Detección de ataques de Phishing utilizando Procesamiento de Lenguaje Natural y Modelo Oculto de Markov

Molina Salavarría, Jhoseph Alberto y Monteros González, Kevin Joel

Departamento de Ciencias de la Computación

Ingeniería en Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniería en Tecnologías de la Información

Ing. Benavides Astudillo, Diego Eduardo, Mgtr.

25 de agosto del 2022

Reporte de verificación de contenido

1. TRABAJO DE INTEGRACIÓN CURRICULAR 2022 MOLIN…

Scanned on: 20:24 August 25, 2022 UTC

Identical Words	174
Words with Minor Changes	120
Paraphrased Words	224
Omitted Words	0

Firma:

Firmado electrónicamente por:
DIEGO EDUARDO BENAVIDES ASTUDILLO

.....

Ing. Benavides Astudillo, Diego Eduardo, Mgtr.

Director

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Comunicación

Certificación

Certifico que el trabajo de integración curricular: "**Detección de ataques de Phishing utilizando Procesamiento de Lenguaje Natural y Modelo Oculto de Markov**" fue realizado por los señores **Monteros González, Kevin Joel** y **Molina Salavarría, Jhoseph Alberto,** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además, fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo, 25/08/2022

Firma:

Firmado electrónicamente por:
DIEGO EDUARDO BENAVIDES ASTUDILLO

.....

Ing. Benavides Astudillo, Diego Eduardo, Mgtr.

C.C 1712883063

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Responsabilidad de Autoría

Nosotros, **Monteros González, Kevin Joel** y **Molina Salavarría, Jhoseph Alberto**, con cédulas de ciudadanía n°**2300557275** y n°**2350660441**, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Detección de ataques de Phishing utilizando Procesamiento de Lenguaje Natural y Modelo Oculto de Markov** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo, 25/08/2022

Monteros González, Kevin Joel
C.C.: 2300557275

Molina Salavarría, Jhoseph Alberto
C.C.: 2350660441

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Autorización de Publicación

Nosotros, **Monteros González, Kevin Joel** y **Molina Salavarría, Jhoseph Alberto**, con cédulas de ciudadanía n°**2300557275** y n°**2350660441**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Detección de ataques de Phishing utilizando Procesamiento de Lenguaje Natural y Modelo Oculto de Markov** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Santo Domingo, 25/08/2022

.....

Monteros González, Kevin Joel

C.C.: 2300557275

.....

Molina Salavarría, Jhoseph Alberto

C.C.: 2350660441

Dedicatoria

El presente trabajo está dedicado a mi familia que con gran esfuerzo, sacrificio y amor estuvieron apoyándome incondicionalmente. En especial a mi madre que jamás permitió que me rindiese, a pesar de los obstáculos que se presentaron en el camino.

A todos aquellos amigos que con sus palabras de aliento lograban motivarme para poder culminar con éxito mi carrera profesional.

A los docentes que compartieron su conocimiento en este largo recorrido, en especial a aquellos que lograron formar parte de mi círculo amistoso.

Finalmente, agradezco el apoyo incondicional de mis dos amigos Kevin Monteros y Aldair Puco, que con gran suerte pude conocer en esta etapa de la vida y siempre estuvieron presentes en los buenos y malos momentos.

Jhoseph Molina

A mis padres y hermanos quienes me apoyaron incondicionalmente durante todo el tiempo de mi formación personal y profesional, siendo una de mis mayores fuentes de motivación.

A mis amigos, Jhoseph Molina y Aldair Puco quienes fueron un gran apoyo emocional y profesional durante todo el tiempo de la carrera, con quienes he compartido grandiosas experiencias y éxitos.

A mis maestros de toda mi formación académica, quienes se esforzaron en instruirme de la mejor manera, compartiéndome sus conocimientos, dedicación, tiempo y experiencia.

Para todo ellos les dedico este logro, pues es a ellos a quienes les debo su apoyo incondicional.

Kevin Monteros

Agradecimiento

Quiero agradecer principalmente a Dios por darme salud y sabiduría para poder culminar con éxito esta etapa de mi vida.

Mi más profundo agradecimiento a mis padres José Molina y Yanina Salavarria por haberme brindado su apoyo incondicional desde el comienzo, a mis hermanos y a mi pareja por ayudarme a superar los obstáculos encontrados a lo largo de este recorrido.

Agradezco a la Universidad de las Fuerzas Armadas ESPE, por permitirme estudiar la Carrera de Ingeniería en Tecnologías de la Información y darme acogida en sus instalaciones durante toda esta etapa de estudio.

Finalmente agradezco a mi Tutor de tesis, el Ing. Eduardo Benavides, Mgtr., quien con su gran enseñanza impartida he podido finalizar este trabajo con el mejor de los éxitos.

Jhoseph Molina

A mis padres, Alex Monteros y Laura González; a mis hermanos, Dante Monteros y Danilo Monteros, quienes han sido el motor que impulsa mis sueños y esperanzas a lo largo de mi vida, siendo un apoyo incondicional durante todo este proceso.

A mi compañero de tesis Jhoseph Molina, con quien logré entablar una gran amistad en todo este proceso de formación lleno de malas noches, momentos inolvidables y por supuesto, grandes éxitos.

A la Universidad de las Fuerzas Armadas ESPE, por encargarse de que este proceso de enseñanza y aprendizaje haya sido tan enriquecedor y retador a la vez. Además, por darme la oportunidad de conocer excelentes docentes, de entre los cuales se encuentra el tutor a cargo de esta tesis, el Ing. Eduardo Benavides, Mgtr.

Kevin Monteros

Índice de contenidos

Índice de tablas

Índice de Figuras

Resumen

Con la llegada de Covid-19, han aumentado los correos electrónicos de Spam que intentan engañar a las personas para hacerlas víctimas de estafas o algún engaño. Es posible invertir en software y hardware, pero al final basta con el descuido de un usuario o su desconocimiento para ser víctima de algún ciberataque. Así, este estudio ofrece principalmente dos cosas: primero, hace una revisión de la literatura sobre las soluciones que utiliza Natural Language Processing: y segundo, analiza los resultados obtenidos por los algoritmos Hidden Markov Model (HMM) versus los obtenidos por Machine Learning (ML) en la detección de estos ataques. Con la revisión literaria se puede afirmar que no se han encontrado muchos artículos que utilicen HMM para solucionar este tipo de ataques con una gran precisión; esto se debe a que este tipo de modelo no cuenta con una amplia investigación previa para poder ser aplicado con gran efectividad, sin embargo, su presencia ha abierto una nueva línea de investigación para poder prevenir este tipo de ataques. Para esto, se realizó un estudio comparativo sobre la efectividad y precisión que tienen hoy en día los mejores algoritmos de Machine Learning en cuanto a la detección de ataques por correo y se evidenció que su porcentaje de precisión podría variar según su escenario y la cantidad de datos a procesar. Finalmente, se determinó que los algoritmos de Machine Learning ofrecen mayor precisión en este tipo de detección. Como trabajo futuro se propone el desarrollo de un algoritmo que realice un preprocesamiento mediante HMM y luego utilice algoritmos de Deep Learning para mejorar la precisión.

Palabras clave: Spam, correo electrónico, Hidden Markov Model, Machine Learning, Ingeniería Social.

Abstract

With the arrival of Covid-19, there has been an increase in Spam e-mails that try to trick people into becoming victims of scams or deception. It is possible to invest in software and hardware, but in the end a user's carelessness or lack of knowledge is enough to become a victim of a cyberattack. Thus, this study offers mainly two things: first, it makes a literature review on the solutions that use Natural Language Processing: and second, it analyzes the results obtained by Hidden Markov Model (HMM) algorithms versus those obtained by Machine Learning (ML) in the detection of these attacks. With the literature review it can be stated that not many articles have been found that use HMM to solve this type of attacks with high accuracy; this is because this type of model does not have extensive previous research to be applied with great effectiveness, however, its presence has opened a new line of research to prevent this type of attacks. For this, a comparative study was carried out on the effectiveness and accuracy that the best Machine Learning algorithms have nowadays regarding the detection of mail attacks and it was evidenced that their percentage of accuracy could vary according to their scenario and the amount of data to be processed. Finally, it was determined that Machine Learning algorithms offer higher accuracy in this type of detection. As future work, we propose the development of an algorithm that performs preprocessing using HMM and then uses Deep Learning algorithms to improve accuracy.

Keywords: Spam, email, Hidden Markov Model, Machine Learning, Social Engineering.

Capítulo I

Introducción

A lo largo de esta sección se detallan los antecedentes del proyecto con base en la revisión literaria de los algoritmos tradicionales para la detección de correos electrónicos de Phishing. Una vez planteados los antecedentes de la investigación y los objetivos, se logró determinar la problemática latente y el enfoque del proyecto.

Antecedentes

"La Ingeniería Social en la actualidad, continúa siendo el método de propagación de ciberataques más utilizado por los creadores de malware, quienes buscan aprovecharse de las ventajas de cualquier medio de comunicación a fin de engañar a los usuarios y lograr sus objetivos maliciosos" (Borghello, 2009). Los ataques de Ingeniería Social son especialmente peligrosos debido a que no pueden ser mitigados por ningún tipo de firewall o hardware de seguridad informática. Pero esto no significa que sean totalmente efectivos, de hecho, para poder cumplir su propósito primero deben engañar a la víctima por alguna especie de medio, principalmente, por correos electrónicos.

El Spam por correo electrónico es uno de los métodos más utilizados para realizar un ataque de Ingeniería Social. Según (Benavides et al, 2022) "mediante la Ingeniería Social, el atacante abusa de un comportamiento desprevenido o de rasgos de personalidad ingenuos para engañar y, por lo general, estafar al remitente". El medio más común para evadir un correo electrónico de Spam, es comprobar si este correo está en una lista negra. El problema de utilizar una Lista Negra es que el correo Spam ya debe haber sido identificado como Spam anteriormente, por ende, alguien ya tuvo que haber sido víctima de este ataque. Para identificar este tipo de ataques informáticos nuevos se utiliza la Inteligencia Artificial.

Los ataques de Ingeniería Social se centran exclusivamente en atacar al usuario, quien es el eslabón más débil de la seguridad informática. En este punto (Albladi, S. y Weir, G., 2020)

muestran su preocupación por el aumento de los ataques informáticos de Ingeniería Social. Estos autores coinciden en que es importante que los usuarios comprendan los factores que influyen en las competencias, la conciencia y las acciones que involucra un ciberataque, y destacan la necesidad de una investigación integral sobre la Ingeniería Social, sus impactos y riesgos; además, proponen estrategias y modelos para una mejora en la concienciación y actuación positiva para la detección de ciber amenazas.

Existen múltiples investigaciones e implementaciónes de técnicas de ML y Deep Learning (DL) en la detección de correos de Phishing (Ozcan et al., 2021), (K. Haynes, 2021) y (S. Bagui, 2021), en donde se proponen modelos híbridos de aprendizaje profundo basados en algoritmos de memoria a corto plazo y de redes neuronales profundas para la detección de URLs de Phishing, con lo cual se ha obtenido una precisión entre 95% y 98% de exactitud. Estos modelos utilizan listas negras para su propósito, e incluso, los que ya son más avanzados, utilizan el análisis semántico profundo para capturar las características inherentes al cuerpo del texto en los correos electrónicos.

Definición de la problemática

Las tecnologías de la información han evolucionado a partir de la búsqueda de acceso y producción de información que se evidenció durante el Covid-19, y actualmente tienen un gran impacto tanto en el aspecto individual como organizacional. El uso del correo electrónico trae consigo amenazas y riesgos que surgen de vulnerabilidades explotadas por atacantes que buscan obtener acceso ilegal a información diversa, y causar daño dentro de una organización o directamente a un individuo. Debido al acceso no autorizado o el robo de información por medio de un correo Phishing, algunos investigadores se han comprometido a combatir los ataques de Ingeniería Social.

Hoy en día la principal brecha de seguridad aún no solucionada radica en los usuarios, esto se debe a que ya sea por desconocimiento, ingenuidad o descuido están expuestos a

distintos tipos de ataques de Ingeniería Social. Uno de los más frecuentes es el de Phishing, el cual actúa de distintos modos y en distintos escenarios. A raíz del Covid-19, gran parte de la población mundial optó por usar el correo electrónico como medio de comunicación; empresas, instituciones educativas o áreas financieras, exigieron a su comunidad el uso del correo electrónico, sin antes realizar la debida socialización o capacitación en cuanto a su correcto o seguro uso. Es por ello que en los últimos años los ataques de Ingeniería Social han incrementado descontroladamente, ya que este tipo de seguridad ya no depende de equipos de alto rendimiento o buenos sistemas, sino que depende en gran medida del usuario y de su capacidad para no entrar en el juego de este tipo de ciberdelincuentes.

Justificación

Aunque la eficacia de los softwares, firewalls y medidas de seguridad basadas en hardware para contrarrestar delitos informáticos se han incrementado a la par con los tipos de ciberataques, la exposición por parte del usuario, que es el eslabón más débil de la seguridad informática ante ataques de Ingeniería Social, no ha cambiado; así, no existen mecanismos eficaces para anticipar las acciones del usuario, por lo cual, quedan los usuarios expuestos y vulnerables.

En la actualidad, Machine Learning cuenta con una variedad de algoritmos que permiten detectar estos tipos de ataques, que han demostrado una alta efectividad. Sin embargo, la evolución de los ataques de Ingeniería Social ha llegado a un punto en el cual no basta con tener herramientas o sistemas informáticos como medio de seguridad.

Por lo expuesto anteriormente, la seguridad informática se ve en la necesidad de buscar nuevos métodos o mecanismos capaces de mitigar los ataques Phishing en correos electrónicos. En este proyecto se presenta a HMM como una alternativa para ayudar a contrarrestar este tipo de delitos informáticos, debido a su avanzada capacidad para realizar análisis semánticos en el contenido de un Spam de Phishing.

Objetivos

Objetivo General

Realizar un estudio comparativo entre Hidden Markov Model (HMM) y los algoritmos tradicionales de Machine Learning (ML) para la detección de correos electrónicos Phishing.

Objetivos Específicos

1. Realizar una revisión de la literatura.
2. Realizar el preprocesamiento del Dataset.
3. Implementar los algoritmos de Machine Learning.
4. Implementar el algoritmo de HMM.
5. Realizar y detallar un análisis comparativo.

Alcance

Este proyecto se plantea el objetivo de determinar Este proyecto se plantea el objetivo de determinar el grado de exactitud en la detección de ataques, basando esta detección en el cuerpo contenido en los correos electrónicos de Phishing, usando algoritmos de Machine Learning versus el algoritmo de HMM. Por ende, el alcance del mismo está comprendido entre la recolección y presentación de los resultados obtenidos de esta comparación desde un planteamiento cuantitativo.

Planteamiento de la hipótesis

La precisión obtenida en la ejecución de los algoritmos para la detección de correos spam es mayor en ML que en HMM.

Capítulo II

Marco Teórico

El contenido de este capítulo se construye en base a dos partes fundamentales: el estado del arte y los conceptos relacionados con la línea de investigación de nuestro proyecto. Así, para realizar la primera parte, se realiza una revisión de la literatura de artículos científicos y otras investigaciones asociadas con nuestro tema de proyecto; y la segunda parte consiste en exponer todos los conceptos necesarios para comprender esta investigación.

Estado del arte

El estado del arte está comprendido por las principales investigaciones que tienen relación y relevancia para nuestro tema de proyecto. Además, son los estudios que tienen relación con la problemática mencionada en el Capítulo I.

A continuación, se muestran los estudios e investigaciones que ayudaron a resolver la problemática descrita anteriormente, los mismos que permitieron obtener: métodos, conceptos, referencias, palabras clave y análisis que ayudaron al desarrollo de este trabajo.

En la Tabla 1, se muestran los artículos principales revisados para la realización de este proyecto, para lo cual se presenta cada artículo con los algoritmos utilizados en su realización, junto a la cantidad de datos utilizados en cada Dataset para la obtención de los resultados de sus pruebas.

Tabla 1.

Investigaciones relacionadas a la detección de correos maliciosos, con los algoritmos y cantidad de datos utilizados.

Título	Algoritmos utilizados	Cantidad de datos
Lightweight URL-based Phishing detection using natural language processing transformers for mobile devices **(K. Haynes, 2021)**	Decision Tree (DT), Gaussian Naive Bayes (GNB), Random Forest (RF), Gradient Boosting (GB), k-Nearest Neighbor (KNN), Support Vector Machine (SVM)	21,910
Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding **(S. Bagui, 2021)**	Decision Tree (DT), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM), CNN LSTM	18,366
A hybrid DNN-LSTM model for detecting Phishing URLs **(A. Ozcan, C. Catal, E. Donmez, and B. Senturk, 2021)**	Decision Tree (DT), Random Forest (RF) XgBoost, AdaBoost k-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), LightGBM Ridge Regression LASSO, CNN, DNN RNN, LSTM, BiLSTM, DNN+LSTM, DNN+BiLSTM	99,575
Phishing Attacks: Detecting and Preventing Infected E-mails Using Machine Learning Methods **(D. Ona, 2019)**	Algoritmo de selección de características Algoritmo de redes neuronales	3,000

Título	Algoritmos utilizados	Cantidad de datos
Phishing attack detection: a solution based on the typical Machine Learning modeling cycle **(B. Espinoza, 2019)**	Naive Bayes (NB), Decision Tree (DT), ML Random Forest (RF), Logistic Regression (LR), Fictitious Classifier (FC)	1,325
A Discrete Hidden Markov Model for SMS Spam Detection **(Tain Xia, 2020)**	HMM	5574

Nota. Esta tabla muestras las investigaciones principales en las que se centra nuestra revisión de la literatura que son: (K. Haynes, 2021), (S. Bagui, 2021), (A. Ozcan, C. Catal, E. Donmez, and B. Senturk, 2021), (D. Ona, 2019), (B. Espinoza, 2019) y (Tain Xia, 2020).

Los autores del artículo “Lightweight URL-based Phishing detection using natural language processing transformers for mobile devices” (K. Haynes, 2021) proponen un algoritmo que permite detectar Phishing en sitios web legítimos, únicamente usando su URL. El algoritmo utiliza trasformadores pre entrenados principalmente por los temas de actualización, y el tiempo de respuesta que estos pueden ofrecer, a diferencia de un trasformador creado a partir de una lista negra. Se desarrolla una comparativa entre distintos algoritmos para la detección de sitios webs Phishing, de los cuales ninguno supera el 96% de precisión resultante en esta investigación.

En la investigación “Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding” (S. Bagui, 2021), se plantea un nuevo método para realizar análisis semántico profundo en textos y poder detectar de manera eficiente cualquier tipo de ataque Phishing mediante correo electrónico. Los autores utilizan codificación One-HotSe en conjunto con técnicas de DL y ML para lograr una mayor eficacia. Al finalizar el estudio, los autores establecen que los algoritmos DL funcionan mejor que los de ML en términos de precisión, pero los modelos de ML funcionan mejor que los de DL en términos de tiempo de

cálculo. Se presenta el algoritmo CNN con Word Embedding y se evidencia que su porcentaje de precisión supera el 95% en comparación con otras pruebas realizadas.

La investigación realizada por (A. Ozcan, C. Catal, E. Donmez, and B. Senturk, 2021) titulada “A hybrid DNN–LSTM model for detecting Phishing URLs”, propone nuevos modelos híbridos basados en corta memoria y algoritmos de redes neurales capaces de detectar el patrón usado en las URLs Phishing, logrando así un porcentaje de precisión superior al de otros modelos de detección. Los modelos presentados utilizan una variedad de funciones de incorporación de caracteres, en conjunto con Programación Neurolingüística (NLP), lo que permite obtener como resultado un porcentaje superior al 98%.

Los autores del artículo “Phishing Attack Detection: A Solution Based on the Typical Machine Learning Modeling Cycle” (B. Espinoza, 2019), proponen un modelo para la detección de ataques Phishing mediante técnicas de ML supervisado. El modelo que presentan los autores es una combinación entre los algoritmos Naive Bayes y Decision Tree, sin embargo, para lograr un mayor porcentaje de precisión se realizaron pruebas con los algoritmos más aceptados en el campo científico, con lo cual lograron obtener el 96,7% de precisión.

La investigación de (D. Ona, 2019) titulada “Phishing Attacks: Detecting and Preventing Infected E-mails Using Machine Learning Methods” plantea el uso de una herramienta capaz de detectar ataques Phishing y encontrar una solución para contrarrestar posibles ataques. La implementación que realizan los autores está basada en Scrum. La información para realizar las pruebas de presión la obtienen de una lista negra descargada de Phishtank. El porcentaje de precisión resultante supera el 90%, por lo cual definen que el empleo de redes neuronales es de gran importancia para establecer un nivel óptimo de aprendizaje sin redundancia alguna.

En la investigación “A Discrete Hidden Markov Model for SMS Spam Detection” (Tain Xia, 2020) se propone un Modelo de Oculto de Markov (HMM) discreto, para la detección de spam en SMS, el rendimiento general del algoritmo propuesto es compatible con el aprendizaje

profundo mediante el empleo de modelos CNN y LSTM. Para la evaluación de rendimiento, los autores utilizan un conjunto de datos de spam de SMS chinos con 2000 mensajes. La precisión obtenida supera el 95% en distintas escalas de pruebas, y se establece que dicha precisión podría mejorar si el modelo HMM se entrena lo suficiente. Por último, se establece que el modelo presentado no es sensible al idioma.

A continuación, se describen los conceptos necesarios básicos para la comprensión de este documento:

Ingeniería Social

La Ingeniería Social es una técnica que consiste en la manipulación de usuarios, explotando sus errores por medio de engaños para obtener información privada, acceso u objetos de valor. Estas estafas de delincuencia informática tienden a atraer a usuarios desprevenidos para que expongan datos de cualquier tipo, propaguen infecciones de malware o den acceso a sistemas restringidos de ciberseguridad. Los ataques pueden producirse por distintos medios como: en línea, en persona o a través de otras interacciones.

Kalnins (2017) afirma: “la Ingeniería Social tiene como objetivo manipular a individuos y empresas para que divulguen datos valiosos y sensibles, en interés de los ciberdelincuentes”, mientras que Kaabouch (2019) plantea: “la Ingeniería Social desafía constantemente la seguridad de todas las redes, independientemente de la fortaleza de sus cortafuegos, métodos criptográficos y sistemas de software antivirus”.

Spam

El Spam es cualquier tipo de comunicación digital no deseada y no solicitada, que se envía en masa. Kujama (2022) escribe: “a menudo, el Spam se envía por correo electrónico, pero también puede distribuirse a través de mensajes de texto, llamadas telefónicas o redes sociales”.

Tipos de Spam

Los Spammers utilizan muchas formas de comunicación para enviar sus mensajes no deseados de forma masiva. Algunos de ellos son mensajes de marketing que venden productos no solicitados. Según Kujama (2022): “Otros tipos de mensajes de Spam pueden propagar malware, engañar al usuario para que divulgue información personal o asustarle haciéndole creer que tiene que pagar algo para salir del apuro”.

Correos electrónicos de Phishing. Los correos electrónicos de Phishing son un tipo de Spam que los ciberdelincuentes envían a muchas personas con la esperanza de "enganchar" a unas pocas. Los correos electrónicos de Phishing engañan a las víctimas para que faciliten información sensible, como los datos de acceso a sitios web o la información de las tarjetas de crédito, entre otros datos.

(Kujama, 2022) afirma: "El Phishing es el tipo de ciberataque más sencillo y, al mismo tiempo, el más peligroso y eficaz. Esto se debe a que ataca al ordenador más vulnerable y poderoso del planeta: la mente humana".

Falsificación de correos electrónicos. Los correos electrónicos falsos imitan a un correo electrónico de un remitente legítimo, y le piden que realice algún tipo de acción. Las falsificaciones bien ejecutadas contienen marcas y contenidos familiares, a menudo de una gran empresa conocida, como PayPal o Apple. Los mensajes de Spam de falsificación de correo electrónico más comunes son los siguientes:

- Una solicitud de pago de una factura pendiente.
- Una solicitud para restablecer su contraseña o verificar su cuenta.
- Verificación de compras que no ha realizado.
- Solicitud de información de facturación actualizada.

Estafas de soporte técnico. En una estafa de soporte técnico, el mensaje de Spam indica que el usuario tiene un problema técnico y que debe ponerse en contacto con el soporte

técnico, llamando al número de teléfono o haciendo clic en un enlace del mensaje. Según Kujama (2022): “Al igual que en el spoofing de correo electrónico, estos tipos de Spam suelen mencionar que son de una gran empresa tecnológica, como Microsoft, o de una empresa de ciberseguridad como Malwarbites”.

Estafas de actualidad. Los temas de actualidad pueden utilizarse en los mensajes de Spam para llamar su atención. En 2020, cuando el mundo se enfrentaba a la pandemia de Covid-19 y había un aumento de los trabajos desde casa, algunos estafadores enviaron mensajes de Spam que prometían trabajos a distancia pagados en Bitcoin. Durante el mismo año, otro tema de Spam muy popular estaba relacionado con la oferta de ayuda financiera para las pequeñas empresas, pero los estafadores acababan pidiendo los datos de la cuenta bancaria. Los titulares de las noticias pueden ser pegadizos, pero tenga cuidado con ellos en lo que respecta a posibles mensajes de Spam.

Estafas por adelantado. Este tipo de Spam es probablemente familiar para cualquiera que haya utilizado el correo electrónico desde los años 90 o 2000. A veces llamados correos electrónicos del "príncipe nigeriano", ya que ese fue el supuesto remitente del mensaje durante muchos años. Este tipo de Spam promete una recompensa económica si primero se proporciona un adelanto en efectivo. El remitente suele indicar que este adelanto en efectivo es una especie de tasa de tramitación o dinero en efectivo para desbloquear una suma mayor, pero una vez que se paga ese adelanto, el estafador desaparece. Para hacerlo más personal, un tipo de estafa similar consiste en que el remitente se hace pasar por un miembro de la familia que tiene problemas y que necesita dinero, lamentablemente también resulta en una estafa.

MalSpam. Abreviatura de "Spam de malware" o "Spam malicioso", el malSpam es un mensaje de Spam que envía malware a tu dispositivo. Los lectores desprevenidos que hacen clic en un enlace o abren un archivo adjunto al correo electrónico, acaban recibiendo algún tipo

de malware, como ransomware, troyanos, bots, ladrones de información, criptomineros, spyware y keyloggers. Un método de entrega habitual, es incluir scripts maliciosos en un archivo adjunto de tipo familiar, como un documento de Word, un archivo PDF o una presentación de PowerPoint. Una vez abierto el archivo adjunto, los scripts se ejecutan y recuperan la carga útil del malware.

Llamadas y mensajes de Spam. ¿Ha recibido alguna vez una llamada robótica? Eso es Spam de llamadas. Por otro lado, ¿Ha recibido un mensaje de texto de un remitente desconocido que le insta a hacer clic en un enlace desconocido?, eso se llama Spam de mensajes de texto o "smishing", que es una combinación de SMS y Phishing.

Phishing

Según Benavides (2020), “El Phishing es un intento fraudulento, generalmente realizado a través del correo electrónico o de páginas web, para robar información personal”. Por otra parte: El Phishing es la herramienta más popular entre los hackers para ejecutar ataques con el fin de obtener información sensible, como las credenciales de nuestra cuenta personal, la información de la cuenta bancaria y, a veces, la información de las redes sociales, engañando al usuario para que pague en la cuenta del hacker (Indrasiri, 2021).

Se produce cuando un atacante, haciéndose pasar por una entidad de confianza, engaña a la víctima para que abra un correo electrónico, un mensaje instantáneo, un mensaje de texto o acceda a un sitio web. Según Indrasiri (2021) “el destinatario es engañado para que haga clic en un enlace malicioso, lo que resulta en la instalación de malware, la congelación del sistema como parte de un ataque de ransomware o la divulgación de información sensible”.

Principales vectores de ataque

Falsificación de correos electrónicos de Phishing.

La mayoría de los correos electrónicos de Phishing utilizan técnicas sociales, en luagra de trucos técnicos para engañar a los usuarios finales.

La transmisión de urgencia es un método bien conocido, utilizado por los delincuentes para desviar la atención de la gente; un ejemplo es hacerse pasar por un administrador de sistemas que advierte a los usuarios de un nuevo ataque, instándoles a instalar el parche adjunto. Otro, es notificar a las personas que varios inicios de sesión en su cuenta han fallado y que deben comprobar su cuenta o arriesgarse a ser comprometidos. (Hong, 2012)

Sitios web falsos.

La mayoría de los ataques de Phishing intentan convencer a los usuarios de que vayan a un sitio falso en el que se recoge información personal. Hong (2012): “Los estafadores pueden alojar un sitio falso utilizando espacio web gratuito y una máquina comprometida o registrar un nuevo dominio “.

Machine Learning

Es una colección de algoritmos que aprenden y predicen a partir de datos registrados, su funcionamiento radica en optimizar una función de utilidad dada bajo incertidumbre, extraen estructuras ocultas de los datos y realizan descripciones de los mismos. Machine Learning suele aplicarse cuando la programación explícita es poco práctica o demasiado rígida. A diferencia del código normal realizado por los desarrolladores de software para generar una salida de código de programa específico a partir de una entrada determinada, Machine Learning utiliza los datos para generar un código estadístico que incluye la salida. Los algoritmos utilizados en nuestro estudio son los siguientes:

Naive Bayes (NB)

Los clasificadores Naive Bayes, son una familia de clasificadores basados en el famoso teorema de la probabilidad de Bayes, conocidos por crear modelos simples y potentes, especialmente en la clasificación de documentos y la predicción de enfermedades. Según Abbas (2019): "La clasificación textual NB es la más utilizada para clasificar textos, ya que es

rápida y fácil de implementar. Los algoritmos con menos fallos suelen ser más lentos y complejos".

Support Vector Classifier (SVC)

Según Jordan (2018): "El clasificador de vectores de soporte (SVC) es una poderosa herramienta para resolver problemas de reconocimiento de patrones", de tal manera que "la solución está totalmente descrita como una combinación lineal de varias muestras de entrenamiento, llamadas vectores de soporte" (Cruz et al, 2000). El procedimiento de entrenamiento para resolver problemas de vectores de soporte, se basa normalmente en la programación cuadrática, con algunas limitaciones inherentes; principalmente la complejidad computacional y los requisitos de memoria para grandes conjuntos de datos de entrenamiento.

Decision Tree (DT)

El árbol de decisión es un algoritmo de aprendizaje supervisado que puede utilizarse para resolver problemas de regresión y clasificación. El objetivo de utilizar un árbol de decisión es crear un modelo de entrenamiento que pueda predecir la clase o el valor de la variable objetivo mediante el aprendizaje de reglas de decisión simples deducidas de datos anteriores (entrenamiento). "La idea básica de este algoritmo de varias etapas, es descomponer una decisión compleja en varias decisiones más sencillas, con la esperanza de que la solución final así obtenida, se parezca a la solución deseada" (Landgrebe, 1991).

K-Nearest Neighbors Classifier (K-NN)

K-NN es un algoritmo de gran versatilidad que es utilizado para imputar valores perdidos y remuestrear conjuntos de datos. K-NN, para su rendimiento, considera a los K vecinos más cercanos para predecir la clase o el valor continuo del nuevo punto de datos.

K-NN es un algoritmo de aprendizaje supervisado que realiza procesos de clasificación sin construir modelos de antemano; espera los datos sin clasificar y luego trabaja en el algoritmo para hacer la predicción de la clasificación. Por lo tanto, consume mucho tiempo, ya

que cada vez que hay que hacer una predicción, hay que rehacer todo el esfuerzo de construcción del modelo. (Pandya, 2016).

Logistic Regression (LR)

La regresión logística es un algoritmo de Machine Learning supervisado, desarrollado para aprender problemas de clasificación. Un problema de aprendizaje de clasificación es cuando la variable objetivo es categórica. Atlman (1992): “El objetivo de la regresión logística es asignar una función de las características del conjunto de datos a los objetivos para predecir la probabilidad de que un nuevo ejemplo pertenezca a una de las clases objetivo”.

Random Forest (RF)

Según Louppe (2014): “Los bosques aleatorios son, sin duda, una de las herramientas más robustas, precisas y versátiles para resolver tareas de Machine Learning”. Este algoritmo de clasificación consiste en muchos árboles de decisión individuales que operan como un conjunto. Cada árbol del bosque aleatorio brinda una predicción de clase, y la clase con más votos se convierte en la predicción del modelo.

Natural Language Processing (NLP)

Natural Language Processing (NLP), es una disciplina que estudia los problemas del lenguaje en la interacción hombre-ordenador, utilizando técnicas de Machine Learning para entender la estructura y el significado de las palabras. Con las aplicaciones de NLP, las organizaciones pueden analizar textos, extraer información sobre personas, lugares y eventos, y obtener más información sobre el sentimiento de las publicaciones en las redes sociales y las conversaciones con los clientes.

Natural Language Processing es una rama de la inteligencia artificial, que permite a los ordenadores entender el lenguaje humano (escrito y hablado). Como disciplina científica, el NLP implica la identificación de la estructura y los límites de las frases en los documentos, la detección de palabras clave o frases en las grabaciones, la extracción de relaciones entre los

documentos y el descubrimiento de significados en patrones informales o de jerga. (S. Salloum, 2021, p.19).

NLP basado en Machine Learning tradicional. NLP puede clasificar las tareas de procesamiento y formar múltiples subtareas. Los métodos tradicionales de Machine Learning pueden utilizar SVM. Crowston (2012): “Métodos como Markov (Markov Model) y Conditional Random Field Model (CRF Model), procesan múltiples subtareas en el lenguaje natural para mejorar la precisión de los resultados del procesamiento”.

Técnicas y métodos de NLP

“NLP utiliza la sintaxis para evaluar el significado de un lenguaje en términos de reglas gramaticales” (Crowston, 2022). Las técnicas de sintaxis incluyen:

- Parsing: Se aplica a la parte gramatical de una frase.
- Segmentación de palabras: Se selecciona una cadena de texto para derivar diferentes estructuras de palabras.
- Fragmentación de oraciones: Divide las frases en fragmentos.

Pasos en la ejecución de la NPL. La clave del NLP es que el ordenador entienda el lenguaje natural, por lo que NLP también se denomina comprensión del lenguaje natural, también conocido como lingüística computacional. Por otro lado, es uno de los temas centrales de la inteligencia artificial.

Básicamente, un algoritmo de Natural Language Processing se divide en tres fases:

- Preprocesamiento
- Construcción del modelo
- Ejecución / Resultados finales

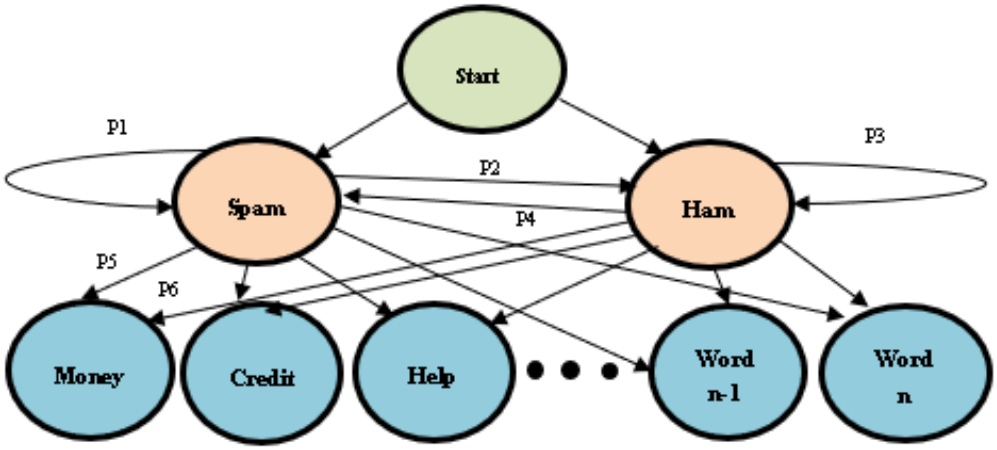
Hidden Markov Model (HMM)

HMM es un tipo de modelo gráfico, comúnmente utilizado para simular datos temporales. A diferencia de los modelos de Markov tradicionales, HMM supone que los datos

observados no son el estado real del modelo, sino que son generados por la capa oculta subyacente (H en HMM). Mientras que esto suele dificultar la inferencia, los HMM basados en Markov (M primaria en los HMM) hacen que la inferencia sea eficiente. La Figura 1 muestra un diseño descriptivo del modelo aplicado a este estudio.

Figura 1.

Diseño de un HMM para la detección de correo basura



Nota. Esta figura muestra el diseño de un algoritmo HMM para la detección de correo spam mediante la segmentación de palabras Ham o Spam.

Para entender mejor, veamos el modelo HMM de la Figura 1, adaptado a nuestra investigación. Para empezar, tenemos un estado inicial que luego indica si un correo electrónico es Spam o Ham. La capa que indica si es Spam o Ham, estará oculta para cualquier agente externo; por lo tanto, la única capa visible es la inferior, donde se describen las palabras Dinero, Crédito y Ayuda, hasta la n-ésima palabra. Además, las flechas del modelo indican las posibles transiciones y probabilidades de pasar de un estado oculto a otro.

Por ejemplo, en el estado Spam, se observa que hay una línea de salida hacia Ham con probabilidad P2 y otra línea que hace un bucle hacia sí misma con probabilidad P1. Esta suma de probabilidades será del 100%. El mismo análisis se haría con las líneas que salen de Ham.

Ahora analicemos de la Figura 1 la palabra Dinero. Si bien se sabe que la palabra Dinero aparece en un gran número de correos electrónicos Spam, también esta palabra puede aparecer en los mensajes Ham, pero después del entrenamiento, se puede definir que Dinero aparece mucho más en los mensajes Spam que los de Ham (Por ejemplo, la palabra dinero podría tener un 80% de probabilidades de aparecer en Spam y un 20% de pertenecer a un mensaje de Ham). Así, se puede definir que la suma de probabilidades P5 + P6 sería del 100%. Finalmente, como esta última capa de palabras es la capa visible, podríamos predecir si la capa anterior nos da como resultado Spam o Ham, en función del texto contenido en cada mensaje de correo electrónico.

Proceso HMM

Si un modelo se puede describir como HMM, entonces se pueden resolver tres problemas. Los dos primeros son problemas de reconocimiento de patrones (Landgrebe, 1991, p.674):

1. Obtener la probabilidad (evaluación) de una secuencia de estados observables.
2. Encontrar una secuencia de estados ocultos que maximice la probabilidad de que esta secuencia produzca una secuencia de estados observables (decodificación).
3. El tercer problema consiste en generar un HMM a partir de un conjunto de secuencias de estados observables.

Métricas

Accuracy (Exactitud)

Representa la relación entre las muestras predichas correctamente y el número total de muestras. La métrica Accuracy tiene un buen funcionamiento para conjuntos de datos equilibrados (Han, 2012). El accuracy para los modelos se calcula mediante la fórmula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Precision (Precisión)

Representa una medida de precisión o exactitud, es decir, que porcentaje de datos etiquetados como positivos lo son realmente (Han, 2012). Se calcula mediante la siguiente fórmula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

False Positive Rate (Tasa de falsos positivos)

Representa la tasa de datos negativos que fueron clasificados o predichos incorrectamente como positivos (Han, 2012). Se calcula mediante la siguiente fórmula:

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \tag{3}$$

False Negative Rate (Tasa de falsos negativos)

Representa la tasa de datos positivos que fueron clasificados o predichos incorrectamente como negativos (Han, 2012). Se calcula mediante la siguiente fórmula:

$$\text{FNR} = \frac{\text{FN}}{FN + TP} \tag{4}$$

Error Rate

Representa la relación entre el número total de muestras predichas incorrectamente y el número total de muestras (Han, 2012). Se calcula mediante la siguiente fórmula:

$$\text{ERR} = \frac{FP + FN}{\text{TP+TN+FP+FN}} \tag{5}$$

Capítulo III

Metodología

Para llevar a cabo esta investigación, se siguieron los siguientes pasos:

1. Se realizó una Revisión de la Literatura y se determinaron las características de las soluciones orientadas a enfrentar ataques de Phishing utilizando NLP.

2. Se buscó un Dataset adecuado y se procedió a limpiarlo y pre procesarlo para entrenar los algoritmos de Inteligencia Artificial.

3. Se accedió al servidor RIG de la ESPE-SD y se instalaron y levantaron los servicios necesarios para realizar la ejecución de los algoritmos.

4. Se realizó la ejecución del algoritmo basado en HMM con el Dataset.

5. Se realizó la ejecución de los algoritmos de Machine Learning con el mismo Dataset.

6. Se procedió a comprar los resultados obtenidos de la ejecución de los algoritms HMM vs los algoritmos de Machine Learning.

Revisión sistemática de la literatura

Para ubicar nuestra investigación, se realizó una revisión del estado actual del arte en la lucha contra los ataques de Spam utilizando NLP. Así, se efectuó una Revisión de la Literatura utilizando la metodología de Barbara Kitchenham (B. Kitchenham, 2009).

Definición de la cadena de búsqueda

La cadena de búsqueda utilizada en las diferentes bases de datos científicas fue (Spam and NLP) o (Spam and "Natural Language Processing"), que dio como resultado 24 artículos científicos. Los artículos científicos se introdujeron en la herramienta Rayyan.ai (Rayyan, 2018) para clasificar los que eran relevantes para nuestra investigación.

Criterio de Calidad

Como criterio de calidad, se verificó que los artículos se encontraran en bases de datos científicas de alto impacto como Scopus o en Web of Science, y que estuvieran escritos en inglés.

Criterios de Inclusión y Exclusión

Como criterios de inclusión y exclusión, se valoró que no se tratase sólo de un estudio o propuesta, sino que fueran trabajos con resultados, usen alguna forma de aplicación de NLP para detectar el Spam, y que utilizaran alguna técnica de Inteligencia Artificial. Finalmente, se preseleccionaron 19 artículos, de los cuales haciendo una lectura completa, solo hay cuatro similares a nuestra propuesta y uno de ellos propone un método HMM similar al implementado en este estudio, sin embargo, el algoritmo propuesto maneja una técnica de etiquetado según la propiedad de cada mensaje, lo cual es distinto en el caso del algoritmo implementado en esta investigación. La Tabla 2 muestra los principales resultados de la revisión de la literatura.

Selección del Dataset

Tras la revisión de los artículos científicos, se determinó que el Dataset más apropiado para nuestro estudio es el ofrecido por Shantanu en Kaggle (Dhakad, 2018). Este Dataset consta de 5172 registros, de los cuales el 87% son ham y el 13% son spam. En este punto, previo a la utilización del Dataset, se eliminaron todos los correos cuyo contenido sea menor a 20 caracteres, para que al pasarlos por los procesos de tokenización, stemmer, stop words, estos registros se queden sin contenido, es decir, sin ninguna palabra. Finalmente, fue posible obtener alrededor de 600 registros de spam y 600 de ham para analizar el comportamiento de la detección, con lo que además se obtuvo dos clases balanceadas.

En base a la revisión de la literatura y al tiempo designado para la realización de este estudio, se determinó que los 1200 registros obtenidos del proceso de filtrado del Dataset eran

suficientes para realizar las pruebas de rendimiento tanto para los 6 algoritmos de ML como para el algoritmo de HMM.

Evaluamos previamente la alternativa de trabajar con Datasets provenientes de escenarios reales, sin embargo, en base a la revisión de la literatura, en la mayoría de los otros proyectos, no se vio la necesidad de trabajar con datos en escenario reales. Por otro lado, el recolectar esta cantidad de información, hubiese llevado más tiempo del planificado para esta tesina.

Levantamiento de entorno de pruebas

En este punto, se procedió acceder al servidor RIG de la Universidad de Las Fuerzas Armadas ESPE de la Sede Santo Domingo, mediante los permisos y credenciales de acceso tramitadas por el tutor a cargo. Una vez ingresados en el servidor, se procedió a instalar y configurar Jupyter y Python para la implementación de los algoritmos, además de las librerías necesarias para la ejecución.

Ejecución de algoritmos

Una vez configurado el entorno de pruebas e instaladas las herramientas necesarias, se procedió a ejecutar los algoritmos de ML y el algoritmo de HMM.

- Para ejecutar los algoritmos de ML escribimos el código el especificado en el link que sigue a continuación, y que está subido a github como un aporte de nuestra investigación:

  https://github.com/kevinmonteros1/Algoritmos-ML/tree/master.

  Los algoritmos de ML utilizados fueron adaptados para nuestro estudio, a partir de los algoritmos encontrados en:

  https://www.milindsoorya.com/blog/build-a-spam-classifier-in-python

- Para ejecutar el algoritmo de HMM, usamos el especificado en el link que sigue a continuación, y que está subido a github como otro aporte de nuestra investigación:

  https://github.com/kevinmonteros1/Algoritmo-HMM/tree/master.

El algoritmo de HMM utilizado fue adaptados para nuestro estudio, a partir del algoritmo encontrado en:

https://github.com/FantacherJOY/Hidden-Markov-Model-for-NLP/blob/master/hmm.py

Recopilación de resultados

Para poder realizar el análisis de los resultados obtenidos y plantear la discusión correspondiente al estudio, se procedió a detallar los porcentajes de accuracy en diagramas estadísticos para una mejor visualización, adicional a esto, se presenta la matriz de confusión que se obtuvo para ambos casos.

Tipo de investigación

La investigación presentada como trabajo de tesis es de tipo cuantitativo y descriptivo. La línea de investigación a la que se apega el presente trabajo maneja valores concretos en cuanto a exactitud y precisión, por ende, la metodología usada para la recopilación de información de tipo cuantitativo es presentada mediante la tabulación de resultados y comparación de estos resultados.

En base a los objetivos planteados previamente, se realizó una revisión de la literatura, para poder establecer el estado en el que se encuentra el tema investigado, luego se realizó una búsqueda y análisis para seleccionar el Dataset a utilizar en la implementación de los algoritmos de ML y DL. Una vez realizada la implementación de dichos algoritmos, se realizó una comparativa entre los resultados obtenidos.

El resultado de la investigación es descriptivo, esto se debe a que se presenta un análisis profundo a los datos obtenidos en la ejecución de los algoritmos. El enfoque aplicado en este proyecto abre una nueva línea de investigación para contrarrestar los ataques de Ingeniería Social, específicamente de Spam de Phishing.

Capítulo IV

Resultados y Discusión

Resultados

Resultados de la revisión sistemática de la literatura

En la Tabla 2, se muestran los resultados de la revisión de la literatura. Para esto, se presenta cada artículo científico con los siguientes datos: email o web page (corresponde a la fuente del ataque, ya sea por correo electrónico o por página web), siglas del algoritmo de ML aplicado, porcentaje de Accuracy obtenido en el artículo, fuente del Dataset utilizado y el número de registros del Dataset con el que se obtuvieron los resultados.

Tabla 2.

Resultados de la revisión sistemática de la literatura.

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
1	(Haynes, 2021)	Web page	DT GNB RF GB KNN SVM	96.00%	Phishtank.com OpenPhish.com CommonCrawl.org	21,910
2	(Bagui et al, 2021)	E-mail	DT GNB SVM	96.34%	App River	18,366

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
3	(Ozcan, et al, 2021)	Web page	DT RF XgBoost AdaBoost KNN GNB Light GBM Ridge Regression LASSO	99.21%	UCI Machine Learning Repository: Phishing Websites Dataset Alexa, openphish, Spamhaus.org, techhelplist.com Alexa, hphosts, Joewein, malwaredomains, and Phishtank	99,575
4	(Xiao, 2021)	Web page	CNN	95.60%	5000 Best Websites homepage (2012) Phishtank homepage (2019)	80,033
5	(Alsufyani, 2021)	E-mail	KNN MNB DT	90.00%	Wikileaks archives, Democratic	10,306

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
			AdaBoost		National Committee, Hacking Team, Sony emails	
6	(Junnarkar, 2021)	E-mail	GNB SVM DT RF	94.64%	ND	110
7	(Indrasiri et al, 2021)	Web page	DT RF XgBoost AdaBoost GB KNN LR	98.27%	Buber et al. (2019) NetSec Explained website UCI database	173,575
8	(Yaseen, 2021)	E-mail	ND	98.67%	Spambase del repository of Machine Learning of UCI Spam filter with Kaggle	11,297

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
9	(Sirigineedi, 2020)	Web page	KNN LR SVM GBC ABC RFC	96.60%	Dataset of git-hub (Phishing_ Detection)	73,575
10	(Kumar, 2020)	E-mail	SVM NN	98.00%	ND	1,705
11	(Sahingoz, 2019)	Web page	NB RF KNN Adaboost K-star SMO RT	97.00%	Dataset of github (Ebbu2017)	73,575
12	(Verma, 2019)	E-mail	A total of 17 models were used, the two that stand out are:	80.00%	Dataset of IWSPA	3,865

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
			DT MNB			
13	(Thakur, 2018)	E-mail	SVM	87.00%	WordNet VerbNet PhishMonger	286,000
14	(Buber, 2018)	Web page	RF SMO NB	97.20%	Phishtank	7,357
15	(Buber, Dırı, & Sahingoz, 2017)	Web page	RF SMO NB	89.90%	ND	10,572
16	(R. Verma, 2012)	E-mail	ND	97.00%	Dataset of github (PhishCatch)	2,000
17	(Ona et al, 2019)	E-mail	ND	93.90%	Phishtank	3,000
18	(Espinoza, 2019)	E-mail	NB DT RF LR	96.77%	Phishtank	1,325

Nro	Paper	E-mail / web page/SMS	Algoritmo de ML	Accuracy	Dataset	Nro de Registros
19	(Xia y Chend, 2020)	SMS	HMM	98.50%	Repositorio UCI	2000

Nota. Esta tabla muestra las principales características de las 19 investigaciones abarcadas en este estudio, en que se abarca el combate de Spam de Phishing con NLP.

Ejecución de los algoritmos de Machine Learning

La Figura 2, muestra los resultados obtenidos al aplicar los seis algoritmos de ML sobre el Dataset:

Figura 2.

Resultados de accuracy obtenidos con los algoritmos de ML.

La Figura 3, muestra la matriz de confusión que se obtuvo al ejecutar el algoritmo de ML que obtuvo mayor Accuracy:

Figura 3.

Matriz de confusión en algoritmos de ML.

Ejecución de los algoritmos HMM

La Figura 4, muestra los resultados obtenidos al aplicar el algoritmo HMM junto a los valores obtenidos con los seis algoritmos de ML sobre el mismo Dataset:

Figura 4.

Resultados de accuracy obtenidos con HMM.

La Figura 5, muestra la matriz de confusión que se obtuvo al ejecutar el algoritmo HMM:

Figura 5.

Matriz de confusión en algoritmos HMM.

Discusión

Cabe destacar que este trabajo está orientado a diferenciar la exactitud obtenida con un algoritmo aplicando NLP, concretamente HMM; frente a seis algoritmos tradicionales de Machine Learning. Así, con base en los resultados obtenidos, se puede observar que el algoritmo HMM da como resultado una exactitud del 76%, es decir, es inferior a cualquiera de los algoritmos de Machine Learning que implementamos (DT = 96%, SVC = 95 %, KNN = 89 %, NB = 98%, RF = 95%, y LR = 90%). A primera vista, se puede observar que HMM da resultados más bajos que ML; sin embargo, hay que tener en cuenta que también en ML se realizó un análisis semántico utilizando NLTK.

Cabe mencionar que los tiempos de procesamiento en la ejecución de los algoritmos de ML fueron de aproximadamente cinco minutos, mientras que en el caso de la ejecución del algoritmo de HMM tardó un total de dos días de procesamiento con el mismo dataset. Esta diferencia de tiempo tan grande se debe a la manera de trabajar de HMM con su análisis semántico y sintáctico, que incrementa exponencialmente los tiempos de procesamiento.

Capítulo V

Conclusiones, Trabajo Futuro, y Recomendaciones

Conclusiones

En este trabajo primero se realizó una revisión de la literatura, de la que se obtuvieron las principales características obtenidas de las soluciones orientadas a resolver problemas de Spam utilizando NLP, en segundo lugar, se realizó un estudio comparativo del uso del algoritmo HMM versus los algoritmos de Machine Learning tradicionales.

Los resultados obtenidos en la ejecución de los algoritmos no tuvieron mayor variación en cuanto a la exactitud según la cantidad de datos, sin embargo, el algoritmo de HMM implementado varía su porcentaje de precisión en gran escala al momento de reducir o aumentar la cantidad de datos.

El tiempo de procesamiento de datos en el caso del algoritmo HMM es muy alto en comparación con ML. Esto se debe a que analiza de manera profunda cada palabra, y realiza predicciones en base a dicho análisis. Los algoritmos de ML manejan un tiempo de respuesta bastante corto, debido a que el funcionamiento de estos, no se basa en un análisis semántico o sintáctico profundo.

En base al estudio realizado, se puede afirmar que, los algoritmos de ML destacan principalmente en su exactitud obtenida y en su tiempo de procesamiento. El porcentaje de exactitud en los algoritmos de ML varía mucho, según parámetros como: cantidad de datos usados y medios de procesamiento utilizados.

Trabajo futuro

Como trabajo futuro, pretendemos desarrollar un algoritmo que ofrezca mayor precisión, en el que el texto de entrada sea analizado primero por el algoritmo HMM y luego introducido en un conjunto de algoritmos de ML, concretamente DL.

Otra propuesta es la de implementar alguna solución de software a manera de plug-in de los navegadores, que permita la implementación de HM, DL o HMM, a fin, de que dicho programa pueda detectar un correo de Spam antes de ser abierto. De esta manera, se podría advertir a tiempo al usuario sobre un posible ataque de Ingeniería Social.

Recomendaciones

En base a la investigación realizada, hasta el momento se siguen recomendando las alternativas que ofrece ML en la detección de correos de Spam debido a su alto porcentaje de exactitud. Sin embargo, HMM mantiene una línea de investigación latente y no explorada, a diferencia de muchos algoritmos que con el pasar del tiempo, han llegado a quedar obsoletos al trabajar de manera autónoma. Es recomendable que se sigan expandiendo más investigaciones alrededor de HMM, ya que es un modelo bastante prometedor por sus capacidades de análisis semántico y sintáctico.

Se recomienda, además, contar con equipos de cómputo con la suficiente capacidad para realizar la ejecución de algoritmos y procesamiento de datos en gran volumen y de gran velocidad, ya que para realizar la ejecución del algoritmo HMM se contó con un servidor de excelentes características.

Bibliografía

A. Abbas, K. A. (2019). *Multinomial Naive Bayes Classification Model for Sentiment Analysis.* IJCSNS Int. J. Comput. Sci. Netw. Secur.

A. Junnarkar, S. A. (2021). *E-mail spam classification via machine learning and natural language processing.* Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV.

A. Kumar, J. M. (2020). *A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing.* Int. J. Electr. Comput. Eng.

A. Ozcan, C. Catal, E. Donmez, and B. Senturk. (2021). *A hybrid DNN-LSTM model for detecting phishing URLs.* Neural Comput. Appl.

al, E. B.-A. (2022). *Analysis of Vulnerabilities Associated with Social Engineering Attacks Based on User Behavior.*

al., E. B.-A. (2022). *A Framework Based on Personality Traits to Identify Vulnerabilities to Social Engineering Attacks.*

Albladi, S. y Weir, G. (2020). *Predicting individuals' vulnerability to social engineering in.*

Atlman, N. S. (1992). *An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression.* AM. Stat.

Azzopardi, D. E. (2008). *Assessing multivariate Bernoulli models for information retrieval.* ACM Trans. Inf. Syst.

B. Espinoza, J. S. (2019). *Phishing attack detection: A solution based on the typical machine learning modeling cycle.* Proc. - 6th Annu. Conf. Comput. Sci. Comput. Intell. CSCI.

B. Kitchenham, O. P. (2009). *Systematic literature reviews in software engineering .* Inf. Softw. Technol.

Borghello, C. (2009). *El arma infalible: la Ingeniería Social.* ESET Latinoamérica.

Buber, E., Dırı, B., & Sahingoz, O. K. (2017). *Detecting phishing attacks from URL by using NLP techniques.* Antalya, Turkey.

D. Ona, L. Z. (2019). *Phishing Attacks: Detecting and Preventing Infected E-mails Using Machine Learning Methods.* 3rd Cyber Secur. Netw. Conf. CSNet.

Dhakad, S. (2018). *Email Spam Detection Dataset (classification).* https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-dataset-classification.

E. Benavides, W. F. (2020). *Caracterización de los ataques de phishing y técnicas para mitigarlos. Ataques: una revisión sistemática de la literatura.* Cienc. y Tecnol.

E. Buber, B. D. (2018). *NLP Based Phishing Attack Detection from URLs.* Adv. Intell. Syst. Comput.

F. Pérez-Cruz, P. A.-D.-V.-R. (2000). *Fast training of support vector classifiers.* Adv. Neural Inf.

Han, J. (2012). *Data mining concepts and techniques .* Data Mining (Third Edition) (Third Edit). Morgan Kaufmann.

Hong, J. (2012). *The state of phishing attacks.* ACM.

Islam, F. (29 de Junio de 2018). *GitHub*. Obtenido de https://github.com/FantacherJOY/Hidden-Markov-Model-for-NLP/blob/master/hmm.py

Jakkula, V. (2017). *Tutorial on support vector machine (svm).*

K. Crowston, E. E. (2012). *Using natural language processing technology for qualitative data analysis.*

K. Haynes, H. S. (2021). *Lightweight URL-based phishing detection using natural language processing transformers for mobile devices.* Procedia Comput. Sci.

K. Thakur, J. S.-S. (2018). *Innovations of Phishing Defense: The Mechanism, Measurement and Defense Strategies.* Int. J. Commun. Networks Inf. Secur.

Kaabouch, F. S. (2019). *Social Engineering Attacks: A Survey.* vol. 11, no. 4, p.89.

Kujama, A. (19 de Junio de 2022). *Definition & Types of Spam*. Obtenido de What is Spam?: https://www.malwarebytes.com/spam

L. Jiang, S. W. (2016). *Structure extended multinomial naive Bayes.* vol. 329, pp. 345-356.

Landgrebe, S. R. (1991). *A survey of decision tree classifier methodology.* IEEE Trans. Syst. Man. Cybern.

Louppe, G. (2014). *Accelerating random forests in scikit-learn.*

M. Jordan, S. L. (2018). *Statistics for Engineering and Information Science.*

Noble, W. S. (2006). *What is a support vector machine?* Nat. Biotechnol.

O. K. Sahingoz, E. B. (2019). *Machine learning based phishing detection from URLs.* Expert Syst. Appl.

P. L. Indrasiri, M. N. (2021). *Robust ensemble machine learning model for filtering phishing URLs: Expandable Random Gradient Stacked Voting Classifier (ERG-SVC).* IEEE Access.

Pandya, V. J. (2016). *Comparing handwritten character recognition by AdaBoostClassifier and KNeighborsClassifier.* 8th International Conference on Computational Intelligence and Communication Networks (CICN).

R. Kalnins, J. P. (2017). *Security Evaluation of Wireless Network Access Points.* vol. 21, no. 1, pp. 38-45.

R. Verma, N. S. (2012). *Detecting Phishing Emails the Natural Language Way.* Lect. Notes Comput. Sci.

Rayyan. (2018). *Intelligent Systematic Review.*

S. Bagui, D. N. (2021). *Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding.* J. Comput.

S. M. A. Alsufyani, A. A. (2021). *Social Engineering Attack Detection Using Machine Learning: Text Phishing Attack.* Indian J. Comput. Sci. Eng.

S. S. Sirigineedi, J. S. (2020). *Learning-based models to detect runtime phishing activities using URLs.* ACM Int. Conf. Proceeding Ser.

S. Salloum, T. G. (2021). *Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey.* Procedia Comput. Sci.

Schnyer, D. A. (2020). *Support vector machine.* Elsevier.

Soorya, M. (13 de Septiembre de 2021). *How to build a Spam Classifier in python and sklearn*. Obtenido de https://www.milindsoorya.com/blog/build-a-spam-classifier-in-python

Tain Xia, X. C. (2020). *A Discrete Hidden Markov Model for SMS Spam Detection.* Shanghai.

Verma, G. E. (2019). *Phishing email detection using robust NLP techniques.* IEEE Int. Conf. Data Min. Work. ICDMW.

X. Zhou, X. Z. (2016). *Online support vector machine: A survey.* Springer.

Xiao, X. (2021). *Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets.* Comput. Secur.

Yaseen, I. A. (2021). *Spam Email Detection Using Deep Learning Techniques.* Procedia Comput. Sci.