UNIVERSIDAD SAN FRANCISCO DE QUITO

USFQ

Colegio de Ciencias e Ingenierías

Detección Avanzada de Phishing Basado en Procesamiento de Lenguaje Natural Mediante Modelo de Lenguaje BERT

Jorge Esteban Alvarado Borja

Computación

Trabajo de fin de carrera presentado como requisito

para la obtención del título de Ingeniero en Ciencias de la Computación

Quito, 31 de enero de 2024

Universidad San Francisco de Quito USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN

DE TRABAJO DE FIN DE CARRERA

Detección Avanzada de Phishing basado en Procesamiento de Lenguaje Natural mediante Modelo de Lenguaje BERT

Jorge Esteban Alvarado Borja

Alejandro Proaño, PhD Electrical and Computer Engineering

Quito, 31 de enero de 2024

© DERECHOS DE AUTOR

Nombres y apellidos: Jorge Esteban Alvarado Borja

Código: 00215138

Cédula de identidad: 1750908657

Lugar y fecha: Quito, 31 de enero de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

El phishing se ha convertido en uno de los ciberataques más usados y peligrosos hoy en día. Anualmente hay pérdidas millonarias en las empresas debido a la continua dificultad de detectar este tipo de ataques en el momento. No obstante, el emergente desarrollo del campo de la inteligencia artificial (IA) ha permitido introducir diversas herramientas efectivas para el combate a esta problemática, entre las que se incluyen los modelos de lenguaje. El presente proyecto ha propuesto entrenar el modelo de lenguaje enmascarado (MLM) BERT para la detección avanzada de phishing, a través de un proceso conocido como finetuning. Las métricas de rendimiento muestran que este método, basado en procesamiento de lenguaje natural, brinda resultados favorables para la detección de phishing en varios contextos, entre los que se incluyen: correos, sitios web, mensajes SMS y URL. Se espera que el modelo entrenado resultante pueda prevenir con precisión y eficacia potenciales ataques de phishing hacia individuos y empresas, y que la información provista en este documento sea útil para futuras investigaciones.

Palabras clave: Phishing; Clasificación; IA; NLP; MLM; BERT; Finetuning.

ABSTRACT

Phishing has become one of the most used and dangerous cyber-attacks today. Annually there are millions of dollars in losses in companies due to the continuous difficulty of detecting this type of attacks at the time. However, the emerging development in the field of artificial intelligence (AI) has allowed the introduction of several effective tools to combat this problem, including language models. The present project has proposed to train the masked language model (MLM) BERT for advanced phishing detection, through a process known as Finetuning. Performance metrics show that this method, based on natural language processing, provides favorable results for phishing detection in various contexts, including: mails, websites, SMS messages and URLs. It is expected that the resulting trained model can accurately and effectively prevent potential phishing attacks on individuals and businesses, and that the information provided on this document will be useful for future research.

Key words: Phishing; Classification; AI; NLP; MLM; BERT; Finetuning.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El vertiginoso desarrollo de la tecnología digital ha traído consigo numerosos desafíos; uno de ellos, y tal vez uno de los más importantes, es la protección de la información. En cualquier empresa u organización el activo más valioso es la información (Najar, 2017), pues esta suele estar asociada a datos personales sensibles de clientes, usuarios o empleados, que si se revelan podrían perjudicar a muchas personas. Por ese motivo la seguridad informática se ha convertido en un pilar fundamental en todo tipo de industrias. Esto es de esperarse, pues un informe de la Interpol (2020) reveló que la delincuencia digital tras los estragos que causó la pandemia de COVID-19, ha hecho un cambio sustancial en sus objetivos. Antes atacaba empresas particulares y pequeñas, ahora tienden a ser grandes multinacionales, administraciones estatales e infraestructuras esenciales. Además, en el reporte de la empresa multinacional Check Point Software (2023) se menciona que los ataques mundiales aumentaron un 28% en el tercer trimestre de 2022 en comparación con el mismo periodo de 2021, y la media de ataques semanales por organización en todo el mundo superó los 1,130. El reporte termina enfatizando que para el 2023, esta tendencia no parece que vaya a disminuir, todo lo contrario, probablemente habrá un aumento de explotación de vulnerabilidades.

En el marco de un ataque informático el ser humano es el “eslabón más débil”. Esto es porque todas las personas comparten debilidades comunes que pueden ser explotadas por técnicas de engaño, las cuales existen desde los inicios de la humanidad. Aún dentro de un entorno digital, estas técnicas no dejan de ser funcionales (Navarro, 2019). La ingeniería social, justamente, aprovecha esos errores o debilidades inherentes del ser humano, y utiliza un “conjunto de técnicas … para engañar a los usuarios

incautos para que les envíen datos confidenciales, infecten sus computadoras con malware o abran enlaces a sitios infectados" (Kaspersky, 2023a).

El phishing es la forma más común de ingeniería social. Los ataques de phishing se caracterizan por emplear correos electrónicos, mensajes de texto, llamadas telefónicas o sitios web fraudulentos para manipular a las personas a que realicen acciones que los exponga a ellos mismos o a sus organizaciones (IBM, 2023b). Es el artefacto ideal para los cibercriminales pues este tipo de ataques no requieren programación sofisticada o la búsqueda exhaustiva de brechas, sino más bien emplear correctamente técnicas de manipulación y persuasión en las personas para cumplir el objetivo.

Fabio Assolini, director del Equipo Global de Investigación y Análisis para América Latina en Kaspersky (2023b), manifestó que el "phishing continúa siendo el vector más importante para el robo de datos personales y es el primer paso de los ciber incidentes que resultan en fugas de datos masivas". La empresa Kaspersky (2023b) asimismo expuso un reporte donde registró 286 millones de ataques de phishing en los últimos 12 meses desde agosto del 2022, lo que representa un aumento del 617% en comparación con los 12 meses anteriores. Y un informe de seguridad de IBM del 2023 revela que el phishing fue el vector de ataque más frecuente y el segundo más caro con 4.76 millones de dólares.

Las consecuencias de un ataque exitoso de este tipo pueden llegar a ser nefastas para un individuo: robo de dinero, cargos fraudulentos en tarjetas de crédito, pérdida del acceso a archivos o la suplantación de identidad. En el ámbito laboral, una empresa podría sufrir la pérdida de fondos corporativos, filtraciones de datos de clientes y empleados, pérdida de archivos confidenciales, deterioro de la reputación de la empresa,

entre otros (Microsoft, 2023). Por ejemplo, en el año 2013 atacantes robaron las credenciales de más de 110 millones de clientes de la cadena de supermercados Target a través de una estafa de phishing por correo electrónico. El ataque a Target fue una de las mayores filtraciones de datos de la historia, y todo comenzó con un mensaje de phishing enviado a un contratista que estaba conectado al sistema de la empresa (Regan, 2018). Por eso, ahora más que nunca es necesario seguir fortaleciendo la seguridad de los datos mediante nuevas y mejoras formas de detectar preventivamente ataques de phishing.

OBJETIVO Y CONTRIBUCIÓN DEL PROYECTO

Motivado por la sección anterior, el objetivo principal del proyecto fue la implementación de un sistema integral para la detección de phishing, Este sistema es capaz de identificar intentos de phishing en diversas formas, incluyendo enlaces, mensajes SMS, correos electrónicos y sitios web, utilizando técnicas de procesamiento de lenguaje natural. Para lograrlo se usó el modelo de lenguaje BERT, que fue entrenado con un conjunto de datos diverso y robusto. La eficacia de este método se evaluó comparando el rendimiento de BERT con otros modelos de clasificación. Además, se realizó un análisis exploratorio de los datos de phishing generados para identificar patrones maliciosos asociados al phishing. Las contribuciones del proyecto al combate contra el phishing se resumen a lo siguiente:

- El estudio de Basit et al. (2021) deja entrever que la gran mayoría de proyectos que usan técnicas de inteligencia artificial para detectar phishing no consideran el lenguaje natural; cuando, dada la naturaleza lingüística del phishing, debería ser todo lo contrario. Por ende, este proyecto innova en los métodos de detección de phishing basándose en el procesamiento de lenguaje natural con el modelo de lenguaje BERT.

- Este proyecto no limita el análisis del lenguaje únicamente a correos o URL, tal como sucede en Salloum et al. (2021) y Shirazi et al. (2022), sino que combina ambos tipos y agrega también la posibilidad de detectar phishing en mensajes SMS y sitios web, ampliando así su alcance y flexibilidad en potenciales ataques.
- El desarrollo de este proyecto ha generado un conjunto de datos combinado de phishing, disponible en la plataforma HugginFace, que contiene muestras de URL, SMS, correos y sitios web malignos. Este conjunto de datos podrá utilizarse para futuras investigaciones relacionadas a la detección de phishing.
- La sección de análisis del dataset de phishing proporciona una guía empírica para identificar patrones maliciosos de phishing en URL, sitios web, correos y mensajes de texto.
- El método propuesto en este proyecto, basado en procesamiento de lenguaje natural con BERT, demuestra superioridad en rendimiento sobre otros métodos de clasificación con modelos como XGBoost, Naive Bayes Multinomial y LSTM-CNN.

DESARROLLO DEL TEMA

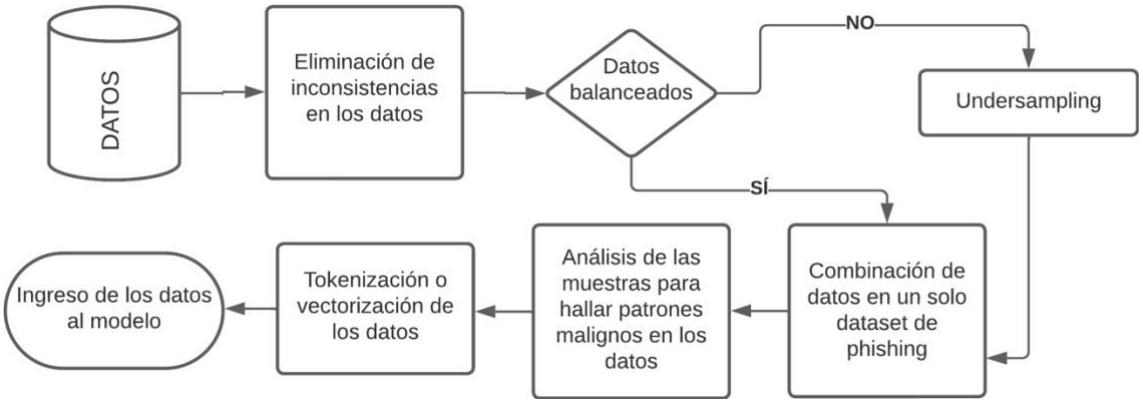
1. Manejo y Análisis de Datos de Phishing.

Los conjuntos de datos con los que el modelo de lenguaje se alimenta provienen de varias fuentes; cada uno aporta con información relevante para el sistema de detección y corresponden a los vectores de ataque más comunes del phishing: enlaces, mensajes SMS, correos y sitios web (Krombholz et al., 2015). Esta variedad de datos permite a BERT mejorar su capacidad de reconocer el phishing en varios contextos. Los

datasets etiquetan las muestras como benignas o malignas y comparten una estructura similar. El flujo de procesamiento sobre esta información se especifica a continuación:

Figura 1

Flujo de procesamiento de los conjuntos de datos.



Nota. El flujo detalla cómo se manejaron y procesaron los datos utilizados a lo largo del desarrollo de este proyecto.

1.1. Descripción de los datasets

1.1.1. Dataset de correos. Conjunto de datos que especifica el cuerpo de texto de varios correos electrónicos seguros y de phishing. Contiene más de 18 mil correos generados por empleados de la Corporación Enron (Chakraborty, 2023).

1.1.2. Dataset de mensajes SMS. Conjunto de aproximadamente 6 mil mensajes de texto o SMS etiquetados como Legítimos (Ham), Spam o Smishing. Los datos se recopilaron convirtiendo imágenes obtenidas de Internet en texto mediante código Python (Mishra & Soni, 2022).

1.1.3. Dataset de URL. Se trata de una colección de más de 800 mil URL malignas y legítimas de diversas fuentes. Las URL se recogieron del sitio web de JPCERT, conjuntos de datos existentes de Kaggle, repositorios de Github (donde las

URL se actualizan una vez al año) y algunas bases de datos de código abierto, incluidos archivos Excel (Sudhan, 2023).

1.1.4. Dataset de sitios web. Colección de 80 mil instancias entre sitios web legítimos y de sitios web con phishing. Cada instancia contiene la URL y la página HTML. Los datos legítimos se recogieron de dos fuentes: (a) búsqueda simple de palabras clave en el motor de búsqueda de Google, donde se recopilaron las 5 primeras URL de cada búsqueda, y (b) recopilación de casi 25,874 URL activas del repositorio Ebbu2017 Phishing Dataset. Para los datos de phishing se utilizaron tres fuentes: PhishTank, OpenPhish y PhishRepo. (Ariyadasa, et al., 2021).

1.2. Preprocesamiento de Datos

El preprocesamiento es un paso fundamental que debe realizarse antes de cualquier tarea de procesamiento de lenguaje natural, debido a que los datos por lo general vienen con errores como: datos faltantes, ruido e inconsistencias. Con frecuencia, el preprocesamiento impacta de forma significativa en el desempeño de los algoritmos de inteligencia artificial (Hernandez & Rodriguez, 2013). Los conjuntos de datos descritos se preprocesaron en su totalidad, con excepción del dataset de sitios web. Cada registro de este dataset es una página web con un peso considerable, si a esto se suma que son una gran cantidad de registros (aproximadamente 10 GB de datos), fue inviable procesarlos todos con los recursos disponibles. Por ello de las 80,000 páginas solo se importaron 30,000, de las cuales se eligieron únicamente aquellas que tuvieran un peso menor a 100KB, quedando unas 20,887 muestras.

Durante el preprocesamiento se eliminaron registros con valores NaN, nulos, vacíos o duplicados; en total fueron eliminados 20,235 registros entre todos los datasets. También se minificaron las muestras del dataset de sitios web; un proceso que consiste

en eliminar caracteres innecesarios del código, sin cambiar su funcionalidad. Esta optimización del código HTML reduce tiempos de espera pues permite la rápida lectura de los datos por parte de los modelos (Surática, s.f.). Luego se formatearon ligeramente los datasets para que todos cumplan con una estructura de dos columnas: texto y etiqueta, para facilitar su combinación y análisis.

Antes de proceder a unificar los datasets, se pudo notar que el tamaño del dataset de URL era casi 40 veces mayor a los demás. Unificar en ese estado haría que las URL comprendan el 97% de las muestras, y los correos, SMS y sitios web apenas el 3%. Si faltan tipos de datos de poblaciones específicas el modelo podría sesgarse y no reflejar las realidades del entorno en el que se ejecuta. Los datos de entrenamiento deben incluir un tamaño de muestra proporcionalmente exacto de cada miembro de una población, incluidos todos los tipos de instancias y combinaciones (Shaked, 2021); de lo contrario el modelo tendrá un desempeño negativo e ignorará los tipos de datos que están infrarrepresentados (Apéndice A). Para solucionar este inconveniente se recurrió a un método simple de muestreo aleatorio conocido como Undersampling que envuelve eliminar aleatoriamente ejemplos del tipo de dato mayoritario (Brownlee, 2021). En base a esto se descartó aleatoriamente el 95% de las URL, dejando como resultado una cantidad razonable alrededor de 41 mil URL.

Después del preprocesamiento se combinaron todos los datasets: URL, correos, mensajes SMS y códigos de sitios web, en un solo conjunto de datos que de ahora en adelante se llamará dataset de phishing (Tabla A1). Con esta información robusta y variada, BERT tiene la capacidad de identificar phishing en varios contextos y con gran flexibilidad, sin importar si la muestra es una URL, texto o sitio web.

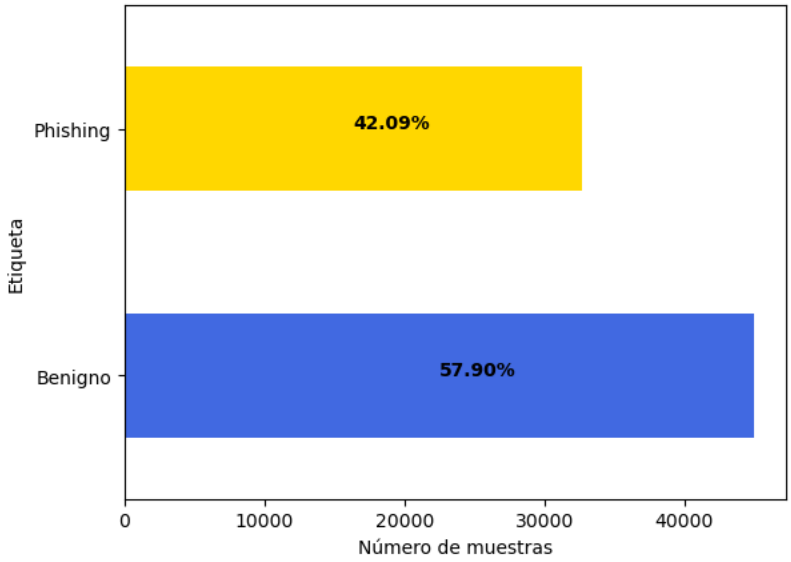
1.3. Análisis del Dataset de Phishing

Esta sección está dedicada al análisis exploratorio del dataset de phishing. Aquí se exponen características, asociadas al lenguaje natural, que son determinantes para que los modelos de aprendizaje automático clasifiquen muestras como phishing o benignas. A su vez, brinda al usuario una base sobre cómo detectar phishing mediante una observación superficial del lenguaje y su estructura.

El dataset de phishing tiene más de 75 mil registros de cuatro tipos: sitios web, URL, SMS y correos. La mayoría son URL (41 mil), seguido de correos (17 mil), sitios web (15 mil), y SMS (2 mil). El dataset está equilibrado ya que lleva una proporción de clases con un 57.90% de muestras benignas y 42.09% malignas:

Figura 2

Proporción de clases en el dataset de phishing.



Como grupo mayoritario, las URL juegan un papel crucial en la detección de phishing. Estas tienen ciertas propiedades textuales que las relaciona con ello, por ejemplo: presencia de una dirección IP en el dominio, número de puntos, presencia de

palabras clave ‘login’, ‘server’ o ‘admin’, presencia del sufijo o prefijo “-” en el dominio, etc (Tabla B1). Se extrajeron dichas características de las URL, pero para conseguir un análisis más cercano a la realidad en este paso se optó por utilizar el dataset original de URL con más de 800,000 muestras, obteniendo los siguientes resultados:

Tabla 1

Información correspondiente a las características de las URL.

Característica	**Proporción Benigna (%)**	**Proporción Phishing (%)**	**Diferencia (%)**
Usa servicio de acortamiento	7.0722%	5.166%	-1.9062%
Tiene el prefijo-sufijo ‘-‘ en el dominio	0.6268%	11.062%	10.43%
Tiene un script de redirección	0.019%	0.6313%	0.6123%
Tiene un ‘@’	0.0567%	1.1868%	1.1301%
Tiene un número de puerto	0.0172%	0.0264%	0.0092%
Tiene la palabra clave ‘admin’	0.1015%	1.9061%	1.8046%
Tiene la palabra clave ‘server’	0.2166%	0.4859%	0.2693%
Tiene la palabra clave ‘login’	0.1923%	9.0762%	8.8839%
Tiene la palabra clave ‘client’	0.1027%	0.5275%	0.4248%
Dirección IP en el dominio	0.0002%	0.5612%	0.561%
Está codificada	2.5445%	1.5583%	-0.9862%
Característica	**Promedio Benigno**	**Promedio Phishing**	**Diferencia**
Longitud de la URL	47.0041	47.5751	0.5709
Longitud de la ruta	41.4234	25.1904	-16.2330
Longitud del host	1.8158	9.8135	7.9977
Entropía de la URL	3.6417	3.5023	-0.1393
Número de dígitos en la URL	3.0659	5.0170	1.9510

Número de subdirectorios	3.3200	2.5336	-0.7863
Número de puntos	1.8258	2.4675	0.6417
Número de parámetros	0.1836	0.2592	0.0755

Muchas URL de phishing tienen el prefijo-sufijo "-" en el dominio. De hecho, el 11.06% de las URL de phishing tienen un guion, mientras que solo el 0.62% de las URL seguras lo tienen. Algunas palabras que frecuentan usar las URL de phishing son "login" y "admin". Estas palabras aparecen en el 9.07% y el 1.90% de las URL de phishing, respectivamente. Además, las URL de phishing tienen en promedio nombres de dominio o host muy largos. El promedio de longitud del host en muestras malignas es cinco veces mayor que el de las URL seguras.

Las URL en sí mismas no son dañinas, pero pueden serlo dependiendo de a donde dirijan a los usuarios. Los atacantes usan las URL para llevar a sus víctimas a sitios web dañinos. Algunos estudios revelan características importantes relacionadas a la estructura y contenido de las páginas web maliciosas como: número de enlaces internos, número de enlaces externos, número de funciones que modifican el DOM, número de scripts, etc (Tabla B2). Se extrajeron estas características de las muestras de sitios web presentes en el dataset de phishing, obteniendo los siguientes resultados:

Tabla 2

Información correspondiente a las características de los sitios web.

Característica	**Promedio Benigno**	**Promedio Phishing**	**Diferencia**
Número de funciones sospechosas	0.3125	0.2740	-0.0385
Entropía de la página	4.4785	4.3166	-0.1619
Número de etiquetas 'script'	15.1260	6.2447	-8.8813

Longitud de la página	14183.5757	7956.9136	-6226.6621
Número de palabras	1157.0611	550.0260	-607.0351
Número de oraciones	236.1877	159.8381	-76.3496
Número de signos de puntuación	2224.9793	1394.5386	-830.4407
Número de letras mayúscula	1352.0863	976.6509	-375.4353
Promedio de tokens en las oraciones	8.1698	9.9748	1.8050
Número de etiquetas HTML	361.2852	129.6899	-231.5953
Número de etiquetas ocultas	0.2980	0.2263	-0.0717
Número de Iframes	0.4109	0.1893	-0.2215
Número de objetos	0.0167	0.0074	-0.0092
Número de elementos embebidos	0.0045	0.0045	-0.00002
Número de links internos	26.7312	1.2294	-25.5018
Número de links externos	27.5605	6.5961	-20.9643
Número de espacios en blanco	2460.2432	1703.7536	-756.4896
Número de elementos incrustados	8.3171	3.7532	-4.5638
Número de funciones que modifican el DOM	7.2105	4.0480	-3.1624
Promedio de longitud (caracteres) de los scripts	549.0055	878.3920	329.3864
Promedio de entropía de los scripts	1.7409	1.6034	-0.1375

Las muestras benignas de sitios web mantienen en su mayoría un promedio superior al de las malignas en: la longitud de la página, número de palabras, número de signos de puntuación, número de etiquetas HTML, número de etiquetas ‘script’, etc. Lo que significa que en casi todos los casos los sitios web benignos llevan más contenido e información que los malignos. Otra característica que también destaca sobre los sitios web malignos es el promedio de longitud de los scripts; en páginas maliciosas este es 1.5 veces mayor que el que se calculó en las muestras benignas. Esta diferencia podría

estar relacionada a la ejecución de ataques como Cross-Site Scripting (XSS) que requieren cierta complejidad para evadir las soluciones de seguridad informática. Además, conforme a lo que menciona Opara et al. (2024), se observa que los sitios web malignos tienen en promedio más enlaces externos que internos, posiblemente para enriquecer el contenido del sitio y hacerle creer a la víctima que la página es legal.

Las URL y sitios web alcanzan una mayor efectividad en ataques de phishing cuando se conectan con un mensaje que emplea técnicas de ingeniería social. Un informe de Firewall Times (2023) declara que el 98% de ataques informáticos envuelve algún tipo de ingeniería social. Por eso analizar los correos y mensajes SMS que contiene el dataset de phishing fue uno de los pasos más importantes en esta sección. Para el análisis del lenguaje natural se utilizó el algoritmo de vectorización TF-IDF que transforma el texto en una representación significativa de números (Chaudhary, 2020). La agencia de marketing digital y desarrollo web, Kiwop (2023), manifiesta que este método se enfoca en resaltar palabras que son relevantes y únicas para un documento específico, en vez de limitarse a calcular la frecuencia absoluta de términos habituales. TF-IDF se compone de dos partes (Kiwop, 2023):

- Frecuencia de términos (TF): mide la frecuencia de un término en un documento.
- Frecuencia de términos inversa (IDF): mide la rareza o importancia de un término en un conjunto de documentos.

TF-IDF es el producto de estos dos cálculos. El valor TF-IDF aumenta cuando una palabra clave específica tiene una frecuencia alta en un documento y la frecuencia de documentos que contienen la palabra clave entre todos los documentos es baja. Este principio puede utilizarse para encontrar las palabras clave que aparecen con frecuencia

en los documentos. Por consiguiente, utilizando la puntuación TF-IDF es posible averiguar qué palabras clave son importantes en cada documento (Kim & Gil, 2019).

Figura 3

Top 10 palabras relevantes en correos y SMS benignos basado en puntuación TF-IDF.

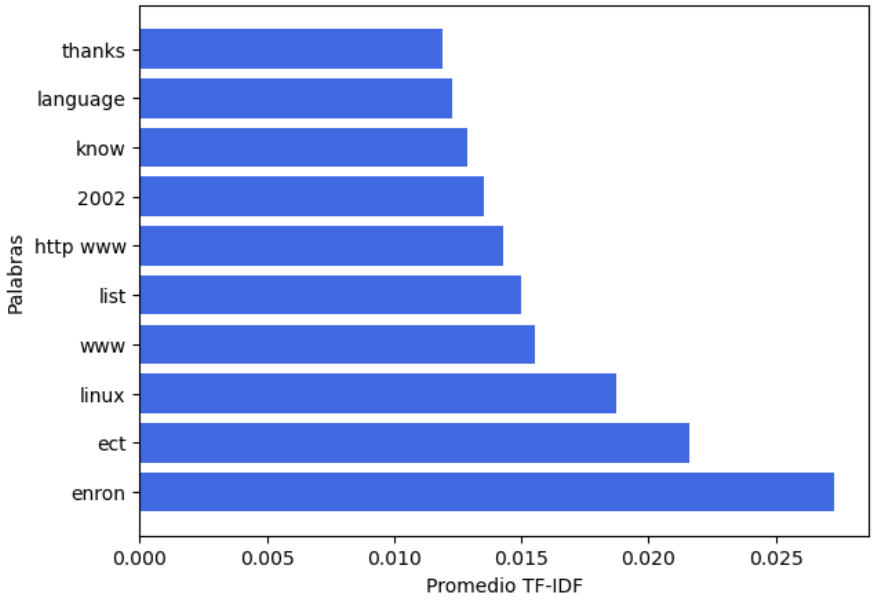
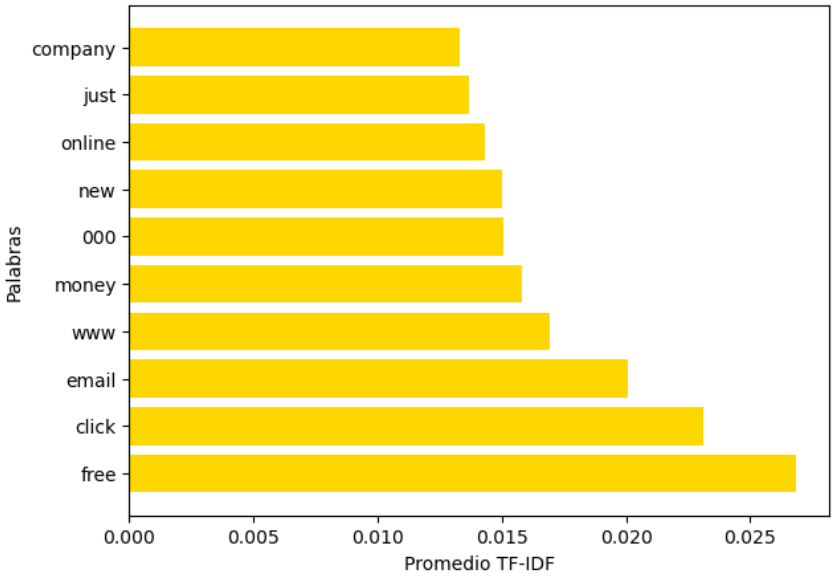


Figura 4

Top 10 palabras relevantes en correos y SMS malignos basado en puntuación TF-IDF.



La Figura 3 muestra que las palabras más relevantes en textos benignos son “enron”, “know”, “language”, “thanks”, etc. Mientras tanto la Figura 4 evidencia que las palabras más importantes en textos malignos son: “free”, “click”, “email”, “money”, “company”, entre otras. En esencia, esta información conduce a la conclusión de que los textos de phishing generalmente usan un lenguaje que ofrece productos o servicios gratuitos, incita a dar clicks y habla de dinero y negocios. Asimismo, se deduce que los textos benignos usan un lenguaje formal y educado, donde se solicita cierta información o requerimiento; esto por los términos “know” y “thanks”. Además, muy posiblemente incluyan el nombre correcto de una empresa, en este caso, la corporación Enron, pues de allí provienen las muestras de correos.

1.4. La Heterogeneidad Aplicada al Dataset de Pishing

El dataset de phishing tiene una gran variedad de datos, como se ha podido observar hasta ahora. Esta diversidad introduce un concepto explicado por Kamm et al. (2023): la heterogeneidad; que ocurre cuando las muestras presentan diferencias relacionadas a sus propiedades como el formato, presentación, significado, etc. La heterogeneidad se puede clasificar en cinco clases, pero en el dataset solo se aplican tres de ellas: sintáctica, semántica y estadística.

La sintáctica se refiere a las diferencias en los tipos de datos o formatos. Sin duda esto es así debido a que el dataset incluye tipos de datos disímiles entre sí y con un formato específico, como sucede en las URL y sitios web que contienen código HTML. La heterogeneidad semántica cubre los distintos significados e interpretaciones de los datos. Una URL por sí sola podría no tener el mismo significado que cuando se utiliza dentro de un correo o mensaje SMS. En último lugar, la heterogeneidad estadística implica la diferencia en las propiedades estadísticas entre las muestras dentro de un

conjunto de datos. Como ejemplo, el promedio de palabras en una URL (asumiendo que se separan las URL en palabras) será muy diferente al que poseen los correos.

Es crucial considerar este concepto para el entrenamiento de BERT, ya que según Wenz et al. (2021) la heterogeneidad generalmente se considera un problema de calidad en los datos porque es más difícil procesar datos que no están claramente estructurados. Más adelante, se evaluará cómo BERT se comporta con la heterogeneidad que presenta el dataset de phishing en cada uno de los tipos de datos y en la combinación de estos.

2. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) es una rama de la inteligencia artificial que "brinda a las computadoras la capacidad de entender, manipular y comprender el lenguaje humano" (Amazon, 2023). En el contexto del phishing, el NLP puede aprovecharse para hallar patrones de lenguaje que indiquen casos de textos maliciosos. Estos indicadores pueden incluir el uso excesivo de términos financieros, una gramática incorrecta, urgencia inapropiada, etc. De hecho, las mejores tecnologías de detección de correo no deseado utilizan soluciones de clasificación de texto que provee el NLP (IBM, 2023c).

Gracias al actual avance de esta rama, Google propuso en el 2017 una arquitectura de red neuronal apodada "Transformers" que jugó un papel crítico en el desarrollo de los modelos lingüísticos para tareas de NLP. Incluso, modelos actuales famosos que están surgiendo basados en NLP consisten de decenas de Transformers o algunas de sus variantes, como GPT o BERT. Los Transformers permiten a los modelos de lenguaje aprender relaciones complejas entre las palabras y oraciones. Se caracterizan por su capacidad para prestar atención a diferentes partes del texto de

manera selectiva mediante un mecanismo de atención auto regresiva, que brinda al modelo la habilidad de determinar qué palabras son relevantes al clasificar un texto (Romero, 2023).

2.1. BERT (Bidirectional Encoders Representations from Transformers)

BERT, cuya abreviatura en español se traduce a: Representaciones Codificadoras Bidireccionales a partir de Transformers, es un modelo de aprendizaje automático (ML) para el procesamiento del lenguaje natural que sirve para más de 11 de las tareas lingüísticas más comunes, como el análisis de sentimientos y el reconocimiento de entidades con nombre. Se pre-entrenó en el conjunto de datos de Wikipedia (2.500 millones de palabras) y el corpus de libros de Google (800 millones de palabras). Estos grandes conjuntos de datos informativos han contribuido al profundo conocimiento de BERT no sólo de la lengua inglesa, sino también del mundo (Muller, 2022). BERT se encuentra disponible en la plataforma HuggingFace de forma gratuita. Para este proyecto se eligió entrenar una versión del modelo denominada: “bert-large-uncased” que consta de 336 millones de parámetros, 24 capas, 1024 dimensiones ocultas y 16 cabezas de atención (Devlin et al., 2018). Las distintas capas de BERT capturan diferentes niveles de información sintáctica y semántica (Qiu et al., 2019), que es justo lo que se requiere para la heterogeneidad sintáctica y semántica del dataset de phishing.

2.2. Finetuning

Un modelo pre-entrenado comprende una red neuronal grande entrenada en un vasto cuerpo de texto, usualmente proveniente del internet. El finetuning es un paso transcendental para optimizar estos modelos para tareas específicas. Algunos modelos pre-entrenados populares son BERT, GPT-3 y RoBERTa. Básicamente, finetuning es

un proceso que consiste en adaptar modelos de propósito general para realizar tareas especializadas con precisión y eficiencia (Banjara, 2024). Si se desea detectar phishing, es necesario ajustar el modelo pre-entrenado de BERT para que sea capaz de entender los matices de esa tarea y dominio específicos. Con ese fin, se ocupó el dataset de pishing para especializar a BERT en la clasificación y detección de muestras malignas.

Dichos datos no pueden ingresarse a BERT en su forma original, sino que deben someterse a un proceso de transformación llevado por un tokenizador. Un tokenizador es un programa que separa el texto crudo en palabras, y las transforma a valores numéricos que los modelos pueden procesar. Hay distintas maneras de hacer esto: basado en palabras, basado en caracteres; incluso hay algunas que tienen reglas específicas para la puntuación (Hugging Face, 2024). Durante su pre-entrenamiento, BERT aplicó una forma particular de tokenización, por lo que se debe importar el mismo tokenizador que usó BERT para que este sea capaz de procesar los datos que son ingresados. El tokenizador asimismo evita que las muestras superen la longitud limitada de 512 tokens que BERT acepta en las secuencias (Devlin et al., 2018).

2.3. El Poder de la Bidireccionalidad

Este modelo posee una ventaja significativa sobre otros modelos de lenguaje populares en el mercado. Tal como lo explican sus autores (Devlin et al., 2018), los modelos de lenguage estándar como ELMo y GPT, tienen la limitación de ser unidireccionales, es decir, que solo pueden analizar texto en una dirección: de derecha a izquierda o de izquierda a derecha. En el caso de GPT, los autores emplean una arquitectura de izquierda-derecha que solo puede ver lo que hay antes de cada pedazo de texto, pero no lo que hay después. Tal restricción podría ser perjudicial para el finetuning de un modelo que se desea especializar para tareas de NLP como la respuesta

a preguntas o, en este caso, la clasificación de phishing, donde se necesita un entendimiento del lenguaje profundo analizando todo el contexto de la muestra. Por ende, Devlin et al. (2018) mejora el enfoque de finetuning para tareas de NLP proponiendo BERT. BERT alivia la restricción de unidireccionalidad enmascarando u ocultando aleatoriamente algunos de los pedazos de texto que se ingresan al modelo durante el pre-entrenamiento, con el objetivo de intentar predecir el pedazo original oculto revisando el contexto que le rodea. Esta táctica de pre-entrenamiento cataloga a BERT como un modelo de lenguaje enmascarado (MLM por sus siglas en inglés), que le faculta la habilidad de aprender un lenguaje mucho más rico y profundo en comparación a otros modelos.

En definitiva, a pesar de que el MLM de BERT no cuenta con una capacidad similar a grandes modelos como GPT, su arquitectura bidireccional le permite desempeñarse eficientemente en la tarea de NLP de clasificación de phishing, puesto que puede analizar las muestras maliciosas de izquierda a derecha, y de derecha a izquierda, al mismo tiempo; hallando patrones de lenguaje que toman en cuenta el contexto de toda la muestra y que están implicados en ataques de phishing.

3. Finetuning de BERT en el Dataset de Phishing

El finetuning de BERT se ejecutó en los cuatro tipos de datos del dataset de phishing por separado: correos y mensajes SMS, URL y sitios web, finalizando con el finetuning en el dataset final que combina todos estos datos. Este enfoque de entrenamiento permite reconocer en qué tipo de datos BERT se desempeña mejor y cómo maneja la heterogeneidad de los datos del dataset de phishing. Al terminar el finetuning de BERT en el dataset de phishing, se evaluará su rendimiento en cada tipo

de dato para observar si efectivamente el modelo es capaz de identificar con precisión el phishing en todas sus formas.

Durante el finetuning se comparó el desempeño de BERT con los siguientes modelos de clasificación: XGBoost, Naive Bayes Multinomial (MNB) y LSTM-CNN. Las motivaciones que condujeron a la decisión de usar estos tres modelos para la comparativa son las siguientes:

- XGBoost, acrónimo en español de Refuerzo de Gradiente Extrema, es un modelo de aprendizaje automático basado en árboles de decisión reforzados por gradiente. Provee de refuerzo de árboles en paralelo y es la principal librería hoy en día para problemas de regresión y clasificación. Muchos individuos y equipos han ganado competencias de datos estructurados de Kaggle gracias a este modelo (NVIDIA, 2023b).
- Los clasificadores de Naive Bayes (NB) son una familia de clasificadores basados en el popular teorema de la probabilidad de Bayes, conocidos para crear modelos sencillos y potentes, sobre todo en las áreas de clasificación de documentos y predicción de enfermedades. La clasificación textual de NB es la más utilizada para categorizar texto, ya que es rápida y fácil de implementar. Los algoritmos menos defectuosos suelen ser más lentos y complejos (Abbas et al., 2019).
- La arquitectura híbrida LSTM-CNN incluye los métodos de redes convolucionales (CNN) y larga memoria a corto plazo (LSTM) para aprovechar las ventajas de ambos métodos y lograr un rendimiento excelente. CNN y LSTM muestran un alto rendimiento en la superación de tareas de clasificación, detección y reconocimiento, por ello utilizar este

método para la tarea de detección de phishing es prometedor (Alshingiti et al., 2023).

3.1. Métricas de Rendimiento

Las métricas de rendimiento son esenciales para evaluar la eficacia de un modelo de aprendizaje automático. Con ellas se puede cuantificar la calidad de las predicciones, identificar áreas de mejora y comparar el modelo con otros modelos o enfoques. Este proyecto considera las siguientes métricas:

- **Accuracy**: mide el porcentaje global de valores correctamente clasificados, tanto de phishing como benignos (Diaz, 2024).
- **Precisión**: muestra el porcentaje de muestras clasificadas como phishing que realmente son phishing (Diaz, 2024).
- **Recall**: es la tasa de muestras de phishing que se clasifican correctamente (Diaz, 2024).
- **AUC**: se interpreta como la probabilidad de que las puntuaciones dadas por un clasificador clasifiquen una instancia de phishing elegida al azar más alto que una benigna elegida al azar (He & Ma, 2013).

Según Divakaran y Oest (2022), en una tarea de detección de phishing una de las métricas más importantes es el recall. El recall se define como:

$$Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos} \quad (1)$$

Cuando se habla de detectar phishing, es primordial capturar la mayor cantidad posible de verdaderos positivos, es decir, instancias de phishing reales. Paralelamente también es vital minimizar el número de falsos negativos o instancias de phishing clasificadas incorrectamente como benignas, ya que esto reduciría la posibilidad de que

un ataque de phishing se perciba como benigno. En ese sentido, el recall es la métrica ideal para cuantificar estos datos en los modelos que se implementaron. Mientras mayor sea su valor mejor será el modelo para clasificar correctamente muestras malignas.

También se provee de una visualización del desempeño de BERT y demás modelos a través de las curvas ROC-AUC y Precision-Recall. La curva ROC-AUC muestra la tasa de verdaderos positivos (en el eje Y) contra la tasa de falsos positivos (en el eje X); o, en otras palabras, la fracción de las predicciones de phishing correctas versus la fracción de errores en las predicciones benignas. El mejor clasificador posible es aquel que tiene una tasa de verdaderos positivos igual a uno y la tasa de falsos positivos igual a cero. En cuanto a la curva Precision-Recall, esta presenta un gráfico de la precisión en el eje Y, y el recall en el eje X. En general, cuando aumenta el recall, disminuye la precisión, y viceversa; por ello, un modelo perfecto es aquel cuya curva muestra una precisión y recall igual a uno (Brownlee, 2020).

3.2. Resultados del entrenamiento

A continuación, se muestran los resultados del entrenamiento de BERT, XGBoost, Naive Bayes Multinomial y LSTM-CNN en los distintos tipos de datos: correos y mensajes SMS, URL y sitios web, y en el dataset final de phishing. Así como también el desempeño de BERT, una vez ya entrenado en el dataset de phishing, para cada tipo de dato. Los detalles de cómo se realizó este entrenamiento se describen en el Apéndice C.

Tabla 3

Métricas de rendimiento para clasificación de correos y mensajes SMS.

Modelo	Accuracy	Precision	Recall	AUC

XGBoost	0.943148	0.946648	0.901629	0.988147
MNB	0.955313	0.940794	0.942020	0.993090
LSTM-CNN	0.960775	0.975812	0.919870	0.994740
BERT-Base	0.604270	0.466817	0.270358	0.572976
BERT-Finetuned	0.990318	0.990170	0.984365	0.999146

Figura 5

Curvas ROC-AUC y Precision-Recall para clasificación de correos y mensajes SMS.

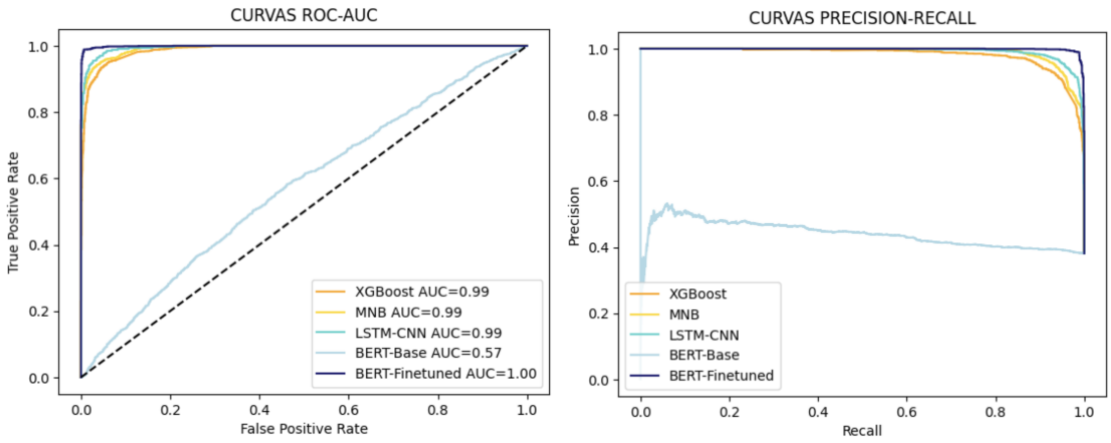


Tabla 4

Métricas de rendimiento para clasificación de URL

Modelo	Accuracy	Precision	Recall	AUC
XGBoost	0.832848	0.901290	0.722898	0.931936
MNB	0.892328	0.922771	0.840880	0.973834
LSTM-CNN	0.532047	1.000000	0.002587	0.501301
BERT-Base	0.469167	0.469160	0.999741	0.560019
BERT-Finetuned	0.976815	0.985979	0.964295	0.996684

Figura 6

Curvas ROC-AUC y Precision-Recall para clasificación de URL.

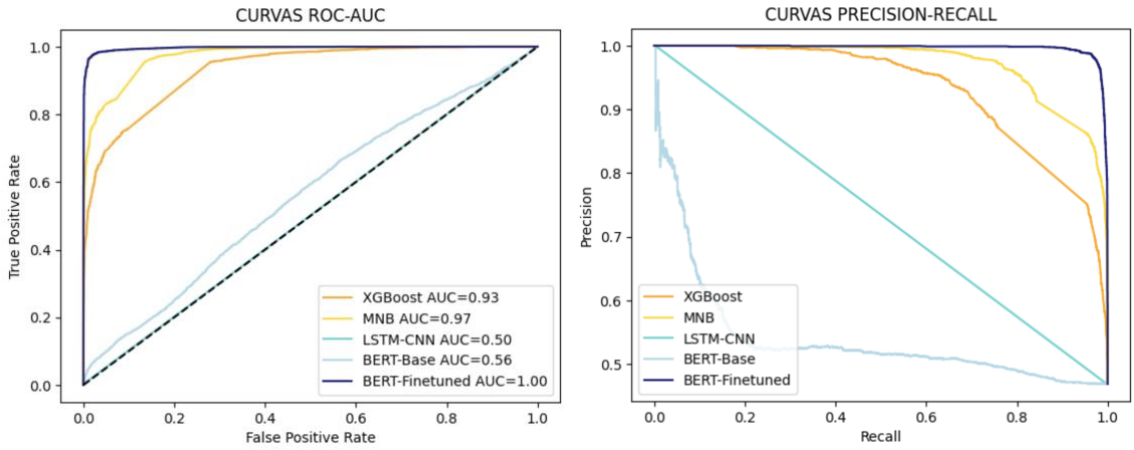


Tabla 5

Métricas de rendimiento para clasificación de sitios web.

Modelo	Accuracy	Precision	Recall	AUC
XGBoost	0.952411	0.934126	0.929026	0.989995
MNB	0.926713	0.896709	0.892630	0.974139
LSTM-CNN	0.937500	0.937016	0.879891	0.976354
BERT-Base	0.494289	0.331058	0.441310	0.462573
BERT-Finetuned	0.651332	0.000000	0.000000	0.644219

Figura 7

Curvas ROC-AUC y Precision-Recall para clasificación de sitios web.

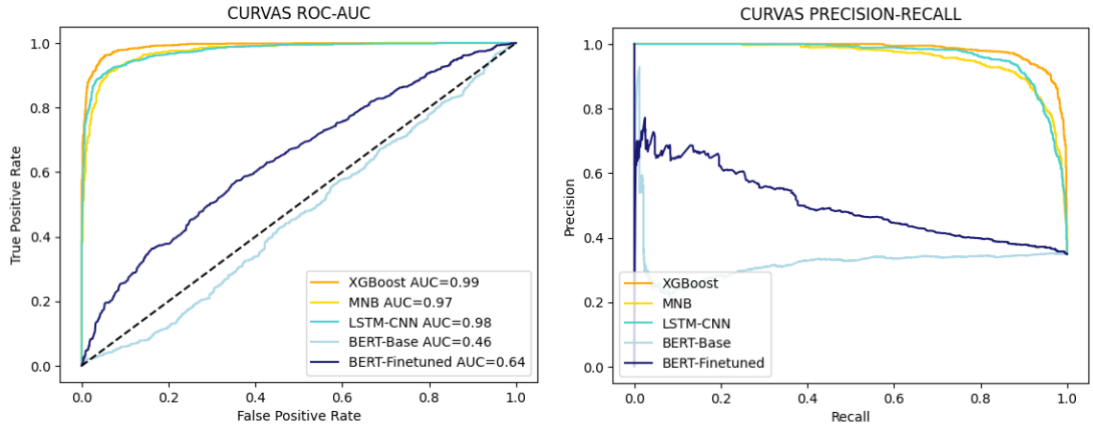


Tabla 6

Métricas de rendimiento para clasificación en el dataset final de phishing.

Modelo	Accuracy	Precision	Recall	AUC
XGBoost	0.850036	0.919944	0.705602	0.942189
MNB	0.875008	0.891957	0.800400	0.957777
LSTM-CNN	0.724979	0.935907	0.373038	0.825239
BERT-Base	0.579425	0.718750	0.003540	0.587870
BERT-Finetuned	0.984433	0.984215	0.978763	0.996928

Figura 8

Curvas ROC-AUC y Precision-Recall para clasificación en el dataset final de phishing.

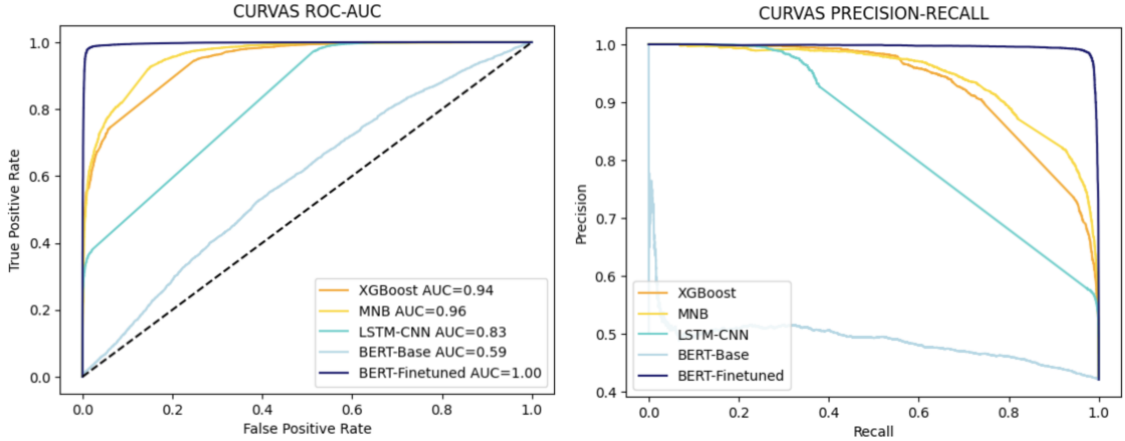


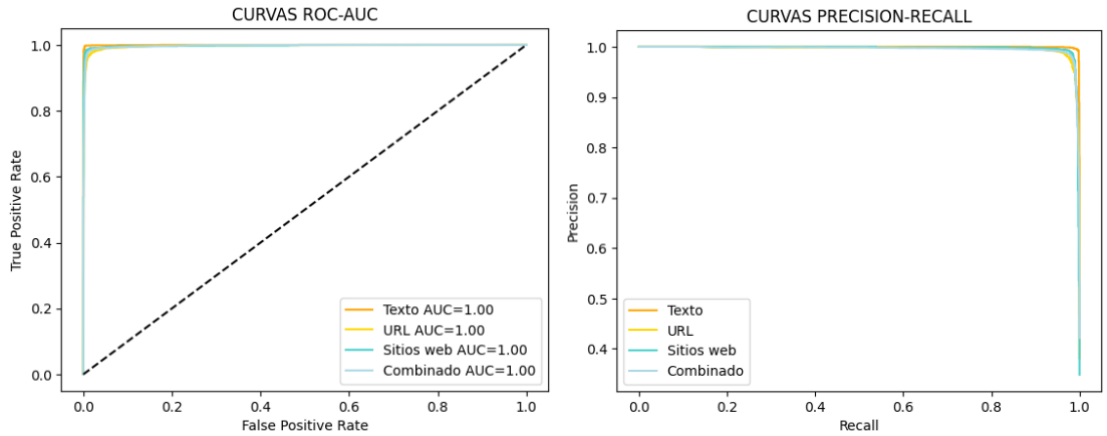
Tabla 7

Métricas de rendimiento de BERT finetuned en cada tipo de dato.

Dataset	Accuracy	Precision	Recall	AUC
Texto (Correos y SMS)	0.995780	0.993498	0.995440	0.999741
URL	0.978757	0.979470	0.975162	0.996458
Sitios web	0.989213	0.988981	0.979982	0.997819
Combinado	0.984433	0.984215	0.978763	0.996928

Figura 9

Curvas ROC-AUC y Precision-Recall de BERT finetuned en cada tipo de dato.



3.3. Discusión de Resultados

El fintetuning de BERT muestra resultados prometedores para la detección y clasificación de phishing. En los experimentos realizados, BERT, cuando se entrena para la detección de phishing, mantiene métricas de accuracy, precision, recall y AUC superiores al 96%. Es especialmente preciso y eficaz en la detección de correos y SMS malignos, con un recall del 98.43%. Pero su rendimiento fue decepcionante en el experimento con las muestras de sitios web. Lo curioso es que estos mismos datos están

presentes en el dataset final de phishing, y, sin embargo, allí BERT finetuned sí tuvo buenos resultados con un recall del 97.87%.

Los sitios web por naturaleza son datos altamente heterogéneos, ya que incluyen lenguaje humano, URL, scripts, y además de ello código HTML. Esta complejidad podría ser la causa de que a BERT se le dificulte tanto identificar patrones malignos en ellos. No obstante, cuando estos datos se combinan con más información relacionada al phishing, es posible que se genere un efecto sinérgico de complementariedad. Este efecto, descrito por Wang et al. (2023), tiene el potencial de estimular interacciones que proporcionan un contexto adicional, mejorando, en consecuencia, la capacidad de BERT para detectar phishing. De esta manera se puede explicar el comportamiento de BERT observado en los experimentos con sitios web y por qué pudo clasificar correctamente este tipo de datos en el dataset final de phishing.

Por otra parte, se constató que BERT base pre-entrenado no es capaz de clasificar y detectar phishing. En el experimento con URL y sitios web, BERT base tuvo un recall alto pero una precisión insuficiente. Los gráficos de Precision-Recall corroboran esta información, mostrando resultados que se desvían significativamente de las curvas ideales. No cabe duda de que, aunque BERT base pre-entrenado puede ser valioso para tareas de propósito general, es imperativo realizar finetuning para optimizar su eficacia en la identificación de phishing.

XGBoost demostró su potencial superando a BERT, MNB y LSTM-CNN en la clasificación de sitios web con un recall del 92.26%. MNB no se quedó atrás, a pesar de ser un modelo sencillo y menos complejo. Su velocidad de ejecución fue excepcional pudiendo entrenarse en un tiempo significativamente más corto, no mayor a 200 milisegundos, y aun así alcanzó resultados razonables. Lamentablemente LSTM-CNN

no tuvo mayor distinción sobre los otros modelos. Su mejor rendimiento se dio en las muestras de correos y mensajes SMS con recall del 91.98%.

En última instancia, se pudo comprobar que BERT finetuned tiene el mejor desempeño de todos en el dataset de phishing con métricas de: 98.44% en accuracy, 98.42% en precisión, 97.87% en recall y 99.69% en AUC. Incluso, al experimentar su rendimiento en cada uno de los tipos de datos, BERT finetuned obtuvo resultados óptimos por encima del 97% en todas las métricas, y por poco cumple con las expectativas de las curvas ideales en los gráficos de ROC-AUC y Precision-Recall. Mientras los demás modelos destacaban en un solo tipo de dato particular, BERT demuestra su capacidad para detectar phishing con exactitud y eficacia en más de un solo tipo de dato como correos, mensajes SMS, sitios web y URL, gracias a la complementariedad y heterogeneidad de la información durante su entrenamiento.

CONCLUSIONES

Se ha propuesto un método basado en procesamiento de lenguaje natural que ha mostrado resultados alentadores para la detección eficaz y fiable de phishing en correos, mensajes SMS, sitios web y URL. Empleando el modelo de lenguaje BERT y un conjunto de datos robusto y variado se ha podido crear un sistema de detección con mucho potencial para el continuo combate contra el phishing en el mundo. De hecho, BERT ha demostrado su superioridad en tareas de clasificación de texto frente a otros modelos de vanguardia gracias a su arquitectura bidireccional, que le permite entender el contexto del lenguaje humano usado en ataques de phishing. Además, el análisis exploratorio realizado en el dataset con el que se entrenó BERT reveló ciertas propiedades en URL, sitios web y palabras clave en correos y mensajes SMS que suelen estar presentes en ataques de phishing comunes. En resumen, se ha cumplido con los

objetivos de este proyecto y se espera que tanto la información recolectada como el modelo de detección de phishing puedan contribuir de manera significativa a la prevención de ataques de phishing dirigidos hacia individuos y empresas.

Apéndice A. Dataset de Phishing

El dataset de phishing es un conjunto de datos que incluye URL, mensajes SMS, correos y sitios web. Fue diseñado para entrenar a BERT en la detección de phishing en varios contextos. La forma final de este dataset puede construirse de dos maneras: utilizando todas las muestras disponibles en el dataset de URL, o reduciendo el número de URL hasta lograr un equilibrio en la representatividad con los otros tipos de datos.

En un entrenamiento previo de BERT con la primera versión del dataset, se produjeron resultados negativos y erráticos, como lo demuestra la siguiente gráfica de la función de pérdida durante el entrenamiento:

Figura A1

Función de pérdida durante el entrenamiento de BERT en la primera versión del dataset de phishing.

La función de pérdida, también denominada función de error, es un componente crucial en el aprendizaje automático que ofrece una métrica clara para evaluar el rendimiento de un modelo, cuantificando la diferencia entre las predicciones y los resultados reales (Alake, 2023). Aunque los algoritmos de un modelo de aprendizaje

automático buscan minimizar la pérdida, en este caso, el valor de la pérdida aumentó en lugar de disminuir, y luego se mantuvo constante. Esto implicó que el modelo no estaba aprendiendo, y, por lo tanto, sus predicciones eran deficientes.

En cambio, como se observa en la sección de resultados de entrenamiento, reducir el número de URL produjo resultados positivos en las predicciones y métricas de rendimiento. Por esta razón, se decidió entrenar a BERT con la segunda versión del dataset. Finalmente, así fue como se construyó el dataset de phishing que se utiliza en este proyecto.

Tabla A1

Dataset final de phishing con muestras de correos, SMS, sitios web y URL.

text	label
You will receive your tone within the next 24hrs. For Terms and …	1
tftp://93.190.142.201/7up	1
checking build system type … i686-pc-linux-gnu checking host …	0
guiadeiraquara.com.br/doc/Gdoccc/	1
quality software for less. the no.1 source for software superstore …	1
let’s go to dinner at lagrasta’s	0
<title> You are being redirected … </title><noscript></noscript …	0
www.interal.com/en/index.asp	0
…	…

Nota. Esta tabla no expone todas las muestras del dataset de phishing, y algunas muestras no exponen todo el texto debido a su longitud. Esta tabla sirve tan solo para que el lector tenga una idea de cómo se ve el dataset final con el cual se entrena BERT.

Apéndice B. Características de URL y Sitios Web Relacionadas al Phishing

Varios estudios han identificado ciertos patrones y propiedades recurrentes en los ataques de phishing. Este apéndice se enfoca en las características específicas que se derivan de las URL y sitios web, las cuales fueron obtenidas durante el análisis exploratorio de un conjunto de datos de phishing. El propósito es ilustrar cómo estas características están vinculadas con el phishing y explicar su utilidad en la detección de patrones de malignos.

Tabla B1

Características en URL relacionadas al phishing.

Características	Descripción
Servicio de acortamiento	Los servicios de acortamiento de URL, como TinyURL, hacen que los enlaces sean más cortos y fáciles de compartir. Existe evidencia contundente que indica que los atacantes explotan estos servicios para ejecutar sus ataques[a].
Presencia del prefijo-Sufijo "-" en el dominio y del "@"	Generalmente en ataques de URL codificadas, los atacantes usan símbolos especiales como el "-", "@", "&", "/" para sobrepasar la lógica de validación[b]. Además, los resultados de un estudio confirman que la presencia de más de un guión en el dominio puede indicar que se trata de URL de phishing[c].
Tiene un script de redirección	Un método para construir URL maliciosas que parezcan legítimas es utilizando un script de redirección, como: http://www.google.com/url?q=http://www.badsite.com[d]
Presencia de palabras clave como "login", "admin", "client" y "server"	Autores de páginas web maliciosas con frecuencia explotan la familiaridad de los usuarios hacia una página web, incluyendo palabras en la URL que podrían hacer pensar al usuario que se trata de un sitio web legítimo[e].
Dirección IP en el dominio	Algunos ataques de phishing se alojan en computadoras comprometidas. Estas máquinas no tienen entradas DNS, y la forma más sencilla de referirse a ellas es a través de la dirección IP. Las compañías rara vez enlazan sus páginas con una dirección IP[d].
URL codificada	Unicode provee de un número único para cada carácter. El estudio de Jeeva & Rajsingh (2016) menciona que la mayoría de URL malignas están codificadas con caracteres unicode.

Longitud de la URL	Las URL largas son usadas por el atacante para esconder la parte dudosa de la URL[c].
Número de puntos	Al construir URL maliciosas, los atacantes suelen incluir un gran número de puntos en la URL[d].

Nota. [a]McGrath & Gupta (2008), [b]Aljabri et al. (2022), [c]Jeeva & Rajsingh (2016), [d]Fette et al. (2007), [e]Opara et al. (2024).

Tabla B2

Características en sitios web relacionadas al phishing.

Características	Descripción
Número de funciones sospechosas	Ciertas funciones de JavaScript como `eval()` y `unescape()` se usan comúnmente para el descifrado y la ejecución de explotaciones "drive-by-download"[a].
Número de scripts	Los atacantes usan JavaScript para esconder información del usuario, y potencialmente lanzar ataques sofisticados[b].
Número de Iframes	Los scripts maliciosos suelen inyectar varios Iframes en una página web, y, si el script no está ofuscado, es posible identificar cuando un script modifica el DOM para inyectar un elemento Iframe[a].
Número de links internos y externos	Los sitios web legítimos tienen links internos que apuntan a subdirectorios en el sitio web y pocos links externos apuntando a sitios web legítimos. En cambio, los sitios web maliciosos usan recursos externos para enriquecer el contenido y hacer creer a los usuarios que la página es legítima[c].
Número de espacios en blanco	Los sitios web legítimos tendrán más espacio entre el código para asegurarse de brindar una buena experiencia incluso para el usuario encargado de la seguridad. Como resultado, mientras más espacios en blanco haya en el contenido HTML, más probable es que el sitio web sea legítimo[c].
Número de elmentos incrustados	Elementos como `script`, `iframe`, `frame`, `embed`, `form` y `object` pueden ser usados para incluir contenido externo en la página web[a].
Número de funciones que modifican el DOM	Las explotaciones "drive-by-download" usualmente llaman varias veces estas funciones para instanciar componentes vulnerables y/o crear elementos en la página con el propósito de cargar scripts y páginas externas[a].

Nota. [a]Canali et al. (2011), [b]Fette et al. (2007), [c]Opara et al. (2024).

Apéndice C. Metodología de Entrenamiento

El primer paso fue entrenar BERT en el dataset final de phishing. Para ello se utilizó la librería de Transformers de Python y el tokenizador de BERT. Todo esto se ejecutó en un servidor Linux con una GPU NVIDIA A100 de 80GB, y los siguientes parámetros de entrenamiento:

- Tasa de aprendizaje (learning rate): 2×10^{-5}
- Tamaño de batch de entrenamiento (train batch size): 16
- Tamaño de batch de evaluación (test batch size): 16
- Semilla (seed): 42
- Optimizador (optimizer): Adam
- Número de épocas: 4
- Penalización de pesos (weight decay): 0.1
- Ritmo de aprendizaje (learning rate scheduler): linear

Después del entrenamiento se subió el modelo BERT finetuned a la plataforma de HuggingFace con acceso público. Esto evita que sea necesario reentrenar el modelo para hacer inferencias y permite que otros puedan beneficiarse del mismo.

La comparativa entre modelos se llevó a cabo en la plataforma de nube Colaborativa de Google, con una GPU NVIDIA Tesla T4 de 15GB y RAM ampliada de 51GB. Debido a las limitaciones de GPU, se tuvieron que ajustar los parámetros de entrenamiento de BERT a los siguientes valores:

- Tamaño de batch de entrenamiento (train batch size): 4
- Tamaño de batch de evaluación (test batch size): 4
- Número de épocas: 1

Los otros modelos que se compararon con BERT fueron: XGBoost, MNB y LSTM-CNN. Estos modelos requieren transformar el texto en valores numéricos mediante algún método de vectorización. Se utilizó el vectorizador TF-IDF para XGBoost y MNB, y el vectorizador TextVectorization de la librería de Keras para LSTM-CNN. Este último tiene un funcionamiento similar al TF-IDF, pero con un enfoque distinto.

Después de tener los datos listos, se definieron los modelos con los siguientes parámetros y estructura:

- XGBoost: se establecieron parámetros por prueba y error, y el objetivo se fijó en binario-logístico ya que solo hay dos posibles salidas 1 (phishing) o 0 (benigno).

```
xgb = XGBClassifier(
      colsample_bytree = 0.7,
      gamma = 0.2,
      learning_rate = 0.1,
      max_depth = 12,
      min_child_weight = 2,
      n_estimators = 100,
      subsample = 0.8,
      objective = 'binary:logistic'
)
```

- MNB: se definió un solo parámetro, de igual manera por prueba y error.

```
mnb = MultinomialNB(alpha=0.01)
```

- LSTM-CNN: se construyó una red neuronal con varias capas: vectorización, incrustación o “embedding”, deserción espacial, LSTM, CNN, agrupación máxima global, capa densa y una capa final de dos neuronas con función de activación “softmax”:

```
tv = TextVectorization()
tv.adapt(text)
```

```
lstm_cnn = Sequential()
lstm_cnn.add(tf.keras.Input(shape=(1,), dtype=tf.string))
lstm_cnn.add(tv)
lstm_cnn.add(Embedding(MAX_WORDS_NUM, EMBEDDING_DIM,
input_length=MAX_SEQUENCE_LENGTH))
lstm_cnn.add(SpatialDropout1D(0.2))
lstm_cnn.add(LSTM(100, return_sequences=True))
lstm_cnn.add(Conv1D(50, kernel_size=3, activation='relu'))
lstm_cnn.add(GlobalMaxPooling1D())
lstm_cnn.add(Dense(32))
lstm_cnn.add(Dense(2, activation="softmax"))

lstm_cnn.compile(
    optimizer=Adam(learning_rate=1e-4),
    loss="binary_crossentropy",
    metrics=['accuracy', Precision(), Recall(), AUC()]
)
```

Finalmente se entrenaron los modelos y se hicieron predicciones sobre los conjuntos de prueba. Con los resultados se calcularon las métricas de rendimiento y se generaron las gráficas de ROC-AUC y Precision-Recall, que se pueden visualizar en el experimento asociado.

REFERENCIAS BIBLIOGRÁFICAS

Abbas, M., Ali, K., Jamali, A., Memon, S., & Ahmed, A. (2019). Multinomial Naive Bayes Classification Model for Sentiment Analysis. *International Journal of Computer Science and Network Security*, 19(3), 62.

Alake, R. (2023). Loss Functions in Machine Learning Explained. Obtenido de Datacamp: https://www.datacamp.com/tutorial/loss-function-in-machine-learning

Aljabri, M., Alhaidari, F., Mohammad, R. M. A., Rami Mustafa, A. S. M., Alhamed, D. H., Altamimi, H. S., & Sara Mhd, B. C. (2022). An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models. *Computational Intelligence and Neuroscience : CIN, 2022*. https://doi.org/10.1155/2022/3241216

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., UI Haq, Q. E., Saleem, K., & Faheem, M. H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics*, 12(1).

Amazon. (2023). ¿Qué es el Procesamiento de lenguaje natural (NLP)? (Amazon Web Services) Recuperado el 24 de Septiembre de 2023, de https://aws.amazon.com/es/what-is/nlp

Ariyadasa, S., Fernando, S., & Fernando, S. (2021). Phishing Websites Dataset. Obtenido de Mendeley Data: https://doi.org/10.17632/n96ncsr5g4.1

Banjara, B. (2024). A Comprehensive Guide to Fine-Tuning Large Language Models. Obtenido de Analytics Vidhya: https://www.analyticsvidhya.com/blog/2023/08/fine-tuning-large-language-models

Basit, A., Zafar, M., Javed, A., Jalil, Z., & Kifayat, K. (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems*, 76, 139-154.

Brownlee, J. (2020). ROC Curves and Precision-Recall Curves for Imbalanced Classification. Obtenido de Machine Learning Mastery:

https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

Brownlee, J. (2021). Random Oversampling and Undersampling for Imbalanced Classification. Obtenido de Machine Learning Mastery: https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

Canali, D., Cova, M., Vigna, G., & Krueger, C. (2011). Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages. *Proceedings of the 20th international conference on World wide web*, (págs. 197-206).

Chakraborty, S. (2023). Phishing Email Detection. (Kaggle) Recuperado el 28 de Septiembre de 2023, de https://doi.org/10.34740/KAGGLE/DSV/6090437

Chaudhary, M. (2020). TF-IDF Vectorizer scikit-learn. Obtenido de Medium: https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a

Check Point Software. (2023). 2023 Cybersecurity Report. Check Point.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional Transformers for language understanding. Obtenido de arXiv.org: https://arxiv.org/abs/1810.04805

Diaz, R. (2024). Métricas de Clasificación. Obtenido de The Machine Learners: https://www.themachinelearners.com/metricas-de-clasificacion/

Divakaran, M. D., & Oest, A. (2022). Phishing Detection Leveraging Machine Learning and Deep Learning: A review. *IEEE Security & Privacy*, 20(5), 86-95.

Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to Detect Phishing Emails. *Proceedings of the 16th International Conference on World Wide Web* (págs. 649-656). Banff: WWW2007.

He, H., & Ma, Y. (2013). Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley-IEEE Press.

Hernandez, C., & Rodriguez, J. E. (2013). Preprocesamiento de datos estructurados. *Revista Vínculos*, 4(2), 27-48.

Hugging Face. (2024). Tokenizadores. Obtenido de NLP Course Documentation: https://huggingface.co/learn/nlp-course/es/chapter2/4

IBM. (2023a). Cost of a Data Breach. IBM Security.

IBM. (2023b). ¿Qué es el phishing? Obtenido de IBM: https://www.ibm.com/es-es/topics/phishing

IBM. (2023c). ¿Qué es el procesamiento del lenguaje natural (NLP)? Obtenido de IBM: https://www.ibm.com/es-es/topics/natural-language-processing

Interpol. (2020). Ciberdelincuencia: Efectos de la Covid-19. Lyon: Secretaría General de Interpol.

Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing url detection using association rule mining. *Human-Centric Computing and Information Sciences, 6*(1), 1-19. https://doi.org/10.1186/s13673-016-0064-3

Kamm, S., Veekati, S., Müller, T., Jazdi, N., & Weyrich, M. (2023). A survey on machine learning based analysis of heterogeneous data in industrial automation. *Computers in Industry*, 149(103930).

Kaspersky. (2023a). Ingeniería Social. Obtenido de Kaspersky: https://latam.kaspersky.com/resource-center/definitions/what-is-social-engineering

Kaspersky. (2023b). Nueva epidemia: el phishing se sextuplicó en América Latina con el reinicio de la actividad económica y el apoyo de la IA. (Panorama de amenazas) Recuperado el 23 de Septiembre de 2023, de https://latam.kaspersky.com/blog/panorama-amenazas-latam-2023/26586

Kim, S.-W., & Gil, J.-M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences*, 9(1), 1-21.

Kiwop. (2023). TF IDF: ¿Qué es y por qué es tan importante para posicionar? Obtenido de Kiwop: https://www.kiwop.com/blog/tf-idf-que-es-y-su-importancia-en-el-seo

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and Applications*, 22, 113-122.

McGrath, D.K., & Gupta, M. (2008). Behind Phishing: An Examination of Phisher Modi Operandi. *USENIX Workshop on Large-Scale Exploits and Emergent Threats*.

Microsoft. (2023). ¿Qué es el phishing? Obtenido de Microsoft: https://www.microsoft.com/es/security/business/security-101/what-is-phishing

Mishra, S., & Soni, D. (2022). SMS Pishing Dataset For Machine Learning And Pattern Recognition. (Mendeley Data) Recuperado el 28 de Septiembre de 2023, de https://doi.org/10.17632/f45bkkt8pr.1

Muller, B. (2022). BERT 101 State Of The Art NLP Model Explained. Obtenido de Hugging Face: https://huggingface.co/blog/bert-101

Navarro, J. M. (2019). Ingeniería Social ¿Por qué eres el eslabón más débil? Obtenido de LinkedIn: https://www.linkedin.com/pulse/ingenier%C3%ADa-social-por-qu%C3%A9-eres-el-eslab%C3%B3n-m%C3%A1s-d%C3%A9bil-james-navarro

NVIDIA. (2023b). XGBoost. Obtenido de NVIDIA Glossary: https://www.nvidia.com/en-us/glossary/data-science/xgboost/

Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. Expert Systems with Applications, 236(121183).

Pachecho Najar, J. C. (2017). Exposición del activo más valioso de la organización, la información. Visión Electrónica, 11(1), 3.

Reed, C. (2023). 30 Social Engineering Statistics – 2023. Obtenido de Firewall Times: https://firewalltimes.com/social-engineering-statistics/

Regan, J. (2018). ¿Qué es el phishing? La guía definitiva sobre las estafas y los correos electrónicos de phishing. Obtenido de AVG: https://www.avg.com/es/signal/what-is-phishing

Romero, H. (2023). Inteligencia Artificial y sus maravillas: Descubriendo los LLM, transformers y MLLM. (LinkedIn) Recuperado el 25 de Septiembre de 2023, de https://www.linkedin.com/pulse/inteligencia-artificial-y-sus-maravillas-descubriendo-romero-pico

Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. IEEE Access, 10, 65703-65727.

Shaked, S. (2021). Overcome Data Shortages for ML Model Training with Synthetic Data. Obtenido de TDWI: https://tdwi.org/articles/2021/06/14/adv-all-overcome-data-shortages-for-ml-model-training-with-synthetic-data.aspx

Shirazi, H., Haynes, K., & Ray, I. (2022). Towards Performance of NLP Transformers on URL-Based Phishing Detection for Mobile Devices. Journal of Ubiquitous Systems & Pervasive Networks, 3(1), 35-42.

Sudhan, H. (2023). Phishing and Legitimate URLS. (Kaggle) Recuperado el 28 de Septiembre de 2023, de https://www.kaggle.com/datasets/harisudhan411/phishing-and-legitimate-urls/data

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? Chinese Computational Linguistics: 18th China National Conference, CCL 2019 (págs. 194-206). Kunming, China: Springer International Publishing.

Surática. (s.f.). ¿Qué es la minificación? Obtenido de Surática: https://www.suratica.es/que-es-la-minificacion/

Talebi, S. (s.f.). Fine-Tuning Large Language Models (LLMs). (Medium) Recuperado el 28 de Septiembre de 2023, de https://towardsdatascience.com/fine-tuning-large-language-models-llms-23473d763b91

Wang, D., Zhao, T., Yu, W., Chawla, N., & Jiang, M. (2023). Deep Multimodal Complementarity Learning. IEEE Transactions on Neural Networks and Learning Systems, 34(12), 10213-10224.

Wenz, V., Kesper, A., & Taentzer, G. (2021). Detecting Quality Problems in Data Models by Clustering Heterogeneous Data Values. ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C) (págs. 150-159). Fukuoka, Japón: IEEE.