FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Detección de phishing por envenenamiento del servidor de nombre de dominio para evitar el robo de información en aplicaciones web de microempresas peruanas utilizando aprendizaje de máquina

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor (es)

Bach. Garcia Gutierrez Kevin Gianmarco

ORCID: https://orcid.org/0000-0002-3517-6001

Bach. Guevara Ramirez Cesar Alberto

ORCID: https://orcid.org/0000-0002-6104-8652

Asesor

Dr. Ramos Moscol Mario Fernando

ORCID: https://orcid.org/0000-0003-3812-7384

Línea de Investigación

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú

2023

DETECCIÓN DE PHISHING POR ENVENENAMIENTO DEL SERVIDOR DE NOMBRE DE DOMINIO PARA EVITAR EL ROBO DE INFORMACIÓN EN APLICACIONES WEB DE MICROEMPRESAS PERUANAS UTILIZANDO APRENDIZAJE DE MÁQUINA

Aprobación del Jurado

MG. BANCES SAAVEDRA DAVID ENRIQUE

Presidente del Jurado de Tesis

MG. BRAVO RUIZ JAIME ARTURO

Secretario del Jurado de Tesis

DR. TUESTA MONTEZA VICTOR ALEXCI

Vocal del Jurado de Tesis

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(imos) la **DECLARACIÓN JURADA**, soy(somos) GARCIA GUTIERREZ KEVIN GIANMARCO Y GUEVARA RAMIREZ CESAR ALBERTO. del Programa de Estudios de Ingeniería de Sistemas. de la Universidad Señor de Sipán S.A.C, declaro (amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

DETECCIÓN DE PHISHING POR ENVENENAMIENTO DEL SERVIDOR DE NOMBRE DE DOMINIO PARA EVITAR EL ROBO DE INFORMACIÓN

EN APLICACIONES WEB DE MICROEMPRESAS PERUANAS

UTILIZANDO APRENDIZAJE DE MÁQUINA

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Garcia Gutierrez Kevin Gianmarco	77423798	
Guevara Ramirez Cesar Alberto	48317275	

Pimentel, 01 de marzo del 2023

Dedicatoria

A nuestros Padres

A nuestros padres que siempre estuvieron allí desde nuestro primer aliento de vida, formándonos como seres y personas de bien, por la dura tarea que les tocó hacer para que nosotros podamos estudiar y alcanzar nuestro sueño profesional, dedicado a ellos porque siempre estuvieron a nuestro lado en esta dura carrera profesional y hoy siguen aquí en esta meta, seguirán viéndonos triunfar y ser profesionales de éxito.

Los Autores.

Agradecimiento

A DIOS.

En primer lugar, agradecer a dios todo poderoso por ser guía y darnos la vida para seguir luchando.

A nuestra Familia.

A todos nuestros familiares que fueron nuestro soporte y apoyo en esta carrera profesional, que siempre nos aconsejaron y brindaron una mano para seguir logrando nuestros objetivos de llegar a la meta en este gran paso de culminar con éxito la carrera profesional, gratos agradecimientos a todos ellos.

A los Docentes

A todos nuestros maestros de la Escuela Profesional de Ingeniería de Sistemas, que, a lo largo de nuestra dura carrera profesional, nos inculcaron los valores, nos transmitieron sus conocimientos y nos guiaron hasta este momento, en especial un sincero agradecimiento a nuestro docente y asesor el Mg. Ing. Mejia Cabrera Heber Ivan por su gran apoyo y excelente labor profesional siempre estuvo allí guiándonos en todo momento para ser realidad este proyecto.

A la Universidad

A la Universidad Señor de Sipán por brindarnos su excelente ambiente y formarnos profesionalmente con calidad y alta excelencia educativa y ser un gran aporte para la sociedad.

Índice

Índice de Tablas, Figuras

Resumen

A través de los últimos años los atacantes cibernéticos han venido mejorando la manera de ejecutar ataques, es así, que existen ya muchas técnicas hoy en día para el robo de información confidencial, tal es el caso de las técnicas de ingeniería social, es la táctica más utilizada por los ciberdelincuentes para manipular a las personas y así mismo divulgar información confidencial, existen ciertos tipos de ataques Phishing, como es el caso de los ataques de tienen como nombre Envenenamiento DNS, que es un tipo de ataque Phishing. El Envenenamiento DNS es un tipo de ataque especial donde el atacante no apunta a un solo usuario si no que envenena o ataca al servidor del Sistema de Nombres de Dominio (DNS), es así, que todos los usuarios que utilizan el servicio DNS serán víctimas de un ataque Phishing de este Tipo. Es por ello que numerosos trabajos de investigación se han venido desarrollando para la identificación de tipos de ataques de phishing por envenenamiento de DNS, Sin embargo, cada año los ciber delincuentes siguen cambiando sus estrategias de distintas nuevas formas, además de que son difíciles de detectar, es así que suelen aparecer también nuevos métodos para detectar ataques de tipo Phishing. Por esta Razón en este trabajo de investigación se realizó un estudio para detectar ataques de phishing por envenenamiento del servidor DNS en aplicaciones web, para esto se utilizaron algoritmos de Machine Learning en base a la mejor precisión que tuvieron en sus respectivos estudios. Los resultados obtenidos demuestran que de entre los algoritmos de detección como Naive Bayes, XGBoost, Random Forest, Perceptrón Multicapa, el que mejor resultados obtuvo fue Naive Bayes ya que este arrojó un 99.04% de precisión para la detección de ataques de envenenamiento a servidores DNS, seguido de Perceptrón Multicapa con un 80%, dejando atrás a los algoritmos de XGBoost y Random Forest con un 63% y 75% respectivamente. Entonces queda evidenciado que el algoritmo Naive Bayes puede detectar ataques de Phishing de una manera eficaz.

Palabras Clave: Artículo, Delito Ataque, Método, Técnica, Detección, Ciber Delincuente, Algoritmo, Phishing, Pharming, Servidor, Envenenamiento, Aprendizaje de Máquina**.**

Abstract

Throughout the last years cyber attackers have been improving the way of executing attacks, thus, there are already many techniques today for the theft of confidential information, such is the case of social engineering techniques, is the tactic most used by cybercriminals to manipulate people and also disclose confidential information, there are certain types of Phishing attacks, as is the case of attacks have as name DNS Poisoning, which is a type of Phishing attack. DNS Poisoning is a special type of attack where the attacker does not target a single user but poisons or attacks the Domain Name System (DNS) server, so all users who use the DNS service will be victims of a Phishing attack of this type. However, every year cybercriminals keep changing their strategies in different new ways, and they are difficult to detect, so new methods for detecting phishing attacks tend to appear. For this reason in this research work a study was conducted to detect phishing attacks by poisoning the DNS server in web applications, for this Machine Learning algorithms were used based on the best accuracy they had in their respective studies. The results obtained show that among the detection algorithms such as Naive Bayes, XGBoost, Random Forest, Multilayer Perceptron, the one that obtained the best results was Naive Bayes since it yielded 99.04% accuracy for the detection of DNS server poisoning attacks, followed by Multilayer Perceptron with 80%, leaving behind the XGBoost and Random Forest algorithms with 63% and 75% respectively. It is then evident that the Naive Bayes algorithm can effectively detect phishing attacks.

Keyword: Article, Crime Attack, Method, Technique, Detection, Cybercriminal, Algorithm, Phishing, Pharming, Server, Poisoning, Machine Learnin.

I. INTRODUCCIÓN

1.1. Realidad Problemática.

A través de los últimos años los atacantes cibernéticos han venido evolucionando sus técnicas de ataques, es así, que existen muchas técnicas hoy en día para el robo de información confidencial, tal es el caso de las técnicas de ingeniería social, es la táctica más utilizada por los ciberdelincuentes para manipular a las personas y a sí mismo divulgar información confidencial.

[1] Dentro de las técnicas de ingeniería social existentes tenemos la técnica que tiene como nombre Phishing, cuyos atacantes que utilizan estas técnicas phishing, tienen como nombre phishers.

El phishing es un tipo de ataque o delito cibernético, se basa principalmente, en que, el atacante o phisher, imita a una persona e institución real al presentarlo como persona o entidad de existencia auténtica a través mensajes por medio de los correos electrónicos y otros medios de comunicación.

[2] En este tipo de ataque cibernético, el atacante envía enlaces de carácter malicioso o archivos adjuntos a través de mensajes de correos electrónicos de tipo phishing, es así, que pueden realizar diversas funciones maliciosas, logrando así, la captura de los datos de inicio de sesión o la información privada de la cuenta de la víctima.

Bharat & Chandrasekaran [3] Dan a conocer que, existen ciertos tipos de ataques Phishing, como es el caso de los ataques de tienen como nombre Pharming, que es un

tipo de ataque Phishing. Este ataque puede realizarse explotando vulnerabilidades en el servidor de nombre de dominio, con sus siglas DNS, que es un tipo de ataque más avanzado que el phishing normal.

El Pharming es un tipo de ataque especial donde el atacante no apunta a un solo usuario si no que envenena o ataca al servidor del Sistema de nombres de dominio, es así, que todos los usuarios que utilizan el servicio DNS serán víctimas de un ataque Phishing de este tipo.

Sahoo, P [4] nos expresa que, la detección de ataques de phishing con alta precisión es un problema de investigación desafiante. Los phishers engañan a los usuarios para que ingresen su información confidencial en los sitios web clonados creados por ellos, por lo tanto, roban así las credenciales vitales del usuario.

Los sitios web de tipo phishing normalmente se logran detectar mediante el uso de enfoques basados en las listas negras, pero este enfoque reporta fallas, ya que, los sitios de tipo phishing en la lista blanca no se logran detectar mediante este enfoque. Con el uso cada vez mayor de internet la información en los navegadores web y servidores son altamente vulnerables a sufrir diferentes ataques de seguridad, aunque se utilizan medidas de seguridad en los navegadores web y servidores aún son propensos a ser atacados.

Vijayan, J. [5] nos expresa que las técnicas para la protección a las estafas de phishing, están obligando a los Phishers a adaptar y evolucionar sus métodos cada año, es así que, un análisis que realizó Microsoft con los datos recopilados de usuarios, de sus productos y servicios entre enero del año 2018 y finales de diciembre del mismo año mostró que el phishing fue el principal vector de ataque durante un año más. La proporción de correos electrónicos entrantes, que contienen mensajes con técnicas

phishing incrementó un 25%.

El Perú no es indiferente a esta ola de ciberataques de Phishing, ya que muchos de las empresas están a la vanguardia de la tecnología, dado que cuentan con plataformas en la red de internet, aplicativos webs, móviles, etc. Es por ello, que también son muy vulnerables a este tipo de ataques de robo de credenciales en sus servidores. Paz [6] describe que en el momento de esta pandemia por el COVID-19, solo Perú se registró más de 433 millones de tentativas de ciberataques en los primeros meses del 2020.

Cisco [7] expresa, dentro del informe web publicado, del reporte anual de ciberseguridad de Cisco, que los expertos defensores en sus reportes informan una dependencia mayor de la automatización, así como, por la inteligencia artificial. Los jefes de seguridad de la información que fueron entrevistados en el reporte que se hizo para el Estudio de Referencias dentro de las Capacidades para la Seguridad de la información de Cisco del año 2018, informaron que están deseosos de agregar herramientas útiles que usan técnicas de inteligencia artificial y también de aprendizaje de máquina, es por ello que, su infraestructura de seguridad está incrementando en sofisticación e inteligencia.

Por otra parte, también sienten frustración por la porción de falsos positivos que generan estos dichos sistemas, dado que, como tenemos el conocimiento que los falsos positivos incrementan la carga de trabajo del equipo de seguridad. De otro lado, cuando se les ha realizado la entrevista, acerca ¿De qué tecnologías automatizadas dependen más sus organizaciones?, 39% de los profesionales de seguridad cuestionados respondieron que tienen una cierta dependencia por la automatización, mientras que, por otro lado, el 34% dependen completamente del aprendizaje de máquina, por último, 32% de los cuestionados, dijeron que dependen completamente de la inteligencia artificial.

Es así, como detectar los ataques de tipo phishing es y será un notable desafío en el futuro tecnológico, ya que, como sabemos las características maliciosas van evolucionando, se introducen cada vez diariamente características nuevas continuas y desconocidas.

Zhang X. , Shi, Zhang.H., Liu, & Li [8] argumentan dentro de su artículo una herramienta integral contra el robo de suplantación de identidad para detectar sitios phishing en un tiempo real, que aumenta la proporción de la predicción de los sitios web, este modelo solo puede detectar sitios web de tipo phishing con una capacidad de autoaprendizaje que explora las combinaciones de nuevas características, pero sin extracción de características profesionales, el modelo implementado con mejor exactitud en clasificación de los sitios maliciosos phishing que lograron identificar, fue desarrollado con redes neuronales que llegó hasta un 99%, comparado con otros modelos este puede aumentar su precisión media del 4% al menos.

Jaspher, Paradise, Amrutha, & Eligious [9] Hacen el conocimiento, dentro de su artículo que, se presentó un marco para prevenir y detectar ataques phishing. Donde se utiliza una combinación de técnicas de aprendizaje automático, supervisados y no supervisados, para detectar ataques conocidos y desconocidos, tales como, algoritmo de árbol de decisión, algoritmo de Sim-hash, algoritmos de aprendizaje automático, algoritmo de recuperación de información TF-IDF, técnica de logotipos Web y algoritmo de minería de datos difusos para detectar ataques phishing. Obteniendo, así como resultados, que aquellos enfoques basados en Machine Learning proporcionan una buena identificación de los verdaderos positivos, así también la técnica de algoritmo de minería de datos difusos, ambos con más de 98% de exactitud.

H. Kim & J. Huh [10], Por otro lado, estos autores, hacen el conocimiento de que la mayoría de todas las técnicas de detección de tipo phishing existentes son débiles contra los ataques phishing de tipo Pharming, basados en el sistema DNS. Es así que se han observado más de 10 000 elementos de información de enrutamiento del mundo real durante un período de una semana dentro de su artículo. Llegando así al resultado experimental, que nos muestra al método de clasificación de mejor rendimiento, que es el algoritmo K-vecino más cercano, que tiene la capacidad de lograr una tasa de verdaderos positivos de 99% y una tasa de falsos positivos de 0.7%.

El Phishing de tipo pharming viene siendo una de los principales problemas para las empresas que cuentan con aplicaciones web de comercio electrónico y banca en línea. Es así como, para protegerse contra esta grave amenaza, se requiere una gran medida sofisticada conocida como anti-pharming.

Li, Chu, & Xiao [11] frente a esto proponen, un modelo híbrido de detección de ataques phishing de tipo pharming de los clientes, que se basa en el contenido de las páginas web y en las direcciones IP. Es así que, la solución propuesta en su artículo frente a estos ataques es dividido en dos pasos, como primer paso utilizan múltiples servidores de Nombre de dominio, para verificar la autenticidad de la dirección IP resultante, que corresponde a la URL de la página web, y en el segundo paso identificar la dirección IP como maliciosa o sospechosa.

Por lo tanto, se emplea un algoritmo para construir un clasificador que realice la detención si el usuario sufre ataques de pharming, llegando así a obtener resultados de la simulación una tasa de detección del ataque del modelo híbrido propuesto es superior al 90%.

1.1.2. Antecedentes de Estudio

Jaspher, Amrutha, Mercy &. Kalaivani [12], realizaron la investigación Variants of Phishing Attacks and Their Detection Techniques, en el Department of Computer Science and Engineering Karunya Institute of Technology and Sciences. El phishing es un ataque dirigido al usuario en lugar de al sistema, este ha causado estragos afectando a muchos usuarios debido a la falta de seguridad en la internet, obligándolos a revelar información confidencial como por ejemplo detalles de tarjetas de crédito, los detalles de cuentas bancarias, contraseñas de compras en línea, etc. Por esta razón, se presentó un marco para prevenir y detectar ataques phishing. Se utiliza una combinación de técnicas variadas de aprendizaje automático ya sean supervisadas o no supervisadas para detectar ataques conocidos y desconocidos como: Decision tree algorithm, Sim-hash algorithm, Machine learning algorithms, TF-IDF information retrieval algorithm, Web logo technique y Fuzzy data mining algorithm para detectar ataques phishing. Los enfoques que están basados en el aprendizaje automático proporcionan una buena identificación de los verdaderos positivos, siendo la técnica más efectiva Fuzzy rule-based approach con un 100% de exactitud. Gracias al marco de detección de ataques de tipo phishing se aumentará la seguridad ya que verificará los enlaces en el código fuente de los correos electrónicos o sistemas de páginas web.

Zhang Xu, Wang & Jajodia [13], realizaron la investigación, Gemini: An Emergency Line of Defense against Phishing Attacks, en la Universidad Haining Wang de Delaware. Los ataques a través de internet son más frecuentes hoy en día provocando pérdidas financieras significativas, con el objetivo de robar las credenciales de los usuarios en internet, el phishing juega un papel importante en muchos fraudes en línea. En un ataque phishing, el phisher así llamada la persona que efectúa estos ataques, recrea un sitio web falso que imita al sitio web original, luego atrae a los usuarios a visitar este sitio de phishing publicando anuncios o enviando correos no deseados, adquiriendo

las credenciales de los usuarios víctimas como las contraseñas. Por esta razón, se presentó un enfoque antiphishing basado en extensión de navegador llamada Gemini para elaborar una línea de defensa contra ataques de phishing, el enfoque aprovecha una nueva fuente, como el nombre de usuario para identificar un sitio phishing de alta precisión. Gemini consta de cuatro componentes principales: los cuales son el motor de extracción de nombre de usuario que identifica si la página actual es una página de inicio de sesión, el motor de verificación el cual funciona en segundo plano y escucha los mensajes del motor de extracción de nombre de usuario, recibir el nombre de usuario del motor de extracción, verifica y busca el actual, los empareja a través de la base de datos de mapeo, el motor de reacción el cual asume la responsabilidad de proporcionar comentarios adecuados a un usuario cuando se detecta el sitio web de tipo phishing o permite que el usuario siga navegando con normalidad, si es que el sitio es identificado como legítimo y por último el almacenamiento local que es el encargado de almacenar permanentemente la información como las asignaciones de nombres de usuario, y los nombres de dominio. Los resultados experimentales muestran que el enfoque antiphishing llamado Geminis es capaz de identificar con éxito todos los sitios web de phishing por tanto logra una tasa de los Falsos Negativos (FN) de 0% y con una tasa de los Falsos Positivos (FP) de menos de 1%. Geminis como línea de defensa actúa de una manera eficaz contra ataques phishing entregando resultados positivos.

Nisha & Madheswari [14] realizaron la investigación, Prevention of phishing attacks in voting system using visual cryptography, en el Mahendra Engineering College. Las elecciones juegan un rol importante para la democracia en cualquier país. El phishing tiene como objetivo obtener información confidencial de usuarios, para esto se aloja en sitios web falsos los cuales son idénticos a los originales, la votación por internet se centra en cuestiones de seguridad, privacidad y secreto, así como en los desafíos para la participación y observación del proceso. Por esta razón, se presentó un enfoque para el sistema de votación y de esta manera evitar ataques phishing, antes de dividir la imagen

en dos partes, la imagen se convierte primero en Imagen monocroma (Imagen en blanco y negro). Dada una imagen S, un grupo P de “n” participantes y una estructura de acceso fuerte, un esquema criptográfico visual (VCS) que está encargado de las estructuras de acceso general (GVCS) que codifica a la imagen S en “n” secciones de transparencia. Los resultados alcanzados señalan que este enfoque de sistema de votación propuesto tiene una efectividad del 97% y será útil tanto para los votantes como para las organizaciones de muchas maneras, además que permitirá reducir el costo y el tiempo respectivamente. El sistema de votación en línea propuesto es muy efectivo y será útil para los votantes que estén lejos, así como para personas con discapacidad física, dado que se utiliza la técnica de criptografía visual, el usuario puede averiguar fácilmente si está en el sitio de phishing o en el sitio original.

Sahoo. P [4] realizó la investigación, Data Mining a Way to Solve Phishing Attacks, en el Instituto Sreenidhi de Ciencia y Tecnología, Yamnampet, Ghatkesar. Con el uso cada vez mayor de internet la información en los navegadores web y servidores son altamente vulnerables a sufrir diferentes ataques de seguridad, aunque se hayan dado medidas de seguridad tanto en navegadores web como servidores, estos aún están expuestos a sufrir ataques, el phishing es uno de estos tipos de ataques, los phishers engañan a las personas para que ingresen datos confidenciales en estos sitios creados por ellos, y así robarles información. Por esta razón, se propuso un modelo arquitectónico para diferenciar entre el correo electrónico falso y el correo electrónico real con una alta precisión y utilizar la clasificación bayesiana para dicho propósito, El algoritmo propuesto funciona en varias etapas para la detección de correos electrónicos falsos y, por lo tanto, trata de proteger a los usuarios de la filtración de su información confidencial. Los resultados obtenidos del modelo propuesto muestran que el 90% de las URL de sitios web de phishing fueron identificadas con éxito, y que solo se perdieron 4 URL. Los ataques de phishing en general son una amenaza muy tanto para usuarios como también organizaciones.

Celestine, & otros [15] realizaron la investigación, KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks, en la University of Forestry and Technology. Los ciberataques han evolucionado a un ritmo sorprendente, las violaciones de los datos privados, los ataques de tipo ransomware, los cryptojacking, el malware y los ataques de phishing ahora son más comunes. A esto se les sumaron los problemas con los proveedores y usuarios de software en donde especialmente se tiene que evitar ataques. Por esta razón, se propuso un enfoque de detección de marca de agua que se basa en código cero conocido como KeySplitWatermark, que es utilizado en la protección de software contra ciberataques, este algoritmo de código cero agrega una marca de agua al código, puede utilizar las propiedades adjuntas y se encarga de brindar una solución sólida, el algoritmo de incrustación hace uso de palabras clave para que de este modo los segmentos del código puedan crear una clave dependiente de marca de agua, de manera que los algoritmos de extracción puedan usar esta clave que eliminará la marca de agua y detectara la manipulación, cuando la manipulación aumente en un cierto tiempo definido por el usuario, el código del software original será restaurado, así será resistente a los ataques. Los resultados generados muestran que KeySplitWatermark detecta correctamente marcas de agua, lo que resultará en hasta un 15.95% y un 17.43% de disminución en tiempo de ejecución. Este enfoque KeySplitWatermark es robusto, seguro y eficiente con requisitos informáticos mínimos, tuvo un rendimiento superior en términos de tiempo de ejecución, capacidad y tamaño.

Kishan, Mukul, Palash, & Jayashri [16] realizaron la investigación, A Novel Approach to Detect Phishing Attack Using Artificial Neural Networks Combined with Pharming Detection, en el Department of Computer Engineering St. Francis Institute of Technology Mumbai, India. Phishing hace referencia a la práctica más común pero importante. En esta práctica donde se realizan innumerables fraudes, el autor se hace pasar por una entidad legítima, este correo contiene el enlace hacia el sitio web de

phishing en donde el usuario es inducido a revelar información confidencial, este sitio web de phishing es exactamente parecido al original tanto que los usuarios no sospechan. Por esta razón, en esta investigación se implementó un sistema en el cual se entrenaron las características de URL en una Red Neuronal Artificial (ANN) para una clasificación precisa, para lograr esto se usa la ingeniería de características para extraer variables importantes, la Red Neuronal Artificial se utiliza para construir el modelo y luego predecir un valor entre el rango (0,1) para predecir la legitimidad de la URL. En la fase II, se realizarán los pasos para detectar el pharming que es donde se consulta un DNS local y global para poder obtener Direcciones IP y comprobar si estas coinciden. Los resultados obtenidos muestran que las redes neuronales artificiales tienen un 98,77 % de precisión. Gracias a este sistema se puede detectar pharming y phishing con bastante precisión.

Bharat & Chandrasekaran [17], realizaron la investigación, A Client-Side Anti-Pharming (CSAP) Approach, en el Dept. of Computer Science Engineering National Institute of Technology Karnataka Surathkal, India. Pharming es un tipo de técnica de ataque avanzada de phishing, en donde el atacante quiere robar información confidencial de los usuarios de internet, estos ataques pharming pueden realizarse explotando vulnerabilidades en los servidores, Sistemas de Nombre de Dominio (DNS), el pharming es una técnica de ataque especial ya que el atacante no tiene que apuntar necesariamente a un usuario en específico. Por esta razón, se proponen un enfoque anti - pharming del lado del cliente (CSAP), se consulta al servidor DNS de terceros, luego se calcula la frecuencia del mapeo de IP de IP distinta devuelta por servidores DNS de terceros, se comprueba la IP, si la IP es igual a la de mayor frecuencia la respuesta del servidor se considera legítima. Los resultados obtenidos muestran que el enfoque es eficiente, con un tiempo de ejecución de alrededor del 67% aunque el usuario puede enfrentar un retraso en el servicio, pero podrá explorar Internet sin ser víctima de un ataque de pharming. Las empresas proveedoras de internet están ya están tomando

medidas drásticas para hacer que los servidores de sistemas de nombre de dominio (DNS) sean mucho más seguros.

Manhas, Taterh, & Singh [18] realizaron la investigación, CLAS: A Novel Communications Latency based Authentication Scheme, en el Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. Durante el año 2015 las noticias sobre ataques informáticos, señalaron que el 80% de estos ataques son dirigidos a explotar credenciales de autenticación, como las contraseñas siempre han sido las credenciales de autenticación más notorias para los servicios web, sin embargo, los atacantes siempre están buscando formas de robarlas. Por esta razón, se diseñó e implementó un nuevo esquema de autenticación basado en la Latencia de las Comunicaciones denominado CLAS, para fortalecer la seguridad de los enfoques de autenticación web aprovechando la latencia de las comunicaciones de red de ida y vuelta (RTL) entre clientes y autenticadores. El esquema CLAS está comprendido en tres tipos de entidades, las cuales son, el Usuario, el Servidor y el Relé sigiloso (SR). Los usuarios se encargan de iniciar sesión en los servidores para conectarse con los servicios, los servidores se encargan de la autenticación para con los usuarios según sean las credenciales tradicionales y las mediciones que se almacenaron anteriormente de las comunicaciones de la red. Los análisis y los ensayos realizados señalan que el esquema CLAS llega a lograr una tasa de falsos positivos (FP) de 0.0017%, mientras tanto la tasa de falsos negativos (FN) está por debajo del 0.007%. Claramente notamos que el esquema CLAS es altamente seguro y utilizable, además complementa los mecanismos de autenticación de vanguardia y fortalece la seguridad de la autenticación web.

Azeez & Oluwatosin [19] realizaron la investigación, Cyber Protector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification, en el Department of Computer Sciences, University of Lagos, Nigeria. La incrustación de URL ’s maliciosas

en correos es una de las amenazas más comunes que enfrentan los usuarios de internet, estas son utilizadas para llevar a cabo diversos ataques informáticos, como los ya conocidos pharming, phishing. Por esta razón, se propuso un sistema para identificar URL 's comprometidas en correos electrónicos usando Naïve Bayesian Classifier para detectar si una URL es falsa o verdadera. El clasificador Naïve Bayesian Classifier será utilizado en el conteo del total de apariciones que hay por cada característica en un correo electrónico y calculará la puntuación acumulativa. Si el puntaje acumulado es mayor que la entrada dada, las URL serán consideradas maliciosas, y si se da el caso de ser lo contrario, las URL serán consideradas legítimas. Como resultados se mostró un marco para la detección de phishing probabilístico que es capaz de adaptarse rápidamente a nuevos tipos de ataques con tasas positivas buenas y tasas cercanas a un 0% de falsas positivas.

Li, Chu, & Xiao [11] realizaron la investigación, A pharming attack hybrid detection model based on IP addresses and web content, en la School of Control and Computer Engineering, North China Electric Power University. Pharming es una forma de ataque que está destinada a redirigir el tráfico de red de un sitio web a otro sitio el cual es un sitio web falso. El uso de las tecnologías tradicionales de detección de phishing no es tan eficiente al momento de ataques pharming, el pharming se ha convertido en una gran amenaza para las empresas que alojan servicios web de comercios o banca en línea. Por esta razón, se propuso un modelo de detección híbrida de ataque de pharming de cliente que se basa en el contenido web y direcciones IP. Este modelo fue realizado con una división en dos fases, siendo la primera fase, en donde se utilizan conjuntos de servidores DNS para la verificación de la autenticidad de las direcciones IP resueltas que corresponden a la URL de la página web, por otro lado, la segunda fase se da cuando la dirección IP es identificada como sospechosa. Los resultados muestran que la simulación de la tasa de detección del modelo híbrido propuesto para la detección de ataques de pharming de clientes puede llegar a alcanzar un 99.50% de efectividad. El modelo de

detección puede detectar efectivamente ataques de pharming para clientes de red así protegerá el entorno de red de manera que esta será segura y confiable.

Chandra, Deb, & Nurul [20] realizaron la investigación, Know Your Customer (KYC) based authentication method for financial services through the internet, en e1 Department of Computer Science and Engineering, United International University. Los servicios financieros a través de internet como transferencias de saldo, pagos de facturas, transacciones de comercio electrónico están siendo ejecutadas bajo amenazas de ser atacadas por diversos ciberataques como de tipos phishing. Por esta razón, se propuso un método dinámico de autenticación Multifactorial (MFA) basado en KYC para garantizar el acceso seguro de los servicios financieros a través de Internet. Para que sea una transacción apropiada y segura, el método de autenticación propuesto ejecuta dos operaciones. La primera es el cálculo del factor de riesgo y la segunda es la asignación de una pregunta de desafío (CQ) en función del resultado para evaluación del riesgo. Los resultados de análisis y simulaciones obtenidas muestran que el método propuesto proporciona al 100% el mismo control que las técnicas de Autenticación Multifactorial (MFA) y la técnica de Autenticación de dos factores (2FA) existentes. Este método de autenticación multifactorial puede ser usado en cualquier dispositivo sea privado o público. El método certifica seguridad ya que aborda la mayoría de las vulnerabilidades de transacciones financieras realizadas en Internet.

Şentürk, Yerli, & Soğukpınar [21] realizaron la investigación, Email phishing detection and prevention by using data mining techniques, en el Computer Engineering Department Gebze Technical University. La ingeniería social a lo largo del tiempo se ha convertido en una amenaza en las comunidades virtuales siendo un medio para atacar sistemas de información. El phishing es un tipo de ataque en el que los phishers utilizan correos electrónicos con link fraudulentos y sitios web falsos para lograr engañar al usuario y que de esta manera ellos sin saberlo brinden información personal. Por esta

razón, se propuso un método de detección de phishing utilizando técnicas de aprendizaje automático como la técnica antiphishing de aprendizaje automático (MLAPT) está diseñada para manejar la detección de comportamientos similares al phishing con sistemas de correo electrónico, el método de reconocimiento de patrones, técnicas relacionadas con los protocolos que se utilizan al enviar correos electrónicos como el Protocolo Simple de Transferencia de Correo (SMTP) y minería de datos. Los resultados que se obtuvieron con la detección señalan una tasa de éxito con un 89% contra los ataques de tipo phishing que se dan en mensajes de correo electrónico. El phishing es una forma típica de estafas por Internet que resulta en la propagación de la información financiera y personal de los usuarios afectados.

Nathezhtha, Sangeetha, & Vaidehi [22] realizaron la investigación, WC-PAD: Web Crawling based Phishing Attack Detection, en el Madras Institute of Technology. El phishing es un delito que implica el robo de datos confidenciales del usuario, los sitios web de phishing son dirigidos especialmente a usuarios, empresas, sitios de almacenamiento en la nube, los enfoques de detección de phishing actuales no proporcionan una solución a estos problemas de ataques como los sitios de phishing. Por esta razón, se propuso una detección de ataques de tres fases llamada “Detector de ataque de phishing basado en Web Crawler (WC-PAD)”. En primer lugar, se realiza la detección basada en la lista negra de DNS, en segundo lugar, se realiza una detección basada en Web Crawler seguida de una detección basada en heurística, la IP de phishing utilizada con frecuencia se detecta fácilmente en las pruebas de lista negra de DNS. Los resultados que se pudieron obtener hacen muestra que el detector de ataques de phishing tiene una precisión de un 98.9% tanto para phishing como para detectar los ataques de phishing de día cero.

Li & Wang [23] realizaron la investigación, PhishBox: An Approach for Phishing Validation and Detection, en el Department of Electrical Engineering National Taiwan University. Los ataques de phishing se han vuelto muy comunes en nuestra vida diaria como el crecimiento de dispositivos móviles y la IoT en Internet. Aunque existen muchas herramientas y técnicas para detectar sitios web maliciosos o para evitar que los usuarios brinden su información personal, sigue siendo algo difícil mantener a los usuarios totalmente seguros. Por esta razón, se propuso un enfoque, llamado PhishBox, para recopilar eficazmente datos de phishing y generar modelos para la validación y detección de phishing, para esto en primer lugar, se diseñó un modelo de conjunto para validar los datos de phishing y aplicarlos al aprendizaje activo para reducir el costo del etiquetado manual. A continuación, los datos de phishing validados se utilizarán para entrenar un modelo de detección. La tasa de detección de phishing falsos positivos (FP) se descartan en un 43,7% en promedio por lo que con este resultado se muestra que nuestro modelo de dos etapas es eficaz para verificar los sitios web de phishing. El enfoque propuesto integra la recopilación, validación y detección de sitios web de phishing en una herramienta en línea, que puede monitorear la lista negra de Phishtank y validar y detectar sitios web de phishing en tiempo real.

Zhang X., Shi, Zhang, Liu, & Li [24] realizaron la investigación, Efficient Detection of Phishing Attacks with Hybrid Neural Networks, en la Universidad de Zhengzhou. El Internet se ha convertido en una parte sumamente indispensable en nuestro día a día, donde los usuarios de Internet pueden intercambiar información confidencial como por ejemplo nombres de usuario, y las contraseñas, datos sobre sus cuentas bancarias, etc.

El phishing es considerado como una de las amenazas web que se encarga de persuadir a los usuarios para que dejen libre su información personal. Por esta razón, se propuso un modelo híbrido de aprendizaje profundo para la identificación de ataques phishing que incorpora dos componentes: uno que es el auto codificador (AE) y otro que

es una red neuronal convolucional (CNN). Para este modelo híbrido son utilizadas dos operaciones de agrupación en la red neuronal convolucional CNN: una que es la operación de agrupación máxima y otra que es la agrupación promedio. La operación de agrupación máxima y la operación de agrupación promedio son utilizadas para la reducción del tamaño del mapa de características. Se Adopta la función SoftMax como función de activación del clasificador y la unidad lineal del rectificador con fugas (Leaky-Relu) tanto para la AE como para la CNN. Los resultados obtenidos muestran que la mejor precisión de clasificación de sitios web de phishing de las redes neuronales híbridas alcanza hasta el 99%. Este modelo es capaz de aumentar la precisión promedio en al menos un 4%. Este modelo no solo puede detectar sitios web de phishing en un tiempo mínimo, sino que también es capaz de realizar un autoaprendizaje que busca la nueva combinación de características.

1.2.Formulación del Problema

¿Cómo detectar en forma precisa ataques phishing por envenenamiento del servidor de nombre del dominio en el robo de información en aplicaciones web de microempresas en el Perú?

1.3.Hipótesis.

Mediante el uso de algoritmos de aprendizaje de máquina con mayor precisión se detectarán ataques phishing por envenenamiento del servidor de nombre del dominio aplicado en el robo de la información en aplicaciones web de microempresas en el Perú.

1.4.Objetivos.

1.4.1. Objetivo General.

Detectar de forma precisa ataques phishing por envenenamiento del servidor de nombre de dominio mediante aprendizaje de máquina, para evitar el

robo de información en aplicaciones web de microempresas en el Perú.

1.4.2. Objetivos Específicos.

a. Identificar tipos de vulnerabilidades Phishing por envenenamiento del servidor de nombre de dominio en aplicaciones web del caso de estudio.
b. Seleccionar aplicación web de microempresa peruana como caso de estudio.
c. Seleccionar algoritmos de clasificación de aprendizaje de máquina para la detección de ataques Phishing por envenenamiento del servidor de nombre de dominio en aplicaciones web.
d. Implementar los algoritmos de detección de aprendizaje de máquina para la detección de ataques Phishing por envenenamiento del servidor de nombre de dominio en aplicaciones web.
e. Realizar pruebas de detección de phishing por envenenamiento del servidor de nombre de dominio para aplicaciones web.

1.4.3. Justificación e importancia del estudio.

En estos últimos años hay un número de sitios web que son utilizados para gestionar compras y ventas de productos, o bien para realizar cualquier tipo de operación va en incremento. Sin embargo, las estafas y los fraudes a través de internet también han aumentado de manera radical, los delincuentes informáticos haciendo uso de técnicas comunes de ataques phishing, con engaños han podido obtener información confidencial de usuarios a través de sitios web falsos.

Esto en gran medida ha generado un enorme problema para los usuarios en internet que utilizan específicamente estos sitios web para gestionar sus operaciones.

En este proyecto se busca detectar de forma precisa los ataques de tipo phishing que son realizados mediante el envenenamiento de servidores DNS específicamente en el robo de información en aplicaciones web de microempresas en el Perú.

1.5.Teorías Relacionadas al Tema

1.5.1. Delito Informático.

Se puede definir como delito informático, a aquel hecho delictivo que molesta con actos ilícitos realizados por medio de internet cuyo objetivo es el robo de información personal para hacer operaciones fraudulentas suplantando la identidad de usuarios y empresas. Estos actos delictivos se han vuelto de alguna manera más frecuentes siendo una gran molestia para la sociedad en general. [25].

1.5.2. ¿Qué es el Phishing?

Phishing es una de cientos de forma de ataque cibernéticos basado principalmente en técnicas engañosas de ingeniería social, mediante la utilización de código malicioso, en donde el delincuente, suplanta la identidad de empresa vulnerables o institución de confianza de fácil engaño a sus usuarios, ya que, utilizan tecnologías de la información y comunicaciones, es así que, este ataque trata de lograr engañar al usuario para que le facilite su información confidencial personal, cumpliendo con su objetivo posteriormente será implementado e utilizada para cometer algún acto fraudulento. Es por ello que se debe andar con cuidado cuando se navega por la red, verificar siempre las páginas si son confiables y seguras para navegar [26].

1.5.3.Técnicas Phishing.

a) Spear Phishing.

El Spear Phishing es un tipo de fraude muy similar al phishing tradicional, pero lo diferente en este tipo de fraude o estafa, es que, los ataques son dirigidos a usuarios o empresas ya seleccionadas, donde los fraudes o ataques se hacen a través de correos electrónicos de los usuarios a los cuales les quieren robar su

información, mandando un mensaje con una URL fraudulenta con código maliciosos [27].

b) Deceptive Phishing

Este es el tipo de phishing mayormente más utilizado entre todos, en donde el usuario víctima recibe un correo malicioso que reemplaza a alguna organización, en donde se le informa a la víctima de algún tipo de problema en alguna actividad realizada a través de internet [28].

c) Smishing

Smishing es una otro tipo de ataque de phishing, es una forma distinta de suplantar la identidad de usuarios, estos intentos de suplantación se hacen mediante mensajes que mayormente son mensajes de texto de organizaciones las cuales son suplantadas y buscan infundir miedo en los usuarios de esta forma haciéndoles brindar datos confidenciales [28].

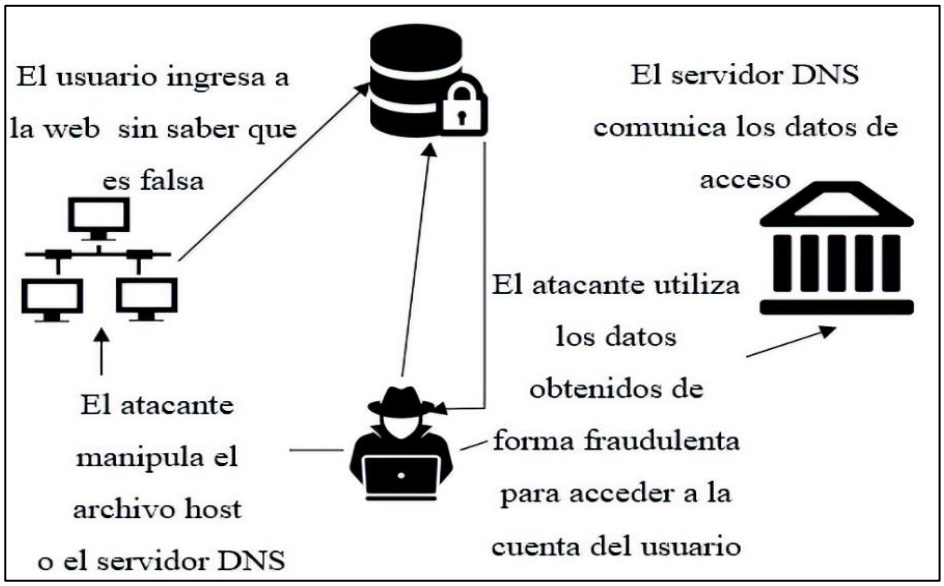
d) Machine Learning.

El Machine Learning o Aprendizaje de Máquina es una parte de la Inteligencia Artificial que crea técnicas las cuales les permite a las computadoras aprender comportamientos en base a información dada como ejemplos [30].

e) Pharming

Pharming es un tipo fraude en donde los servidores DNS que usan los usuarios al navegar por internet son manipulados de esta manera se reduce la navegación que éste hace a sitios web parecidos al original, pero que en realidad son falsos y fueron creados con fines delictivos [29].

Figura 1 *Ataque Pharming hacia un Servidor DNS*



Nota. Fuente: Elaboración Propia

f) Machine Learning.

El Machine Learning o Aprendizaje de Máquina es una parte de la Inteligencia Artificial que crea técnicas las cuales les permite a las computadoras aprender comportamientos en base a información dada como ejemplos [30].

g) Regresión Logística.

La Regresión Logística viene a ser un modelo lineal utilizado para hacer clasificación que permite la predicción de resultados de variables categóricas en base a una variable independiente o predictora [31].

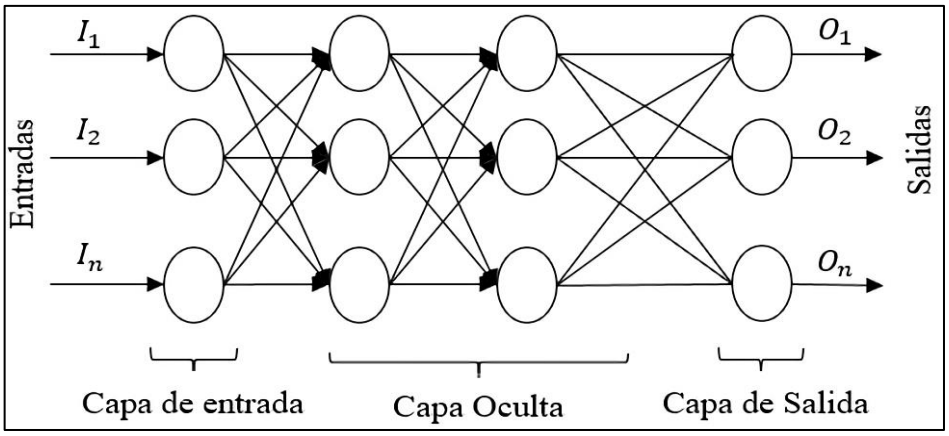
h) Máquina de Soporte Vectorial (SVM).

Support Vector Machine (SVM) o Máquinas de Soporte Vectorial es una agrupación de los algoritmos para el aprendizaje supervisado que están vinculados a problemas de clasificación y regresión [31].

i) Redes Neuronales.

Una Red Neuronal Artificial es aquel sistema que está formado por una conjunción de elementos interconectados los cuales se encargan de procesar información cuyo objetivo principal es el de emular el comportamiento del cerebro humano [32].

Figura 2 *Red Neuronal Conectada*



Nota. Fuente: Elaboración Propia

j) Clasificador Naive Bayes.

Según Javier Di Deco [33, p. 19] define que, los Naive Bayes es un clasificador probabilístico que contiene una función la cual está definida en base a la regla del Máximo Posteriori (MAP) y ofrece un rendimiento bueno ya que no hay la necesidad de estimar distribuciones de probabilidades.

k) Minería de Datos.

Wilford, I [34, p. 2] nos dice que, la minería de datos viene a ser un proceso el cual se encarga de la identificación válida de patrones los cuales se encuentran ocultos en los grandes almacenes de datos los cuales proporciona la data necesaria para la operación de algoritmos de minería de datos.

l) Regresión Lineal.

Kazmier & Diaz [35] Nos dicen que, la finalidad de la Regresión Lineal es el de estimar los valores de variables dependientes en base a que ya se conoce el valor de la variable independiente.

$$b_1 = \frac{XY - nXY}{X^{2-nX^2}}$$

$$b_0 = Y^{-b_1 X}$$

m) Regresión Lineal Simple.

Kazmier & Diaz [35] expresan que, la regresión lineal simple contiene una sola variable dependiente cuyos datos se pueden representar a través de pares de observación.

$$X_{i,y_i}; i = 1,2, \dots, n$$

II. MÉTODO

2.1.Tipo y Diseño de la Investigación.

El Proyecto llevado a cabo tiene como tipo de investigación cuantitativa, cuya tecnología aplicada Antecede de conocimiento científico para realizar los objetivos planteados, también se hará el uso de métricas que luego estas ya aplicadas se obtendrá un resultado favorable que servirá para brindar solución a la situación problemática.

El diseño de investigación es cuasi experimental, debido a que tendremos que manipular la muestra para obtener resultados óptimos, se llevará a cabo la toma de estrategias existentes para así realizar y demostrar con precisión que algoritmo de detección de ataques de tipo phishing por envenenamiento del servidor de nombre de dominio es más preciso.

2.2. Variables, Operacionalización.

2.2.1.Variables.

a) Variable Independiente.

- Algoritmos de aprendizaje de máquina.

b) Variable Dependiente.

- Ataques Phishing por envenenamiento del servidor de nombre de dominio.

2.2.2.Operacionalización

Tabla 1 *Operacionalización del proyecto*

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
VI: técnicas de aprendizaje de máquina.	Consumo de Recursos	Tiempo de Respuesta	$Tr = \sum_{j}^{n} tf_j - tf_i \quad /n$	Instrumento Electrónico y Observación
		Grado de consumo CPU	$Cc = \sum_{j}^{n} Cc_j/n$	
		Consumo de memoria	$Cm = \sum_{j}^{n} Cm_j/n$	
VD: Ataques Phishing por envenenamiento del servidor de nombre de dominio.	Rendimiento o Desempeño	Exactitud	$\frac{VN + VP}{VN + FP + FN + VP}$	
		Precisión	$\frac{VP}{FP + VP}$	
		Especificidad	$\frac{VN}{FP + VN}$	
		Recall	$\frac{VP}{FN + VP}$	

Nota. Fuente: Elaboración Propia.

2.3. Población de estudio, muestra, muestreo y criterios de selección.

2.3.1. Población

La población en esta investigación se realizó una revisión de artículos científicos los cuales fueron ubicados en diferentes bases de datos fidedignas, de esta manera la población está compuesta por todos los algoritmos de Machine Learning que estarán dentro de 12 grupos los cuales son los siguientes. El grupo de los Algoritmos de Clasificación, dentro de este grupo tenemos, al Vecino más cercano (regla NN), los k-vecinos más cercanos (regla k-NN) y Árboles de clasificación.

El grupo de los Algoritmos de Regresión, dentro de este tenemos, Regresión Lineal Múltiple, Random, Forest Regresión, Regresión Polinómica y Regresión Lineal Simple. El grupo de los Algoritmos de Árbol de Decisión, dentro de este tenemos, Árboles de decisión de Microsoft, CART, ASSISTANT, CLS y ID3/4/5.

El grupo de los Algoritmos de Redes Neuronales, dentro de este tenemos, Redes recurrentes simples, Redes Neuro-Fuzzy, Red neuronal probabilística (PNN), Redes de Patrón para Productoras de Composición, Redes de Contra Propagación, Redes Neuronales Oscilantes, Redes Neuronales Híbridas, Redes Neuronales Físicas, Red neuronal óptica, Redes Neuronales Estocásticas, Redes Neuronales Dinámicas, Red Neuronal Feed-Forward (FNN), Red Neuronal Recurrente (RNN), Redes Neuronales de Retardo en Tiempo (TDNN), Redes de Realimentación Reguladora (RFNN), Redes de Función de Base Radial (RBF), Red Neuronal de Convolución (CNN), Red Neuronal Modular, Redes Neuronales Asociativas (ASNN), Redes Neuronales ya Entrenadas Instantáneamente (ITNN), Redes Neuronales de Impulsos (SNN) e Impulsión codificada de Redes Neuronales (PCNN).

El grupo de los Algoritmos Bayesianos, dentro de este tenemos, clasificador bayesiano simple, Redes Bayesianas Dinámicas, Naive Bayes, Redes Gaussianas y DBNs. El grupo de los Algoritmos de Clustering (agrupación), dentro de este tenemos, K-Means, K-Medians y Hierarchical Clustering.

El grupo de los Algoritmos basados en Instancia, dentro de este tenemos, CART, Árbol de Decisión Conditional, Random Forest. El grupo de los Algoritmos de Aprendizaje Profundo, dentro de este tenemos, Backpropagation, DNNs y Random Forest.

El grupo de los Algoritmos de Reducción de Dimensión, dentro de este tenemos, Principal Component Analysis (PCA), T-SNE, Algoritmos de Aprendizaje por Reglas de Asociación, Algoritmos de Conjunto, Computer Vision.

2.3.2. Muestra

Para la elaboración de la Muestra se realizó una revisión de artículos científicos, se hizo un listado de algoritmos que fueron utilizados en situaciones similares al caso estudio obteniendo un listado total de 18 algoritmos, el listado se encuentra en la Tabla 18 del apartado de aportes prácticos, de este listado deriva un top de los 4 mejores algoritmos con mejor precisión entre ellos quedaron Naive Bayes (NB), Random Forest (RF), XGBoost y Perceptrón Multicapa (MLP) esto se muestra en la Tabla 19 en el apartado de aportes prácticos.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

2.4.1. Instrumentos de recolección de datos

Los instrumentos para la recolección de los datos que fueron utilizados son, la observación y todos los registros electrónicos necesarios para su implementación de este proyecto, entre ellos tendremos a los softwares que tiene como nombres, Excel del paquete de Office, así mismo Statistical Package for the Social Sciences (SPSS), también TensorFlow y Weka que en el idioma inglés significa Waikato Environment for Knowledge Analysis [36].

Méndez & Cuevas [37] nos dan por definición que el SPSS es un avanzado programa computacionales para el análisis estadístico, dado que se podría tener una representación de los datos y así crear sencillamente una numerosa multiplicidad de efectos visuales, así como, gráficos de caja radiales y también gráficos de densidad, que nos ayudarán mucho en la recolección de los datos y la traficación de ellos.

Witten, y otros [36] Definen que, weka es una plataforma utilizada para el aprendizaje automático que es de código libre y se permite ingresar mediante una interfaz gráfica hecha para el usuario, es un conjunto completo de clases de bibliotecas de Java, donde implementa muchas de los más modernos algoritmos de aprendizaje de máquina, a si también minería de datos con aplicación de un terminal o desde una API para Java. Es muy utilizado para el aprendizaje, igualmente para hacer investigación, en aplicaciones industriales y mucho más. Abarca un gran conjunto de herramientas que están integradas para poder realizar tareas de Aprendizaje Automático.

Otra de las técnicas alternativas es Tensor Flow. (Tensor Flow, 2015) en su página oficial define que es una librería también de código libre orientada a entrenar y desarrollar modelos de Aprendizaje Automático donde Mapea los nodos de un gráfico de flujo de datos a través de muchas máquinas en un clúster y a si también dentro de una máquina a través de múltiples dispositivos tecnologías de información, incluidas la Unidad Central de Procesamiento Multinúcleo (CPU), también la Unidad de Procesamiento Gráfico (GPU) de uso general y diseño personalizado conocidos como Unidades de procesamiento de tensor (TPU).

2.4.2. Técnicas

Las técnicas utilizadas son técnicas estadísticas, donde se verá un análisis para las variables cuantitativas, donde se desarrollarán tablas de frecuencia ya que son el punto de partida al momento de analizar datos agrupados, tienen grandes usos en la estadística y especialmente en lo que a las probabilidades se refiere. Por el momento se empleará ya que ayudará a resumir grandes cantidades de datos.

Rehill, G [38] da entender que la frecuencia de un valor de datos es igual a la cantidad de veces que ocurre el valor de datos. puesto que de una observación particular es el número de veces que la observación ocurre en los datos. La distribución de una variable es el patrón de frecuencias de la observación. Las distribuciones de frecuencia se representan como tablas de frecuencia, histogramas o polígonos.

Por otro lado, obteniendo lo datos recopilados se utilizaron los indicadores de la matriz de confusión Anexo 11, para visualizar el desempeño que se da en la detección de los ataques maliciosos de tipo phishing con los principales algoritmos de Machine Learning haciendo el uso de los instrumentos especificados en el punto

(2.4.1) anteriormente, los cuales son la herramienta del Software de scikit-learn, Tensor Flow.

Posteriormente se construyó una Matriz Anexo 9 [39], donde colocaremos los valores visualizadas y extraídas de los instrumentos mecánicas, tales como la Exactitud, Precisión, Especificidad y Recall, es así que se obtuvo el algoritmo de aprendizaje de máquina más preciso para la detección de ataques Phishing por envenenamiento del servidor de nombre del dominio.

2.5. Procedimiento de análisis de datos.

2.5.1. Consumo de recursos

a) Tiempo de Respuesta

Es el tiempo que se tarda desde el instante en que se percibe una acción hasta el momento en que se da la respuesta.

$$Tr = \sum_{j}^{n} tf_j - tf_i \, /n$$

Tabla 2 *Tiempo de Respuesta*

Variables	Descripción
Tr	Tiempo de Respuesta
Tf_j	Tiempo inicial
Tf_i	Tiempo final
n	Total, del tiempo

Nota. Fuente: Elaboración Propia

b) Grado de consumo CPU

Se refiere al tiempo de respuesta que es usado al ejecutar un proceso.

$$Cc = \sum_{j}^{n} Cc_j/n$$

Tabla 3 *Grado de consumo de CPU*

Variables	Descripción
Cc	Consumo de CPU
Cc_j	Uso de CPU en ejecución.
n	Total, de uso del CPU.

Nota. Fuente: Elaboración Propia

c) Consumo de memoria

Se refiere al consumo de la memoria RAM al momento de ejecutar un algoritmo el cual está denotado por la siguiente fórmula.

$$Cm = \sum_{j}^{n} Cm_j/n$$

Tabla 4 *Consumo de Memoria RAM*

Variables	Descripción
Cm	Consumo de Memoria
Cm_j	Consumo de memoria en uso.
n	Total, de consumo de la Memoria.

Nota. Fuente: Elaboración Propia

2.5.2. Rendimiento o Desempeño.

a) Exactitud

Es el grado de acercamiento que hay entre los resultados que ya fueron medidos con el valor de referencia o conocido también como el valor verdadero, el cual está definido con la siguiente formulación.

$$Ex = \frac{VN + VP}{VN + FP + FN + VP}$$

Tabla 5 *Grado de Exactitud*

Variables	Descripción
Ex	Exactitud.
VP	Verdaderos Positivos
VN	Verdaderos Negativos.
FP	Falsos Positivos
FN	Falsos Negativos

Nota. Fuente: Elaboración Propia

b) Precisión

Es el grado en el que se expresa una cantidad numérica, permite medir la calidad de un modelo de machine Learning para una clasificación, la cual está definida con la siguiente formulación.

$$Pr = \frac{VP}{VP + FP}$$

Tabla 6 *Grado de Precisión*

Variables	Descripción
Pr	Precisión.
VP	Verdaderos Positivos
FP	Falsos Positivos

Nota. Fuente: Elaboración Propia.

c) Especificidad

Es la proporción de los verdaderos negativos (VP) con respecto de la suma de falsos positivos (FP) con consecuencia en los verdaderos negativos (VN), la cual está definida con la siguiente formulación.

$$Es = \frac{VN}{FP + VN}$$

Tabla 7 *Especificidad*

Variables	Descripción
Es	Especificidad.
VN	Verdaderos Negativo
FP	Falsos Positivos

Nota. Fuente: Elaboración Propia.

d) Recall

Es la métrica que informa la cantidad de datos en la que un modelo de Machine Learning es capaz de reconocer, la cual está definida con la siguiente formulación.

$$Rec = \frac{VP}{FN + VP}$$

Tabla 8 *Recall*

Variables	Descripción
Rec	Recall.
VP	Verdaderos Positivos
FN	Falsos Negativo

Nota. Fuente: Elaboración Propia

2.6. Criterios Éticos.

Confidencialidad: La investigación tendrá como estrategia la asignación de un alias para garantizar la protección en la identificación personal para las personas involucradas en el proyecto. Además, cabe recalcar que la información obtenida se recopiló manteniendo las normas y los valores que un profesional debe tener.

Derechos de Autor: El presente proyecto respeta la autoría de cada investigación empleada para respaldar la veracidad de la información, los cuales están citados y referenciados en todo el informe.

2.6.1. Criterios de Rigor Científico

Fiabilidad. El proyecto tendrá resultados estables y consistentes debido a que se hará uso de fórmulas establecidas para poder reducir los márgenes de error en los datos obtenidos y poder garantizar la fiabilidad en la investigación, además de respetar las políticas al realizar la implementación.

Validez. Para esta investigación se usarán indicadores que se especifican en la tabla de Operacionalización. Estos indicadores ayudarán a poder medir las variables y poder obtener datos que luego serán evaluados por personal experto de la materia.

Consistencia. La investigación se fundamenta con pruebas consistentes y demostrables como son los artículos científicos mencionados en el punto 1.2 Trabajos previos.

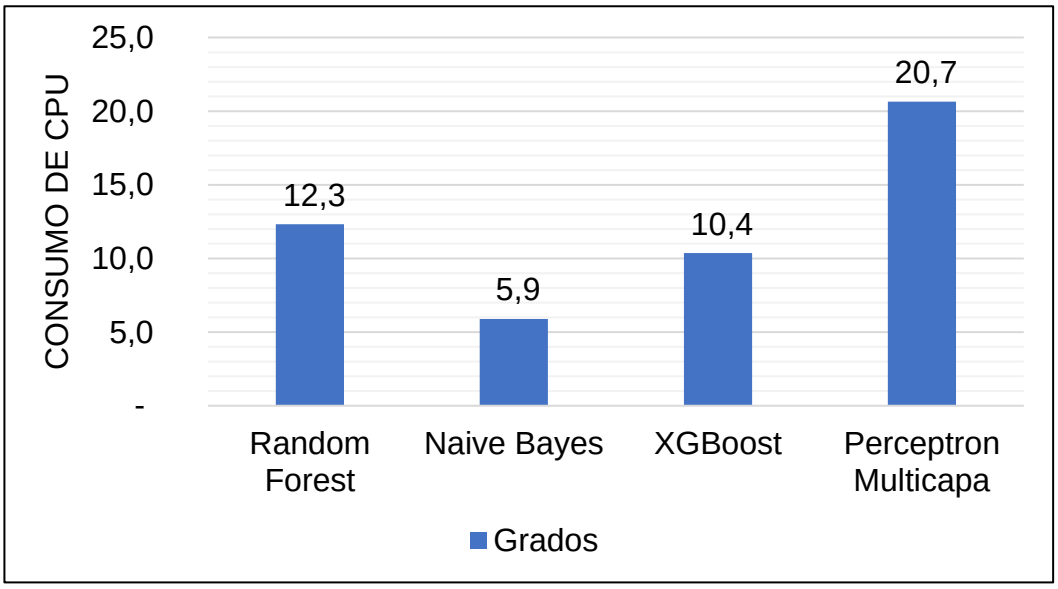
III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

3.1.1. Grado de Consumo de CPU

Los resultados que se observan a continuación hacen referencia al consumo de CPU, al tiempo promedio de respuesta y al grado de consumo de memoria estas mediciones son obtenidas al momento de ejecutarse los algoritmos.

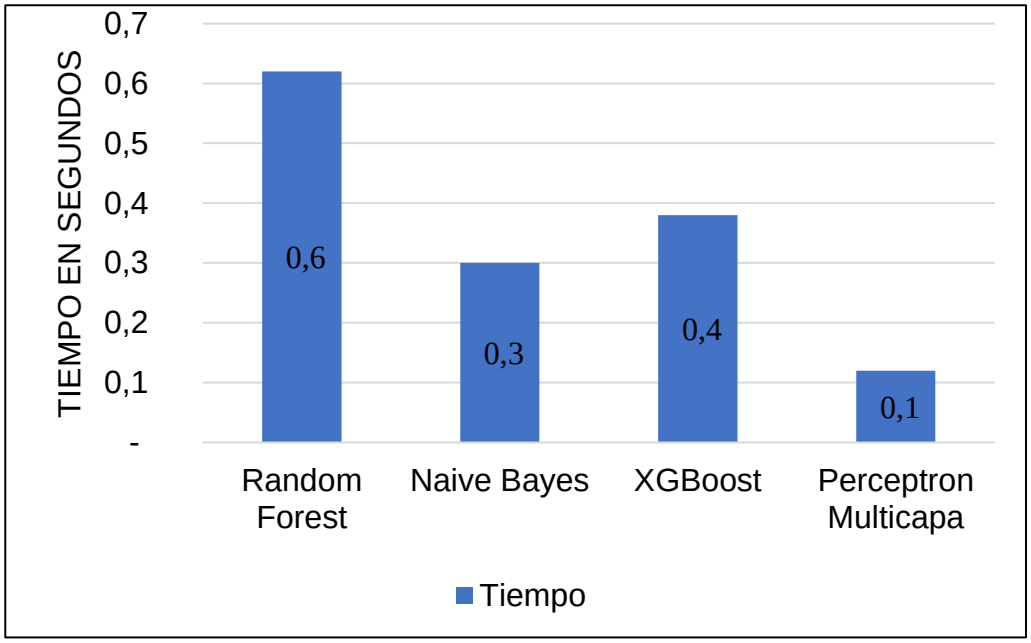
Figura 3 *Grado y Consumo de CPU por cada algoritmo utilizado*



Nota. Fuente: Elaboración Propia.

En la figura 3 se muestra resultados en referencia al consumo de CPU, al momento que, de ejecución de los algoritmos, obteniendo para cada algoritmo diferentes resultados de consumo para el algoritmo Random Forest se obtuvo un 12,3% de consumo de CPU, el algoritmo de Perceptrón Multicapa obtuvo un 20,7% siendo el algoritmo que más consumo de CPU obtuvo, el algoritmo XGBoost obtuvo un 10,4% de consumo de CPU y finalmente el algoritmo de Naive Bayes con un 5,9% fue el algoritmo que menos consumo de CPU hizo.

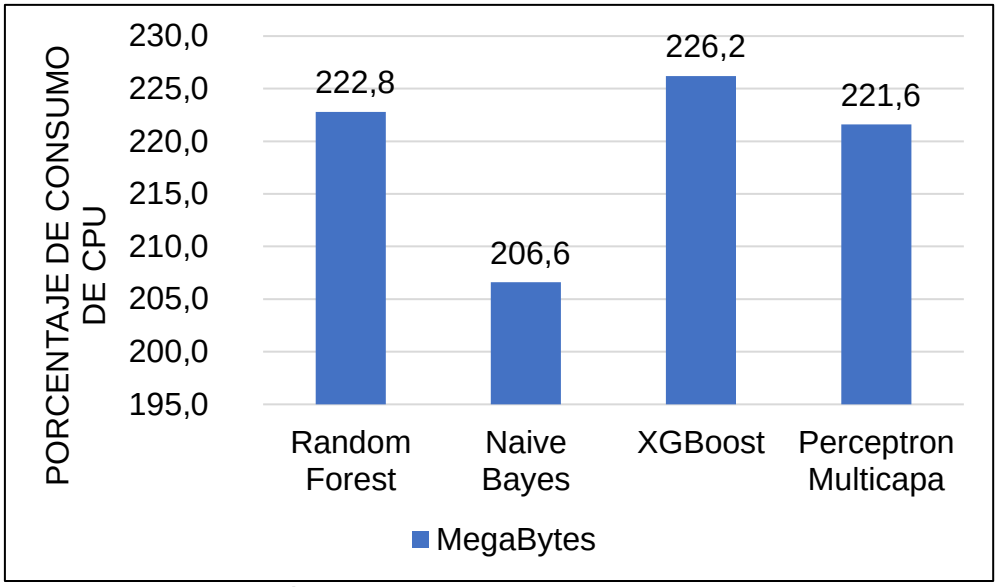
Figura 4 *Tiempo Promedio de Respuesta*



Nota. Fuente: Elaboración Propia.

La figura 4, muestra la gráfica del tiempo promedio de respuesta para los algoritmos Random Forest con un 0,6, Naive Bayes con un 0,3 seguido del algoritmo XGBoost con un 0,4 y finalmente el algoritmo de Perceptrón Multicapa con un 0,1 siendo el mejor algoritmo en para el entrenamiento de los datos ya que su tiempo de respuesta es mucho mejor que los otros algoritmos.

Figura 5 *Grado de consumo de Memoria.*



Nota. Fuente: Elaboración Propia.

En la figura 5 se puede observar el grado de consumo de memoria RAM que se obtienen de los algoritmos ejecutados y que están expresados en Megabyte, al ejecutar el algoritmo para el entrenamiento del conjunto de datos este consume menores cantidades de memoria.

Los resultados obtenidos muestran que para Random Forest su consumo de memoria es de 22.8, para el algoritmo Naive Bayes tiene un consumo de 206,6 siendo el que menos consumo de memoria obtuvo dando una diferencia de 16,2MB, el algoritmo de XGBoost fue el que más memoria consume obteniendo un 226,2MB finalmente seguido del algoritmo Perceptrón Multicapa con un 221,6 MB de consumo de memoria.

Para poder evaluar la precisión de los algoritmos Random Forest, Naive Bayes, XGboost y Perceptrón Multicapa se realizó con los indicadores de Exactitud, Precisión, Especificidad, Recall.

Para poder obtener estos resultados fueron ejecutados los algoritmos antes mencionados y así poder obtener la matriz de confusión así mismo se pudo obtener los porcentajes para cada indicador.

3.1.2. Resultados en la matriz de confusión según cada algoritmo ejecutado:

3.1.2.1. Algoritmo Naive Bayes

Tabla 9 *Matriz de confusión Algoritmo Naive Bayes*

		Clasificación	
		Positivo	Negativo
Reales	Positivo	58	14
	Negativo	1	4

Nota. Fuente: Elaboración Propia.

Teniendo un total de 77 registros en el conjunto de datos, el algoritmo de Naive Bayes pudo detectar 59 intentos de ataque que fue tomado como positivos y solo 18 que fueron tomados como negativos.

3.1.2.2. Algoritmo Random Forest.

Tabla 10 *Matriz de confusión Algoritmo Random Forest*

		Clasificación	
		Positivo	Negativo
Reales	Positivo	39	9
	Negativo	13	16

Nota. Fuente: Elaboración Propia.

Con un total de 77 registros en el conjunto de datos, el algoritmo Random Forest pudo detectar 52 intentos de ataque que fueron tomados como positivos y sólo 25 que fueron tomados como negativos.

3.1.2.3. Algoritmo XGBoost

Tabla 11 *Matriz de Confusión Algoritmo XGBoost*

		Clasificación	
		Positivo	Negativo
Reales	Positivo	37	15
	Negativo	22	3

Nota. Fuente: Elaboración Propia.

Con un total de 77 registros en el conjunto de datos, el algoritmo de XGBoost pudo detectar 59 intentos de ataque que fueron tomados como positivos y 18 que fueron tomados como negativos.

3.1.2.4. Perceptrón Multicapa

Tabla 12 *Matriz de Confusión Perceptrón Multicapa*

		Clasificación	
		Positivo	Negativo
Reales	Positivo	44	9
	Negativo	11	13

Nota. Fuente: Elaboración Propia.

Teniendo un total de 77 registros para el conjunto de datos, el algoritmo de XGBoost pudo detectar 55 intentos de ataque que fueron tomados como positivos y 22 que fueron tomados como negativos.

3.1.3. Resultados del entrenamiento del conjunto de datos, obtenidos según los diferentes algoritmos

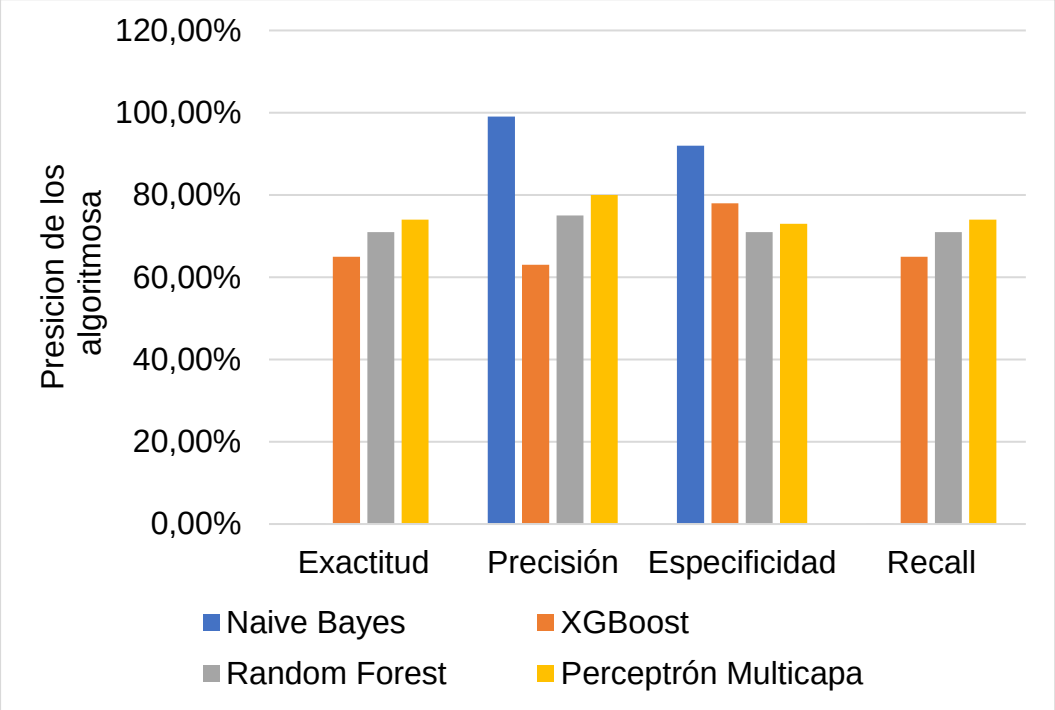
Tabla 13 *Resultados de entrenamiento obtenidos según los algoritmos usados*

Algoritmo Clasificador	Exactitud	Precisión	Especificidad	Recall
Naive Bayes	0.08%	99.04%	92%	0.08%
XGBoost	65%	63%	78%	65%
Random Forest	71%	75%	71%	71%
Perceptrón Multicapa	74%	80%	73%	74%

Nota. Fuente: Elaboración Propia.

Los datos que se visualizan en la tabla 13 muestran que el algoritmo Naive Bayes obtuvo mejores resultados en indicadores de rendimiento para la detección de ataques por envenenamiento del servidor DNS esto se debe a que el algoritmo Naive Bayes es capaz de detectar con mayor precisión los ataques por envenenamiento del servidor DNS.

Figura 6 *Precisión de los Algoritmos de Clasificación*



Nota. Fuente: Elaboración Propia.

Los datos muestran que la precisión del algoritmo de Naive Bayes es por mucho superior a la precisión de los otros algoritmos que fueron probados en esta investigación.

3.2. Discusión

Hajaramusa, Gital, Zambuk, Umar, Usmanwaziri [40] en la investigación que realizaron denominada “A Comparative Analysis Of Phishing Website Detection Using XGBoost Algorithm” se centraron en phishing proponiendo un modelo de detección de sitios web de phishing basándose en el algoritmo de XGBoost obteniendo resultados de precisión del 97.27%, superando a la red neuronal probabilística y también a Random Forest arrojando una precisión de 96.79% y 95.56% respectivamente de esta manera se probó que el algoritmo XGBoost es muy eficiente en la precisión de sitios web de phishing.

En base a los resultados que se pudieron obtener en esta investigación el algoritmo XGBoost arrojó una precisión de 63% de esta manera este algoritmo es muy malo para detectar ataques de phishing.

Cuzzocrea, Martinelli, Mercaldo [41] en la investigación que realizaron denominada “Application of automatic learning techniques to detect and analyze web phishing attacks” enfocada en el análisis y la detección de ataques phishing a webs utilizando diferentes algoritmos de aprendizaje automático entre ellos Random Forest el cual obtuvo 91,2% de exactitud.

Para la presente investigación también se hizo la medición de precisión utilizando el algoritmo Random Forest, pero a diferencia de la investigación antes mencionada para este arrojó una precisión muy baja del 75% siendo muy ineficiente para la detección de ataque de phishing.

Oladimeji [42] en su investigación denominada “Text Analysis and Machine Learning Approach to Phished Email Detection” se enfoca en la extracción de textos de correos electrónicos de phishing y ham, se utilizaron tres técnicas de aprendizaje automático que son Naive Bayes, K-Nearest Neighbor (KNN) y Support Vector Machine (SVM) siendo Naive Bayes el que obtuvo una mejor precisión con un 99% superando a las otras dos técnicas de aprendizaje automático.

Para esta investigación se utilizó el algoritmo de aprendizaje automático Naive Bayes de igual manera se sobrepuso a las otras técnicas utilizadas ya que este fue el que obtuvo una precisión del 100% lo que indica que Naive Bayes es muy eficiente en la detección de ataques de phishing.

Krishnaveni & Sathiyakumar [43] en su investigación realizada denominada “Multiclass Classification of XSS Web Page Attack using Machine Learning Techniques” hace referencia a los ataques de Cross-Site Scripting (XSS) que son dirigidos a aplicaciones web y presentaron una técnica de aprendizaje automático para clasificar páginas web maliciosas y detectar el script XSS, los experimentos que se realizaron fueron hechos con tres métodos de aprendizaje automático como Naive Bayes (NB), Decision Tree (DT) y Multi-Layer Perceptron (MLP), siendo Decision Tree el clasificador que obtuvo un 100% de precisión seguido de Multi-Layer Perceptron con un 96,20% y dejando a Naive Bayes al final.

Para esta investigación solo se utilizó el clasificador de Perceptrón Multicapa (NV) para detectar el envenenamiento de servidores DNS en aplicaciones web el cual obtuvo una precisión de 80% en detección quedando en segundo lugar ya que el que quedó en primer lugar de entre los 4 clasificadores utilizados fue Naive Bayes con un 100 de precisión.

3.3. Aporte de la Investigación.

Las vulnerabilidades son debilidades o defectos en donde la seguridad puede ser explotada específicamente en las aplicaciones web de microempresas, en donde los atacantes comprometen la seguridad de estas aplicaciones y el acceso a información confidencial de los usuarios para realizar acciones maliciosas.

Para identificar los tipos de vulnerabilidades phishing se hizo una revisión literaria científica buscando e indagando en artículos científicos acerca de cuáles son los tipos de vulnerabilidades que hay en las aplicaciones web de microempresas como resultado se generó la siguiente lista de vulnerabilidades.

Palabras clave: vulnerabilidad de seguridad, aplicación web, riesgos de seguridad.

Tabla 14 *Lista de tipos de vulnerabilidades en aplicaciones web de microempresa*

N.º	Tipo	Vulnerabilidad	Autor(es)
1	Cross Site Request (CSRF)	Entradas no válidas	Hernández & Mejia (2017)
2	Configuraciones de seguridad incorrectas	Gestión Incorrecta de Errores Configuración de seguridad incorrecta	Cova, Felmetsger &Vigna
3	Inyección SQL, NoSQL, OS y Envenenamiento DNS	Ejecución de comandos no intencionados o acceso a datos	Kaur & Preet (2015)
4	Autenticación Comprometida	La Gestión de las sesiones de usuario	Hernández & Mejia (2017)
5	Control de Acceso	Modificación de Parámetros en la URL	Hernández & Mejia (2017)
N.º	**Tipo**	**Vulnerabilidad**	**Autor(es)**
6	Entradas externas XML (XXE)	Explotación de código vulnerable, dependencias o integraciones, robo de datos confidenciales	Gonzales & Montesino (2018)
7	Scripts de sitios cruzados (XSS) basados en DOM	Modificación del script DOM mediante el robo de datos confidenciales.	Talebzadeh & Ghodrat (2017)
8	Inclusión de cualquier archivo	Ejecución de código en el servidor web (archivos locales y remotos)	Hernández & Mejia (2017)
9	Insuficiente registro y monitoreo	Fugas constantes de datos	Gonzales & Montesino (2018)
10	Exposición de datos sensibles	Protección inadecuada de datos de usuarios	Rojas (2017)
11	Gestión Incorrecta de Errores	Muestra mensajes de error como salida después de que se procesa la aplicación.	Yadav, Gupta, Singh, Kuma & Sharma
12	Entrada no válida	Ingreso de información maliciosa en la aplicación, evitando así	Yadav, Gupta, Singh, Kuma & Sharma

		la seguridad del sitio web.	
13	Uso de componentes con vulnerabilidades conocidas	Obtención de Exploits	Gonzales & Montesino (2018)
14	Registro y Monitoreo Insuficientes	Cambio de los sistemas, alteración, extracción y/o destrucción de los datos	Gonzales & Montesino (2018)

Nota. Fuente: Elaboración Propia.

Las vulnerabilidades identificadas en las tablas 14, son causantes de problemas en entornos aplicados en la red que se basan en servidores los cuales interactúan con los usuarios a través de sistemas web, se observó que la mayoría de ellas son causadas por intentar entregar un servicio lo más rápido posible o quizá por mantener una confianza excesiva en el software que se utiliza, inclusive por el propio desconocimiento.

Para conocer cuáles de estas vulnerabilidades son las más graves o peligrosas en las aplicaciones web a partir de la información obtenida en la tabla 15, se realizó un análisis según el impacto, la prevalencia y la detección de estas vulnerabilidades llegando a un top de 5 tipos de vulnerabilidades mencionados en la siguiente lista.

Tabla 15 *Top 5 de vulnerabilidades más peligrosas en aplicaciones web de microempresas*

N.º	Tipo de Vulnerabilidad	Impacto	Prevalencia	Detección de Vulnerabilidad
1	Inyección SQL, NoSQ, Envenenamiento DNS	Grave	Común	Fácil
2	Autenticación Comprometida	Grave	Común	Fácil
3	Exposición de datos sensibles	Grave	Generalizada	Media
4	Entradas externas XML (XXE)	Grave	Común	Media
5	Control de Acceso	Grave	Común	Media

Nota*. Fuente: Elaboración Propia.*

3.3.1. Seleccionar aplicación web de microempresa peruana como caso de estudio

Para la presente investigación se hizo una búsqueda de microempresas que se encuentren ubicadas en la ciudad de Chiclayo específicamente en el rubro de ventas, utilizando como fuente los siguientes sitios (Anexo 3).

1. La web de la cámara de comercio.
2. La web del Ministerio de Economía y Finanzas.
3. La Plataforma digital única del estado peruano.
4. La Página trabajo.gob.pe

se realizó un listado de todas las que fueron encontradas para ser tomadas como posibles candidatas, fueron evaluadas cuidadosamente y solo una de ellas fue seleccionada como caso de estudio.

Tabla 16 *Lista de microempresas seleccionadas para ser evaluadas.*

	Nombre de Empresa	Rubro	Aplicación Web
1	Ópticas Romero	Ventas	http://www.opticasromero.com/index.php
2	Brujhas	Ventas	https://www.brujhas.com/
3	Nowlovers	Ventas	https://www.nowlovers.com/
4	Ópticas Premium	Ventas	https://www.opticaspremium.com/
5	Moda Urbana	Ventas	https://moda-urbana-clothing-store.negocio.site/
6	Leonisa	Ventas	https://www.leonisa.com/pe/
7	Noemi	Ventas	http://www.confeccionesnoemi.com/
8	Joaquim Miro	Ventas	https://www.joaquimmiro.com/
9	Bottero Perú	Ventas	https://www.botteroperu.com/

Nota. Fuente: Elaboración Propia.

Luego de identificar las microempresas como posibles candidatas de caso de estudio se hizo una evaluación basada en tres criterios de selección para que de este modo se pueda determinar cuál de todas cumplía con todos o la mayoría de los criterios propuestos.

C1. La microempresa seleccionada debe contar con una aplicación web de ventas.

C2. La microempresa seleccionada debe contar con un servidor DNS.

C3. La microempresa seleccionada debe ceder el acceso a la información.

Tabla 17 *Microempresas seleccionadas con mayor aceptación*

	Empresa	C1	C2	C3
1	Ópticas Romero	x	x	X
2	Joaquim Miro	x		X
3	Leonisa		x	X
4	Bottero Perú	x		X

Nota. Fuente: Elaboración Propia.

En la tabla 17 se muestran las 4 microempresas que fueron evaluadas según los criterios propuestos, fue Ópticas Romero la que mejor se adecuo a los criterios de selección siendo elegida como caso de estudio.

En esta investigación se hizo una revisión de la literatura científica buscando en diversas bases de datos, usando distintas cadenas de búsqueda como “técnicas de clasificación, aprendizaje automático, detección de phishing” y palabras clave como “Machine Learning, Classification, Phishing, ciberseguridad”, obteniendo 8.360 artículos como resultados, sin embargo fueron omitidos aquellos trabajos donde sí fueron utilizados clasificadores de Machine Learning pero sin embargo el caso en que fue aplicado no estaba relacionado con el tema de investigación.

Tabla 18 *Algoritmos de Detección usados en casos similares con el tema de investigación*

N.º	Algoritmos de Detección	En qué caso fue aplicado	Desempeño	Autor
1	Rotation Forest (RFT)	Prevención del fraude al permitir la detección automática de sitios web maliciosos	97,28%	Ghatasheh (2016)
2	Support Vector Machine + SMO	Detección de ataques de inyección de SQL a nivel de base de datos	95,67%	Priyaa & Devi (2016)
3	Decision Trees	Detección de ataques de phishing en web	93%	Şanliöz, Kara, Aydin & Balik (2020)
4	Extreme Learning Machine (ELM)	Detección de sitios web de phishing según la clasificación de funciones ELM	96,93%	Kandi & Agarkar (2020)
5	Random Forest	Detección de URL maliciosas	98,40%	Pradeepthi & Kannan (2020)

6	Perceptrón Multicapa	Detección eficiente de sitios web de phishing	98,5%	Odeh & Keshta & Abdelfattah
7	J48	Detección y análisis de ataques phishing	91,3%	Cuzzocrea & Martinelli & Mercaldo (2018)
8	Naive Bayes	Análisis de texto y enfoque de aprendizaje automático para la detección de correo electrónico phishing	99,0%	Oladimeji (2019)
9	Extreme Gradient Boosted Tree (XGBOOST)	Análisis comparativo y detección de sitios web de Phishing	97,30%	Musa, Gita, Zambuk, Umar, Umar & Waziri (2019)
10	Perceptrón Multicapa	Clasificación multiclase de ataques a páginas web XSS	96,20%	Krishnaveni & Sathiyakumari (2013)

N. º	**Algoritmos de Detección**	**En qué caso fue aplicado**	**Desempeño**	**Autor**
11	Gradient Tree Boosting.	Detección automática de fraudes de clics	97.20%	Dash & Pal (2020)
12	Random Forest con funciones FLN	Clasificación de Sitios web de Phishing	87,98%	Kalayci (2018)
13	Red neuronal artificial (ANN)	Detección de páginas web maliciosas: un enfoque de aprendizaje automático	96.01%	Siragel din, Baharu din & Tang Jung (2020)
14	Modelo de Espacio Vectorial para Clasificación (VSMA)	Clasificación de ataques web	98%	Yadav, Satyan arayan a, Vasum athi (2016)

Nota. Fuente: Elaboración Propia.

Tomando como base las tablas anteriores en donde se encuentra los algoritmos de detección los cuales fueron usados en casos similares al caso de estudio, se optó por realizar un top de los 4 mejores algoritmos usando como referencia el desempeño que obtuvieron en cada uno de los casos en los que fueron propuestos, como resultado se obtuvo la siguiente tabla.

Tabla 19 *Top 4 de Algoritmos de Detección con mayor desempeño en casos similares*

N.º	Algoritmos de Detección	En qué caso fue aplicado	Desempeño	Autor
1	Naive Bayes	Análisis de texto y enfoque de aprendizaje automático para la detección de correo electrónico phishing	99,0%	Oladimeji (2019)
2	Random Forest	Detección de URL maliciosas	98,40%	Pradeepthi & Kannan (2020)
3	Perceptrón Multicapa	Detección eficiente de sitios web de phishing mediante	98,5%	Odeh & Keshta & Abdelfattah
4	Extreme Gradient Boosted Tree (XGBOOST)	Análisis comparativo y detección de sitios web de Phishing	97,30%	Musa, Gita, Zambuk, Umar, Umar & Waziri (2019)

Nota. Fuente: Elaboración Propia.

EL conjunto de datos presentado fue preparado con el fin de evaluar varios métodos de detección de ataques de phishing, el marco de automatización del navegador se emplea para mejorar el método de extracción de características de esta manera sería más preciso y sólido.

Las funciones de este conjunto de datos se clasificaron en tres grupos, que son la barra de direcciones, basados en las propiedades del dominio y basada en anomalías.

1. Las funciones basadas en la barra de direcciones son las características de la URL como el tráfico de red de la página.
2. Las funciones basadas en las propiedades de dominio URL en formato de dirección IP.
3. Las Funciones basadas en anomalías son características de acciones anormales en la web.

Para la evaluación de los modelos, existen muchas herramientas de evaluación. Pero en este caso se planteó evaluar el modelo usando la ecuación de precisión, por lo tanto, calcular las tasas de precisión fue suficiente y preciso. Para la aplicar la fórmula de precisión, se debe recalcar que existen dos tipos de métodos de detección de acuerdo con el número de clases que son la detección binaria y la detección multiclase, para este proyecto se utilizó la detección binaria ya que para esta solo se utilizan dos clases, P para la clase positiva y N para la clase negativa.

Figura 7 *Matriz de Confusión*

Verdaderos Positivos	Falsos Positivos
Falsos Negativos	Verdaderos Negativos

Nota*. Fuente: Elaboración Propia.*

1. Verdaderos Positivos (TP): es la tasa de predicción real de las muestras positivas. El valor predicho es positivo y el valor real también positivo.
2. Falso Positivo (FP): es el valor negativo clasificado incorrectamente como positivo.
3. Verdaderos Negativos (TN): es la verdadera predicción de muestras negativas. El valor predicho es negativo y el valor real también es negativo.
4. Falso Negativos (FN): es el valor positivo clasificado incorrectamente como negativo.

La precisión se refiere a la proporción de las instancias que fueron clasificadas de manera correcta. Se sabe que es la métrica de evaluación más utilizada para el desempeño de problemas de clasificación binaria. Además, está determinando la precisión del modelo, para calcular la precisión se utiliza la siguiente ecuación:

$$Pr = \frac{VP}{VP + FP}$$

Para la construcción del dataset se hizo un estudio a profundidad de la literatura, hay dos métodos con los cuales se pueden combatir los ataques phishing que son la lista negra y el método heurístico que a diferencia de la lista negra este enfoque es capaz de reconocer sitios web de phishing en tiempo real.

Existe una manera generalmente utilizada para saber cómo obtener los datos o qué tipo de características extraer de ellos.

Extraer y calcular automáticamente características de URL o sitios web de phishing luego extraer y aplicar algunas reglas heurísticas.

De esta manera las características que fueron extraídas pertenecen a estos grupos los cuales son las funciones basadas en dominios, a continuación, se muestran

algunas de las características más importante.

Tabla 20 *Funciones más importantes para detectar phishing*

Grupo	Características
Funciones que están basadas en propiedades del dominio	Edad del dominio Registro del DNS El tráfico web

Nota. Fuente: Elaboración Propia.

Para prepara el conjunto de datos lo que se hizo primero fue recopilar una lista de URL de phishing, de esas URL que se recopilaron se pudo extraer 21 características basadas en el dominio, estas características se extrajeron mediante el recuento de caracteres especiales como son los símbolos también se extrajeron características basadas en el tiempo de búsqueda del dominio de respuesta, el número de servidores, el valor del tiempo de vida que está asociada al host.

Tabla 21 *Lista de símbolos utilizados para extraer funciones basadas en DNS*

Símbolos		
signos "."	signos de "a"	Número total de signos "+"
Número total de signos "-"	signos "&"	Número total de signos "*"
signos "_"	signos "!"	Número total de signos de "a"
"?"	" "	Número total de signos "$"
"/"	Número total de "señales"	Número total de signos "%"
signos "$"	de signos "%	Número total de vocales
signos de "a"	Número total de signos ","	URL en formato de dirección IP

Nota. Fuente: Elaboración Propia.

3.3.2. Descripción del dataset o conjunto de datos

Los datos recopilados consisten en una colección de instancias de sitios web tanto legítimas como de phishing, este conjunto de datos ayudará a prevenir ataques phishing ya que el conjunto de datos presentado se basa en atributos que pueden ser extraídos fácilmente.

El conjunto de datos se recopiló y se preparó con el fin de evaluar algoritmos de detección.

Tabla 22 *Atributos basados en el Nombre del Dominio*

N. º	Atributo	Formato	Descripción
1	qyt_dot_domain	“.”	numérico
2	qyt_hyphen_domain	“-“	numérico
3	qyt_underline_domain	“_”	numérico
4	qyt_slash_domain	“/”	numérico
5	qyt_quiestionmark_domain	“?”	numérico
6	qyt_equal_domain	“=”	numérico
7	qyt_at_domain	“@”	numérico
8	qyt_and_domain	“&”	numérico
9	qyt_exclamation_domain	“!”	numérico
10	qyt_space_domain	“ ”	numérico
11	qyt_tilde_domain	“´”	numérico
12	qyt_comma_domain	“,”	numérico
13	qyt_plus_domain	“+”	numérico
14	qyt_asterisk_domain	“*”	numérico
15	qyt_hashtag_domain	“#”	numérico
16	qyt_dollar_domain	“$”	numérico
17	qyt_percent_domain	“%”	numérico
18	qyt_vowels_domain	Número de vocales	numérico
19	domain_length	Número de caracteres del dominio	numérico
20	domain_ip	Dominio en formato IP	booleano
21	server_client_domain	Servidor en el dominio	booleano

Nota. Fuente: Elaboración Propia.

Tabla 23 *Etiqueta de Clase de Dataset*

N. º	Atributo	Formato	Descripción
1	phishing	Si es un ataque phishing	Booleano

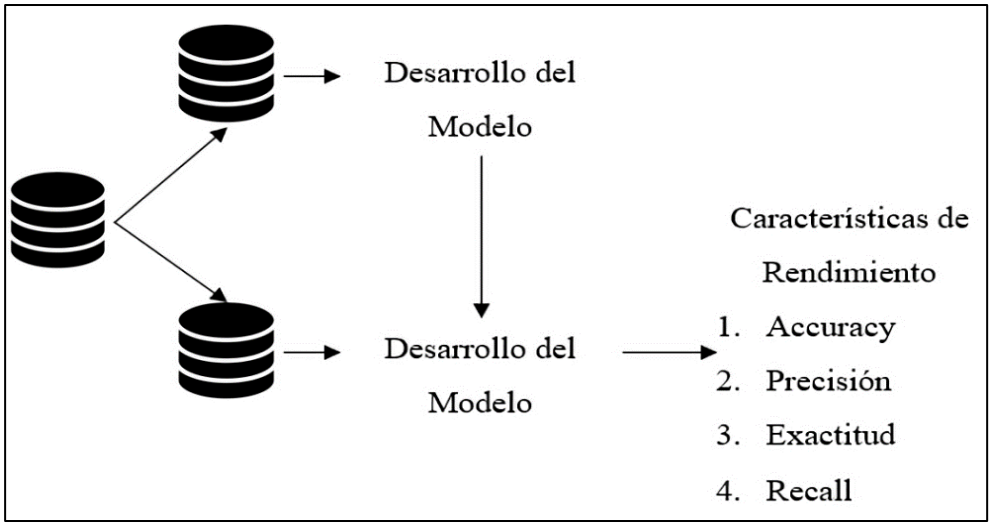
Nota. Fuente: Elaboración Propia.

3.3.3. Algoritmos seleccionados con mayor desempeño que serán probados en este proyecto:

3.3.3.1. Implementación de Naive Bayes

La detección tiene dos fases, una que es de aprendizaje y otra que es de evaluación, en la fase de aprendizaje el clasificador entrena el modelo en el conjunto de datos determinado y para la primera fase que es la de prueba para el rendimiento del detector. El rendimiento es evaluado en base a varios parámetros, como son el error, la precisión y el Recall.

Figura 8 *Parámetros de Evaluación del Rendimiento*



Nota. Fuente: Elaboración Propia.

Naive Bayes es una de los algoritmos de detección que se encuentra basada en el Teorema de Bayes. Es un algoritmo de Machine Learning supervisado más simple, el clasificador Naive Bayes es preciso y confiable ya que tiene alta velocidad y precisión con grandes conjuntos de datos.

Este clasificador supone que el efecto de una entidad determinada en una clase es independiente de otras entidades, esta suposición simplifica el cálculo y es por eso que se considera ingenua.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$P(h)$ es la probabilidad de que la hipótesis h sea verdadera.

P(D) que viene a ser la probabilidad del dato que se conoce como probabilidad previa.

P(h | D) la probabilidad de hipótesis h dados los datos D que se conoce como probabilidad posterior

P(h | D) es la probabilidad de los datos dado que la hipótesis h era verdadera.

Figura 9 *Código Naive Bayes utilizado*

```
#importamos
import pandas as pd
# importar el conjunto de datos! Usaremos
#la función de nombres de Pandas para asegurarnos de que
#los nombres de columna asociados con los datos lleguen a través.
df = pd.read_csv( "dataset_full.csv" ,sep = ",")
df.head()
# a preparar nuestro conjunto de entrenamiento en las siguientes líneas.
#X contendrá conjuntos de características y y contendrá etiquetas de cada fila
X = df.iloc[:, 2:].values
y = df.iloc[:, 1].values
# codificar etiquetas de y
#para el propósito de entrenamiento
y
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
y
```

Nota. Fuente: Elaboración Propia.

Figura 10 *División de datos para entrenamiento y prueba*

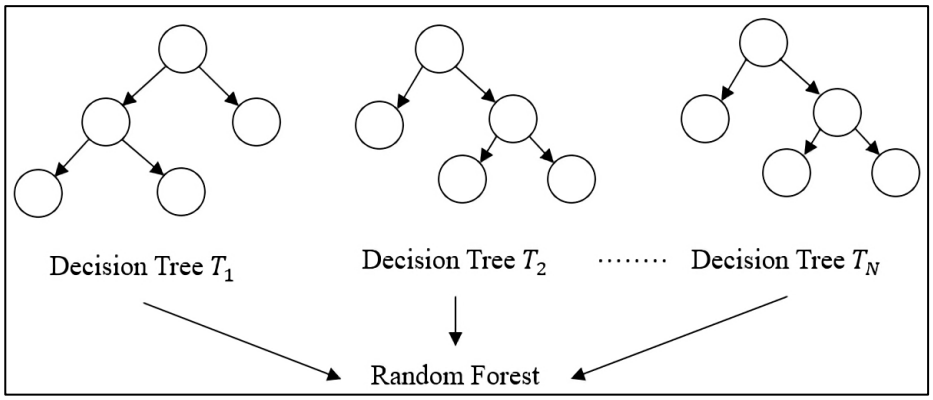
```
#En el siguiente segmento se divide el conjunto de datos
#en conjunto de entrenamiento y prueba con relación 80:20
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20,
random_state=1)
#Y no se olvide de estandarizar sus conjuntos de características
from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()
X_train_std = stdsc.fit_transform(X_train)
X_test_std = stdsc.transform(X_test)
#Así que aquí estamos. Es hora de ajustar nuestro estimador
#con los datos de entrenamiento.
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(X_train_std, y_train)
GaussianNB(priors=None)
#y_pred contiene la etiqueta pronosticada del conjunto de pruebas.
y_pred = clf.predict(X_test_std)
#Finalmente es hora de ver la exactitud de nuestro estimador.
predictions = clf.predict(X_test_std)
print(accuracy_score(y_test, predictions))
#Matriz de confusion
print(confusion_matrix(y_test, predictions))
```

Nota. Fuente: Elaboración Propia.

3.3.3.2. Implementación de Random Forest

Random Forest es un método para clasificación y detección que está basado en el algoritmo de árboles de decisiones. Es apropiado para conjuntos de datos enormes ya que puede abarcar un número considerable de variables en el conjunto de datos. En la fase para el entrenamiento, se construyen diferentes grupos de árboles de decisión, donde cada árbol se ejecuta en un conjunto de atributos predefinidos los cuales son seleccionados al azar. El proceso de clasificación es realizado por la mayoría de votos con los resultados que fueron obtenidos de cada árbol. Random Forest es entrenado en varias partes del conjunto de datos para entrenamiento, una característica de usar este algoritmo es que resolvió el problema de sobreajuste que ocurre de manera común cuando se usan árboles de decisión individuales. Sin embargo, el proceso de reproductibilidad está ausente porque la construcción del bosque es aleatoria.

Figura 11 *Representación del Algoritmo Random Forest*



Nota. Fuente: Elaboración Propia.

Figura 12 *Código Random Forest utilizado*

```
# Import needed packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Si trabajas en el Cuaderno Jupyter, incluye lo siguiente para que se
muestren los gráficos:

%matplotlib inline

#Carga de datos

import pandas as pd

# abrir archivo pd.read_csv

df = pd.read_csv( "dataset_full.csv" )

print(df.shape)

# print head of data set

print(df.head())
```

Nota. Fuente: Elaboración Propia.

Figura 13 *Código Random Forest implementando la librería Scikit-Learn*

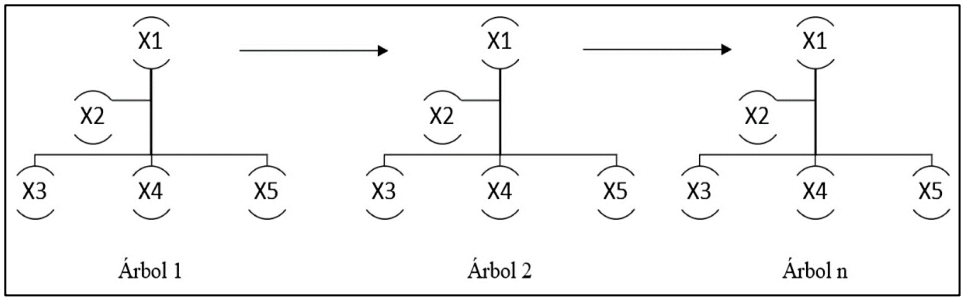
```
# Dividir el conjunto de datos en características y objetivos

y = df['server_client_domain']

X = df.drop('server_client_domain', axis=1)

# Ver el recuento de cada clase

y.value_counts()

# Dividir las características y el objetivo en entrenamiento y conjuntos de prueba

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, stratify=y)

# Instanciar y ajustar el RandomForestClassifier

forest = RandomForestClassifier()

forest.fit(X_train, y_train)

# Hacer predicciones para el conjunto de pruebas

y_pred_test = forest.predict(X_test)

# Ver la puntuación de la precisión

accuracy_score(y_test, y_pred_test)

# Ver matriz de confusión para datos de prueba y predicciones

confusion_matrix(y_test, y_pred_test)

# Ver el informe de clasificación para los datos de las pruebas y las predicciones

print(classification_report(y_test, y_pred_test))
```

Nota. Fuente: Elaboración Propia.

3.3.3.3. Clasificador Xgboost

El clasificador Extreme Gradient Boosted Tree (XGBoost) es una librería de impulso de gradiente optimizado, fue diseñado para ser muy flexible, portátil y eficiente. Principalmente se emplea en tareas de clasificación y detección donde se usa como clasificador para mapeo. XGBoost tiene muchos puntos fuertes en comparación con las implementaciones tradicionales de aumento de gradiente. Entre sus puntos fuertes se encuentran una mejor capacidad de regularización que ayuda a reducir el sobreajuste, la alta velocidad y rendimiento debido a la naturaleza paralela en la que se construyen los árboles, la flexibilidad debido a sus objetivos de optimización, criterios de evaluación y rutinas incorporadas para mejorar los valores faltantes. Estas y muchas otras ventajas de XGboost lo han convertido en una excelente herramienta de elección para muchos investigadores en la ciencia de datos y aprendizaje automático.

Figura 14 *Representación del Algoritmo Xgboost*



Nota. Fuente: Elaboración Propia.

Funcionamiento del algoritmo XGboost:

a) Inicialmente se obtiene un árbolF_0 con este se predice la variable objetivo y, finalmente el resultado se junta con un residuo $(y - F_{0)}$.

b) Luego se obtendrá un árbol nuevo llamado h_1 este será el que ajuste el error del paso previo.

c) Tanto los resultados de F_0 como deh_1 se combinarán para obtener un árbol F_1 , en donde el error cuadrático medio de F_1 deberá ser menor al de F_0.

Figura 15 *Código empleado para el algoritmo XGBOOST*

```
# importer librerias
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.metrics
  import mean_squared_error
# importar el conjunto de datos y almacenarlo en una
variable llamada df

df = pd.read_csv( "dataset_full.csv",sep=",")
df.head()
#Separacion de la variable de destino y el resto de las variables para crear un
subconjunto de los datos.iloc
X, y = df.iloc[:,: -1 ],df.iloc[:,-1]
#Ahora convertirá el conjunto de datos en una estructura de datos optimizada
llamada que XGBoost
#admite y le da aclamadas mejoras de rendimiento y eficiencia
                    data_dmatrix = xgb.DMatrix(data=X,label=y)
#creará los datos para entrenamiento y pruebas para la validación cruzada de
los resultados
#utilizando la función del módulo de sklearn con un tamaño igual al 20% de
los datos
from sklearn.model_selection
import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=123)
```

Nota. Fuente: Elaboración Propia.

Figura 16 *Implementación de librería la Scikit-Learn*

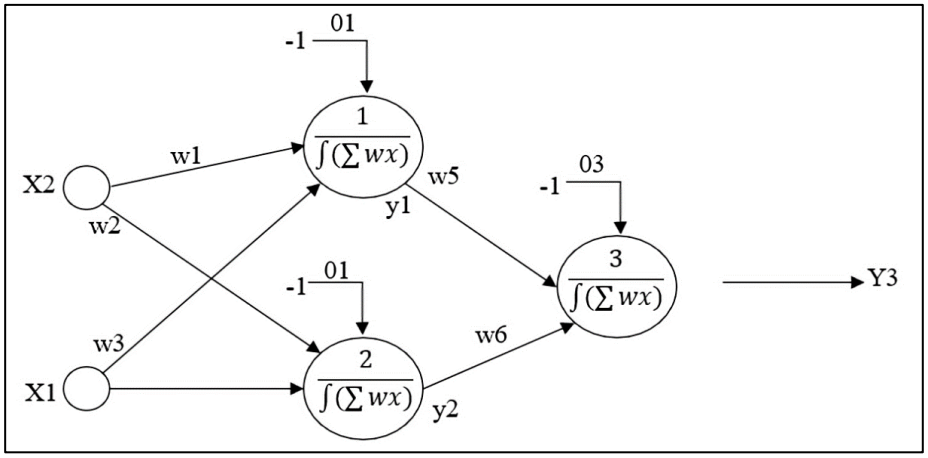
```
#el siguiente paso es crear una instancia de un objeto regressor XGBoost

#llamando a la clase desde la biblioteca XGBoost con los hiperparámetros pasados como argumentos

xg_reg = xgb.XGBClassifier(objective = 'reg:linear',

colsample_bytree = 0.3, learning_rate = 0.1,max_depth = 5, alpha = 10,

n_estimators = 10)

#Ajuste el regresor al conjunto de entrenamiento y realice predicciones en el conjunto de pruebas

xg_reg.fit(X_train,y_train)

preds = xg_reg.predict(X_test)

#Calcular el rmse invocando la función desde el módulo de sklearn

rmse = np.sqrt(mean_squared_error(y_test, preds))

print( "RMSE: %f" % (rmse))

#k-fold Cross Validation usando XGBoost

params = { "objective":"reg:linear",'colsample_bytree': 0.3,'learning_rate': 0.1,

            'max_depth': 5, 'alpha': 10}

cv_results = xgb.cv(dtrain=data_dmatrix, params=params, nfold=3,

                num_boost_round=50,early_stopping_rounds=10,metrics="rmse", as_pandas=True, seed=123)
#contiene métricas de RMSE de entrenamiento y prueba para cada ronda de impulso

cv_results.head()

#Extraccion e impresión de la métrica final de la ronda de impulso.

print((cv_results[ "test-rmse-mean"]).tail(1))

predictions = xg_reg.predict(X_test)

#ver el reporte de clasificacion del modelo

from sklearn.metrics import classification_report

print("Reporte de  clasificacion of Model:"
,classification_report(y_test,predictions))
```

Nota. Fuente: Elaboración Propia.

3.3.3.4. Clasificador Perceptrón Multicapa

El Perceptrón Multicapa (MLP) es la red neuronal artificial más popular y más utilizada. Al igual que una red neuronal, el MLP consiste en componentes multi - interconectados. Están construidos de tres capas diferentes que son una capa de entrada, oculta y la capa de salida, cada una tiene su propia funcionalidad, una capa de entrada se utiliza para obtener la señal, una capa de salida resulta una decisión sobre la entrada, y hay en al menos una capa oculta que es el motor computacional del MLP. Se suele utilizar para problemas de aprendizaje supervisado: se entrena en un grupo de pares de entrada-salida y aprende la correlación y las dependencias entre ellas.

Figura 17 *Estructura para Perceptrón Multicapa*



Nota. Fuente: Elaboración Propia.

Donde:

X = Entradas$y^1 = \int((x1 * w1) + (x2 * w3) - 01))$

W = Pesos$y^2 = \int((x1 * w2) + (x2 * w4) - 02))$

0 = Umbral $y^3 = \int((y1 * w5) + (y2 * w6) - 03))$

A continuación, se muestra un algoritmo para Perceptrón Multicapa

Figura 18 *Importación y visualización de datos*

```
#importars las librerias
import numpy as np ; import pandas as pd ; import pylab as pl
from matplotlib import pyplot as plt
#Primero se importa el conjunto de datos! usando
#la función de nombres de Pandas para asegurarnos de que
#los nombres de columna asociados con los datos lleguen a través.
df = pd.read_csv( "dataset_full.csv",sep =",")
df.head()
#Echemos un vistazo a los datos:
df.describe().transpose()
#Vamos a configurar nuestros datos y nuestras etiquetas:
X = df.drop('phishing',axis=1)
y = df['phishing']
```

Nota. Fuente: Elaboración Propia.

Figura 19 *Importación y visualización de datos*

```
#División de entrenamiento y de prueba
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)
#Preprocesamiento de datos
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
#Sólo se ajusta a los datos de entrenamiento
scaler.fit(X_train)
#Ahora aplica las transformaciones a los datos:
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
#Entrenamiento del modelo
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes= (13,13,13),max_iter=500)
mlp.fit(X_train,y_train)
```

Nota: Código para importación y visualización de los datos

Figura 20 *Evaluación de rendimiento del modelo*

```
#Predicciones y evaluación
predictions = mlp.predict(X_test)
# usar las métricas integradas de SciKit-Learn,
#como un informe de clasificación y una matriz de confusión
para evaluar el #rendimiento de nuestro modelo:
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
```

Nota. Fuente: Elaboración Propia.

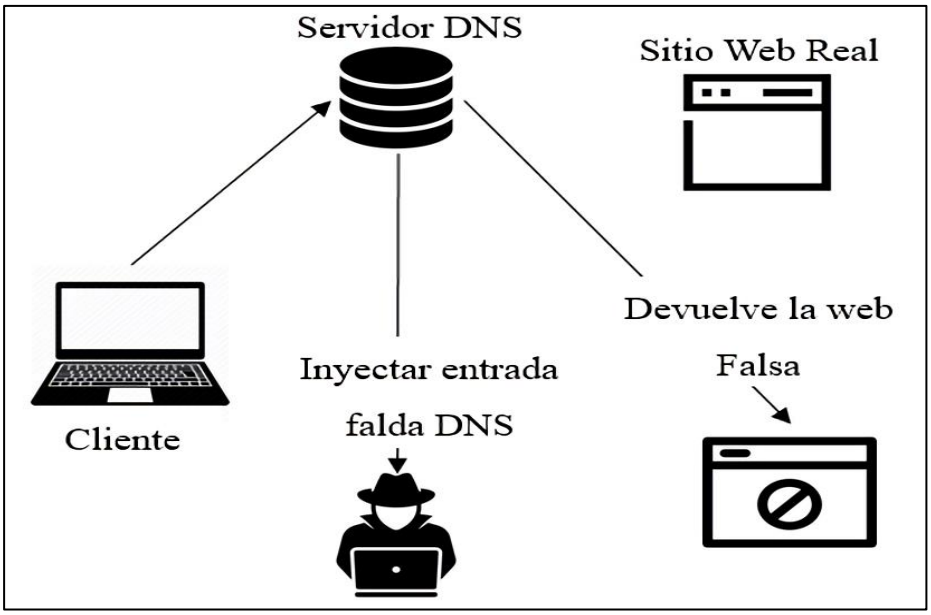
3.3.3.5. Envenenamiento del servidor DNS

El envenenamiento de servidor DNS básicamente hace referencia a los ataques del servidor DNS, estos ataques explotan las vulnerabilidades que se encuentran integradas desde el principio y busca recopilar datos sensibles de los usuarios.

Cuando el navegador o una aplicación web entra en internet, se inicia haciendo petición a un servidor para DNS que busque la dirección para un nombre en específico, el servidor DNS se conectará a los servidores raíz que poseen ese dominio y pedirá la dirección del servidor de nombres autorizado de ese dominio.

El envenenamiento de DNS sucede cuando un atacante interviene en ese proceso y proporciona una respuesta incorrecta.

Figura 21 *Envenenamiento del servidor DNS*



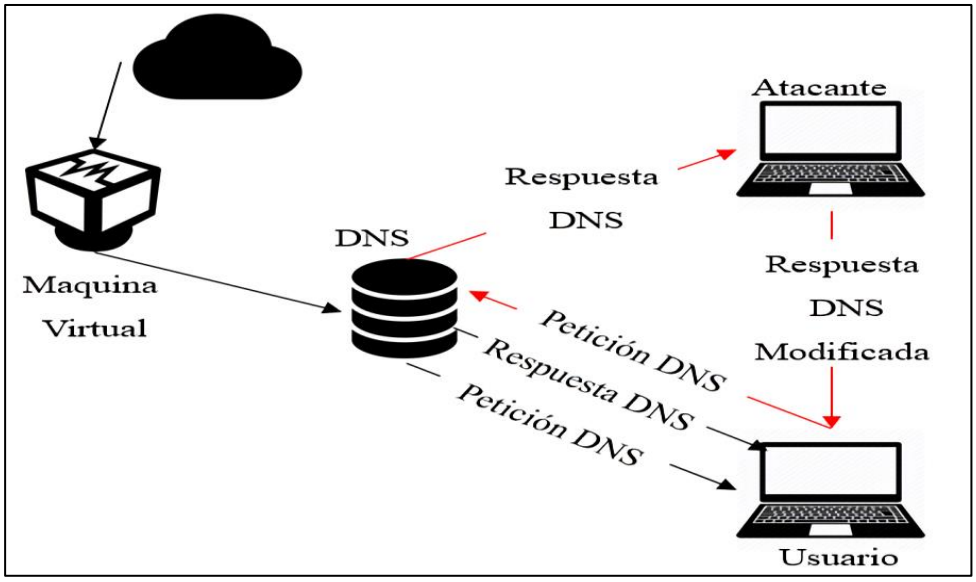
Nota. Fuente: Elaboración Propia.

Como en toda investigación siempre hay una fase de experimentación, para este caso se preparó un escenario de pruebas en donde se simuló un ataque basado en la realidad y las características del caso de estudio.

Debido a que no se cuenta con los recursos necesarios para contar con un servidor DNS propio en el caso de estudio se optó por crear un servidor local en nube el cual proporciona muchas ventajas como la facilidad de implementación que es inmediata de este modo no se requiere de soporte, es escalable ya que se ve reflejado en los planes del servicio, los gastos de infraestructura son reducidos a un plan mensual, es más seguro ya que soporta la encriptación de datos y generan copias de seguridad automáticas por lo que perder datos importantes prácticamente sería imposible.

De esta forma se realizó el escenario de pruebas quedando graficado de la siguiente manera:

Figura 22 *Escenario de Prueba*



Nota. Fuente: Elaboración Propia.

3.3.4. Características del escenario de pruebas:

El escenario propuesto cuenta con un servidor DNS que fue instalado y configurado en una máquina virtual alojada en la nube, este servidor utiliza el sistema operativo Windows Server 2016 versión Datacenter.

Para el usuario se utilizó una laptop con sistema operativo Windows 10 Home 64, contando con un procesador Intel Core i3 (2.3 GHz, 3 MB cache, 2 cores), una memoria RAM de 4gb, un disco duro sata de 1 tb y conexión a internet cableada desde ahí se intentó acceder a la aplicación web del caso de estudio.

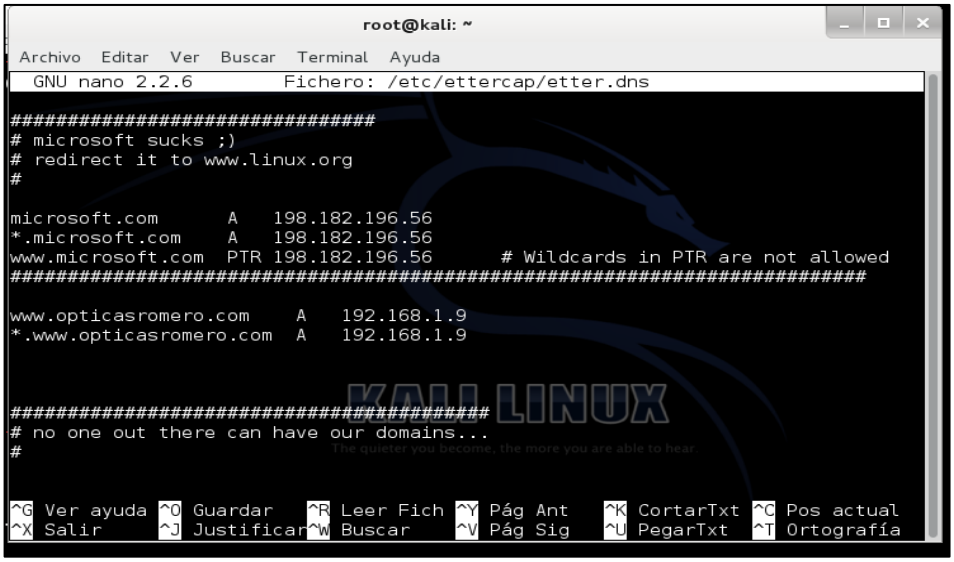
La máquina desde donde se generó el ataque de envenenamiento DNS fue una laptop con el sistema operativo Kali Linux versión 2016, que es usado para realizar pruebas de seguridad, un procesador Intel Core i5 (c/TB hasta 2.70 GHz), memoria RAM de 4gb, un disco duro de 500 gb, y conexión a internet via wifi.

Se utilizó el sniffer/ registrador Ettercap cuya función más destacada en la de inyección para una conexión que este establecida de esta manera emula comandos siempre y cuando la conexión este activa, de la misma manera también intercepta el tráfico y crea ataques en este caso el envenenamiento de DNS al host de la víctima.

3.3.4.1. Descripción de pruebas realizadas:

Las pruebas de ataque realizadas se hicieron en un escenario virtual con Kali Linux, en la máquina del atacante se configuró el fichero “ettert.dns”, es allí donde se modificó el sitio que será redirigido al sitio clonado.

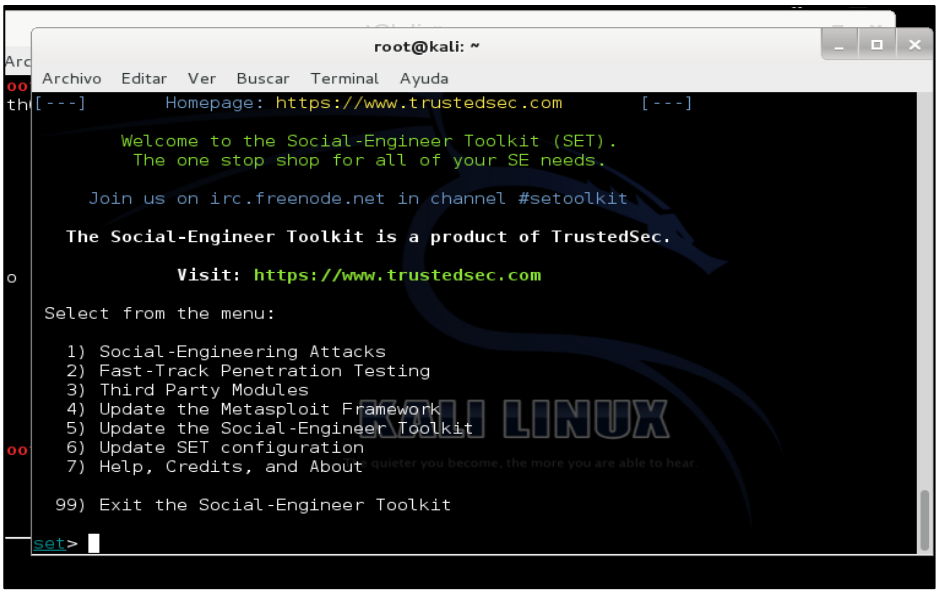
Figura 23 *Configuración del fichero etter.dns*



Nota. Fuente: Elaboración Propia.

Se procedió a clonar la aplicación web con las herramientas de “setoolkit”, ejecutando el comando ‘setoolkit’, en el menú se elige la opción 1 que es ataque de ingeniería social.

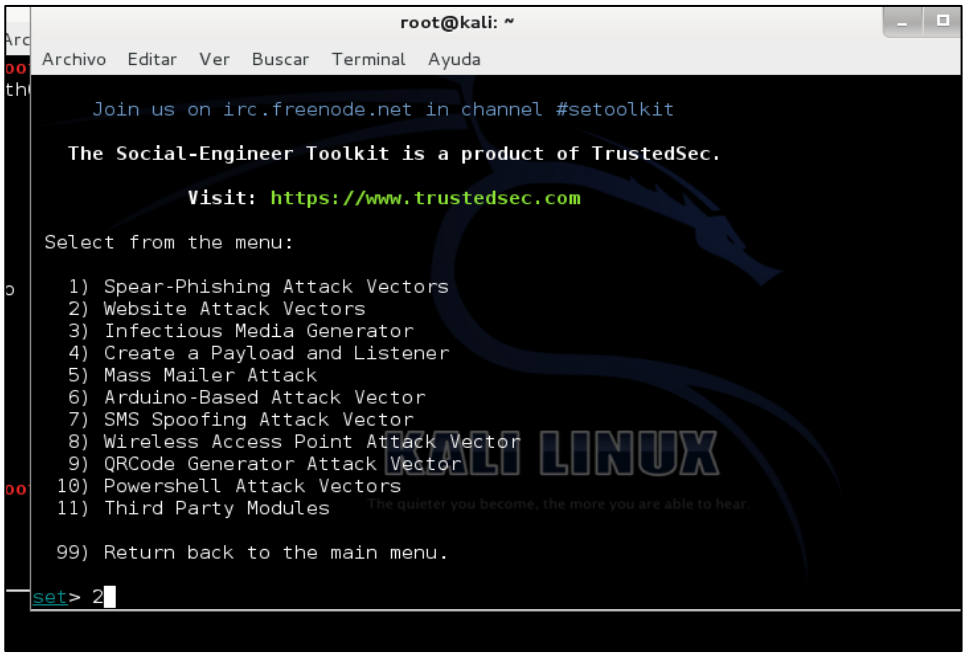
Figura 24 *Menú de Herramientas setoolkit*



Nota. Fuente: Elaboración Propia.

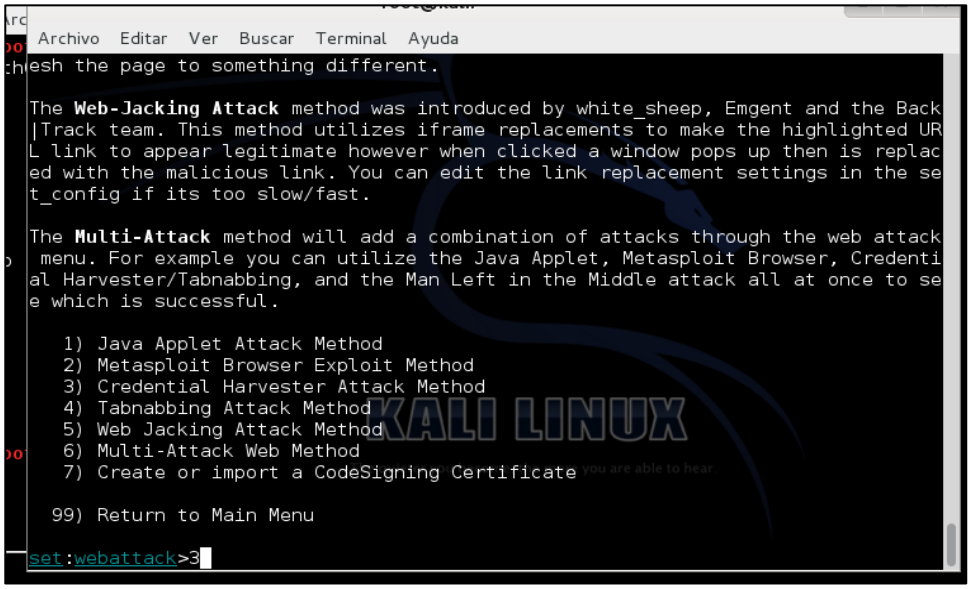
En esta configuración se nos lleva por 3 menús de opciones más, en el menú siguiente se eligió la opción 2 ‘vectores de ataque a sitios web’ Figura 25, en el siguiente menú se optó por la opción 3 ‘Método de Ataque del Cosechador de Credenciales’ Figura 26, finalmente en el tercer y último menú se eligió la opción 2 que permite clonar sitios web Figura 27.

Figura 25 *Vectores de ataques de sitios web*



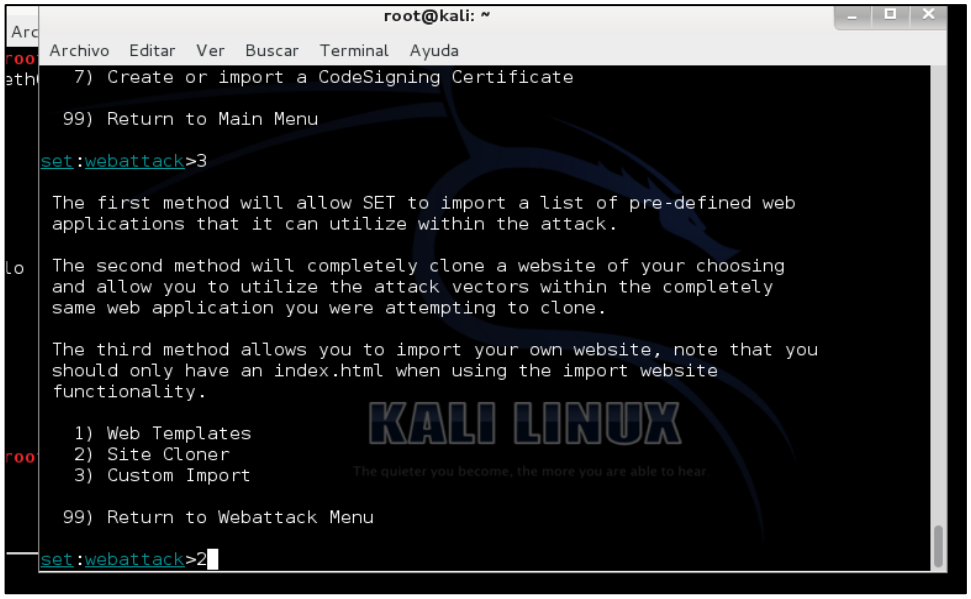
Nota. Fuente: Elaboración Propia.

Figura 26 *Opción 3 Método de Ataque del Cosechador de Credenciales*



Nota. Fuente: Elaboración Propia.

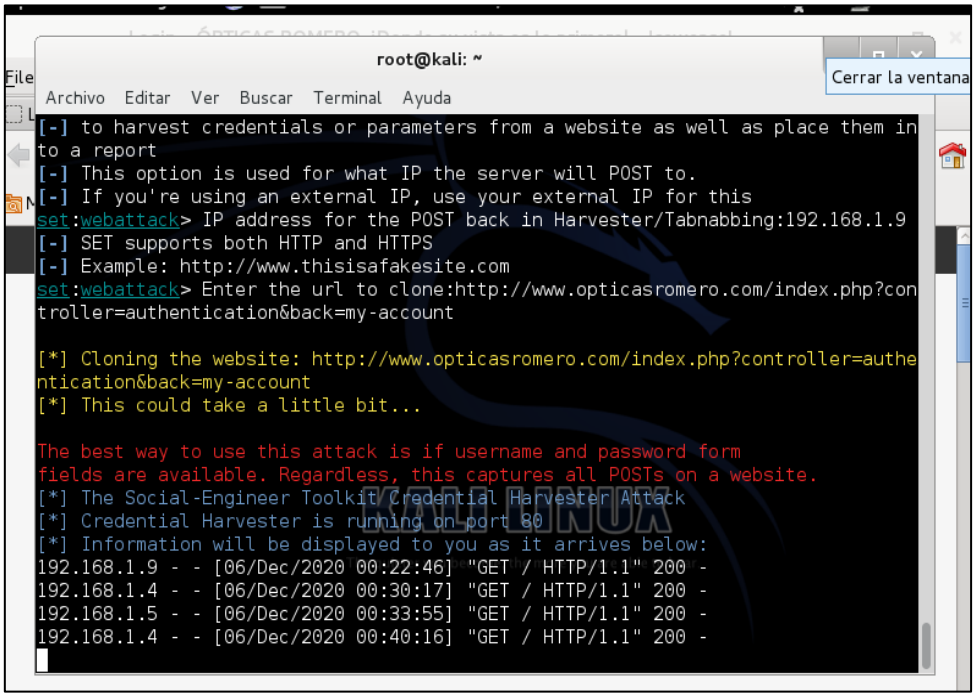
Figura 27 *Opción 2 Clonación de Sitio Web*



Nota. Fuente: Elaboración Propia.

Para realizar la clonación de la aplicación web, luego de escoger la opción 2 ‘site cloner’ Figura 27, pedirá la dirección IP de la interfaz que es atacante para este caso de prueba fue la dirección IP 192.168.1.9, luego nos pedirá la URL del sitio web que será clonado http://www.opticasromero.com/index.php?controller=authentication&back=my-account que es la URL de la aplicación web del caso de estudio, las configuraciones se aprecian en la Figura 28, también se puede observar las direcciones IP de las victimas que ingresaron a la ampliación web falsa Figura 28.

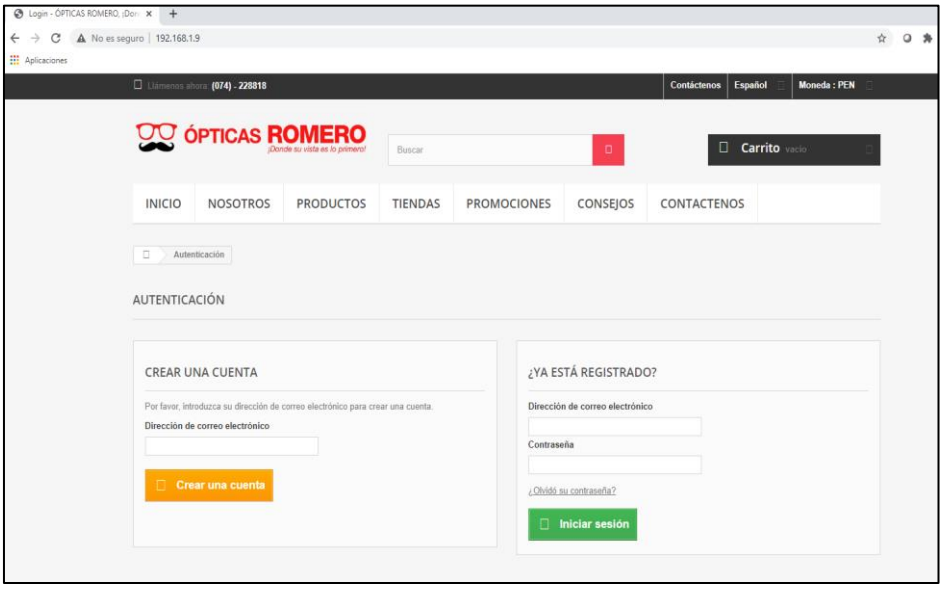
Figura 28 *Clonación de la aplicación web del caso de estudio*



Nota. Fuente: Elaboración Propia.

Haciendo pruebas para comprobar de que en realidad funcionó la clonación de la ampliación web se ingresó al navegador la dirección IP de la máquina atacante que es la 192.168.1.9 es en esa dirección la que se configuró como IP para la aplicación web clonada, se pudo observar que en efecto si se clonó la aplicación web del caso de estudio Figura 29.

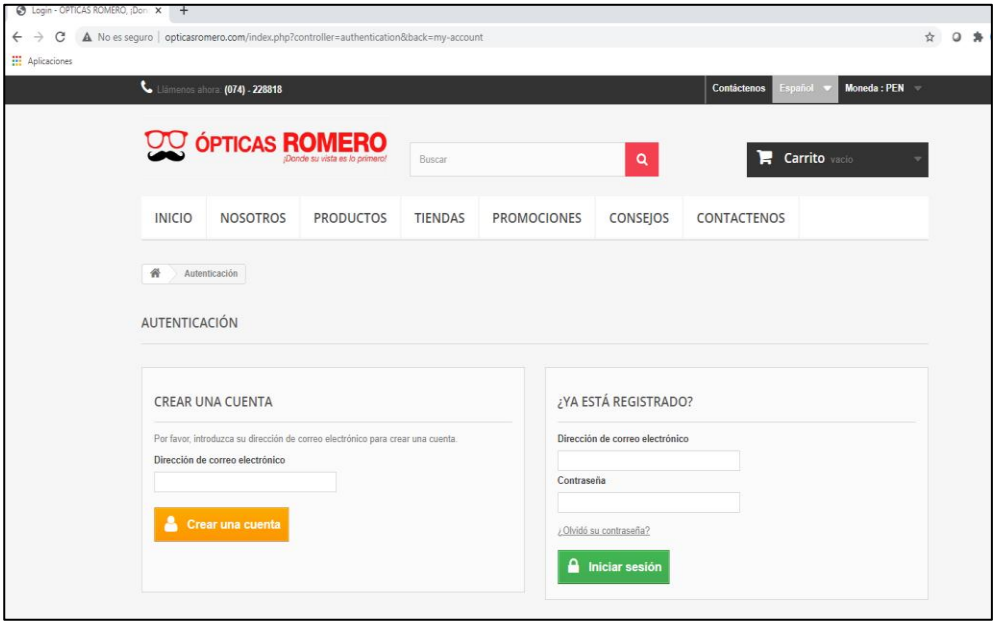
Figura 29 *Aplicación web Clonada*



Nota. Fuente: Elaboración Propia.

Se pudo comprobar porque se ingresó con la dirección IP del atacante y no con la dirección URL original Figura 29.

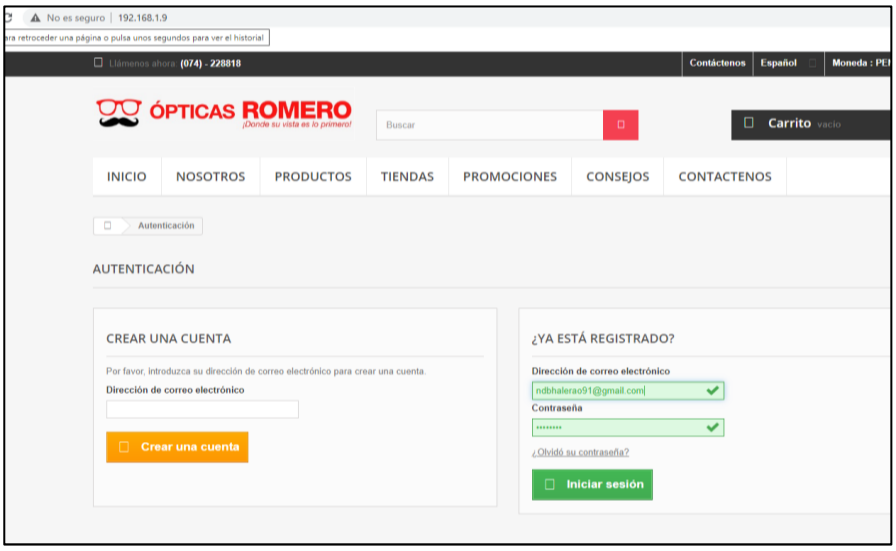
Figura 30 *Aplicación web Original*



Nota. Fuente: Elaboración Propia.

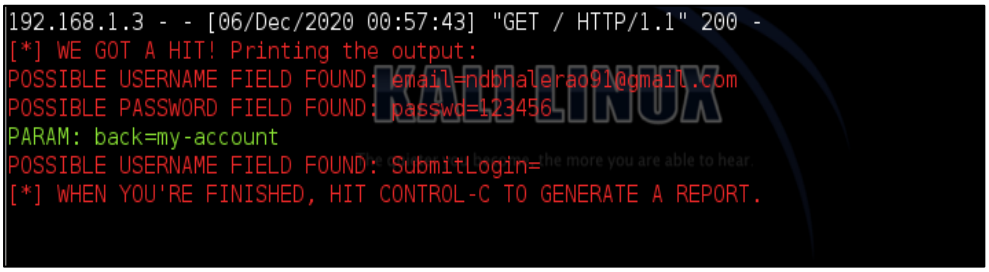
Se realizaron las pruebas de captura de datos, en la Figura 30 se observa la entrada de los datos del usuario de prueba, en la Figura 31 se observa que en efecto los datos de inicio de sesión del usuario fueron capturados.

Figura 31 *Aplicación web Original*



Nota. Fuente: Elaboración Propia.

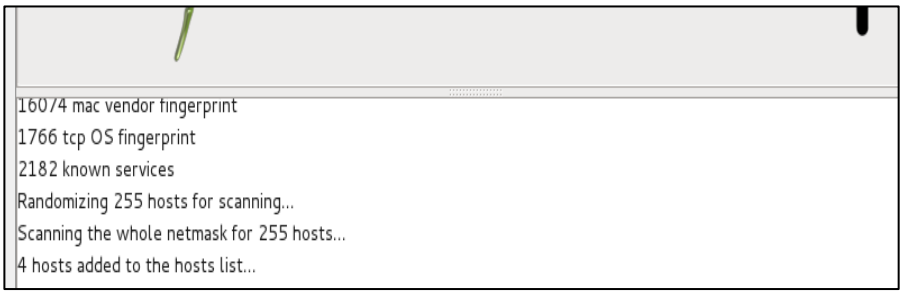
Figura 32 *Datos de Inicio de Sesión capturados*



Nota. *Fuente: Elaboración Propia.*

La herramienta sniffer Ettercap fue utilizado para realizar el envenenamiento al servidor DNS de la máquina de la víctima, se hicieron las configuraciones necesarias para realizar el ataque y de esta manera cuando un usuario víctima se conecte al servidor de solicitado, primero le cargará la página web clonada pero esta vez ya no se visualizará la IP del atacante sino que se verá la URL original de este modo el usuario no sospecharía nada y los datos que ingrese podrán ser robados y enviados a la máquina atacante.

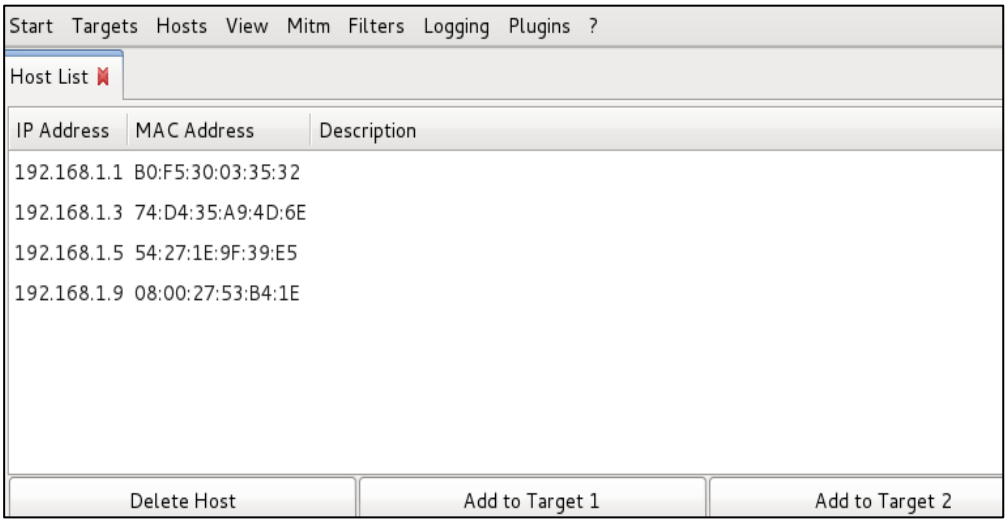
Figura 33 *Scaneo de Host*



Nota. Fuente: Elaboración Propia.

Se hizo el listado de los hosts encontrados Figura 34 y en efecto se encuentra el host de máquina a la que se atacará que fue la 192.168.1.5.

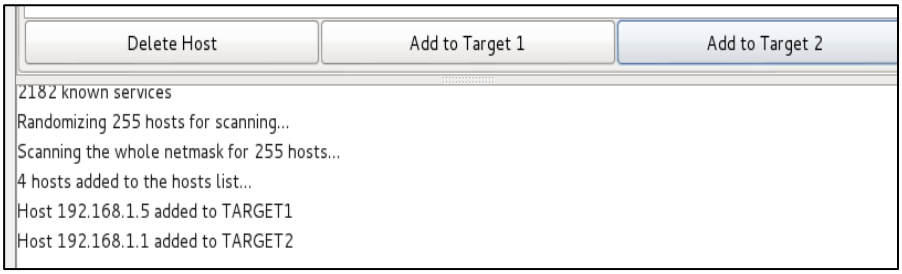
Figura 34 *Listado de Host encontrados*



Nota. Fuente: Elaboración Propia.

La puerta de enlace la cual es la dirección IP 192.168.1.1 se agrega al Target 2 y la dirección IP de la víctima 192.168.5 se agrega al target 2, la configuración se visualiza en la Figura 35.

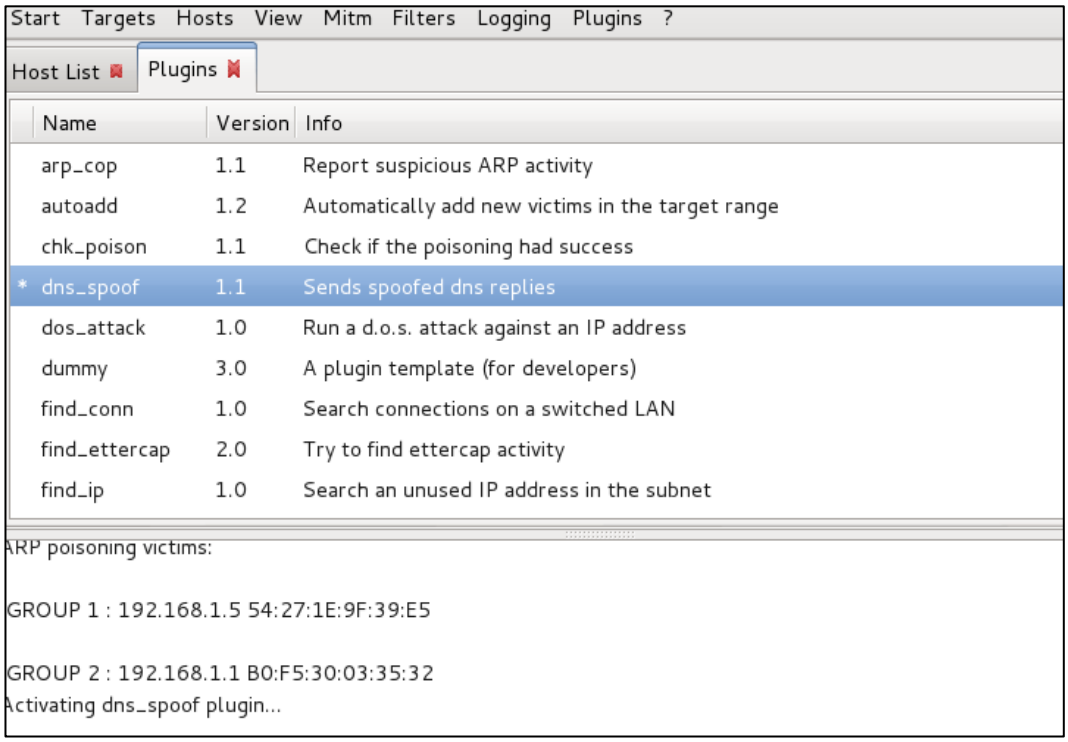
Figura 35 *Configuración de Target*



Nota. Fuente: Elaboración Propia.

Después de realizadas las configuraciones se procedió a elegir el tipo de taque que se realizara que fue el Spoofing DNS o envenenamiento DNS Figura 36.

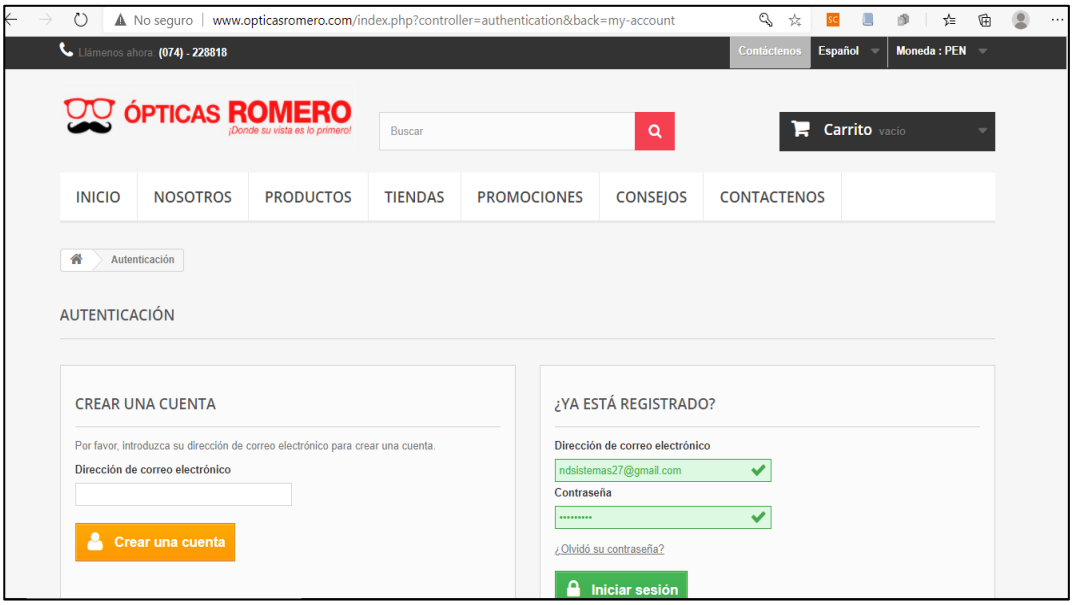
Figura 36 *Selección del ataque de envenenamiento DNS*



Nota. Fuente: Elaboración Propia.

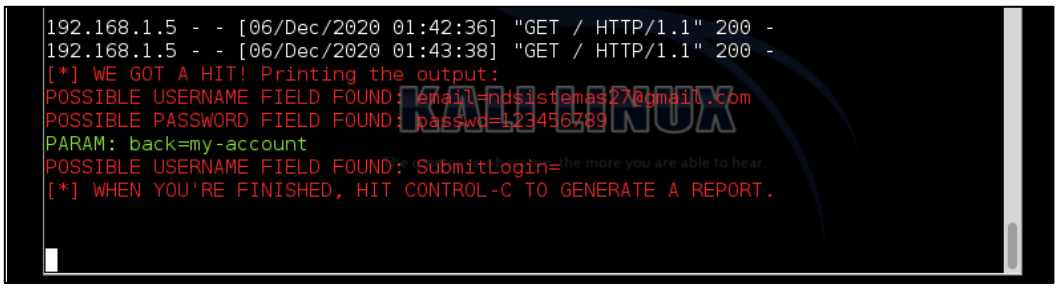
Ahora al ingresar a la aplicación web primero nos redireccionará a la aplicación web clonada el usuario víctima no podrá darse cuenta ya que cargara con la URL original Figura 37, en la Figura 38 se nota que las credenciales fueron extraídas.

Figura 37 *Aplicación web Falsa*



Nota. Fuente: Elaboración Propia.

Figura 38 *Credenciales extraídas de usuarios*



Nota. Fuente: Elaboración Propia.

Se capturó el tráfico de red con el analizador de protocolos Wireshark, de esta forma se pudo obtener características para crear el dataset y realizar clasificación de phishing por envenenamiento del servidor DNS.

3.3.5. Detección con el algoritmo de Naive Bayes

Figura 39 *Código utilizado para la Detección*

```
#importamos
 import pandas as pd
 #Primero se debe importar el conjunto de datos!
 #con la función de nombres de Pandas para asegurarse de que
 #los nombres de columna asociados con los datos lleguen a través.
 df = pd.read_csv("C:\\Users\\kevin\\kevin\\dataset_pruebas1.csv", sep=",")
 df.head()
 #X contendrá conjuntos de características y contendrá etiquetas de cada fila
 X = df.iloc[:, 2:].values
 y = df.iloc[:, 1].values
 #Después de eso tenemos que codificar etiquetas de y
 #para nuestro propósito de entrenamiento
 y
 from sklearn.preprocessing import LabelEncoder
 le = LabelEncoder()
 y = le.fit_transform(y)
```

Nota. Fuente: Elaboración Propia.

Figura 40 *División de datos para entrenamiento y pruebas*

```
#En el siguiente segmento se divide el conjunto de datos
#en conjunto para entrenamiento y prueba con relación 80:20
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20, random_state=1)
#Y no se olvide de estandarizar sus conjuntos de características
from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()
X_train_std = stdsc.fit_transform(X_train)
X_test_std = stdsc.transform(X_test)
from sklearn.metrics import confusion_matrix
#Matriz de confusion
print(confusion_matrix(y_test, predictions))
#Generara el reporte de clasificación
from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```

Nota. Fuente: Elaboración Propia.

3.3.6. Detección con el algoritmo de Random Forest

Figura 41 *Código Random Forest*

```
# Importar paquetes necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Nota. Fuente: Elaboración Propia.

Figura 42 *Código Random Forest*

```
#Carga de datos
import pandas as pd
# abrir acrchivo pd.read_csv
df = pd.read_csv("dataset_pruebas.csv")
print(df.shape)
# impresión del conjunto de datos
print(df.head())
# Dividir el conjunto de datos en características y objetivos
y = df['server_client_domain']
X = df.drop('server_client_domain', axis=1)
# Ver el recuento de cada clase
y.value_counts()
```

Nota. Fuente: Elaboración Propia.

Figura 43 *División de conjunto de datos*

```
# Dividir las características y objetivos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, stratify=y)
# RandomForestClassifier
forest = RandomForestClassifier()
forest.fit(X_train, y_train)
# Hacer predicciones para el conjunto de pruebas
y_pred_test = forest.predict(X_test)
# Ver matriz de confusión para datos de prueba y predicciones
confusion_matrix(y_test, y_pred_test)
# Vea el informe de clasificación para los datos de las pruebas y predicciones
print(classification_report(y_test, y_pred_test))
```

Nota. Fuente: Elaboración Propia.

3.3.7. Detección con el algoritmo de XGBoost

Figura 44 *Código XGBoost*

```
# Importar paquetes o librerias
import pandas as pd
import numpy as np
import xgboost as xgb
df = pd.read_csv("dataset_pruebas1.csv",sep=",")
df.head()
import xgboost as xgb
from sklearn.metrics import mean_squared_error
import pandas as pd
import numpy as np
X, y = df.iloc[:,:-1],df.iloc[:,-1]
#convertir el conjunto de datos en una estructura de datos optimizada llamada XGBoost
#admite y le da aclamadas mejoras de rendimiento y eficiencia
data_dmatrix = xgb.DMatrix(data=X,label=y)
# crear conjunto de entrenamiento y pruebas para la validación cruzada de los resultados
#utilizando la función del módulo de sklearn con un tamaño igual al 20% de los datos
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
#l siguiente paso es crear una instancia de un objeto regressor XGBoost
#llamando a la clase desde la biblioteca XGBoost con los hiperparámetros pasados como argumentos
xg_reg = xgb.XGBClassifier(objective ='reg:linear', colsample_bytree = 0.3, learning_rate = 0.1,max_depth = 5, alpha = 10, n_estimators = 10)
```

Nota. Fuente: Elaboración Propia.

Figura 45 *División de conjunto de datos*

```
#Ajuste el regresor al conjunto de entrenamiento y realice predicciones en el conjunto de pruebas

xg_reg.fit(X_train,y_train)

preds = xg_reg.predict(X_test)

#Calcular el rmse invocando la función desde el módulo de sklearn

rmse = np.sqrt(mean_squared_error(y_test, preds))

print("RMSE: %f" % (rmse))

#k-fold Cross Validation usando XGBoost

params = {"objective":"reg:linear",'colsample_bytree': 0.3,'learning_rate': 0.1,

                'max_depth': 5, 'alpha': 10}

cv_results = xgb.cv(dtrain=data_dmatrix, params=params, nfold=3,

                    num_boost_round=50,early_stopping_rounds=10,metrics="rmse",
as_pandas=True, seed=123)

# Ver matriz de confusión para datos de prueba y entrenamiento

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

confusion_matrix(y_test, predictions)

from sklearn.metrics import classification_report

# Vea el informe de clasificación para los datos de las pruebas y predicciones

print(classification_report(y_test,predictions))
```

Nota. Fuente: Elaboración Propia.

3.3.8. Detección con el algoritmo de Perceptrón Multicapa

Figura 46 *Código utilizado para la detección*

```
# Importar Librerías
%matplotlib inline
import numpy as np
import pandas as pd
import pylab as pl
from matplotlib import pyplot as plt
#Primero se va a importar el conjunto de datos! usando
#la función de nombres de Pandas para asegurar de que
#los nombres de columna asociados con los datos lleguen a través.
df = pd.read_csv("dataset_pruebas1.csv",sep=",")
df.head()
#vistazo a los datos:
df.describe().transpose()
#se va configurar nuestros datos y nuestras etiquetas:
X = df.drop('phishing',axis=1)
y = df['phishing']
#División de entrenamiento y de prueba
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)
#Preprocesamiento de datos
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
#Sólo se ajusta a los datos de entrenamiento
scaler.fit(X_train)
#Ahora aplica las transformaciones a los datos:
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Nota. Fuente: Elaboración Propia.

Figura 47 *Entrenamiento del conjunto de datos*

```
#Entrenamiento del modelo
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(13,13,13),max_iter=500)
mlp.fit(X_train,y_train)
#Predicciones y evaluación
predictions = mlp.predict(X_test)
# usar las métricas integradas de SciKit-Learn,
#como un informe de clasificación y una matriz de confusión para evaluar el rendimiento de nuestro modelo:
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))
# Vea el informe de clasificación para los datos de las pruebas y predicciones
print(classification_report(y_test,predictions))
```

Nota. Fuente: Elaboración Propia.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

Según los resultados obtenidos durante la elaboración de esta investigación, las que derivan de los objetivos específicos propuestos.

Se hizo una revisión de artículos científicos entre los años 2015 y 2020 para identificarlos tipos de vulnerabilidades de phishing que se pueden encontrar en aplicaciones web obteniendo un listado de 14 vulnerabilidades, se realizó un análisis según el impacto, la prevalencia y la detección de estas vulnerabilidades llegando a un top de 5 tipos de vulnerabilidades.

Seguidamente para esta investigación se hizo un listado de microempresas basados en criterios de selección específicamente que tuvieran un aplicativo web, que contarán con su propio servidor DNS y que la microempresa seleccionada pueda dar el fácil acceso a la información, solo una microempresa fue tomada como caso de estudio ya que contaba con todos los criterios de selección antes mencionados.

Al igual que las vulnerabilidades se realizó una revisión de algoritmos de aprendizaje de máquina para hacer detección de ataques phishing por envenenamiento del servidor DNS de los cuales se seleccionaron 4 basados en el criterio de mejor precisión entre ellos están Naive Bayes, Random Forest, XGBoost y el Perceptrón Multicapa.

Se realizó la implementación de algoritmos de detección de aprendizaje de máquina que fueron seleccionados anteriormente usando el Kernel de Python 3 y el entorno Jupyter Notebook para entrenar y probar el modelo de dataset.

Por último, se realizaron las pruebas de detección ataques de envenenamiento al servidor DNS creando un escenario de pruebas y utilizando las distintas herramientas mencionadas anteriormente las cuales fueron de gran ayuda.

El mejor resultado que se pudo obtener luego de haber realizado las pruebas de detección fue el algoritmo de Naive Bayes con un 99.04% de precisión seguido del Perceptrón Multicapa con un 80 % de precisión dejando atrás a los otros dos algoritmos Random Forest y XGBoost con una precisión por debajo del 80%.

4.2. Recomendaciones.

Dado a que en esta investigación solo se realizó el uso de algoritmos de Machine Learning como Naive Bayes, Perceptrón Multicapa, XGBoost y Random Forest, las cuales dieron buenos resultados en la detección de ataques phishing por envenenamiento del servidor de nombres de dominio, existen más algoritmos que pueden dar aún mejores resultados y que no se abordan en esta investigación.

Esta investigación fue realizada con el propósito de poder detectar ataques de phishing por envenenamiento de servidores DNS, por lo que se puede decir que es posible seguir mejorando en cuanto al uso de algoritmos de clasificación para detectar ataques phishing de este tipo.

REFERENCIAS

[1] S. Gupta, A. Singhal y A. Kapoor, «A literature survey on social engineering attacks: Phishing attack,» *International Conference on Computing, Communication and Automation (ICCCA)*, pp. 537-539, 2016.

[2] M. Baykara y Z. Gürel, «Detection of phishing attacks,» *6to Simposio internacional sobre seguridad y análisis forense digital (ISDFS)*, pp. 1-4, 2018.

[3] A. Bharat y K. Chandrasekaran, «A client-side anti-pharming (CSAP) approach,» *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, p. 4, 2016.

[4] P. Sahoo, «Data mining a way to solve Phishing Attacks,» *International Conference on Current Trends towards Converging Technologies*, pp. 1-5, 2018.

[5] J. Vijayan, «InformationWeek It Netword,» 03 07 2019. [En línea]. Available: https://www.darkreading.com/attacks-breaches/phishing-attacks-evolve-as-detection-and-response-capabilities-improve-/d/d-id/1334109.

[6] F. Paz, «Andina.com,» 04 05 2020. [En línea]. Available: https://andina.pe/agencia/noticia-coronavirus-peru-sufrio-mas-433-millones-intentos-ciberataques-2020-795751.aspx.

[7] Cisco, «Reporte Anual de Ciberseguridad,» 23 Noviembre 2018. [En línea]. Available: https://www.cisco.com/c/dam/global/es_mx/solutions/pdf/reporte-anual-cisco-2018-espan.pdf.

[8] X. Zhang, D. Shi, Zhang.H., W. Liu y R. Li, «Efficient Detection of Phishing Attacks with Hybrid Neural Networks,» *IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 844 -848, 2018 .

[9] W. Jaspher, M. Paradise, R. Amrutha y C. Eligious, «Variants of phishing attacks and their detection techniques,» *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 255-259, 2019.

[10] H. Kim y J. Huh, «Detecting DNS-poisoning-based phishing attacks from their network performance characteristics,» *Electronics Letters* , pp. 656 - 658, 2011.

[11] Y. Li, S. Chu y S. Xiao, «A pharming attack hybrid detection model based on IP addresses andweb content,» *School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, PR Chinaa*, pp. 234 - 239, 2015.

[12] G. Jaspher, A. Amrutha, P. Mercy y C. Kalaiyani, «Variants of phishing attacks and their detection techniques,» *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1-5, 2019.

[13] X. Zhang, W. Haining y S. Jajodia, «Gemini: An Emergency Line of Defense against Phishing Attacks,» *IEEE 33rd International Symposium on Reliable Distributed Systems,* pp. 1 -10, 2015.

[14] S. Nisha y M. Neela, «Prevention of phishing attacks in voting system using visual cryptography,» *International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS),* pp. 1- 4, 2016.

[15] I. Celestine, J. Zunera, R. .. Abdul, G. Reddy, G. Kaluri, Srivastava. y J. Ohyun, «KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attackss,» pp. 1 -11, 2020.

[16] G. Kishan, J. Mukul, M. Palash y M. Jayashri, «A Novel Approach to Detect Phishing Attack Using Artificial Neural Networks Combined with Pharming Detection,» *3rd International conference on Electronics, Communication and Aerospace Technology (ICECA),* pp. 1-5, 2019.

[17] A. Bharat y K. Chandrasekaran, «A Client-Side Anti-Pharming(CSAP) ApproachCombined With Pharming Detection.,» pp. 1 - 6 , 2016.

[18] S. Manhas, S. Taterh y D. Singh, «Clas:A novel Communications Latency based Authentication Scheme,» pp. 1 - 21, 2017.

[19] N. Azeez y A. Oluwatosin, «CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification,» *International Conference on Computational Science and Computational Intelligence (CSCI),* pp. 1-7, 2016.

[20] P. Chandra, R. Deb y M. Nurul, «Know your customer (KYC) based authentication method for financial services through the internet,» *19th International Conference on Computer and Information Technology (ICCIT),* pp. 1- 6, 2016.

[21] S. Şentürk, E. Yerli y A. Soğukpınar, «Email phishing detection and prevention by using data mining techniques,» *International Conference on Computer Science and Engineering (UBMK),* pp. 1 - 6, 2017.

[22] T. Nathezhtha, D. Sangeetha y V. Vaidehi, «WC-PAD: Web Crawling based Phishing Attack Detection,» *International Carnahan Conference on Security Technology (ICCST),* pp. 1 - 6, 2019.

[23] J. Li y S. Wang, «PhishBox: An Approach for Phishing Validation and Detection,» *IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech),* pp. 1- 8, 2017.

[24] X. Zhang, D. Shi, H. Zhang, W. Liu y R. Li, «Efficient Detection of Phishing Attacks with Hybrid Neural Networks,» *IEEE 18th International Conference on Communication Technology (ICCT),* pp. 1- 5, 2018.

[25] D. Hernandez, El delito Informatico., España: EGUZZKILORE, 2009.

[26] I. A. Amiri y E. Fazeldehkordi, AMachine-Learning Approach to phishing Detection and Defense, Malasia: Elsevier, 2015.

[27] AndalucíaCERT, «Informe de divulgaciónPhishing,» *Seguridad & confiana digital,* pp. 4-5, 2017.

[28] M. Jakobsson y S. Myers, Phishing and Countermeasures: Understandingthe Increasing Problem of Electronic identity Thef, New Jersey: Wiley, 2007.

[29] N. Oxman, «Estafas informáticas através de Internet: acerca de la imputación penal del "phishing" y el"pharming",» *Revista deDerecho de la Pontificia Universidad Católica de Valparaíso,* pp. 5 - 7, 2013.

[30] R. Gago, «Uso de Algoritmos de aprendizaje automático aplicados a bases de datosgeneticos,» *Universidad oberta de cataluña,* p. 39, 2017.

[31] R. Gago, «Uso de Algoritmos de aprendizaje automático aplicados a bases de datos geneticos ,» *Universidad oberta de cataluña,* p. 41, 2017.

[32] J. Matich, «Redes Neuronales:conceptos Basicos y Aplicaciones,» *Redes neuronales,* p. 8, 2001.

[33] J. Deco, «Estudio y aplicación de técnicas de aprendizaje automático orientadas al ámbitomédico: estimación y explicación de predicciones individuales,» *Estudio y aplicación de técnicas de aprendizaje automático orientadas al ámbitomédico,* p. 19, 2012.

[34] I. Wilford, «minería de datos: herramienta de apoyo en la selección de equipos de proyectos informáticos,» p. 2, 2006.

[35] L. Kazmier y A. Diaz, Estadísticaaplicada a administración y economía, Juáres Mexico: McGRAW-HILL, 2006.

[36] I. Witten, F. Eibe, L. Trigg, M. Hall, G. Holmes y S. Cunningham, «Weka: Practical Machine Learning Tools and Techniques,» *Department of Computer Science, University of Waikato, New Zealand.,* pp. 1- 8, 1999.

[37] S. Méndez y A. Cuevas, «Manual introductorio al SPSS,» Statistics Standard Edition 22, México, 2014.

[38] G. Rehill, «Frequency and Frequency tables,» *Year 8 Interactive Maths. In: Interactive Maths Series Software.,* pp. 1-3, 2012.

[39] M. Gharib y A. Bondavalli, «On the Evaluation Measures for Machine Learning Algorithms for Safety-Critical Systems,» *15th European Dependable Computing Conference (EDCC),* pp. 141 -144, 2019.

[40] G. Z. U. U. Hajaramusa, «A Comparative Analysis Of Phishing Website Detection,» *Revista de tecnología de la información teórica y aplicada,* vol. 97, nº 5, p. 10, 2019.

[41] M. M. Cuzzocrea, «Application of automatic learning techniques to detect and,» de *The 20th International Conference*, Italia, 2018.

[42] O. O. Oladimeji, «"Text Analysis and Machine,» *Revista internacional de aplicaciones informáticas*, vol. 182, nº 36, p. 6, 2019.

[43] K. &. Sathiyakumar, «Multiclass Classification of XSS Web Page Attack using Machine Learning,» *Revista internacional de aplicaciones informáticas*, vol. 74, nº 12, p. 5, 2013.

[44] J. Vijayan, «InformationWeek It Net word,» 03 07 2019. [En línea]. Available: https://www.darkreading.com/attacks-breaches/phishing-attacks-evolve-as-detection-and-response-capabilities-improve-/d/d-id/1334109.

[45] TensorFlow, «www.tensorflow.org.,» 03 November 2015. [En línea]. Available: https://www.tensorflow.org/about/bib.

[46] O. O. Oladimeji, «Text Analysis and Machine,» *Revista internacional de aplicaciones informáticas*, vol. 182, nº 36, p. 11, 2019.

ANEXOS

Anexo 1. Resolución para ampliación de tema de Tesis

Universidad Señor de Sipán

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0139-2023/FIAU-USS

Pimentel, 10 de marzo de 2023

VISTO:
El Acta de reunión N°0803-2023 del Comité de investigación de la Ingeniería de Sistemas remitida mediante Oficio 0059-2023/FIAU-IS-USS de fecha 09 de marzo de 2023, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48º que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24º señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N° 059 - 2023, de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE, a cargo de los estudiantes y /o egresados del Programa de estudios INGENIERÍA DE SISTEMAS, hasta la fecha que indica la presente resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0139-2023/FIAU-USS

Pimentel, 10 de marzo de 2023

ANEXO

AMPLIACION DE VIGENCIA DE TEMA DE TESIS

	APELLIDOS	TESIS	OBSERVACION	RESOLUCION APROBACIÓN/AMPLIACIÓN
1	MERA TAPIA GABRIEL HUGO	ESTRATEGIA TECNOLOGICA PARA MEJORAR LA PLANEACION DE TECNOLOGIA, INFORMACION Y COMUNICACION DE ELECTRO ORIENTE S.A	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	2109-2020/FIAU-USS
2	TORRES CHIMOY EDILBERTO PAÚL DIAZ SANCHEZ RICARDO RICHAR	DESARROLLO DE UN MÉTODO DE IDENTIFICACIÓN DE CRYPTOHACKING MEDIANTE EL ANÁLISIS DE CÓDIGO JAVASCRIPT UTILIZANDO APRENDIZAJE AUTOMÁTICO	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1347-2020/FIAU-USS
3	GUZMAN LOPEZ RICARDO ARTURO	EVALUACIÓN DE LA USABILIDAD DE UNA APLICACIÓN MÓVIL DE SOPORTE TÉCNICO UTILIZANDO LA NORMA ISO/IEC 25010	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	2320-2020/FIAU-USS
4	GARCIA GUTIERREZ KEVIN GIANMARCO GUEVARA RAMIREZ CESAR ALBERTO	DETECCIÓN DE PHISHING POR ENVENENAMIENTO DEL SERVIDOR DE NOMBRE DE DOMINIO PARA EVITAR EL ROBO DE INFORMACIÓN EN APLICACIONES WEB DE MICROEMPRESAS PERUANAS UTILIZANDO APRENDIZAJE DE MÁQUINA	AMPLIAR HASTA EL 31 DE DICIEMBRE DEL 2023	1329-2020/FIAU-USS
5	JARA TUCTO ALEXANDER	IDENTIFICACIÓN AUTOMÁTICA DE NEUMONÍA MEDIANTE EL PROCESAMIENTO DIGITAL DEL	AMPLIAR HASTA EL 31 DE JULIO DEL 2023	0938-2019/FIAU-USS

Anexo 2. Carta de Aceptación

ÓPTICAS ROMERO
¡Donde su vista es lo primero!

"Año de la Universalización de la Salud"

Chiclayo, 17 de diciembre del 2020

Ing. Heber Ivan Mejia Cabrera

Director(e)de la Universidad Señor de Sipan S.A.C

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO UNIVERSIDAD SEÑOR DE SIPÁN

Presente:

Asunto: Trabajo de Investigacion

Tengo el agrado de dirigirme a Usted para saludarle y en atención al asunto de la referencia, hago de su conocimiento que atendiendo lo solicitado por su Institución, los estudiantes **GUEVARA RAMIREZ CESAR ALBERTO** identificado con DNI 48317275, y **GARCIA GUTIERREZ KEVIN GIANMARCO** identificado con DNI 77423798, cuentan con mi autorización para el recojo de información relevante que será usada para la realización de su trabajo de Investigación **"DETECCIÓN DE PHISHING POR ENVENENAMIENTO DEL SERVIDOR DE NOMBRE DE DOMINIO PARA EVITAR EL ROBO DE INFORMACIÓN EN APLICACIONES WEB DE MICROEMPRESAS PERUANAS UTILIZANDO APRENDIZAJE DE MÁQUINA"**.

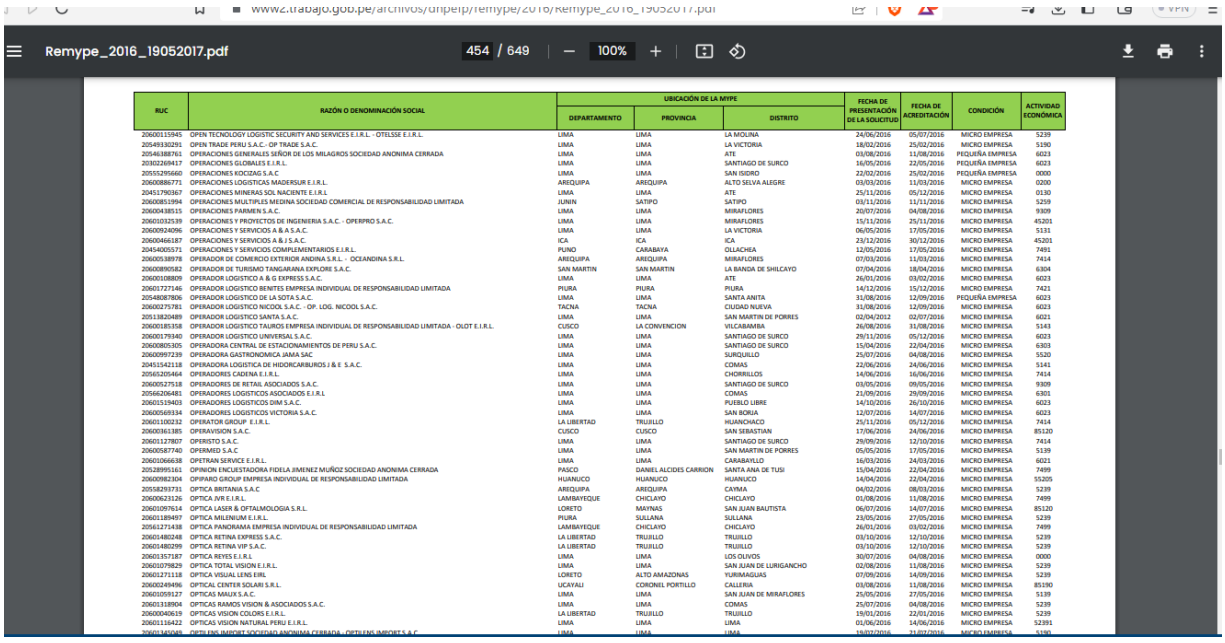
Sin otro particular, quedo de Ud.

Atentamente,

ÓPTICAS ROMERO
Juan Carlos Romero Robles
GERENTE

Anexo 3. Fuentes de donde se Extrajo el listado de Microempresas en el rubro de ventas.

Nota: Pagina web del Ministerio de Economía y Finanzas



Nota: Listado de Microempresas en la página trabajo.gob.pe

Anexo 4. FORMATO PARA INFORME DE CONSUMO DE CPU.

Consumo de CPU	
Ítem	**Valor**
Uso	7%
Velocidad	2.32 GHz
Procesos	96
Subprocesos	1305
Tiempo	0:14:21

Anexo 5. FORMATO DE INFORME DE CONSUMO DE MEMORIA.

Consumo de Memoria	
Ítem	**Valor**
Uso	2.8
Disponibilidad	351 MB
Confirmada	4.0/5.3 GB
En caché	241MB
Tiempo	0:14:21

Anexo 6. FORMATO PARA INFORME DE PROMEDIO DE TIEMPO DE RESPUESTA.

Promedio de Tiempo de respuesta	
Ítem	**Valor**
Velocidad	149 kb/s
Tiempo	10.0 ms
CPU	9%
Memoria	84%
Disco	8%

Anexo 7. Lista de tipos de vulnerabilidades en aplicaciones web de microempresa

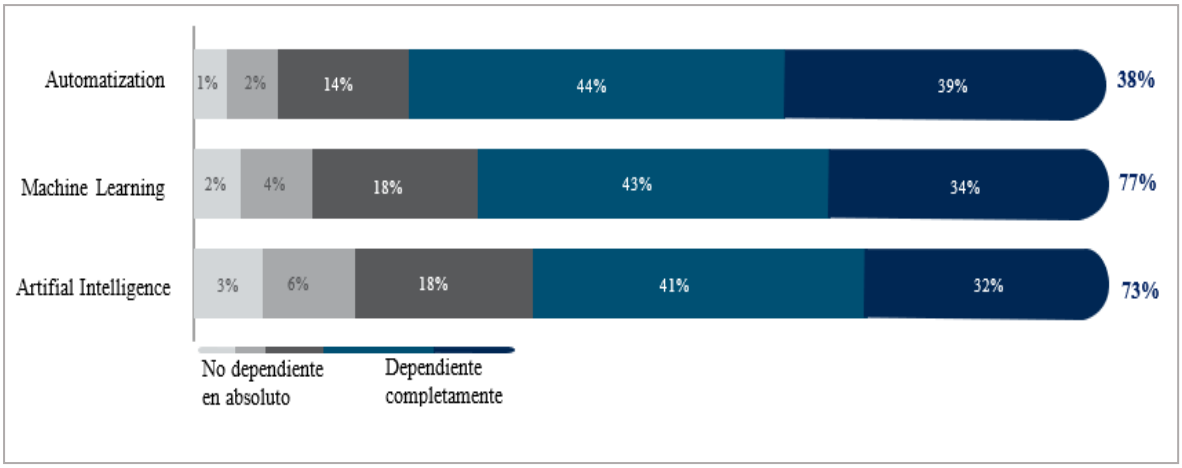
N.º	Tipo	Vulnerabilidad	Autor(es)
1	Cross Site Request (CSRF)	Entradas no válidas	Hernández & Mejia (2017)
2	Configuraciones de seguridad incorrectas	Gestión Incorrecta de Errores Configuración de seguridad incorrecta	Cova, Felmetsger &Vigna
3	Inyección SQL, NoSQL, OS y Envenenamiento DNS	Ejecución de comandos no intencionados o acceso a datos	Kaur & Preet (2015)
4	Autenticación Comprometida	La Gestión de las sesiones de usuario	Hernández & Mejia (2017)
5	Control de Acceso	Modificación de Parámetros en la URL	Hernández & Mejia (2017)

N.º	Tipo	Vulnerabilidad	Autor(es)
6	Entradas externas XML (XXE)	Explotación de código vulnerable, dependencias o integraciones, robo de datos confidenciales	Gonzales & Montesino (2018)
7	Scripts de sitios cruzados (XSS) basados en DOM	Modificación del script DOM mediante el robo de datos confidenciales.	Talebzadeh & Ghodrat (2017)
8	Inclusión de cualquier archivo	Ejecución de código en el servidor web (archivos locales y remotos)	Hernández & Mejia (2017)
9	Insuficiente registro y monitoreo	Fugas constantes de datos	Gonzales & Montesino (2018)
10	Exposición de datos sensibles	Protección inadecuada de datos de usuarios	Rojas (2017)
11	Gestión Incorrecta de Errores	Muestra mensajes de error como salida después de que se procesa la aplicación.	Yadav, Gupta, Singh, Kuma & Sharma
12	Entrada no válida	Ingreso de información maliciosa en la aplicación, evitando así la seguridad del sitio web.	Yadav, Gupta, Singh, Kuma & Sharma
13	Uso de componentes con vulnerabilidades conocidas	Obtención de Exploits	Gonzales & Montesino (2018)
14	Registro y Monitoreo Insuficientes	Cambio de los sistemas, alteración, extracción y/o destrucción de los datos	Gonzales & Montesino (2018)

Anexo 8. Matriz para establecer la técnica más eficiente en la detección de ataques Phishing

N°.	ALGORITMO	Métricas	EXACTITUD	PRECISIÓN	ESPECIFICIDAD	RECALL
1	Naive Bayes	FP: 1 TP: 58 FN: 4 TN: 14	0.08%	99.04%	92%	0.08%
2	Random Forest	FP: 13 TP: 39 FN: 16 TN: 9	65%	63%	78%	65%
3	Perceptrón Multicapa	FP: 11 TP: 44 FN: 13 TN: 9	71%	75%	71%	71%
4	Extreme Gradient Boosted Tree (XGBOOST)	FP: 22 TP: 37 FN: 3 TN: 15	74%	80%	73%	74%

Anexo 9: Reporte de referencia de las capacidades de la seguridad de cisco 2018.



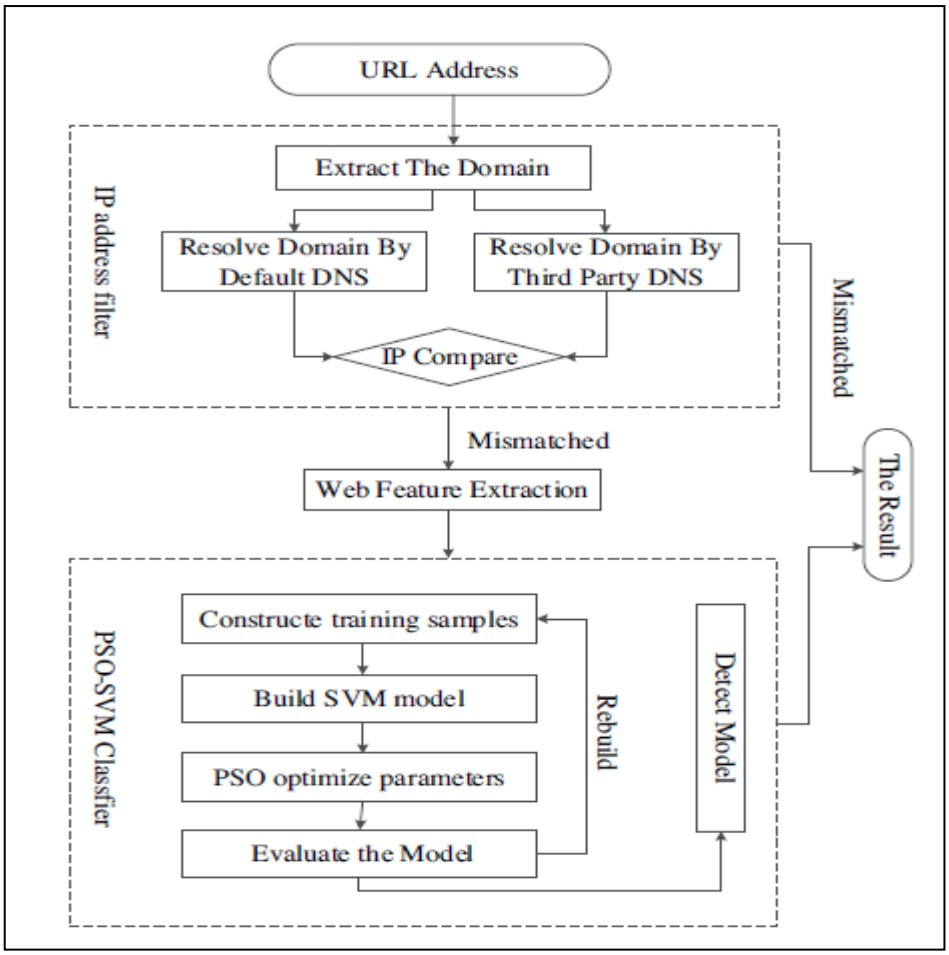
Fuente: Reporte Anual de Ciberseguridad de Cisco

Anexo 10: Los diversos enfoques de los algoritmos de detección.

S.NO.	APPROACH	ALGORITHM	METRICS	ACCURACY
1	Heuristic based approach	Decision tree algorithm	False positive: 5 True positive: 120 False negative: 3 True negative: 72	96.76%
2	Blacklist approach	Simhash algorithm	False positive: 0	84.36%
3	Fuzzy rule-based approach	Fuzzy data mining algorithm	Accuracy: 100%	100%
4	Machine learning approach	Machine learning algorithms	False positive: 1.52% True positive: 98.39%	98.4%
5	Cantina based approach	TF-IDF information retrieval algorithm	False positive: 6%	97%
6	Image based approach	Web logo technique	True positive: 99.8% True negative: 87%	98%

Nota.Fuente. Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019).

Anexo 11: Modelo de detección de híbridos de ataque farmacéutico



Nota: Modelo de detección de híbridos de ataque farmacéutico.

Fuente: School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, PR China.