Análisis y Diseño de un Modelo Predictivo para Detección de Phishing Basado en Url y Corpus del Correo Electrónico

Albán, Fernanda [1,*]; **Urvina, Menthor** [2]; **Andrade, Roberto** [3]

[1] *Escuela Politécnica Nacional, Facultad de Ciencias, Quito, Ecuador*
[2] *Escuela Politécnica Nacional, Departamento de Matemática, Quito, Ecuador*
[3] *Escuela Politécnica Nacional, Departamento de Informática y Ciencias de la Computación, Quito, Ecuador*

Resumen: Uno de los delitos cibernéticos más reportados a nivel mundial es el phishing. En la actualidad se están desarrollando diversos sistemas anti-phishing (APS) para identificar este tipo de ataque en sistemas de comunicación en tiempo real. A pesar de los esfuerzos de las organizaciones, este ataque continúa creciendo, teniendo como causas: la detección errónea en el ataque de día cero, el alto costo computacional y las tasas altas de falsificación. Aunque el enfoque de Machine Learning (ML) ha logrado una tasa de precisión favorable, se debe considerar que la elección y el rendimiento del vector de características es un punto clave para obtener un nivel de precisión adecuado. En este trabajo, se propone un modelo predictivo basado en ML y en el análisis de la eficiencia de algunos esquemas anti-phishing que sirvieron para entender esta temática. El modelo propuesto consta de un módulo de selección de características que se utiliza para la construcción del vector final. Estas características se extraen de la URL, las propiedades de la página web y del corpus de correo electrónico. El sistema utiliza los modelos de clasificación, Random Forest (RF) y Naïve Bayes (NB), que han sido entrenados en el vector de características. Los experimentos se basaron en Dataset compuestas por instancias de phishing y benignas. Mediante el uso de la validación cruzada, los resultados experimentales indican una precisión del 97,5 % para los dataset utilizados, mientras que para el abordaje de esta investigación a nivel local se obtuvo una precisión del 96,5 %.

Palabras clave: Anti-phishing; Ataques cibernéticos; Phishing; Middleware; Amenaza

Analysis and Design of a Predictive Model for Phishing Detection Based on Url and Email Corpus

Abstract: One of the most reported cyber crimes worldwide is phishing, and various anti-phishing systems (APS) are currently being developed to identify this type of attack on communication systems in real time. Despite the efforts of organizations, this attack continues to grow, due to the erroneous detection in the zero-day attack: the high computational cost and the high rates of forgery. Although the Machine Learning (ML) approach has achieved a favorable accuracy rate, it should be considered that the choice and performance of the feature vector is a key point to obtain an adequate level of accuracy. In this work, a predictive model based on ML and the analysis of the efficiency of some anti-phishing schemes that served to understand this issue is proposed. The proposed model consists of a feature selection module that is used to build the final vector. These characteristics are extracted from the URL, the properties of the web page, and the email corpus. The system uses the Random Forest (RF) and Naïve Bayes (NB) classification models, which have been trained on the feature vector. The experiments were based on datasets composed of phishing and benign instances. Using cross-validation, the experimental results indicate a precision of 97.5 % for the datasets used, while a precision of 96.5 % was obtained for the approach of this research at the local level.

Keywords: Anti-phishing; Cyberattacks; Phishing; Middleware; Threat

1. Introducción

Phishing es una clase de ataque cibernético, del tipo ingeniería social, que se usa frecuentemente para el robo de datos personales, tales como: las credenciales de inicio de sesión y los números de tarjetas de crédito. Las técnicas de ingeniería social pretenden adquirir la identidad de los usuarios ingenuos o información confidencial sensible mediante el uso de correos electrónicos falsificados, sitios web falsos, anuncios/promociones dudosas en línea, SMS falsos de proveedores de servicios o empresas, entre otros. El objetivo principal de los phishers es: atacar a las grandes corporaciones, instituciones financieras y gubernamentales que

dolores.alban@epn.edu.ec
Recibido: 10/03/2022
Aceptado: 27/05/2022
Publicado en línea: 23/12/2022
10.33333/rp.vol50n3.03

generalmente sufren enormes daños en su credibilidad. Los informes de seguridad de Estadísticas y Tendencias en 2021 indicaron que en enero de ese año se registró el pico más alto a nivel histórico en la cantidad de sitios de phishing a nivel mundial, según reporte del *Anti-Phishing Working Group*.

Este tipo de ataque afecta principalmente al sector financiero, con un porcentaje del 24,9 % durante el primer trimestre, seguido por las redes sociales con el 23,6 % y los distribuidores de servicios de sitios web con el 19,6 %. Otro sitio que presentó un crecimiento acelerado en la cantidad correos electrónicos que registraron textos únicos en el campo asunto de los correos, con un total de 172.793 asuntos diferentes en un solo mes.

En la actualidad, los phishers han desarrollado un *ransomware* que ejecuta un código malicioso que afecta negativamente a los recursos informáticos y exige el pago de un rescate, para restaurar los recursos al estado original. La incidencia de correos con presencia de phishing es del 93 % según *Chief Security Officer*. El informe observó que la mayoría de las víctimas tiende a pagar rápidamente debido a la naturaleza sensible de sus recursos observar en Online report on phishing activities (2016).

Aunque cada día los sistemas se actualizan, los phishers buscan nuevas maneras de atacar y efectuar el robo a los usuarios. Varios de los sistemas de phishing desarrollados se enfocan principalmente en la revisión de URL, o la validación de protocolos de seguridad como https. Sin embargo, estas alternativas de control han sido abordadas por los phishers, obteniendo certificados válidos o cambiando continuamente las URL,lo que dificulta la detección a través de las herramientas de seguridad. Motivados por esta premisa, el objetivo de esta investigación es desarrollar un esquema anti-phishing apoyado en las características presentes en el Corpus del email, URL y Dominios, mediante la construcción de modelos de clasificación, utilizando la base de ajuste de parámetros en RF y NB con tres fuentes de datos para detectar phishing.

Como resultado de esta investigación, los principales aportes son:

1) Determinar características para clasificar entre sitios web de phishing, sospechosos y legítimos, obtenidos de los tres Dataset.

2) Identificar el vector de características final que ha ofrecido el mejor rendimiento en comparación con otros en el campo anti-phishing.

3) Identificar el modelo con mayor precisión mediante métricas usadas frecuentemente en la detección de phishing.

El documento también presenta la ventaja de la detectabilidad en la elección del conjunto de características para el corpus del email.

2. Marco Teórico

En esta sección, se describen las nociones teóricas necesarias para comprender la clasificación de URL utilizando métodos estadísticos para descubrir las propiedades léxicas, basadas en *host* de las URL de sitios web maliciosos, con el propósito de entender como clasificar la presencia de un ataque malicioso a gran escala para la construcción del modelo de predicción.

Una vez que los datos han sido entendidos, se puede establecer una idea referente al camino a tomar o sobre la técnica a emplear. Para predecir la probabilidad de ser víctima de robo de información o suplantación de identidad se ha dividido en dos secciones principales: metodología de investigación, análisis de características para detección de phishing, problemas de clasificación. Las referencias principales son Calva (2020), Orunsolu et al. (2019), Rosero (2020) y Hastie et al. (2017), que serán continuamente utilizadas en este trabajo.

2.1 Metodología de Investigación

La problemática que se tiene para esta investigación, toma en cuenta que el phishing es un conflicto social y en la actualidad se busca una solución eficiente. Para este trabajo, se utilizará la metodología CRISP-DM que es un modelo de proceso independiente para la minería de datos. Consta de seis fases iterativas que van desde la comprensión del problema hasta el desarrollo del escrito final, como indica Creswell (2015).

A continuación, se comienza con la investigación, es importante mencionar que para el análisis de características referente a la detección de phishing se realizó una revisión de literatura teniendo como principales fuentes Adebowale et al. (2018), Chin et al. (2018), Orunsolu et al. (2019) y Gansterer and Polz (2009), para posteriormente realizar una comparación de resultados de otras investigaciones relacionadas, esto se visualizará en la sección de resultados.

2.2 Análisis de características para detección de phishing

La definición de ataque phishing es un caso típico de clasificación binaria, ya que una comunicación en línea, por ejemplo: (correo electrónico, sitio web y chat electrónico) puede ser clasificada como: Phishing o benigna.

Tratándolo más formalmente, sea w una solicitud que necesita clasificación, es decir

$$w \underset{\rightarrow}{x} \{\text{Phish, benigna}\}. \tag{1}$$

Entonces x es el sistema anti-phishing que toma características, $f_i \in w$ donde

$$w_i = (f_1, f_2, ...f_i, ..., f_m),$$

Es decir, w_i es un vector no vació.
Por lo tanto, una solicitud contiene al menos una característica, por ejemplo: (enlaces, etiquetas HTML, scripts, certificado SSL, etc.) sobre la cual se puede consultar o clasificar la predicción de su estado. Debido a que estas características pueden variar de simples a complejas, el modelo propuesto utiliza una evaluación de frecuencia de características para la composición de vectores de características representada por $x = \{x_1, x_2, ..., x_n\}$ que asignan la etiqueta y a cada $f_i \in w$, de modo que la etiqueta y es una clase binaria representada como:

$$y = \begin{cases} 1, & \text{si es phishing,} \\ 0, & \text{si es benigna,} \end{cases} \quad (2)$$

Representado (2) como

$$x_i : f(w) \rightarrow y$$

La ecuación (1) describe el problema de clasificación donde, dado un dato de entrenamiento D, que contiene $(w_1, w_2, ..., w_n)$ y cada w_i contiene un conjunto de características $(f_1, f_2, ..., f_m)$. Además, los datos de entrenamiento son un conjunto de clases. $C = (c_1, c_2)$ que representa sitios legítimos y de suplantación de identidad que:

$$c_1 = \{w_i, f_i \mid w_i \in D, y = benigna, i = 1, ..., m\},$$
$$c_2 = \{w_i, f_i \mid w_i \in d, y = phishing, i = m+1, ..., p\}.$$

Por tanto, cada caso $w_i \in D$ se le puede dar una clase $c_i \in C$ y se representa como un par $(w_i, (c_i))$ dónde c_i es una clase de C asociada con el caso w_i en los datos de entrenamiento. Sea H el conjunto de clasificadores para $D \rightarrow C$, donde cada caso $c_i \in C$ se le da una clase y el objetivo es encontrar un clasificador $h_i \in H$ que maximiza la probabilidad de que $h(c_i) = C$ para cada caso de prueba. En el sistema propuesto, se eligen dos clasificadores de aprendizaje automático más comunes para la clasificación de phishing.

2.3 Módulo de selección de funciones

Un proceso de extracción de características implica la identificación de ciertas características en un conjunto particular de datos, por ejemplo: (phishing o benignas). Tales características generalmente se marcan como "huellas digitales", ocurren con poca o ninguna probabilidad, en la mayoría de los casos, estos rasgos suelen excluirse mutuamente de las otras. En este enfoque, se utiliza la evaluación de características basada en el análisis de frecuencia de varias características recopiladas de literatura existente. Esto se define como un módulo de selección de características (FSM) que consta de:

- Las características de la URL
- Las propiedades del documento web
- Las características del email

Estos tres componentes se consideran un filtro en FSM y cada factor se organiza en el enfoque para tener un sistema apoyado en componentes. Con base en esto, los tres filtros se construyen como un filtro unitario y uno compuesto para lograr gradualmente un planteamiento de detección eficiente.

2.4 Las características de la URL (filtro F1)

Las características de la URL representan las características asociadas con las direcciones web donde se puede recuperar una página en particular de Internet. Las características de la URL se extraen ya sea una URL absoluta o una URL relativa mediante el análisis de la estructura de enlaces en el DOM (Dominio). Para la extracción de identidad de URL, FSM considera **href** y **src** atributos de los enlaces de anclaje, en particular las etiquetas $< a >$, $< area >$, $< link >$, $< img >$ y $< script >$ del árbol de DOM, una página web donde normalmente se encuentran las direcciones web. Basándose en el estudio preliminar, se construyó el módulo de selección de características mediante una serie de consultas sobre ciertos rasgos de la URL seleccionadas de las investigaciones existentes (ver, Aburrous et al. (2010), Gowtham and Krishnamurthi (2014), Sonowal and Kuppusamy (2020) y Zouina and Outtaj (2017)).

Basado en la metodología de evaluación de características de frecuencia, se presenta el algoritmo 1.

Algorithm 1 Análisis de frecuencia de evaluación de características de URL

Require: Corpus de phishing actualizado, d_{ph}, URL, d_{be}, Valor umbral predefinido,θ.
Ensure: Vector de dimensión de características basado en URL S_m
Empezar
1. Para $i = 1$ hasta n:
2. $F_{URL(n)} \leftarrow$ el conjunto de todas las n funciones de URL.
3. Si $F_{url\_i} \in d_{ph}$ ó d_{be} Luego
4. $S \leftarrow$ nueva lista de funciones
5. Calcular la frecuencia de $F_{url\_i} \in d_{ph}.d_{be}$
6. Calcular la información de frecuencia, FI, de F_{url_i}
7. Si $FI_{F_{url\_i}} > \theta$, Luego
8. Adjuntar F_{url_i} a S
9. **Caso Contrario**
Rechazar F_{url_i}
10. $i = i + 1$
Continuar
11. Rango $F_{url\_i} \in S$
12. Seleccionar la parte superior F_{url_i} características $\in S$
13. Obtener una medida de desempeño de S
14. Identificar la mejor medida de desempeño como las mejores características
15. $S_m \leftarrow$ mejores características
16. Fin

La Tabla 6 presenta el significado de las notaciones utilizadas en el 1, esta se encuentra disponible en el Apéndice A.

2.5 Las propiedades del documento web (filtro F2)

Las propiedades del documento web de una página web se extraen de la etiqueta de documento. Este proceso de extracción se basa en el concepto de método Término-Frecuencia de un Documento de Frecuencia Inversa (TF-IDF). El método se utiliza para extraer un conjunto de palabras clave del documento d, que se recopila de varias partes de una página web. El TF-IDF refleja la estadística numérica de cuán relevante es una característica para un documento en un corpus de datos. El valor de TF-IDF aumenta proporcionalmente al número de veces que aparece una característica en el documento, pero se compensa con la frecuencia de la característica en el corpus. Por tanto, un término particular definido como t tiene un peso TF-IDF alto si el término tiene una frecuencia alta en un documento D dado y una frecuencia baja si el término es relativamente poco común.

Dado un documento dy su conjunto de identidad de términos t, el FSM usa la medición de la tasa de frecuencia para determinar la inclusión de una característica en la clase discriminatoria. A continuación, se presenta el Algoritmo de Análisis de frecuencia de evaluación de características.

Algorithm 2 Análisis de frecuencia de evaluación de características

Require: Tamaño de datos(p), conjunto de características original(n), umbral(θ), clase(C).
Ensure: Subconjunto de características principales de dimensión $m(fs)$
Empezar
1. Para $i = 1$ hasta n:
2. Para $j = 1$ hasta p:
3. a=1
4. Seleccioanr $s_i \in H_P$
5. Calcular CFS usando {
6. $s(f_i, f_2, ..., f_n, c)$ como entrada
7. Para $i = 1$ hasta n:
8. Inicializar el factor de correlación apropiado, t
9. $r = calcular\_correlacion(f, c)$
10. Si $(t > r)$ luego
11. Adjuntar $f_i \in s_n$ en m
12. Fin
13. Ordenar m en valor descendente de t
14. Quitar f con rango inferior
15. Devolver f predominante como $f(s)$
16. Fin}
17. Si $(f_i(s) > \theta)$ luego agregar a $m(fs)$
18. $c = c + 1$
19. Si $(a \leq n)$ volver a 3
20. Fin para
21. Fin para
22. Conjunto de características de retorno $m(fs)$
23. Fin

El algoritmo 2 presenta el flujo de la metodología del sistema, y las notaciones para este algoritmo se encuentran la Tabla 6.

2.6 Las propiedades del corpus del email (filtro F3)

A veces, hay casos en los que el filtrado de *spam* no tiene éxito en evitar que el *malware* de productos básicos u otros correos electrónicos no solicitados lleguen.

Es por esta razón que, para este filtro se analizará el contenido de los correos tanto en los que presentan phishing como los emails reales mediante la técnica de Text Mining con el propósito de detectar patrones adicionales a los que se conocen por defecto que son:

- El remitente no corresponde con el servicio que envía el correo
- La gramática con fallos
- URL_s falsas camufladas en hipervínculos
- Archivos adjuntos que no son lo que parecen
- Correo de un servicio no utilizado o no contratado

2.7 Problemas de clasificación y su modelamiento

Para este trabajo de investigación, se trabajará con los problemas de clasificación, frecuentemente se divide la base en dos conjuntos de datos que son: entrenamiento/training y prueba/test, siguiendo normalmente el criterio de Pareto de 80 % y 20 %.

2.8 Random Forests

El Bagging o bootstrap es una técnica para procedimientos de varianza grandes y sesgos pequeños, como lo son los árboles que son más simples de entrenar y ajustar. La idea esencial del bagging es promediar muchos modelos ruidosos, pero aproximadamente insesgados y, por lo tanto, reducir la varianza. Los árboles son candidatos ideales para el bagging, ya que pueden capturar interacciones complejas.

Dado que los árboles son notoriamente ruidosos, se beneficia enormemente el promedio. Además, dado que cada árbol generado en el bagging se distribuye de manera idéntica $(i.d.)$, la expectativa del promedio de B árboles es lo mismo que la expectativa de cualquiera de ellos. Esto contrasta con el impulso, donde los árboles se generan de forma adaptativa para eliminar el sesgo y, por lo tanto, no son $i.d.$

2.9 Naïve Bayes

El NB es un algoritmo de clasificación de texto simple y efectivo que usa las probabilidades conjuntas de palabras y categorías para estimar las probabilidades de categorías dado un documento como se menciona en Anwar et al. (2017). El supuesto de independencia condicional se puede expresar formalmente como:

$$P(A \mid C = c) = \prod_{i=1}^{n} P(A_i \mid C = c),$$

donde cada conjunto de atributos o conjunto de características $A = \{A_1, A_2, ..., A_n\}$ consta de n valores de atributo. Con el supuesto de independencia condicional, en lugar de calcular la probabilidad condicional de clase para cada agrupación de A, solo estime la probabilidad condicional de cada A_i, dado C. Para clasificar una muestra de prueba, el clasificador NB calcula la probabilidad posterior para cada clase C como:

$$P(C \mid A) = \frac{P(C) \prod_{i=1}^{n} P(A_i \mid C)}{P(A)}. \quad (3)$$

La ecuación (3) indica que al observar el valor de una característica particular, A_i, la probabilidad previa de una categoría particular, C_i, $P(C_i)$ se puede convertir a la probabilidad posterior, $P(C_i \mid A_i)$, que representa la probabilidad de una característica en particular, A_i siendo una categoría particular, C_i .

3. Marco Metodológico

En esta sección, se centrará en la segunda etapa de la metodología CRISP-DM que se basa en la recopilación de datos para poder abordar el problema, como siguiente paso el análisis de mismos para llegar a verificar la calidad de los datos, en la siguiente etapa se procede a la construcción del vector de características esenciales para la generación del modelo. Como fuente principal para esta sección, véase, Rosero (2020), Martínez (2018) y Zhao et al. (2019).

Para este proceso se ha utilizado el método *Feature Selection*, que permite preprocesar las características de las URL_S y correos electrónicos, teniendo una peculiaridad que es imputar los innecesarios sin correr el riesgo de pérdida de información. Además, se utilizó RF y NB para la construcción del vector de aprendizaje automático. A continuación, en la Figura 1 se ilustra el diagrama de flujo del modelo a realizar.

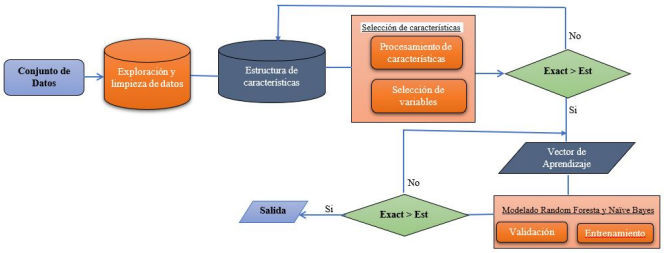

Figura 1. Diagrama de flujo del modelo de detección y mitigación de phishing

Es de vital importancia mencionar que para este proyecto se seleccionó los lenguajes de programación Java para la construcción del vector de características y el software Python para la implementación de los modelos predictivos, se escogió estos lenguajes porque proporcionan librerías eficientes en la manipulación y tratamiento de este tipo de registros.

En la siguiente sección, se va a describir los datos descargables con que se trabajará para el proyecto de investigación.

3.1 Descripción del conjunto de datos

Se utilizan tres conjuntos de datos que se encuentran disponibles al público, estas bases contienen URL malignas y reales, correos con presencia de phishing y correos benignos para evaluar el rendimiento de la arquitectura de detección de phishing propuesta. Los conjuntos experimentales se obtienen de un corpus de datos extraídos de Monkey (2020), Phishtank (2021) y William and Cohen (2019).

El número de datos recolectados en cada conjunto es: 14200 en PhishTank, 3700 en Monkey y para Enron se tiene 16800, dándonos un total general de 31084; estos datos posteriormente se los tratará para obtener una data limpia lista para realizar análisis y visualizar algunos resultados.

Ahora bien, estas tres fuentes de datos se eligen porque contienen conjuntos de datos verificados que se utilizan en la mayoría de las investigaciones anti-phishing para comparar los resultados de la evaluación.

3.2 Tratamiento de datos

Una vez identificados los datos, se sigue con la exploración y análisis de datos, que se lleva a cabo con el uso de técnicas estadísticas, que nos dirán si los datos extraídos son válidos. La etapa de exploración se hizo a través de búsquedas básicas de los Dataset con extensión M.box y CSV.

Los documentos descargables contienen las URL_s y correos electrónicos con Phishing y sin Phishing, por tal razón no requiere de una discriminación entonces se lleva a cabo un proceso aparte, debido a que, aunque se cuenta con Datasets completamente diferentes en sus categorías, y sus variables son parecidas. Se eligen estos datos para hacer el análisis.

1). header['From']

2). header['Subject']

3). header ['Content-Type']

4). header['Date']

5). header['Body']

Siguiendo con este proceso se limpia los datos únicamente para las variables seleccionadas, lo primero que se va a corregir es el porcentaje de valores nulos. Es así que se tiene los siguientes resultados, para PhishTank se tiene 5,9 %, Monkey con 10,8 % y Enron tiene 10 % de valores nulos, estos valores son aceptables y no necesitan un tratamiento especial.

Con los datos que se obtuvieron de este análisis, se podrá determinar que dominios predominan en la falsificación de identidad. Seguidamente, se visualizan estos resultados de las principales fuentes utilizadas para este estudio que son: PhishTank y Monkey.

Se observa en la Tabla 1 que el dominio Bank presenta la frecuencia más alta en el contexto de ser víctima de Phishing para la data de PhishTank con un porcentaje de ocurrencia del 18 %, le sigue Gmail con el 16 %. Ahora bien, para la base Monkey el dominio predominante es Gmail con una representatividad del 45 % y esto se sobreentiende dado que esta data en particular contiene correos infectados. De manera general los phishers apuntan a tener una buena ganancia con el menor tiempo de ejecución.

Posteriormente, se procede a identificar las palabras más usadas tanto en los emails con presencia de phishing y los reales. Mediante el método de exploración de texto (Text Mining) en las bases, esta técnica ayudó a identificar las palabras que presentan una alta frecuencia en los correos electrónicos con presencia de Phishing o benignas, Una vez obtenido el conjunto de palabras, se las tomará como variables en el modelo de detección. Ahora se observarán las palabras identificadas, la frecuencia absoluta y la frecuencia relativa en los emails, lo que permite observar la diferencia para la detección con más claridad.

Nota: Se utilizó esta técnica dado que se puede explorar y descubrir relaciones ocultas dentro de datos no estructurados. Dado que

Tabla 1. Dominios detectados como los más utilizados en la suplantación de identidad (Phishing) en PhishTank

Dominios	PhishTank	R.PhisT	Monkey	R.Monkey
Bank	1300	0,1831	227	0,0289
Gmail	1200	0,1690	3512	0,4472
Run escape	1039	0,1463	0	0,0000
Google	620	0,0873	231	0,0294
PayPal	609	0,0858	1020	0,1299
eBay	388	0,0546	1	0,0001
Facebook	345	0,0486	25	0,0032
office	259	0,0365	138	0,0176
Microsoft	217	0,0306	124	0,0158
Halifax	124	0,0175	0	0,0000
Amazon	119	0,0168	24	0,0031
Android	99	0,0139	393	0,0500
Apple	99	0,0139	695	0,0885
Netflix	93	0,0131	100	0,0127
Adobe	84	0,0118	13	0,0017
WhatsApp	75	0,0106	1	0,0001
YouTube	60	0,0084	54	0,0069
Steam	57	0,0080	0	0,0000
Yahoo!	55	0,0077	993	0,1264
Outlook	52	0,0073	61	0,0078
LinkedIn	40	0,0056	0	0,0000
Instagram	38	0,0054	9	0,0011
Virus total	34	0,0048	0	0,0000
Twitter	29	0,0041	17	0,0022
JPM Chase and Co.	22	0,0031	0	0,0000
Hotmail	12	0,0017	56	0,0071
American express	10	0,0014	100	0,0127
Vodafone	9	0,0013	0	0,0000
HSBCgruop	7	0,0010	0	0,0000
Windows	6	0,0008	60	0,0076

Tabla 2. Frecuencia absoluta de las palabras que se determinaron como características para detección de Phishing.

Palabra	Phishing	F.Rphis	No-Phishing	F.R.Nphis
Actualizar	233	0,006	746	0,073
Confirmar	121	0,003	197	0,019
Usuario	244	0,006	220	0,021
Cliente	45	0,001	245	0,024
Querido	112	0,003	65	0,006
Miembro	44	0,001	258	0,025
Restringir	365	0,009	257	0,025
Sostener	120	0,003	871	0,085
Verificar	242	0,006	18	0,002
Cuenta	402	0,010	179	0,017
Notificación	343	0,009	80	0,008
Login	236	0,006	12	0,001
Sesión	500	0,012	81	0,008
Clic aquí	1200	0,030	194	0,019
Contraseña	930	0,023	81	0,008
Felicidades	700	0,017	38	0,004
Felicitaciones	523	0,013	22	0,0012
Ganaste	189	0,005	0	0,000
Gratis	1200	0,030	934	0,091
Seguridad	118	0,003	40	0,004
Importante	1500	0,037	125	0,012
Aviso	47	0,001	135	0,013
Crédito	5800	0,144	833	0,081
Banco	14500	0,360	219	0,021
En línea	124	0,003	818	0,080
Enviar	448	0,011	593	0,058
Transferir	945	0,023	2057	0,201
Acceso	743	0,018	369	0,036
Contagio	38	0,001	3	0,000
Brote	638	0,016	95	0,0084
Epidemia	735	0,021	1057	0,102
Suspender	68	0,002	26	0,003
Tarjeta	6054	0,150	278	0,027
Sospechosa	435	0,011	89	0,009
Financiera	657	0,016	56	0,005
Actividad	767	0,021	99	0,008
Vulnerable	57	0,005	36	0,004
Pandemia	965	0,036	209	0,011
Vacuna	1260	0,053	509	0,015
Covid-19	25750	0,597	7576	0,191

el 80 % de los datos en el mundo reside en un formato no estructurado, la minería de texto es una práctica extremadamente valiosa dentro de este tipo de procesos.
Con los resultados obtenidos, se observó la influencia del uso de estas palabras en los correos falsos y verídicos. A continuación, se visualiza una gráfica de las frecuencias obtenidas.

Figura 2. Número de términos usados en los emails con Phishing y sin Phishing

3.3 Verificación la calidad de los datos

Luego de realizar el análisis de datos, se concluye que los Datasets obtenidos son adecuados, dado que nos brindan la información requerida para la extracción de características, las mismas que contribuyen para la modelación eficiente, pretendiendo obtener una buena precisión. Con respecto a los correos electrónicos, provienen de una fuente fidedigna, hay que notar que los registros están actualizados, por lo que se puede asegurar que los resultados son un buen punto de referencia para nuevos estudios.

El conjunto de datos que contiene correos no infectados, ayuda a balancear los datos y con ello obtener un mejor resultado en la predicción del modelo de aprendizaje, dando un mejor pronóstico en la identificación de correos electrónicos infectados, debido a que nos brinda información para determinar las características que ayudan a determinar entre un correo infectado y otro no.

PhishTank es una fuente confiable mencionada en varios artículos referentes a este tema en especial Orunsolu et al. (2019), la información proporcionada nos ayuda en demasía en el análisis de URL_s y de dominios. Los registros recopilados, brindan datos muy relevantes y precisos, lo que garantiza la ausencia de riesgo de tener un porcentaje alto de ruido en el procesamiento, y que los resultados serán veraces.

Preparar los datos para ajustarlos de forma adecuada para el proceso de modelamiento, es algo principal y frecuentemente conlleva más tiempo, debido a que es necesario agrupar y elegir de manera correcta las variables que van a ser procesadas.

4. Construcción del vector de características

Para el desarrollo de esta parte del proyecto, se seleccionó el lenguaje de programación Java 11, dado que se tiene datos no estructurados en formato M.box y se procesan para determinar las características para la detección de Phishing, para lo cual se utiliza librerías para parsear el HTML y extraer los datos necesarios. Con esto se exporta la base de mensajes completa del formato JSON a CSV.

4.1 Selección de características

Como se expuso al inicio de esta sección para el diseño del modelo, hay que seleccionar las características apropiadas, basándonos en estudios referentes a la temática y el análisis que se realizó en el capítulo anterior. Además, se utiliza una biblioteca Java llamada Secure Socket Extension para extraer información de terceros relacionada con un dominio en particular durante el proceso de extracción de características. Esto proporciona una forma eficaz de examinar todas las características y etiquetas relevantes de la página analizada para examinar su estado.

Para las funciones de URL, se extraen 7 funciones. En las características del documento web, se extraen otras 17 características, que se extraen para mejorar la detección de phishing en los diversos corpus del email. Aunque algunas otras características todavía están disponibles, se elige especialmente esas características porque las que se omiten se puede deducir de las elegidas (por ejemplo, el número de puntos en la mayoría de las URL) de phishing está asociado con nombres de dominio alargados.

Se designa un nombre a las características extraídas. La fase del clasificador de aprendizaje automático utilizó las características elegidas para entrenar los algoritmos (RF y NB) de aprendizaje automático y seguir con el proceso de validación.

4.2 Identificación del ataque Phishing en emails

Para el entendimiento sobre los patrones que tiene un correo electrónico con presencia de Phishing o a su vez un correo legítimo, se realizó una revisión exhaustiva de documentación relacionada con el tema de estudio, con esto se procede a realizar una comparación de palabras entre correos electrónicos benignos (reales) e infectados mediante la técnica de text mining.

Este proceso permitió la correcta caracterización de las palabras que predominan en email con legítimo o con Phishing. La formación de características, basadas en los términos textuales son unidas y esquematizadas de tal forma que cada palabra tiene homogeneidad entre ellas y así formar un grupo de estas. Se debe mencionar que esta etapa es una parte primordial para la construcción del Dataset que posteriormente se usará para el modelamiento y la validación. Se conformaron 9 grupos de palabras para este proceso.

La variable dependiente y representa la identificación de correos mediante ciertas características, por tanto, será una variable binaria que toma los valores de 1 para los correos con etiqueta de Phishing y 0 para los correos con etiqueta de benignos: de modo que la etiqueta y es una clase binaria representada como:

$$y = \begin{cases} 1, & \text{si es phishing,} \\ 0, & \text{si es benigna.} \end{cases}$$

5. Construcción del modelo Predictivo

En esta sección, se describe el modelo utilizado para el cumplimiento del objetivo planteado, por lo que se utilizará las métricas que indicarán el porcentaje de precisión de los algoritmos de clasificación, en la siguiente subsección, se evalúa estos modelos y se observará si se cumple con los criterios establecidos para los datos de prueba y locales.

5.1 Implementación del Algoritmo Predictivo.

Haciendo uso del software Python explicado a profundidad en una sección anterior, en esta implementación se utilizarán las siguientes librerías para la construcción del Algoritmo Predictivo: *sklearn.metrics*, *matplotlib.pyplot* y *numpy*.
A continuación, se muestran los detalles y observaciones del algoritmo.

En el software Python, se carga el paquete que contiene librerías, de NB, Árboles de decisión y RF son los que se utilizarán para la modelización de esta problemática que tiene como punto fundamental predecir si un correo tiene presencia o no de este ataque.

En la elaboración del modelo, se toma como entrada el documento que contiene la matriz de 0_s y 1_s, estos datos se dividirán mediante un muestreo aleatorio simple, en dos submuestras aleatorias, una para el desarrollo del modelo (train) y la otra para su validación (testeo).

La muestra de modelamiento corresponde aproximadamente al 80 % de la muestra original. En cambio, la muestra de validación corresponde aproximadamente al 20 % de la muestra original, siguiendo el principio de Pareto. Teniendo como objetivo principal detectar la presencia de phishing, las entradas que se consideran para los algoritmos de clasificación son las que se muestran a continuación.

- train_inputs: Variable que abarca el 80 % de los datos para la caracterización de un email con presencia o no de Phishing.
- test_inputs: Variable que abarca el 80 % de los datos para la caracterización de un email con presencia o no de Phishing
- train_outputs: Variable que abarca el 20 % de los datos para la caracterización de un email con presencia o no de Phishing.
- test_outputs: Variable que abarca el 20 % de los datos para la caracterización de un email con presencia o no de Phishing.

En el problema de la detección de phishing, el uso de RF y NB es común en el caso de tener vectores de características con diferentes volúmenes de dimensionalidad de datos (Moghimi and Varjani , 2016). En la literatura, el análisis de frecuencia de diferentes clasificadores indica una alta adopción de estos dos clasificadores, especialmente en la definición de problemas de phishing debido a su simplicidad y alta precisión(Anwar et al., 2017; Dhanalakshmi and Chellappan, 2013). Motivado por las investigaciones anteriores de RF y NB sobre conjuntos de datos de phishing, nuestro método de clasificación emplea estos dos clasificadores en el mismo conjunto de funciones para evaluar su rendimiento.

5.2 Modelo Random Forests

Como otros clasificadores, los clasificadores de bosque deben estar equipados con dos matrices: una matriz X de forma dispersa o densa que contiene las muestras de entrenamiento, y una matriz Y de forma que contiene los valores objetivo, para este modelo se utilizará la librería RandomForestClassifiery, donde cada árbol en el conjunto se crea a partir de una proporción extraída con reemplazo del conjunto de entrenamiento. Sin embargo, RandomForestRegressor usa un número predeterminado de árboles de 100, que normalmente no es suficiente. En consecuencia, esta se subió hasta 1.000 árboles en primera instancia para posteriormente dejarlo en 3.000 árboles. La profundidad predeterminada de cada árbol (max_depth) es 5, lo que significa que se ensambla árboles con profundidad máxima de 5.

Para determinar los parámetros óptimos, primero se ha ejecutado la forma predeterminada, es decir sin cambiar lo que por defecto está determinado para observar los resultados que se obtienen, posteriormente se va jugando con los parámetros para encontrar los que hacen mínima la tasa de mal clasificados.

El propósito de usar RF es que se consigue una variación pequeña al combinar varios árboles, ocasionalmente a costa de un incremento pequeño en el sesgo. En la práctica, la reducción de la varianza es a menudo significativa, por lo que se obtiene un mejor modelo.

5.3 Modelo Naive Bayes

El módulo sklearn. ensemble tiene la librería MultinomialNB que se usará para la implementación del algoritmo de Bayes para datos distribuidos multinominalmente, siendo estas las dos variaciones clásicas de NB, para el uso en la clasificación de texto en el que los datos son recuentos de vectores de palabras normalmente, aunque también se sabe que los vectores tf-idf funcionan bien. La distribución está parametrizada por vectores $\theta_y = (\theta_{y_1}, ..., \theta_{y_n})$ para cada clase y, donde n es el número de características (en la clasificación del texto, el tamaño de la palabra) y θ_{y_i} es la probabilidad $P(x_i \mid y)$ de característica i apareciendo en una porción perteneciente a la clase y.

Para esta implementación no fue necesario utilizar (partial_fit) dado que el conjunto de entrenamiento completo no presentó ningún inconveniente en la memoria.

De forma general, el tiempo de ejecución de este modelamiento depende de los datos que se utiliza por ejemplo, para el conjunto de datos descargables en la construcción de características tuvo una hora y media de procesamiento mientras que para el entrenamiento se demoró alrededor de una hora, ahora bien, para los datos locales (cuentas personales) el tiempo varió sustancialmente, dado que era necesario construir un programa en JAVA 11 para indexar los mensajes de las cuentas, por tal razón, se requirió de cuatro horas para completar el proceso. Hay que mencionar que este periodo de tiempo varía dependiendo del equipo que se utilice en la modelización.

5.4 Evaluación el modelo

En la fase de evaluación, se compara el rendimiento del sistema propuesto a través de un experimento de validación cruzada de 10 veces antes del proceso de evaluación en el conjunto de datos de prueba. Esto implica la división aleatoria del conjunto de datos de prueba en diez submuestras iguales, de las cuales una sola submuestra se usa para la validación final del modelo, mientras que las otras submuestras son usadas para el entrenamiento del sistema. Por tanto, el modelo predictivo propuesto se basó en el 80 % del conjunto de datos y se validó en el 20 % restante.

Las razones para usar la validación cruzada en este modelo son para:

i. i. Verificar el comportamiento del error del modelo predictivo, en este caso, los errores asociados con el modelo predictivo de RF y NB en la detección de phishing.

ii. Validar el Dataset de entrenamiento mediante la validación de cada subconjunto. Esto es para tener un nivel alto de confianza en el modelo entrenado.

En el experimento, se usará tanto conjuntos de datos de prueba como los locales, estas datas contienen sitios web legítimos y de phishing no superpuestos que se procesara previamente para posteriormente obtener un archivo CVS, que contiene el vector de características finales para procesar en el modelo predictivo. A continuación, se explica cómo se recolectaron los datos locales en forma resumida sin dejar de lado lo primordial.

Dado que el enfoque de esta investigación es a nivel local del país, los dataset disponibles en la red pública son demasiado antiguos o en otro idioma, por tal motivo se opta por utilizar los correos de 3 cuentas, una cuenta de Gmail personal, Outlook personal y una institución educativa nacional.

- Se implementa un programa en JAVA 11 que permite ejecutar un cron que se encargará de indexar todos los emails de las cuentas mencionadas para obtener tanto correos categorizados como verdaderos, así como de la carpeta de spam mediante la lectura directa del buzón de mensajes con el uso del protocolo POP3.

- Esos registros se guardan en un Dataset no estructurado (mongodb) en formato JSON y se procesan para determinar las características necesarias para la ejecución, para lo cual se utiliza librerías como JSOUP para parsear el HTML y extraer los datos necesarios.

Con esto se exporta la base de mensajes completa del formato JSON a CSV, las librerías que se utilizaron para este proceso están descritas en la sección “Construcción de características"; En el Servicio "MensajeSrv"se encuentra la lógica descrita en Indexado de mensajes.

Posteriormente, se realiza el análisis respectivo, para esto se procede a dividir el proceso en tres etapas, la primera el entrenado del modelo, como siguiente fase se tiene la predicción, y para finalizar los resultados conseguidos al ejecutar el modelo en un entorno moderado para la detección de phishing en los emails en

las cuentas personales antes mencionadas.

A continuación, se detalla los resultados y el análisis del modelo.

Etapa uno: (Entrenado del modelo) Para esta etapa se tiene alrededor de 7043 correos para el entrenamiento del modelo, teniendo 3620 con presencia de phishing y 3423 correos reales.

Nota: Estos datos representan el 80 % del dataset obtenido.

Hay que mencionar, antes de seguir con la segunda etapa, que en la data de testeo se tiene 1761 emails teniendo 920 emails con phishing y 841 emails sin presencia de phishing, estos datos se utilizarán para la predicción, teniendo un total de 8804 correos en la base original.

Segunda etapa: (Predicción del modelo) Para la fase de predicción, se trabajará con 1629 emails para esta etapa, se hará uso de los modelos de clasificación RF y NB. El rendimiento del sistema propuesto se evalúa mediante el uso de cinco parámetros estándar que consisten en Accuracy, Precisión, Recall_score y Puntación F1. Estas son las métricas de rendimiento estándar para evaluar cualquier sistema de detección de phishing para la evaluación de los resultados.

Adicionalmente, el coeficiente de correlación de Mathew (MCC) nos permite determinar el poder predictivo del modelo de aprendizaje automático en el experimento de validación cruzada.
Estos consisten en:

- Verdadero Positivo (TP): que señala el número de correos identificados como phishing, siendo estos phishing.
- Verdadero Negativo (TN): son los señalados como benigno, siendo estos phishing.
- Falsos Positivos (FP): son los señalados como phishing, siendo estos benignos/reales.
- Falsos Negativos (FN): son señalados como benignos, siendo estos benignos.

Para el cálculo del MCC y determinar la calidad del modelo de predicción, al aproximarse MCC a la unidad, indica que el sistema tiene una predicción casi perfecta y, por lo tanto, es un sistema de detección confiable. La siguiente ecuación representa el MCC.

Donde:

TPRxTNR-FPRxFNR $= Factor_1$
y
(TPR+FPR)(TPR+FNR)(TNR+FPR)(TNR+FNR) $= Factor_2$

$$MCC = \frac{Factor\_1}{\sqrt{Factor\_2}}$$

En el siguiente capitulo, se presentan los resultados para los dos conjuntos de datos, adicionalmente se presenta la comparación con otros trabajos que siguen la misma línea de investigación y de esta manera observar el nivel de predictibilidad del modelo.

6. Resultados y discusión

El rendimiento del sistema propuesto se evalúa utilizando cinco parámetros estándar y la validación cruzada, esto se menciona en el anterior capítulo. Para el proceso de VC se repitió 10 veces y después de la validación, se calcula una sola estimación. Esta estimación es el promedio de las diez iteraciones.

El incentivo para aplicar el experimento de validación cruzada es ajustar el rendimiento de un modelo fuera del conjunto de entrenamiento, a continuación, se analizarán los resultados de los dos conjuntos de datos.

6.1 Resultado para los datos descargables

Tabla 3. Tabla de resultados para el conjunto de datos Locales (Cuentas personales: Gmail, Outlook e Institucional)

Métricas \Modelo	RF	NB	Trees
Accuracy	97,70 %	92,53 %	94,50 %
Precisión	97,24 %	90,57 %	94,30 %
Recall_score	98,55 %	93,06 %	95,70 %
Puntuación F1	97,54 %	92,06 %	95,92 %
Roc_auc	97,50 %	93,50 %	94,60 %
Coef. Mathew	0,971	0,923	0,959

Se procede a interpretar los datos de las métricas obtenidas, se visualiza que el mejor modelo es Random Forests a comparación de NB como un plus se calculó para árboles de decisión, teniendo como resultado que RF presenta el que mejor nivel de predicción para este investigación, ahora bien se analiza las métricas obtenidas, teniendo la exactitud que representa el porcentaje en el cual el modelo ha acertado, obteniendo un valor de 97,70 % de exactitud, el valor obtenido para la precisión es de un 97,24 %. Dado que se tiene un conjunto relativamente balanceado se puede decir que la métrica Accuracy proporciona un resultado confiable para nuestro trabajo.

En el caso de la métrica de Recall es la capacidad del modelo para detectar los casos significativos. En nuestro caso, se tiene 98,55 % que es claramente un valor muy bueno para una métrica. Se puede afirmar que nuestro algoritmo de clasificación es muy sensible.

Para la Puntuación F1, se observa un porcentaje de 97,54 % dado que nos esquematiza la precisión y sensibilidad en una sola métrica, dándonos una gran ayuda cuando la distribución es desigual, así al tener una alta precisión y alto recall, implica que, el modelo elegido maneja perfectamente esa clase.

En el caso del coeficiente de correlación de Mathew, se tiene 0,971 con una buena calidad para el modelo de predicción RF.

En la Figura 3, se presenta la curva ROC para los datos de testeo, se procede a interpretar. Dado que el valor es cercano a uno, se puede decir que el rendimiento del modelo es bastante bueno. Así, se ha encontrado un clasificador con un rendimiento muy bueno sin tener el riesgo de sobreajuste en los datos de las bases: PhishTank, Monkey y Enron.

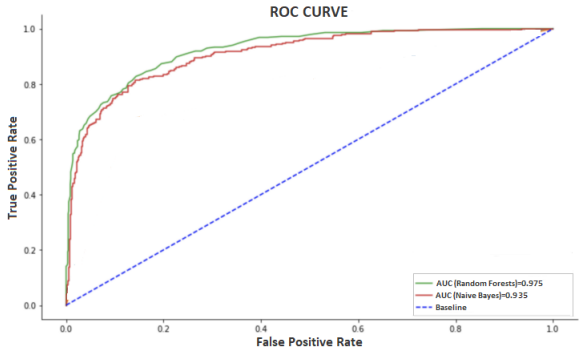

Figura 3. Curva Roc y AUC de datos descargables

Ahora se procede a calcular el valor real del error y el porcentaje de error que tiene el modelo. La ecuación 4 determina el valor real (ERR) y en la ecuación 5, la tasa (TERR), es en otras palabras, el porcentaje que el modelo no etiquetó los correos correctamente.

$$ERR = \frac{\text{FN+FP}}{\text{TN+FN+FP+TP}}, \tag{4}$$

donde, la tasa de error (TERR) se calcula de la siguiente manera:

$$TERR = \frac{\text{FN+FP}}{\text{TN+FN+FP+TP}} \cdot 100, \tag{5}$$

$$= 2,3\,\%.$$

Finalmente, de los resultados que se obtuvieron, se observa que, de los 6217 correos analizados, el 97,7 % que corresponde a 6074 emails fueron clasificados correctamente, por el contrario, el 2,3 % correspondiente a 143 correos fueron etiquetados de forma incorrecta. Por consiguiente, se puede decir que el modelo presenta una predictibilidad alta, teniendo como consecuencia resultados fiables en la clasificación de phishing.

6.2 Resultado para los datos Locales

En esta sección, se analiza los resultados para los datos extraídos de las tres cuentas personales. A continuación, se presenta la Tabla 4 que son las métricas obtenidas en el modelo con este conjunto de datos.

Tabla 4. Tabla de resultados para el conjunto de datos Locales (Cuentas personales: Gmail, Outlook e Institucional)

Métricas\Modelo	RF	NB	Trees
Accuracy	95,40 %	89,35 %	91,50 %
Precisión	92,14 %	85,98 %	89,03 %
Recall_score	96,55 %	89,62 %	92,70 %
Puntuación F1	94,74 %	87,08 %	91,22 %
Roc_auc	94,90 %	90,20 %	91,26 %
Coef. Mathew	0,967	0,932	0,954

De la misma forma que se hizo en la sección anterior para los registros descargables, se procede a interpretar los datos de la Tabla 4 obtenidos, se puede visualizar que el mejor modelo es RF al igual que en el caso anterior, ahora bien, se analiza las métricas obtenidas, teniendo la exactitud que representa el porcentaje de predicciones correctas frente al total por tanto se tiene 95,40 % de exactitud para este modelo, el valor obtenido para la precisión es de un 92,14 %. Por tanto, nuestro modelo es más preciso que exacto, coincidiendo con el experimento anterior.

En el caso de la métrica de Recall (Sensibilidad) es la habilidad del modelo para detectar los casos relevantes. En nuestro caso, se tiene 96,55 % es claramente un valor bueno para una métrica. Se puede decir que nuestro algoritmo de clasificación es sensible.

Para la Puntuación F1, se observa un porcentaje de 94,74 %, dado que nos esquematiza la precisión y sensibilidad en una sola métrica, dándonos una gran ayuda cuando la distribución es desigual, así al tener una alta precisión y alto recall, implica que, el modelo elegido maneja perfectamente esa clase.

En el caso del coeficiente de correlación de Mathew, se tiene 0,967 obteniendo una buena calidad para el modelo de predicción RF. De la misma manera que se visualizó en los datos experimentales, en la Figura 4 se presenta la curva ROC para los datos de testeo, se procede a interpretar. Dado que el valor es cercano a uno, se puede decir que el rendimiento del modelo es bastante bueno. Así, se encontró un clasificador con un rendimiento muy bueno sin tener el riesgo de sobreajuste en los datos locales.

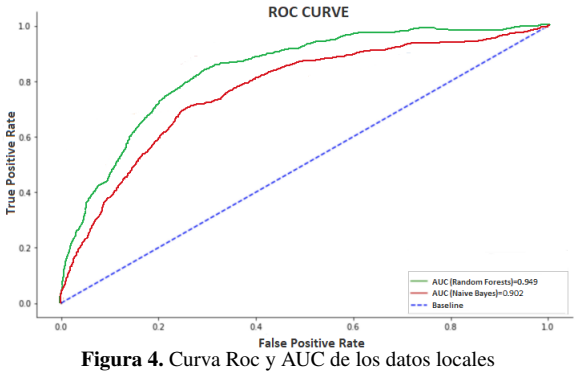

Figura 4. Curva Roc y AUC de los datos locales

Al igual que la sección anterior, se determina el porcentaje de error presente en el modelo, y esto nos proporciona una validez buena para el análisis de resultados.

Se procede a calcular el valor real del error y el porcentaje de error que tiene el modelo. La ecuación 4 determina el valor real (ERR) y en la ecuación 5 la tasa (TERR), es en otras palabras, el porcentaje que el modelo no etiquetó los correos de correctamente.

$$ERRL = \frac{\text{FN+FP}}{\text{TN+FN+FP+TP}}, \tag{6}$$

donde, la tasa de error (TERRL) se calcula de la siguiente manera:

$$TERRL = \frac{\text{FN+FP}}{\text{TN+FN+FP+TP}} \cdot 100, \tag{7}$$

$$TERRL = 4,6\,\%.$$

De los resultados alcanzados, se observa que, de 1.761 correos analizados, el 95,4 % que corresponde a 1680 correos se clasificó de forma correcta, mientras que el 4.6 % correspondiente a 81 correos fueron clasificados de forma errada. Por tal motivo, se determina que el modelo tiene un porcentaje de predicción relativamente alto, dando como consecuencia resultados fiables en la clasificación de phishing.

7. Comparación de resultados obtenidos

Resumiendo, de los resultados obtenidos en los dos conjuntos de datos se observa que, el modelo construido logró un 97.7 % en la detección de phishing en URL_s y en los correos electrónicos para los datos experimentales como son: PhishTank, Monkey y Enron. Teniendo como observación principal que al variar el conjunto de características se puede obtener un mayor o menor porcentaje de predicción, pero esto se debe a la actualización de técnicas de ataque en este tipo de delito, el vector de características debe actualizándose.

Obteniendo que la aplicación de un modelo ayuda a la detección de correos electrónicos con presencia de phishing. Pese a esto también se debe mencionar que la posibilidad de tener el 2,3 % de error al momento de la clasificación, aunque es pequeño no deja de ser un dato que hay que tomar en cuenta para futuros trabajos.

Para los resultados obtenidos en el conjunto de datos locales se visualiza que, en diversas métricas aplicadas, el modelo creado logró un 95.4 % en la determinación de existencia de phishing en URL_s y en los correos electrónicos. Los porcentajes tienen una gran diferencia entre los datos experimentales y los locales, dado que los datos que se tiene en nuestro país no son los mejores o como se menciona anteriormente la detección de delitos cibernéticos en Ecuador está comenzando, es por esta razón que los datos no presentan ciertas características para obtener un porcentaje mayor.

La tasa de error es de 4,6 % dándonos como indicativo que, de 1000 correos analizados, 46 de ellos no se clasificarán de forma adecuada dando un margen de perdida de información o ser víctimas de phishing.

En la siguiente subsección, se mostrarán los resultados de trabajos anteriores, al aplicar técnicas de ML para la detección de este tipo de delito.

7.1 Resultados para el sistema y los trabajos anteriores

El enfoque propuesto en este trabajo de investigación aborda las limitaciones de otras técnicas anti-phishing en términos mediciones de parámetros como la eficiencia computacional y la robustez. En esta sección, la evaluación del desempeño en parámetros de evaluación se compara con otras técnicas existentes. Esta comparación se basa en los trabajos relacionados a técnicas anti-phishing. La Tabla 5 presenta las estadísticas de rendimiento dando a notar que el método propuesto es uno de los mejores modelos anti-phishing existente en los anteriores trabajos.

Tabla 5. Comparación de trabajos relacionados con el método propuesto

Trabajo	Datos phish	Datos benig	Total	TPR	FPR	Exact
Sonowal y Kup	667	995	1.662	90,54	5,82	92,72
Kaur y Kalra	1.078	846	1.924	99,44	0,56	97,51
Chin	3.718	1.185	4.903	96,9	0,03	97,39
Tan y col.	500	500	1.000	99,2	7,8	91,41
El método propuesto	15.964	15.120	31.084	98,7	0,79	97,7

7.2 Validación Estadística

La validación, que se plantea desde el punto de vista estadístico tiene el objetivo de verificar qué modelo de clasificación es el adecuado para el APS teniendo como base el tiempo de ejecución para el desarrollo del mismo.

Las hipótesis contrastadas en un ANOVA para el tiempo de ejecución de los modelos utilizados en este trabajo son:

- H_0: No hay diferencias entre las medias de los tiempos de ejecución de los modelos: $\mu_1 = \mu_2 = \mu$.
- H_1: Las medias son significativamente distintas la una de la otra.

Se tiene 2 grupos y la cantidad de observaciones por grupo es de 10, por lo tanto, se tiene un modelo equilibrado. A continuación, se calcula las medias y las desviaciones típicas de cada grupo.

La media para el primer grupo es de (μ_1): 90 y la desviación estándar (σ_1): 0,83066238629146, para el grupo dos la media (μ_2): 94 y la desviación estándar (σ_2): 1,4494897427832 Por procesos anteriores, de esta investigación se puede decir que existen datos atípicos o diferencia de varianzas. En este caso, los 2 grupos parecen seguir una distribución simétrica.

Se continua con la verificación de las condiciones para un ANOVA, dentro de cada grupo los datos son independientes entre ellos ya que se ha hecho un probado aleatorio. La variable cuantitativa se distribuye de forma normal en cada uno de los grupos, para este estudio de normalidad se realizó mediante la forma gráfica y con el Test de normalidad *Shapiro-Wilk*, dado que el número de observaciones es menor a 50, se tiene para el grupo 1 el valor de W: 0,987166 y un p_val: 0,905192 mientras, que para el grupo 2 el valor de W: 0.970674 y un p_val: 0.706497.

Por lo tanto, no se tiene evidencia de falta de normalidad, para el siguiente paso se procede con la prueba ANOVA obteniendo como resultado el valor del estadístico de prueba, F=1,994349. Es significativamente distinto de 1 para cualquier nivel de significación y por tal razón, se rechaza la hipótesis nula de igualdad de medias. Además, el valor de eta cuadrado (η^2) es de 0.018, lo indica un tamaño de efecto pequeño.

Por lo tanto, las medias son significativamente distintas la una de la otra y por ende se llegaría a concluir que el modelo con el mejor tiempo de ejecución para esta investigación es el que presenta una media de 90 minutos siendo este RF, Ahora se puede aseverar que tanto con el test ANOVA y mediante la matriz de confusión para obtener las métricas RF es el modelo de clasificación que presenta mejor estabilidad para este tipo de desarrollo.

7.3 Discusión

Los esfuerzos que deben seguirse para la detección de phishing deben ser aún mayor, dado la creciente ola de tecnología que no nos deja otra opción de ser precavidos con la información que se divulga en la web, por esta razón se examinó: las limitaciones, el éxito y el impacto de esta investigación a nivel práctico y

teórico. En un enfoque práctico, las personas que revisen este artículo pueden tener una idea del cómo se construyó el vector de características y se aplicó en el modelo, para posteriormente mejorar su resiliencia en la seguridad cibernética al tratar de explicar cómo funciona este delito y prepararse para evitar caer en el phishing.

Desde el punto de vista teórico, esta investigación combina dos mundos el práctico y académico donde los informes de inteligencia de amenazas cibernéticas y los trabajos de investigación académica se utilizaron para aprender y desarrollar la comprensión de las capacidades, tácticas, técnicas y procedimientos de los atacantes.

Las limitaciones de esta investigación están relacionadas con los datos que se tienen para la ejecución de este modelo, dado que son registros que tienen una alta presencia de *NaN* y esto no permite tener los resultados que se tenía en mente. Además, hay que recalcar que los datos utilizados para la evaluación son de 3 cuentas personales, dando como resultado un conjunto de muestra pequeña para el entrenamiento de los modelos RF Y NB y se obtuvo un desempeño menor al que se obtuvo con los datos de experimentales. Sin embargo, durante este trabajo se procuró analizar las características que se están presentando actualmente en los ataques de phishing.

El éxito de esta investigación se basa principalmente en el análisis general de técnicas, tácticas y procedimientos de atacantes modernos que son comúnmente utilizados por los actores de amenazas del mundo real que se obtuvieron a través de la revisión de la literatura. Esto proporcionó más información sobre las acciones específicas y qué capacidades técnicas existen para eludir los controles de seguridad modernos, como el uso del candado de seguridad *Https* para el robo de datos personales. Aunque esta investigación analiza diferentes modelos para visualizar el que nos otorga un mejor nivel de predictibilidad, exactitud y precisión y así dar una pauta a la hora de diseñar e implementar controles de seguridad de compensación, detecciones y procedimientos de respuesta para proteger los datos críticos.

Los resultados y el contenido de este trabajo podrían ser utilizados por estudiantes que deseen aprender más sobre este delito cibernético; las técnicas, tácticas y procedimientos que usan comúnmente los actores de amenazas.

7.4 Abreviaturas y Siglas

APS	Anti-Phishing System
CRISP-DM	Cross Industry Standard Process for Data Mining
FSM	Módulo de selección de características
TF-IDF	Término-Frecuencia Documento Frecuencia Inversa
ROC	Receiver Performance Characteristic
URL	Uniform Resource Locator
TP	Verdadero Positivo
TN	Verdadero Negativo
FP	Falsos Positivos
FN	Falsos Negativos

8. Conclusiones y Recomendaciones

8.1 Conclusiones

(a) Según el análisis del material recopilado para esta investigación, los actores de amenazas modernos se basan más comúnmente en técnicas de phishing para obtener acceso inicial. Algunos de los medios más comunes para obtener este acceso parecen ser la implementación de malware en un documento de Office, que luego se ejecuta una vez que el usuario habilita las macros. Otro método que los atacantes parecen estar usando constantemente es el abuso de vulnerabilidades conocidas públicamente, ya que es mucho más rentable que descubrir vulnerabilidades previamente desconocidas.

(b) En este trabajo, se considera el problema de la detección de phishing utilizando enfoques de aprendizaje automático. En nuestro primer intento, llamado conjunto de prueba, se identifican varias características interesantes al analizar el problema desde el punto de vista estadístico. Se pudo visualizar que ver el problema desde un punto puramente, es insuficiente para resolverlo de manera efectiva. La intención del atacante de phishing también debe tenerse en cuenta para una solución eficaz.

(c) Describir un enfoque hacia el diseño de Funciones basadas en nombres de dominio, análisis tanto de URL como el corpus del email para la detección de phishing mediante aprendizaje automático. Nuestro diseño de funciones hizo hincapié en la eliminación del posible sesgo en la clasificación debido a conjuntos de datos de phishing y páginas legítimas elegidas. Nuestro enfoque difiere de otros trabajos en este espacio, ya que explora la relación del corpus del email con su intención de phishing. Con un conjunto de características balanceado, se obtuvo una tasa de clasificación de 97 % con datos de validación cruzada. Además, se mostró una tasa de detección de 97-97.7 % para URL activas en la lista negra.

(d) Nuestros tiempos de extracción y clasificación de características son muy bajos y muestran que nuestro enfoque es adecuado para la implementación en tiempo real. Es probable que nuestro enfoque sea muy eficaz en las estrategias de phishing modernas, como el phishing extremo, que están diseñadas para engañar incluso a los usuarios experimentados.

8.2 Recomendaciones

Las recomendaciones que se presentan en la sección final son de ayuda para trabajos futuros, teniendo en cuenta que esta investigación es un pequeño aporte a este campo tan extenso y relativamente nuevo.

(a) Los phishers buscan vulnerabilidades constantemente para atacar y a su vez no ser detectados al momento de sus ataques, por tal razón se recomienda ejecutar un análisis constante de las técnicas empleadas por estos atacantes, y así determinar características nuevas o que ayuden a mejorar el vector final y por ende obtener un modelo preciso con un nivel de predictibilidad más elevado.

(b) Se recomienda utilizar Datasets actuales para tener una mejor visión de este tipo de delitos cibernéticos, pero este punto es primordial dado que el tiempo de vida de estos dominios y URL_s es muy corto, por tal razón el conjunto de datos debe ser actualizado al menos cada 6 meses. Así se obtendrán resultados confiables y actuales.

(c) Revisar el trabajo de Orunsolu, Sodiya y Akinwale, en Orunsolu et al. (2019), dado que tiene una buena revisión de literatura para este tema de investigación.

REFERENCIAS

Aburrous, M., Hossain, M., Dahal, K. and Thabtah, F. (2010). Experimental case studies for investigating e-banking phishing techniques and attack strategies *Cognit. Comput.* (2), 242–253 https://doi.org/10.1007/s12559-010-9042-7

Adebowale, M., Lwin, K., Sanchez, E. and Hossain, M. (2018). Intelligent Web-Phishing Detection and Protection Scheme using integrated Features of Images, Frames and Text. *Expert System with Applications.* (115), 300-313 https://doi.org/10.1016/j.eswa.2018.07.067

Amat Rodrigo, Joaquín. (2020). Análisis de texto (text mining) con Python, cienciadedatos.net. Obtenido de: https://www.cienciadedatos.net/. (Diciembre, 2020).

Anwar, T., Abu-Kresha, M. and Bakry A. (2017). An efficient method for web page classification based on text. *International J. Eng. Comput. Sci.*

Barraclough, P. & Sexton, G. (2015). Phishing website detection fuzzy system modelling, *IEEE, London, UK*, 1384-1386, 10.1109/SAI.2015.7237323.

Breiman, L. (2001). Random Forests. Machine Learning *SpringerLink, 45*, 5–32, https://doi.org/10.1023/A:1010933404324

Calva Yaguana, Karen Priscilla. (2020). *Modelo de predicción del rendimiento académico para el curso de nivelación de la Escuela Politécnica Nacional a partir de un modelo de aprendizaje supervisado automatizado en R.* [Tesis de pregrado, Escuela Politécnica Nacional]. Repositorio institucional de la Escuela Politécnica Nacional. https://bibdigital.epn.edu.ec/

Chin, T., Xiong, K. and Hu, C. (2015). PhishLimiter: A Phishing Detection and Mitigation Approach using Software-Defined Networking, *IEEE Access, 6*, 42516-42531, 10.1109/ACCESS.2018.2837889

Cortina, V. G. (2015). *Aplicación de la metodología CRISP-DM a un proyecto de minería de datos en el entorno universitario*. [Universidad Carlos III de Madrid]. Departamento de Informática.

Creswell. (2015). Educational research. Planning, conducting and evaluating quantitative and qualitative research. *USA*.

Dhanalakshmi, R. & Chellappan, C. (2013). Detecting Malicious URLs in E-mails- An Implementation. *AASRI Procedia, 4*, 125-131, https://doi.org/10.1016/j.aasri.2013.10.020

Gansterer, W.N. & Polz, D. (2009). E-mail classification for phishing defense, in Advances in Information Retrieval. *Heidelberg: Springer Berlin Heidelberg*, 449–460.

Gironés Roig, J., Casas Roma, J., Minguillón Alfonso, J., Caihuelas Quiles, R. (2020). *Minería de datos Modelos y algoritmos*, Editorial UOC.

Gowtham, R., Gupta, J. and Gamya, P.G. (2017). Identification of phishing web pages and their target domains by analyzing the feign relationship *J. Informat. Secur. Appl, 35*, 75-84

Gowtham,R. & Krishnamurthi, I. (2014). PhishTackle-a web services architecture for anti-phishing *Cluster Compt, 17*, 1051–1068. https://doi.org/10.1007/s10586-013-0320-5

Gupta, B.B., Tewari, A., Jain, A.K. and Agrawal, P. (2017). Fighting against phishing attacks: state of the art and future challenges. *Neural Comput. Applic, 28*, 3629–3654 https://doi.org/10.1007/s00521-016-2275-y

Hastie, T., Tibshirani, R. and Friedman, J. (2017). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.

Hota, H.S., Shrivas, A.K. and Hota, R. (2018). An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique *Procedia Comput. Sci, 132*, 900-907, 10.1016/j.procs.2018.05.103

Isa, D., Lee, L., Kallimani, V. and Rajkumar, R. (2016). Text document pre-processing using bayes formula for classification based on the vector space model Comput. *Informat. Sci. J, 1* (4), 79-90.

Jain, AK & Gupta, BB. (2016). A novel approach to protect against phishing attacks at client side using auto-updated. *EURASIP Journal on Information Security*(1), 1-11.

Kittler, J., Hatef, M. and Duin, R.P.W. (1998). On Combining Classifiers. Transactions on pattern analysis and machine intelligence. *IEEE, 20*.

Khonji, M., Iraqi, Y. and Jones, A. (2013). Phishing Detection: A Literature Survey. *IEEE Communications Surveys & Tutorials, 15*(4), 2091-2121.

Kuncheva, L. (2004). Combining Pattern Classifiers. Methods and algorithms. *John Wiley & Sons, New Jersey*.

Martínez, M. B. (2018). Minería de Datos. web: http://bbeltran.cs.buap.mx/NotasMD.pdf.

Moghimi, M. & Varjani, A.Y. (2016). New rule-based phishing detection method *Expert systems with applications, 53*, 231-242.

Monkey.org. (2020). Data de correos electrónicos Monkey.org. Web de Monkey.org: https://monkey.org/ jose/phishing/; (2020)

CSO Online report on phishing activities. Accessed 2016 http://www.csoonline.com/articles

Orunsolu, A.A., Afolabi, O., Sodiya, A.S. and Akinwale, A.T. (2019). A Users' Awareness Study and Influence of Socio-Demography Perception of Anti-Phishing Security Tips. Acta Informatica Pragensia, 7(2), 138-151.

Pedregosa. (2011). Scikit-learn: Machine Learning in Python *JMLR 12*, 2825-2830

Phishtank dataset. (2021). http://www.phishtank.com. (2021).

Qabajeh, I., Thabtah, F. and Chiclana, F. (2018). A recent review of conventional vs. automated cybersecurity anti-phishing techniques *Computer Science Review, 29*, 44-55.

Rosero Gomezcoello, Johanna Mishell. (2020). *Detección y mitigación de ataques de ingeniería social tipo Phishing utilizando minería de datos* [Tesis de pregrado, Escuela Superior Politécnica del Ejercito]. Repositorio institucional de la Escuela Superior Politécnica del Ejercito. http://repositorio.espe.edu.ec/

Segal, M. (2004). *Machine learning benchmarks and random forest regression* [Tesis, University of California].

Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C. and Weiss, Y. (2016). Andromaly: a behavioural malware detection framework for android devices. *Journal of Intelligent Information Systems, 38*(1), 161-190.

Sonowal, G. & Kuppusamy, K.S. (2020). PhiDMA- A phishing detection model with a multi-filter approach *Journal of King Saud University-Computer and Information Sciences, 32*(1), 99-112.

Sonowal, G. & Kuppusamy, K.S. (2018). MMSPhiD: A Phoneme based Phishing Verification Model for Persons with Visual Impairments. *Information and Computer Security Journal*.

Tan, C. L., Chiew, K. L. and Sze, S. N. (2017). Phishing Webpage Detection Using Weighted URL Tokens for Identity Keywords Retrieval. *In 9th International Conference on Robotic, Vision, Signal Processing and Power Applications* (pp. 133-139). Springer, Singapore.

Moghimi, M. and Varjani, A.Y. (2016). New rule-based phishing detection method.*Expert Systems with Applications*, (pp. 231-242), https://doi.org/10.1016/j.eswa.2016.01.028..

William, W. and Cohen, MLD.CMU. (2019). Base de datos de correos electrónicos de Enron. Web de https://www.cs.cmu.edu/ ./enron/.

Zhao, J., Wang, N., Ma, Q. and Cheng, Z. (2019). Classifying Malicious URLs Using Gated Recurrent Neural Networks. *Springer, Cham*, 385–394.

Zouina, M. & Outtaj, B. (2017). A novel lightweight URL phishing detection system using SVM and similarity index *Human-centric Computing and Information Sciences, 7*(1), 1-13.

BIOGRAFÍA

Fernanda Albán Toapanta. Egresada en Ingeniería Matemática mención Estadística e Investigación Operativa (Escuela Politécnica Nacional, 2021). Dicto charlas en Arcotel, Sercop, CNT, Women in Data Science y en la sesión de Ciberseguridad del Congreso de Investigación Aplicada a Ciencia de Datos – II Congreso Nacional de R Users Group. Realizo análisis estadísticos para Jedal In Software & Al, We Trust y Sercop. Para su trabajo de titulación me ha enfocado en el área de Ciberseguridad específicamente la detección de Phishing.

Ménthor Urvina Mayorga. Matemático (Escuela Politécnica Nacional, 1990). Magister en Investigación Operativa, mención Gerencia, 1998 (Universidad Andina Simón Bolívar – Escuela Politécnica nacional). Profesor Principal a Tiempo Completo de la Escuela Politécnica Nacional, adscrito al Departamento de Matemática. Autor de libros de Cálculo Vectorial y Ecuaciones Diferenciales Ordinarias. Imparte las cátedras de Cálculo Vectorial, Ecuaciones Diferenciales Ordinarias, Probabilidad y Estadística. Áreas de interés: Cálculo, Ecuaciones Diferenciales Ordinarias, Estadística, Probabilidades, Análisis Numérico, Investigación de Operaciones.

Roberto Omar Andrade. Ingeniero en Electrónica y Telecomunicaciones (Escuela Politécnica Nacional (EPN), 2007). Magister en Gestión de Redes y Telecomunicaciones (Escuela Politécnica del Ejército, 2013), en la actualidad es estudiante de doctorado en Sistemas de Seguridad en la EPN. Oficial de Seguridad del Ministerio de Educación de Ecuador (MINEDUC, 2015), Coordinador de Infraestructura Tecnológica en la Secretaría Nacional de Planificación SENPLADES 2013-2014. Centro de datos, seguridad y administración de redes en SENPLADES y Tecnología Sucre 2009-2013 e Ingeniería Técnica para sistemas VoIP en SERATVoIP 2007-2011. Es instructor técnico certificado de CCNA, CCNP y CCNA Security en EPN desde 2010 hasta la fecha.

Apéndice A. PRIMER APÉNDICE

Tabla 6. Lista de notaciones y sus significados.

Notaciones	Descripciones
n	Número de funciones para el análisis de frecuencia
F_{url_i}	Una instancia de la función de URL
d_{ph}	Una base de datos de URL de phishing confirmadas
d_{be}	Una base de datos de URL legítimas confirmadas
θ	El umbral para el análisis de características
p	Tamaño del conjunto de datos
S_i	Categoría de función para la función F2
H_P	Funciones de phishing de alto impacto
$CFS(s)$	Función de selección de característica de correlación para s
f	Una instancia de característica en una categoría de característica particular
t	Factor de correlación (incertidumbre de simetría o correlación Coeficiente de Pearson)
a	Encimera
$f(s)$	El conjunto de características de alto impacto seleccionadas
$m(fs)$	El subconjunto de funciones de alto impacto seleccionadas para la detección de phishing