page	Phishing	Legitimo	Legitimo	Legitimo
Pinterest	5	Pinterest Login Page	Phishing	Legitimo	Legitimo	Phishing
Snapchat	6	Snapchat Login Page	Phishing	Legitimo	Legitimo	Phishing
LinkedIn	7	Linkedin Login Page	Phishing	Phishing	Legitimo	Legitimo
Quora	8	Quora Login Page	Phishing	Phishing	Legitimo	Legitimo
DeviantArt	9	Davian Art Login Page	Phishing	Phishing	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
Badoo	10	Badoo Login Page	Phishing	Legitimo	Legitimo	Phishing
Vk	11	Vk Login Page	Phishing	Legitimo	Legitimo	Phishing
Netflix	12	Netflix Login Page	Phishing	Phishing	Legitimo	Legitimo
Paypal	13	Paypal Login Page	Phishing	Legitimo	Legitimo	Legitimo
Steam	14	Steam Login Page	Phishing	Legitimo	Legitimo	Legitimo
PlayStation	15	PlayStation Login Page	Phishing	Legitimo	Legitimo	Phishing
Origin	16	Origin Login Page	Phishing	Phishing	Legitimo	Legitimo
XBOX	17	XBOX Login Page	Phishing	Phishing	Legitimo	Legitimo
Twich	18	Twich Login Page	Phishing	Legitimo	Legitimo	Legitimo
eBay	19	eBay Login Page	Phishing	Legitimo	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
Protonmail	20	Protonmail Login Page	Phishing	Legitimo	Legitimo	Legitimo
Yandex	21	Yandex Login Page	Phishing	Phishing	Legitimo	Legitimo
Spotify	22	Spotify Login Page	Phishing	Legitimo	Legitimo	Legitimo
Reddit	23	Reddit Login Page	Phishing	Legitimo	Legitimo	Legitimo
Discord	24	Discord Login Page	Phishing	Pishing	Legitimo	Legitimo
DropBox	25	DropBox Login Page	Phishing	Legitimo	Legitimo	Legitimo
MediaFire	26	MediaFire Login Page	Phishing	Legitimo	Legitimo	Legitimo
WordPress	27	WordPress Login Page	Phishing	Legitimo	Legitimo	Phishing
StackoverFlow	28	Stackoverflow login page	Phishing	Legitimo	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
GitHub	29	GitHub Login Page	Phishing	Phishing	Legitimo	Legitimo
GitLab	30	GitLab Login Page	Phishing	Phishing	Legitimo	Phishing
Google	31	Gmail New Login Page	Phishing	Phishing	Legitimo	Legitimo
Adobe	32	Adobe Login Page	Phishing	Legitimo	Legitimo	Legitimo
Microsoft	33	Outlook Login Page	Phishing	Phishing	Legitimo	Legitimo
Yahoo	34	Yahoo Login Page	Phishing	Phishing	Legitimo	Legitimo
SITIOS WEB BIEN CLASIFICADOS				13		33
SITIOS WEB MAL CLASIFICADOS				21		1

Con los resultados que se obtienen en a la tabla 22, es posible realizar una matriz de confusión que se corresponda con los datos, tal como se muestra en la Matriz de confusión de la tabla 23 correspondiente al primer modelo de Machine Learning.

Tabla 23

Matriz de confusión del primer modelo de ML

	POSITIVO	**NEGATIVO**
POSITIVO	13	33
NEGATIVO	21	1

Ahora se aplica las fórmulas para las métricas Recall, Precision, Accuracy y F1 de la tabla 4 a los resultados mostrados en la tabla 24. La tabla muestra que, para la métrica de Accuracy de la detección de phishing es del 84%, lo que significa que más del 75% de los sitios de todo el conjunto de datos se detectaron correctamente.

En cuanto a la Precision, la exactitud de la detección de phishing es del 82%. Por otro lado, la métrica de Recall es del 85%, lo que significa que la mayoría de los sitios de phishing se identificaron correctamente en comparación con los sitios de phishing reales. La métrica F1 es del 83%, lo que indica que el modelo funciona mejor en la predicción de categorías benignas.

Tabla 24

Métricas de evaluación calculadas

MÉTRICA DE EVALUACIÓN	**RESULTADO**
Recall	84%
Precision	82%
Accuracy	85%
F1	83%

Tabla 25

Resultados pruebas de Phishing Impact con el primer modelo de Machine Learning empleando NexPhisher

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
Facebook	1	Traditional Login Page	Phishing	Phishing	Legitimo	Legitimo
Instagram	2	Traditional Login Page	Phishing	Legitimo	Legitimo	Legitimo
X	3	X Login Page	Phishing	Legitimo	Legitimo	Legitimo
TikTok	4	TikTok Login Page	Phishing	Legitimo	Legitimo	Legitimo
Pinterest	5	Pinterest Login Page	Phishing	Legitimo	Legitimo	Phishing
Snapchat	6	Snapchat Login Page	Phishing	Legitimo	Legitimo	Phishing
LinkedIn	7	Linkedin Login Page	Phishing	Phishing	Legitimo	Legitimo
Quora	8	Quora Login Page	Phishing	Phishing	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
DeviantArt	9	Davian Art Login Page	Phishing	Phishing	Legitimo	Legitimo
Badoo	10	Badoo Login Page	Phishing	Legitimo	Legitimo	Phishing
Vk	11	Vk Login Page	Phishing	Legitimo	Legitimo	Legitimo
Netflix	12	Netflix Login Page	Phishing	Phishing	Legitimo	Legitimo
Paypal	13	Paypal Login Page	Phishing	Legitimo	Legitimo	Legitimo
Steam	14	Steam Login Page	Phishing	Legitimo	Legitimo	Legitimo
PlayStation	15	PlayStation Login Page	Phishing	Legitimo	Legitimo	Legitimo
Origin	16	Origin Login Page	Phishing	Phishing	Legitimo	Legitimo
XBOX	17	XBOX Login Page	Phishing	Phishing	Legitimo	Legitimo
Twich	18	Twich Login Page	Phishing	Legitimo	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
eBay	19	eBay Login Page	Phishing	Legitimo	Legitimo	Legitimo
Protonmail	20	Protonmail Login Page	Phishing	Legitimo	Legitimo	Legitimo
Yandex	21	Yandex Login Page	Phishing	Phishing	Legitimo	Legitimo
Spotify	22	Spotify Login Page	Phishing	Legitimo	Legitimo	Legitimo
Reddit	23	Reddit Login Page	Phishing	Legitimo	Legitimo	Legitimo
Discord	24	Discord Login Page	Phishing	Legitimo	Legitimo	Legitimo
DropBox	25	DropBox Login Page	Phishing	Legitimo	Legitimo	Legitimo
MediaFire	26	MediaFire Login Page	Phishing	Legitimo	Legitimo	Legitimo
WordPress	27	WordPress Login Page	Phishing	Legitimo	Legitimo	Legitimo

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIO WEB LEGITIMO		PRUEBAS SITIO WEB LEGÍTIMO	
			RESULTADOS ESPERADOS	PREDICCIÓN	RESULTADOS ESPERADOS	PREDICCIÓN
StackoverFlow	28	Stackoverflow login page	Phishing	Legitimo	Legitimo	Legitimo
GitHub	29	GitHub Login Page	Phishing	Phishing	Legitimo	Legitimo
GitLab	30	GitLab Login Page	Phishing	Phishing	Legitimo	Legitimo
Google	31	Gmail New Login Page	Phishing	Phishing	Legitimo	Legitimo
Adobe	32	Adobe Login Page	Phishing	Legitimo	Legitimo	Legitimo
Microsoft	33	Outlook Login Page	Phishing	Phishing	Legitimo	Legitimo
Yahoo	34	Yahoo Login Page	Phishing	Phishing	Legitimo	Legitimo
SITIOS WEB BIEN CLASIFICADOS				13		33
SITIOS WEB MAL CLASIFICADOS				21		1

Con los resultados que arrojo la tabla 25 se puede realizar la tabla 26.

Tabla 26

Matriz de confusión con el software NexPhisher

	POSITIVO	**NEGATIVO**
POSITIVO	13	33
NEGATIVO	21	1

Ahora se aplica las fórmulas para las métricas Recall, Precision, Accuracy y F1 de la tabla 4 a los resultados mostrados en la tabla 27. La tabla muestra que, para la métrica de Accuracy de la detección de phishing es del 86%, lo que significa que más del 75% de los sitios de todo el conjunto de datos se detectaron correctamente. En cuanto a la Precision, la exactitud de la detección de phishing es del 92%. Por otro lado, la métrica de Recall es del 82%, lo que significa que la mayoría de los sitios de phishing se identificaron correctamente en comparación con los sitios de phishing reales. La métrica F1 es del 86%, lo que indica que el modelo funciona mejor en la predicción de categorías benignas.

Tabla 27

Métricas de evaluación calculadas

MÉTRICA DE EVALUACIÓN	**RESULTADO**
Recall	86%
Precision	92%
Accuracy	82%
Fi	86%

Análisis de resultados

Los resultados obtenidos se validaron utilizando el modelo de aprendizaje automático implementado, se realizaron diez pruebas de sitio simulado/real, cada una de ellas con 86 sitios de prueba, y el sitio simulado (entorno controlado) contenía 43 sitios creados con la herramienta Zphisher contando con otros 43 sitios creados con la herramienta NexPhisher.

Como se muestra en la columna "SECCIÓN DEL SITIO WEB" de la tabla 22 y la tabla 25, el sitio simulado (entorno controlado) contenía 43 sitios web por tabla creados con las herramientas Zphisher y NexPhisher, y el sitio real (entorno no controlado) contenía los mismos 43 sitios web generados desde los Url nativas de casa sitio, siendo el caso de ser legítimos (sitios web originales). El motivo de las 10 pruebas era garantizar que los resultados predichos se mantuvieran estables y/o tuvieran una pequeña variación.

Se realiza simulaciones y/o pruebas de campo reales utilizando modelos de aprendizaje automático y la correspondiente herramienta de Zphisher. A partir de los resultados de las pruebas, se calcula los valores medios de las métricas de evaluación, obteniendo los siguientes resultados: 84% de precisión, 82% de recall, 85% de recall y 83% de F1. Cabe destacar que mediante el cálculo de estas métricas de evaluación se evalua la precisión del modelo en la predicción del rendimiento del sitio, en otras palabras, estos valores determinan la aceptabilidad de los resultados predichos. Los resultados de las métricas estimadas obtenidas en la fase de entrenamiento son los siguientes. Según estos datos, no hay desviaciones significativas excepto recall y F1, lo que indica la fiabilidad del modelo de aprendizaje automático, que proporciona buenos valores de precisión y recall incluso en la fase de validación (simulación/sitio real).

A continuación, se aplica el mismo procedimiento a la herramienta NexPhisher, utilizando el mismo modelo de aprendizaje automático, y se promedió los valores de cada una de las métricas

evaluadas, obteniendo los siguientes resultados: 86% de precisión, 92% de recall, 82% de recuperación y 86% de F1. Por otro lado, como se ha mencionado anteriormente, las mediciones para la fase de entrenamiento fueron las siguientes: 84% de precisión, 85,72% de recuperación, 85,72% de exactitud, 82% de recuperación y 86% de F1.

Evaluación de las métricas para garantizar que los resultados no sólo se encuentran dentro de un rango aceptable, sino que permanecen estables. Se verificó que los resultados estaban dentro de un rango aceptable y permanecían estables, lo que confirma la eficacia del modelo de aprendizaje automático implementado, para identificar mejor los resultados obtenidos se realizó uno comparación entre las herramientas Zphisher y NexPhisher visibles en la tabla 28.

Tabla 28

Métricas de evaluación calculadas

Etapa de entrenamiento				**Herramienta**	**Campo simulado/real**			
Accuracy	Precision	Recall	F1		Accuracy	Precision	Recall	F1
84%	85%	96%	90%	Zphisher	84%	82%	85%	83%
				NexPhisher	86%	92%	82%	86%

A partir de estos resultados, se puede concluir que existe una diferencia significativa entre los valores obtenidos en las pruebas de simulación/mundo real y los obtenidos en el entrenamiento, lo que confirma las conclusiones extraídas en la sección anterior. Los valores de las métricas de evaluación de NexPhisher muestran una mejora significativa en comparación con el uso correspondiente de la herramienta Zphisher, especialmente la métrica de "precisión" (86%), que se encarga de determinar el porcentaje de sitios legítimos en todo el conjunto de datos de prueba y la proporción de sitios web de phishing categorizados positivamente.

La métrica también está relacionada con los valores obtenidos durante la fase de entrenamiento consistente con una de precisión de (92%), que se utiliza para determinar el porcentaje de sitios de phishing correctamente categorizados, también ha mejorado. La métrica Recall (82%), en cambio, ha disminuido, ya que determina el porcentaje de sitios de phishing correctamente categorizados como porcentaje de los sitios etiquetados como sitios de phishing en los datos de prueba seleccionados, independientemente de los datos etiquetados como sitios legítimos.

En cuanto a la métrica F1 (86%), se observó que aumentó en comparación con la herramienta Zphisher, pero disminuyó en comparación con los valores de entrenamiento, debido a que combina las métricas de precisión y recuerdo, y por lo tanto depende de ambos valores, lo que indica una diferencia en el rendimiento del clasificador, sugiriendo que los algoritmos de clasificación son mejores a la hora de predecir puntuaciones positivas. A diferencia de otros estudios, este se probó en un campo simulado/real (utilizando Zphisher y NexPhisher) y el valor de precisión más alto obtenido fue del 86% y el más bajo del 84%, que se encuentra dentro del rango del 83% encontrado en la literatura (Sonowal, 2022; di Laurea, n.d.).

En base a los resultados obtenidos, se puede concluir que un Sistema de detección de Intrusiones (IDS) contra ataques de phishing utilizando la extensión de Google Chrome utiliza el modelo de "Indicadores de Error" y/o algoritmos de aprendizaje automático, los cuales, según la revisión bibliográfica mencionada anteriormente, muestran resultados dentro de un rango de predicción aceptable.

Conclusiones

En el transcurso del desarrollo del trabajo de investigación se llegaron a varias conclusiones que se presentan a continuación:

- Se llevó a cabo una revisión sistemática de los indicadores de compromiso para recopilar información adecuada sobre los elementos específicos que pueden ayudar a detectar el acceso no autorizado a páginas web, especialmente a los que están soportados por motores de búsqueda, centrándose en el uso del navegador Google Chrome.
- El IDS titulado, "Phishing Impact" se creó para combatir los ataques de phishing. Se entrenó utilizando un conjunto de datos que contenía 13.192 URL de sitios web, divididos en 3.987 sitios web de phishing y 9.205 sitios web legítimos. En este proceso se extrajeron 30 características asociadas a las URL y 10 características de Indicadores de Compromiso (IOC). La elección de los modelos de Machine Learning se basó en una revisión sistemática, y se seleccionaron 4 modelos. Tras evaluar los resultados utilizando varias métricas, se comprobó que el algoritmo Random Forest era el más eficaz en términos de exactitud, recall, precision y F1, que son cruciales para detectar y prevenir sitios web de phishing. Este logro cumplió el segundo objetivo específico: el desarrollo de una extensión de Google Chrome que utiliza técnicas de ML para mejorar la seguridad de los sitios web y las medidas de detección.
- Se puso en marcha un sistema para detectar los intentos de phishing mediante la creación de una extensión de Google Chrome que aprovecha métodos de ML para detectar si un sitio web es auténtico o una amenaza de phishing.
- La aplicación del enfoque Scrum desempeñó un papel importante en la consecución de los objetivos del proyecto, ya que emplea un proceso de desarrollo flexible para la planificación de tareas y mejoras continuas durante los Sprints. Además, mejoró enormemente la experiencia de

trabajo en equipo y la comunicación entre los miembros del equipo, garantizando una progresión fluida del trabajo de investigación.

- Para evaluar la eficacia del IDS (Phishing Impact), se utilizaron dos herramientas de validación: Zphisher y NexPhisher. Estas herramientas se emplearon para crear sitios web de phishing simulados con fines de prueba. La evaluación del rendimiento del modelo se llevó a cabo utilizando métricas específicas, con NexPhisher mostrando resultados superiores en términos de accuracy, recall, precisión y F1. Estas métricas superaron a las obtenidas por la otra herramienta y el entorno de entrenamiento.
- En conclusión, la recién creada extensión de Google Chrome (Phishing Impact) está lista para su despliegue en un entorno práctico.

Recomendaciones

A continuación, se presentan recomendaciones para resaltar las dificultades identificadas y para mejorar futuros trabajos de investigación:

- Al realizar la revisión sistemática, es aconsejable tener en cuenta las palabras clave relevantes asociadas al tema. Este enfoque garantiza la recuperación de resultados valiosos y artículos pertinentes para iniciar el proceso de investigación.
- Al momento de crear el conjunto de datos, se recomienda abastecerse principalmente de información actualizada procedente de sitios web fiables. También es esencial asegurarse de que las URLs sean accesibles. Un aspecto crucial es la inclusión de indicadores de compromiso, que desempeñaron un papel importante en la investigación.
- Se sugiere utilizar la metodología ágil Scrum, ya que proporciona la capacidad de organizar eficazmente las tareas, mejora la comunicación entre los miembros del equipo y contribuye a mejorar el trabajo grupal.
- En todo proyecto de desarrollo de software, es crucial crear un área designada para almacenar los cambios, a menudo denominada repositorio. En este caso concreto, se creó un espacio con el objetivo principal de gestionar el control de versiones, fomentar el trabajo en equipo y promover la coordinación entre los participantes del equipo.
- Se sugiere realizar pruebas unitarias para garantizar una funcionalidad adecuada y evitar problemas durante la fase de despliegue en producción.
- En el proceso de prueba de un modelo de ML, es aconsejable realizar distintas pruebas. Esto ayuda a capturar el rango de variabilidad en las métricas de evaluación, permitiéndonos identificar tanto los valores más altos como los más bajos.

Bibliografía

Abraham, D., Y Raj, N. S. (2014). Approximate string matching algorithm for phishing detection. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2285–2290. https://doi.org/10.1109/ICACCI.2014.6968578

Adil, M., Khan, R., Y Nawaz Ul Ghani, M. A. (2020). Preventive Techniques of Phishing Attacks in Networks. 2020 3rd International Conference on Advancements in Computational Sciences (ICACS), 1–8. https://doi.org/10.1109/ICACS47775.2020.9055943

Al-Khamis, A. K., Y Khalafallah, A. A. (2015). Secure Internet on Google Chrome: Client side anti-tabnabbing extension. 2015 First International Conference on Anti-Cybercrime (ICACC), 1–4. https://doi.org/10.1109/Anti-Cybercrime.2015.7351942

Alhazmi, O. H., Malaiya, Y. K., Y Ray, I. (2007). Measuring, analyzing, and predicting security vulnerabilities in software systems. Computers Y Security, 26(3), 219–228. https://doi.org/10.1016/j.cose.2006.10.002

Alhazmi, O., Malaiya, Y., Y Ray, I. (2005). Security Vulnerabilities in Software Systems: A Quantitative Perspective. In S. Jajodia Y D. Wijesekera (Eds.), Data and Applications Security XIX (Vol. 3654, pp. 281–294). Springer Berlin Heidelberg. https://doi.org/10.1007/11535706_21

Alshabob, O., Ahmed Jabeur, R., Y Alserhani, F. M. (2022). Digital Forensic Analytics In Social Media. In Digital Forensic Analytics In Social Media (p. 10). Vol.100. No 19.

Alzas Hernandez, J. (2023, 07 09). Universitat Oberta de Catalunya. Retrieved from Universitat Oberta de Catalunya: https://openaccess.uoc.edu/handle/10609/148147

Apps, S. (2022, enero 18). Cyberattacks 2021: Statistics from the Last Year. Spanning. Retrieved from https://spanning.com/blog/cyberattacks-2021-phishing-ransomware-data-breach-statistics/

AWS Toolkit for Visual Studio Code. (n.d.). Retrieved July 11, 2023, from https://aws.amazon.com/es/visualstudiocode/.

Axelsson Stefan, (2002, octubre). Sistemas de detección de intrusos: un estudio y taxonomía. https://www.researchgate.net/publication/2597023_Intrusion_Detection_Systems_A_Survey_and_Taxonomy

Aziz M, V., Wijaya, R., Prihatmanto A, S., Y Henriyan , D. (2013). HASH MD5 function implementation at 8-bit microcontroller. 2013 Joint International Conference on Rural Information Y Communication Technology and Electric-Vehicle Technology (RICT Y ICeV-T). doi:https://doi.org/10.1109/rICT-ICeVT.2013.6741530

Aziz, M. V. G., Wijaya, R., Prihatmanto, A. S., Y Henriyan, D. (2013). HASH MD5 function implementation at 8-bit microcontroller. 2013 Joint International Conference on Rural Information Y Communication Technology and Electric-Vehicle Technology (RICT Y ICeV-T), 1–5. https://doi.org/10.1109/rICT-ICeVT.2013.6741530

Bass, L., Clements, P., Y Kazman, R (2021). software Architecture in Pactice, 4th Edition.

Becerril Domínguez, O. A. (2018). Resulta extremadamente valioso al detectar ataques que aún no se han identificado. http://tesis.ipn.mx/handle/123456789/30137

Cabrera, L. E. (2017). Tesis. Retrieved from http://repositorio.ug.edu.ec/handle/redug/19955

Cabrera, L. V. (2013). Introducción a CSS. Recuperado de: https://www. cs. us. es/cursos/bd/temas/BD-Tema-10. pdf.

Cadavid, A. N. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva, 11(2), 30. https://doi.org/10.15665/rp.v11i2.36

Castillo Veloz, Mishell Estefanía. Chuquitarco Velasco, Kevin Jair (2023). Sistema de detección de intrusos en sitios web, usando modelos y, o algoritmos de Machine Learning: caso práctico Phishing Google Chrome. Carrera de Ingeniería en Software. Departamento de Ciencias de la Computación. Universidad de las Fuerzas Armadas ESPE. Extensión Latacunga.

Challenger-Pérez, I., Díaz-Ricardo, Y., Y Becerra-García, R. A. (2014). El lenguaje de programación Python. Ciencias Holguín, 20(2), 1-13.

Colastra, B. (2017). Herramienta para el prototipado de aplicaciones móviles usando mockups.

Coronado Huamán, H. H., Sanz García, L., Y Han, A. (2020, junio). Detección automática de sitios web fraudulentos [Info:eu repo/semantics/bachelorThesis]. Retrieved from https://eprints.ucm.es/id/eprint/68262/

Coste de la vulneración de datos 2022—España. (2023, mayo 11). IBM. Retrieved from https://www.ibm.com/es-es/reports/data-breach.

Coste de la vulneración de datos 2022—España | IBM. (n.d.). Retrieved May 11, 2023, from https://www.ibm.com/es-es/reports/data-breach

Dalton, J. (2019). Burn down chart. Great Big Agile: An OS for Agile Leaders, 143-145.

Deemer, P., Benefield, G., Larman, C., Y Vodde, B. (2009). Información básica de SCRUM. California: Scrum Training Institute.

Detección automática de sitios web fraudulentos Automatic detection of fraudulent websites—E-Prints Complutense. (n.d.). Retrieved May 13, 2023, from https://eprints.ucm.es/id/eprint/68262/

di Laurea, T. (n.d.). Cybersecurity e Gamification.

Dominguez, A. H., Y García, W. B. (2021). Principales mecanismos para el enfrentamiento al phishing en las redes de datos. 15.

Dominguez, A., Y García, W. (2021). Principales mecanismos para el enfrentamiento al phishing en las redes de datos. In Principales mecanismos para el enfrentamiento al phishing en ... - SciELO (p. 15). SciELO.

es, S. [at] rediris [dot]. (2008, November 12). RedIRIS - Sistemas de detección de intrusos. RedIRIS. España. https://www.rediris.es/cert/doc/unixsec/node26.html

Fernandez, A. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1). (Y. Y. Python 3 al descubierto—2a ed. Alfaomega Grupo Editor. Freund, Ed.) doi: https://doi.org/10.1006/jcss.1997.1504

Ferrer, J., García, V., Y García, R. (2013). Curso completo de HTML. Recuperado de: http://es. tldp. org/Manuales-LuCAS/doc-curso-html/doc-curso-html. pdf.

Gomez M, R. (2021). Curso de desarrollo Web. HTML, CSS y JavaScript. Anaya Multimedia. .

Gonzaga, M. K. C., Pazos, W. J. O., Meneses, L. J. U., Y Esteban, J. A. (2019). Metodología Híbrida de Desarrollo de Software combinando XP y SCRUM. MIKARIMIN Revista Multidisciplinaria, 5(2), 109-116.

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media, Inc

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. Genetic Programming and Evolvable Machines, 19(1-2), 305-307. https://doi.org/10.1007/s10710-017-9314-z

Hernández Báez, I., & *CA1234125. (2016). Clasificador bayesiano ingenuo en RapidMiner. https://hdl.handle.net/20.500.12371/1757

Host, run, and code Python in the cloud: PythonAnywhere. (n.d.). Retrieved July 11, 2023, from https://www.pythonanywhere.com/.

HTML: Lenguaje de etiquetas de hipertexto | MDN. (2023, July 24). https://developer.mozilla.org/es/docs/Web/HTML

Indicadores de compromiso (IOC): Cómo los recopilamos y utilizamos. (2022, December 21). https://securelist.lat/how-to-collect-and-use-indicators-of-compromise/97380/

James, G., Witten, D., Hastie, T., Y Tibshirani, R. (2013). An Introduction to Statistical Learning (Vol. 103). Springer New York. https://doi.org/10.1007/978-1-4614-7138-7

Jaramillo Basantes, F. P. (2023). 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]. https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/38430

Jaramillo Basantes, F. P. (2023). 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. Retrieved from Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Tecnologías de la Información.: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/38430

Ken, S., Y Sutherland, J. (2020). The Scrum Guide. Scrum Alliance.

Khayal, S. H., Khan, A., Bibi, N., Y Ashraf, T. (2009). Analysis of password login phishing based protocols for security improvements. 2009 International Conference on Emerging Technologies, 368–371. https://doi.org/10.1109/ICET.2009.5353144

Kikitamara, S., Y Noviyanti, A. A. (2018). A Conceptual Model of User Experience in Scrum Practice. 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 581–586. https://doi.org/10.1109/ICITEED.2018.8534905

Mariz, L. M. R. D. S., Franca, A. C. C., Y Silva, F. Q. B. D. (2010). An Empirical Study on the Relationship between the Use of Agile Practices and the Success of Software Projects that Use Scrum. 2010 Brazilian Symposium on Software Engineering, 110–117. https://doi.org/10.1109/SBES.2010.17

Mariño, S. I., Y Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. 19(4).

Marsland, S. (2014). Machine Learning: An Algorithmic Perspective (2.ª ed.). Chapman and Hall/CRC. https://doi.org/10.1201/b17476

Menzinsky, A., López, G., Palacio, J., Sobrino, M., Álvarez, R., Y Rivas, V. (2018). Historias de usuario. Ingeniería de requisitos ágil.

Montero, B. M., Cevallos, H. V., Y Cuesta, J. D. (n.d.). Agile methodologies against traditional methods in the software development process.

Musa, U. S., Chhabra, M., Ali, A., Y Kaur, M. (2020). Intrusion detection system using machine learning techniques: A review. 2020 International Conference on Smart Electronics and Communication (ICOSEC), 149–155.

Muñoz Cortés, F. J (2019). Desarrollo de una estrategia híbrida para la gestión de alarmas en sistemas detectores de intrusos basados en firmas MunozFrancisco_2019_DesarrolloEstrategiaHibrida.pdf (udea.edu.co)

Naik, N., Jenkins, P., Savage, N., Yang, L., Naik, K., Song, J., Boongoen, T., Y Iam-On, N. (2020). Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging. 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 1138–1145. https://doi.org/10.1109/SSCI47803.2020.9308189

Navarrete, T. (2006). El lenguaje JavaScript. Asignatura: Fundamentos de Cartografia i SIG.

Naík , N., Jenkins, P., Savage, N., Yang, L., Song , j., Song, J., . . . Iam-On, N. (2020). Fuzzy Hashing Aided Enhanced YARA Rules for Malware Triaging. 2020 IEEE Symposium Series on Computational Intelligence (SSCI). doi:https://doi.org/10.1109/SSCI47803.2020.9308189

Pascariu, C., Y Bacivarov, I. C. (2021). Detecting Phishing Websites Through Domain and Content Analysis. 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 1–4. https://doi.org/10.1109/ECAI52376.2021.9515165

Pérez, J. L. R., de Cienfuegos, U., Y Rodríguez, C. R. (n.d.). Técnicas de aprendizaje automático para la detección de intrusos en redes de computadoras.

Puga, J. L., García, J. G., Sánchez, L. D. la F., & Solana, E. I. D. la F. (2007). LAS REDES BAYESIANAS COMO HERRAMIENTAS DE MODELADO EN PSICOLOGÍA. Anales de Psicología / Annals of Psychology, 23(2), Article 2.

Robbins J, N. (2012). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media, Inc.

Robotham, H., Bosch, P., Castillo, J., Y Tapia, I. (2012). Clasificacion acustica de anchoveta (Engraulis ringens) y sardina comun (Strangomera bentincki) mediante maquinas de vectores soporte en la zona centro sur de Chile: Efecto de la calibracion de los parametros en la matriz de confusion. Latin American Journal of Aquatic Research, 40(1), 90–101. https://doi.org/10.3856/vol40-issue1-fulltext-9

Ruiz, R., Riquelme, J. C., Y Aguilar-Ruiz, J. S. (2005, June). Heuristic search over a ranking for feature selection. In International Work-Conference on Artificial Neural Networks (pp. 742-749). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sanglerdsinlapachai, N., Y Rungsawang, A. (2010). Using Domain Top-page Similarity Feature in Machine Learning-Based Web Phishing Detection. 2010 Third International Conference on Knowledge Discovery and Data Mining, 187–190. https://doi.org/10.1109/WKDD.2010.108

Sarmiento, A., Fondón, I., Velasco, M., Qaisar, A., & Aguilera, P. (n.d.). Modelo de Mezcla de Gaussianas Generalizadas para Segmentación de Melanomas.

Shalev-Shwartz, S., Y Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge University Press.

Silveira, M. R., Cansian, A. M., Y Kobayashi, H. K. (2020). Detection of Malicious Domains Using Passive DNS with XGBoost. 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), 1–3. https://doi.org/10.1109/ISI49825.2020.9280552

Silveira, M. R., Cansian, A. M., Y Kobayashi, H. K. (2020). Detection of Malicious Domains Using Passive DNS with XGBoost. 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), 1–3. https://doi.org/10.1109/ISI49825.2020.9280552

Sonmez, Y., Tuncer, T., Gokal, H., Y Avci, E. (2018). Phishing web sites features classification based on extreme learning machine. 2018 6th International Symposium on Digital Forensic and Security (ISDFS), 1–5.

Sonowal, G. (2022). Phishing Kits. Phishing and Communication Channels: A Guide to Identifying and Mitigating Phishing Attacks, 115-135.

Sonowal, G. (2022). Phishing Kits. Phishing and Communication Channels: A Guide to Identifying and Mitigating Phishing Attacks, 115-135.

Srivastava, A., Bhardwaj, S., Y Saraswat, S. (2017). SCRUM model for agile methodology. 2017 International Conference on Computing, Communication and Automation (ICCCA), 864–869. https://doi.org/10.1109/CCAA.2017.8229928

Statcounter Global Stats—Browser, OS, Search Engine including Mobile Usage Share. (n.d.). StatCounter Global Stats. Retrieved July 11, 2023, from https://gs.statcounter.com/.

Sudirman, D., Suharsono, T. N., Y Mardiati, R. (2022). Security Implementation of Wifi Password Asset Sharing With One Way Hash Cryptography Method Sha256 And QR Code. 2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 1–4. https://doi.org/10.1109/TSSA56819.2022.10063886

Tai, A. T., Alkalai, L., Y Chau, S. N. (1998). On-board preventive maintenance for long-life deep-space missions: A model-based analysis. Proceedings. IEEE International Computer Performance and Dependability Symposium. IPDS'98 (Cat. No.98TB100248), 196–205. https://doi.org/10.1109/IPDS.1998.707722

Tan, C. L., Chiew, K. L., Y Sze, S. N. (2014). Phishing website detection using URL-assisted brand name weighting system. 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 054–059. https://doi.org/10.1109/ISPACS.2014.7024424

Tang L. H., M. Q. (2023). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. Machine Learning and Knowledge Extraction,. Retrieved from https://doi.org/10.3390/make3030034

Tang, L., Y Mahmoud, Q. H. (2021). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. Machine Learning and Knowledge Extraction, 3(3), 672–694. https://doi.org/10.3390/make3030034

Tejaswi, B., Samarasinghe, N., Pourali, S., Mannan, M., Y Youssef, A. (2022). Leaky Kits: The Increased Risk of Data Exposure from Phishing Kits. 2022 APWG Symposium on Electronic Crime Research (ECrime), 1–13. https://doi.org/10.1109/eCrime57793.2022.10142092

Verma, M., Kumarguru, P., Brara Deb, S., Y Gupta, A. (2018). Analysing Indicator of Compromises for Ransomware: Leveraging IOCs with Machine Learning Techniques. 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). doi:https://doi.org/10.1109/ISI.2018.8587409

Wardman, B., Y Warner, G. (2008). Automating phishing website identification through deep MD5 matching. 2008 ECrime Researchers Summit, 1–7. https://doi.org/10.1109/ECRIME.2008.4696972

Wardman, B., Y Warner, G. (n.d.). Automating phishing website identification through deep MD5 matching. 2008 ECrime Researchers Summit. doi:https://doi.org/10.1109/ECRIME.2008.4696972

Wong Durand, S., Y Gutarra Meza, F. N. (2017). Análisis y requerimientos de software: manual autoformativo interactivo.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Y Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Xenya, M. C., Y Quist-Aphetsi, K. (2019). A Cryptographic Technique for Authentication and Validation of Forensic Account Audit Using SHA256. 2019 International Conference on Cyber Security and Internet of Things (ICSIoT), 11–14. https://doi.org/10.1109/ICSIoT47925.2019.00008

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Y Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access, 6, 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950

Anexos