Diseño de un modelo predictivo para la detección de URLs maliciosas con técnicas de Machine Learning

JHON SEBASTIAN USUGA FERRARO, jhon.usuga1@udea.edu.co
SEBASTIAN RENTERIA PALACIOS, sebastian.renteria@udea.edu.co

I. Descripción del problema

A. Contexto y utilidad del problema

La detección de sitios *phishing* es un problema crítico, y que cada vez ha ganado más relevancia dentro del ámbito de la seguridad informática, ya que millones de usuarios son víctimas cada año de páginas falsas que suplantan sitios legítimos, resultando en pérdidas financieras, el robo de información sensible y la pérdida de la confianza en las instituaciones que manejan estos datos.

El conjunto PhiUSIIL Phishing URL Dataset [PC24] fue creado con el propósito de ofrecer una base estandarizada para entrenar y evaluar modelos de detección automática de phishing. Este dataset integra características léxicas, de contenido y de red, lo que permite a los algoritmos aprender patrones que distinguen una URL real de una fraudulenta.

Automatizar este proceso mediante *Machine Learning* permite reducir el tiempo de respuesta ante amenazas, permitiendo la detección proactiva de ataques desconocidos (*zero-day attacks*) y complementar las soluciones basadas en listas negras, que suelen quedar obsoletas con rapidez frente a la evolución constante y las sofistificación de las tácticas de los atacantes.

Debido a su estructura limpia y su número elevado de registros, el dataset PhiUSIIL se ha convertido en una referencia para estudios de detección de phishing con modelos de *Machine Learning* y *Deep Learning*.

B. Análisis del conjunto de datos

El conjunto PhiUSIIL Phishing URL Dataset contiene 235,795 registros, de los cuales cada uno representa una URL etiquetada como:

- 1 : Sitio legítimo
- 0 : Sitio de phishing

Cada registro incluye 55 atributos que describen propiedades de la URL y de la página web asociada con su respectiva etiqueta, de estos, 51 son de tipo númerico, y 4 de tipo categórico (FILENAME, URL, TLD y Title). Entre las categorías de características más relevantes se encuentran:

Características léxicas: longitud de la URL (URLLength), presencia de caracteres sospechosos como “@”, “//”, “-” (NoOfOtherSpecialCharsInURL).

Características basadas en contenido HTML: Número total de líneas de código (LineOfCode), la existencia de un título (HasTitle) y su relación con el dominio o la URL (DomainTitleMatchScore, URLTitleMatchScore); la presencia de etiquetas visuales o de contenido como imágenes (NoOfImage), hojas de estilo (NoOfCSS) y scripts (NoOfJS).

Características derivadas del dominio: La detección de una dirección IP en lugar de un nombre de dominio (IsDomainIP), TLD y TLDLenght, y TLDLegitimateProb que evalúa la parte final de las URL (.gov, .com, .net), presencia de HTTPS (IsHTTPS) para validar si le sitio posee un certificado.

El dataset fue recopilado a partir de múltiples fuentes y validado manualmente. Durante la fase de análisis exploratorio se observó que varias variables numéricas presentan distribuciones altamente asimétricas y con una gran concentración de valores en cero, además de algunos valores extremos. Estas características sugieren que, dependiendo del modelo a utilizar, podría ser necesario aplicar algún tipo de transformación o escalado para mejorar la estabilidad del entrenamiento.

El conjunto de datos no contiene valores faltantes y la distribución de clases muestra un leve desbalance, con aproximadamente 57% de URLs legítimas y 43% de URLs de phishing.

Esta estructura lo hace adecuado para evaluar distintos algoritmos dentro del paradigma de aprendizaje supervisado. Los modelos basados en árboles, como *Random Forest*, *XGBoost* y *LightGBM*, pueden entrenarse directamente sobre las variables originales, mientras que modelos sensibles a la escala, como *SVM*, *Logistic Regression* y redes neuronales profundas, requieren previamente un proceso de normalización o escalado.

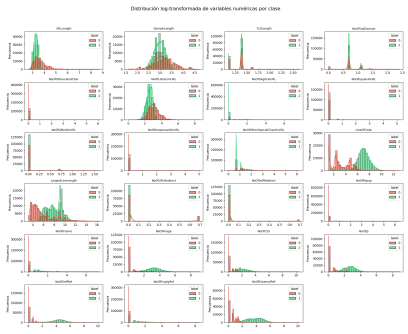

Fig. 1. Distribución log-transformada de variables numéricas por clase

C. Paradigma de aprendizaje y justificación

El proyecto se enmarca en el aprendizaje supervisado con una tarea de clasificación binaria, cuyo objetivo es distinguir entre URLs legítimas y maliciosas (*phishing*). Dado que cada muestra cuenta con una etiqueta previamente asignada (0 o 1), los modelos pueden aprender patrones discriminativos a partir de los atributos descritos.

Las métricas empleadas para la evaluación del rendimiento incluirán:

- Accuracy
- Precision
- Recall
- F1-score

La métrica *F1-score* será especialmente importante para reducir los falsos negativos, es decir, los casos en los que un sitio de phishing se clasifica erróneamente como legítimo, con el uso de este dataset se permitirá comparar diversos modelos de aprendizaje automático, con técnicas de optimización y validación cruzada, asegurando una evaluación rigurosa, reproducible y aplicable a contextos reales de ciberseguridad.

II. Estado del arte

A. Vajrobol et al. [VGG24]

Dado que la base de datos PhiUSIIL contiene ejemplos previamente etiquetados como URLs legítimas y URLs maliciosas, los diferentes trabajos abordados se enmarcan bajo el paradigma de aprendizaje supervisado, en donde el modelo aprende a distinguir entre las clases a partir de las etiquetas proporcionadas. Para ello, se utiliza la técnica de regresión logística en conjunción con selección de características basada en información mutua. Los autores resaltan que la combinación de estas técnicas hacen al modelo simple y eficiente, facilitando su interpretabilidad y manteniendo un equilibrio entre precisión y simplicidad. Como metodología de validación, se particionó el conjunto de datos a un ratio de 80:20. El desempeño del modelo fue evaluado mediante las métricas de *Accuracy*, *Precision*, *Recall* y *F1-score*, y se complementó con matrices de confusión. En la conclusión, lograron una *accuracy* del 99.97% alcanzando esos resultados sólo con 5 características: "URLSimilarityIn- dex," "LineOfCode," "NoOfExternalRef," "NoOfImage," y "NoOfSel- fRef.". Con ello, concluyeron que la técnica de selección de características Mutual Information, en conjunto con la regresión logística, puede ser un modelo muy útil, confiable, preciso y fácil de entender para tareas de identificación de URLs maliciosas.

B. Sruthi K. & Manohar Naik S. [KN25]

Este estudio también emplea un enfoque supervisado, por las mismas razones descritas anteriormente, dado que se usó el mismo conjunto de datos. En este caso, se utiliza una red neuronal siamesa con subredes LSTM. Esta red neuronal procesa pares de URLs que, a través de cada subred LSTM, generan una representación que se compara posteriormente mediante una medida de distancia euclidiana, con el fin de determinar la similitud entre ellas. De este modo, el modelo aprende a reducir la distancia entre pares similares y a aumentarla entre pares disimilares. Se empleó la metodología de validación cruzada *5-fold cross-validation*, la cual fue complementada con estudios de ablación y sensibilidad para medir la estabilidad del modelo. El desempeño del modelo fue evaluado mediante las métricas de *Accuracy*, *Precision*, *Recall* y *F1-score*. En los conjuntos estándar, se obtuvo una *accuracy* del 99.68%, y además, el modelo demostró tener un desempeño similar para muestras generadas usando GPT 4.0, lo que sugiere que podría ser una herramienta efectiva para mitigar los ciberataques de phishing basados en IA.

C. Manguli et al. [MKPR25]

Para este trabajo, se utilizaron los conjuntos de datos PhishTank para la obtención de URLs maliciosas y de Alexa para la obtención de URLs legítimas, lo que, por ende, lo clasifica como un problema de aprendizaje supervisado. La metodología empleada es una Graph Neural Network, específicamente el algoritmo GraphSAGE, que tiene la capacidad de generalizar a nuevos nodos o grafos no vistos durante el entrenamiento. Se empleó la metodología de validación cruzada 5-fold cross-validation. Asimismo, se evaluó el desempeño mediante las métricas de *Accuracy*, *Precision*, *Recall*, *F1-score* y *AUC-ROC*. El modelo GraPhish obtuvo una *accuracy* del 98.96%, por lo que resultó ser una gran alternativa cuando se busca reducir la dependencia en el contenido del mensaje.

D. Fatma Hendaoui & Saloua Hendaoui [HH24]

Propusieron un sistema denominado *SENTINEY*, que combina enfoques de aprendizaje supervisado y no supervisado. Para el aprendizaje no supervisado se implementaron técnicas como *Isolation Forest*, *One-Class SVM*, *Elliptic Envelope*, *K-Means*, *DBSCAN*, *Agglomerative Clustering*, *Birch* y *GMM*; mientras que para el aprendizaje supervisado se emplearon *MLP*, *Random Forest* y *Gradient Boosting*. Dado el diseño del sistema basado en *SMPC*, no se realizó una validación estadística formal, sino pruebas en entornos controlados y distribuidos. Las métricas utilizadas fueron *Accuracy*, *Precision*, *Recall*, *F1-score*, *Average Detection Time* y *Silhouette Score*. El sistema alcanzó una *accuracy* del 99.4% y un tiempo promedio de detección de 0.89 segundos por correo, concluyendo que el módulo supervisado fue más eficaz en la detección de correos de *phishing* conocidos, mientras que el módulo no supervisado resultó más eficiente para identificar *zero-day attacks*.

III. Entrenamiento y evaluación de los modelos

A. Configuración experimental

Para el entrenamiento de los modelos se utilizaron únicamente las características numéricas del conjunto PhiUSIIL, manteniendo la etiqueta `label` como variable objetivo. El conjunto de datos se particionó mediante una división estratificada en proporción 80/20, donde el 80 % se empleó para entrenamiento y validación, y el 20 % restante se reservó exclusivamente para pruebas. La estratificación

garantiza que la relación entre clases (aprox. 57 % URLs legítimas y 43 % URLs de *phishing*) sea similar en los tres subconjuntos.

Dado que algunos algoritmos son sensibles a la escala de las variables, se aplicó una estandarización mediante *StandardScaler* sobre las características numéricas. Este preprocesamiento se utilizó para los modelos lineales (regresión logística y SVM) y para la red neuronal MLP, mientras que los modelos basados en árboles (árbol de decisión y Random Forest) se entrenaron directamente sobre las variables originales.

No se aplicaron técnicas de submuestreo ni sobremuestreo, puesto que el desbalance de clases es moderado y, en las pruebas preliminares, los modelos lograron altos niveles de desempeño sin necesidad de ajustar la distribución de la variable objetivo. Sin embargo, en el caso de la SVM se incluyó el parámetro `class_weight` dentro de la malla de búsqueda para permitir que el modelo compensara automáticamente la clase minoritaria cuando fuera beneficioso.

Para la selección de hiperparámetros se empleó validación cruzada estratificada *K-fold* con $K = 5$ sobre el conjunto de entrenamiento. En cada modelo se realizó una búsqueda exhaustiva mediante *GridSearchCV*, utilizando la exactitud (*accuracy*) promedio de validación como criterio principal de selección. Una vez encontrado el mejor conjunto de hiperparámetros, el modelo se reentrenó con todos los datos de entrenamiento y se evaluó sobre el conjunto de prueba mantenido al margen del proceso de ajuste.

En total se evaluaron cinco modelos de aprendizaje supervisado, de acuerdo con los requerimientos del proyecto:

- **Modelo paramétrico**: Regresión Logística, utilizado como base lineal para establecer un punto de comparación contra métodos más complejos.
- **Modelo no paramétrico**: Árbol de Decisión, empleado como representante de los modelos que no asumen una forma funcional fija y que dividen el espacio de características mediante reglas jerárquicas.
- **Modelo de ensamble**: Random Forest, como ejemplo de métodos basados en el agregado de múltiples árboles de decisión para mejorar la estabilidad, reducir el sobreajuste y capturar relaciones más complejas.
- **Red neuronal artificial**: Perceptrón Multicapa (MLP), incorporado para evaluar un modelo de aprendizaje profundo capaz de aproximar funciones no lineales mediante múltiples capas ocultas.
- **Máquina de vectores de soporte**: SVM lineal (*LinearSVC*), seleccionada por su eficiencia computacional y su capacidad para encontrar hiperplanos óptimos de separación en espacios de alta dimensión.

La Tabla I resume la malla de hiperparámetros considerada para cada uno de los algoritmos durante el proceso de búsqueda.

TABLE I
HIPERPARÁMETROS EVALUADOS PARA CADA MODELO

Modelo	Malla de hiperparámetros
Regresión logística	$C \in \{0.01, 0.1, 1, 10\}$; `solver` = `liblinear`; `penalty` = ℓ_2; $max\_iter \in \{300, 500\}$.
Árbol de decisión	$max\_depth \in \{3, 5, 7, 10\}$; $min\_samples\_split \in \{2, 5, 10\}$; `criterion` $\in \{\texttt{gini}, \texttt{entropy}\}$.
Random Forest	$n\_estimators \in \{100, 200\}$; $max\_depth \in \{\texttt{None}, 10, 20\}$; $min\_samples\_split \in \{2, 5\}$; $max\_features \in \{\texttt{sqrt}, \texttt{log2}\}$.
SVM lineal	$C \in \{0.01, 0.1, 1, 10\}$; `class_weight` $\in \{\texttt{None}, \texttt{balanced}\}$; `penalty` = ℓ_2; `loss` = `squared_hinge`; $max\_iter = 2000$.
Red neuronal MLP	$hidden\_layer\_sizes \in \{(64,), (64, 32)\}$; `activation` = `relu`; `solver` = `adam`; $\alpha \in \{10^{-4}, 10^{-3}\}$; `batch_size` = 256; $max\_iter \in \{50, 100\}$.

B. Métricas de desempeño

El desempeño de los modelos se evaluó mediante métricas clásicas de clasificación binaria: *accuracy*, *precision*, *recall* y *F1-score*. La *accuracy* se utilizó como métrica global para seleccionar los hiperparámetros durante la validación cruzada, ya que resume de manera compacta la proporción de ejemplos correctamente clasificados.

Sin embargo, en problemas de detección de *phishing* resulta particularmente importante controlar los errores asociados a la clase maliciosa. Por ello, sobre el conjunto de prueba se reportan también la *precision*, el *recall* y el *F1-score* de la clase *phishing*. Un *recall* alto implica que la mayoría de las URLs maliciosas son detectadas (pocos falsos negativos), mientras que una *precision* alta indica que la mayoría de las URLs marcadas como maliciosas lo son realmente (pocos falsos positivos). El *F1-score* integra ambas cantidades en una única medida armónica que resulta útil para comparar modelos cuando existe un compromiso entre estos dos tipos de error.

Adicionalmente, para cada modelo se estimó un intervalo de confianza del 95 % para la *accuracy* de validación a partir de las cinco particiones de la validación cruzada, utilizando la desviación estándar de los puntajes obtenidos en cada *fold*. Esto permite cuantificar la variabilidad del desempeño y comparar los modelos no solo por su valor medio, sino también por la estabilidad de sus resultados.

IV. RESULTADOS DEL ENTRENAMIENTO Y EVALUACIÓN DE LOS MODELOS

En esta sección se presentan los resultados obtenidos tras el proceso de entrenamiento, optimización y evaluación de los cinco modelos considerados: regresión logística, árbol de decisión, Random Forest, SVM lineal y red neuronal MLP. Para cada uno se reporta el desempeño base, el desempeño tras la optimización mediante *GridSearchCV* y el intervalo de confianza del 95 % de la *accuracy* de validación, estimado a partir de la validación cruzada estratificada con cinco particiones.

Regresión logística: El modelo base de regresión logística, entrenado sobre los datos estandarizados, obtuvo *accuracies* cercanas al 100 % en los tres subconjuntos. Tras la búsqueda de hiperparámetros, el mejor modelo correspondió a $C = 10$ y $max\_iter = 300$, con *accuracy* de prueba de 0.99994. El intervalo de confianza del 95 % sobre la *accuracy* de validación fue estrecho ($\pm 5.95 \times 10^{-5}$), lo que indica una alta estabilidad del modelo entre particiones.

TABLE II
RESULTADOS DE LA REGRESIÓN LOGÍSTICA

Modelo	Train Acc.	Val. Acc.	Test Acc.
Base	0.99988	0.99986	0.99983
Optimizado	0.99994	0.99990	0.99994

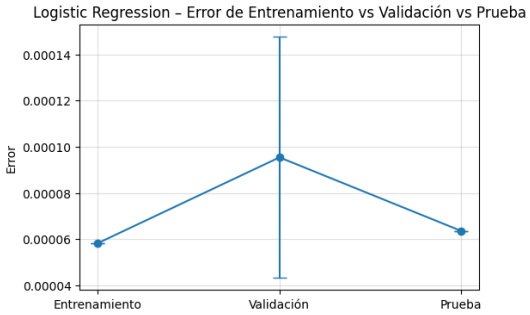

Fig. 2. Error de entrenamiento, validación y prueba para la regresión logística

Árbol de decisión: El árbol de decisión alcanzó un desempeño perfecto, con *accuracy* igual a 1.0 en entrenamiento, validación y prueba, tanto para el modelo base como para el optimizado (siendo este último un árbol con profundidad máxima $max\_depth = 5$). El intervalo de confianza del 95 % fue prácticamente nulo, lo cual refleja que todas las particiones de la validación cruzada obtuvieron resultados idénticos. Esto sugiere que, para este conjunto de datos, las características permiten una separación casi determinística entre URLs legítimas y de *phishing*.

TABLE III
RESULTADOS DEL ÁRBOL DE DECISIÓN

Modelo	Train Acc.	Val. Acc.	Test Acc.
Base	1.0	1.0	1.0
Optimizado	1.0	1.0	1.0

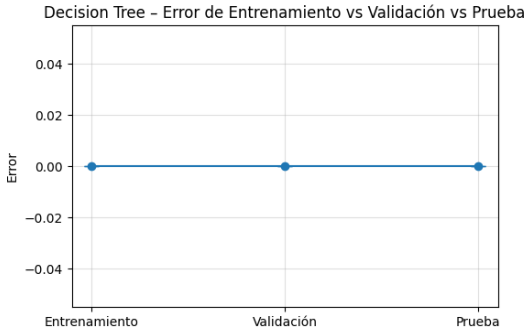

Fig. 3. Error de entrenamiento, validación y prueba para el árbol de decisión

Random Forest: El modelo de ensamble Random Forest también obtuvo un desempeño muy alto. El modelo base alcanzó una *accuracy* de prueba de 0.99983, mientras que el modelo optimizado, con $max\_depth = 10$, $n\_estimators = 200$ y $max\_features = \texttt{sqrt}$, logró una *accuracy* de prueba de 0.99985. La desviación estándar de la *accuracy* de validación fue del orden de 2.7×10^{-4}, lo que indica una variación baja entre particiones y confirma la robustez del ensamble.

TABLE IV
RESULTADOS DEL MODELO RANDOM FOREST

Modelo	Train Acc.	Val. Acc.	Test Acc.
Base	1.0	0.99843	0.99983
Optimizado	0.99981	0.99843	0.99985

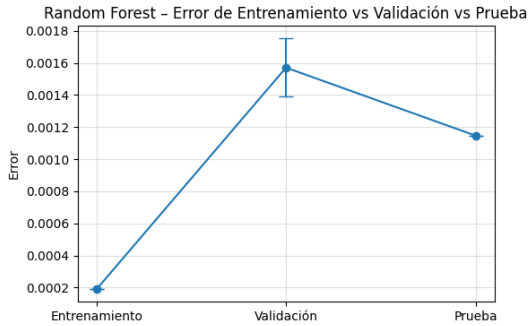

Fig. 4. Error de entrenamiento, validación y prueba para Random Forest

SVM lineal: La SVM lineal entrenada sobre datos estandarizados obtuvo un rendimiento ligeramente inferior al de los modelos anteriores, aunque aún muy alto. El mejor modelo, con $C = 0.01$, alcanzó una *accuracy* de prueba de 0.99669. La desviación estándar de la *accuracy* de validación fue del orden de 4.0×10^{-4}, evidenciando una variación algo mayor entre particiones, pero sin comprometer la capacidad de generalización.

TABLE V
RESULTADOS DE LA SVM LINEAL

Modelo	Train Acc.	Val. Acc.	Test Acc.
Base	0.99660	0.99619	0.99669
Optimizado	0.99660	0.99649	0.99669

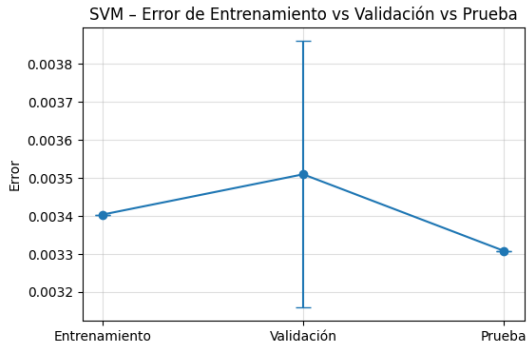

Fig. 5. Error de entrenamiento, validación y prueba para la SVM lineal

Red neuronal MLP: La red neuronal MLP, también entrenada sobre datos estandarizados, mostró un comportamiento competitivo con respecto a la regresión logística. El modelo base alcanzó una *accuracy* de prueba de 0.99961, mientras que el modelo optimizado, con dos capas ocultas $(64, 32)$, activación $\texttt{relu}$ y $\alpha = 0.001$, obtuvo una *accuracy* de

prueba de 0.99956. El intervalo de confianza del 95 % para la *accuracy* de validación fue reducido (orden de 1.4×10^{-4}), indicando una buena estabilidad del entrenamiento.

TABLE VI
RESULTADOS DE LA RED NEURONAL MLP

Modelo	Train Acc.	Val. Acc.	Test Acc.
Base	0.99957	0.99913	0.99961
Optimizado	0.99953	0.99929	0.99956

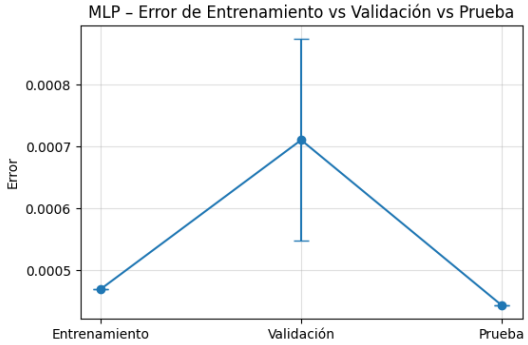

Fig. 6. Error de entrenamiento, validación y prueba para la red neuronal MLP

Comparación general: La Tabla VII resume el desempeño final de los modelos optimizados, permitiendo comparar directamente su capacidad de generalización sobre el conjunto de prueba.

TABLE VII
COMPARACIÓN FINAL DEL DESEMPEÑO DE LOS MODELOS OPTIMIZADOS

Modelo	Train Acc.	Val. Acc.	Test Acc.
Regresión logística	0.99994	0.99990	0.99994
Árbol de decisión	1.00000	1.00000	1.00000
Random Forest	0.99981	0.99843	0.99985
SVM lineal	0.99660	0.99649	0.99669
MLP	0.99953	0.99929	0.99956

En general, todos los modelos alcanzaron niveles de desempeño excepcionalmente altos debido a la alta separabilidad del conjunto de datos. Los modelos basados en árboles (especialmente el árbol de decisión) obtuvieron un ajuste perfecto, mientras que la SVM lineal presentó el desempeño relativamente más bajo. La regresión logística y el MLP ofrecieron un excelente compromiso entre precisión y estabilidad, con errores muy bajos y variaciones reducidas entre particiones de validación.

V. REDUCCIÓN DE DIMENSIÓN

A. Análisis individual de variables

Para hacer el análisis se creó una matriz de correlación con las 50 variables númericas del dataset, sin hacerle limpieza ni ningún tipo de escalado; En la figura se observa una marcada presencia de relaciones fuertes entre las variables, por lo que podría ser una señal de que el uso de técnicas de reducción de la dimensionalidad como PCA y UMAP pueden ser efectivas, y ayudaría a explicar los altos valores que se presenta tanto en entrenamiento, validación y pruebas.

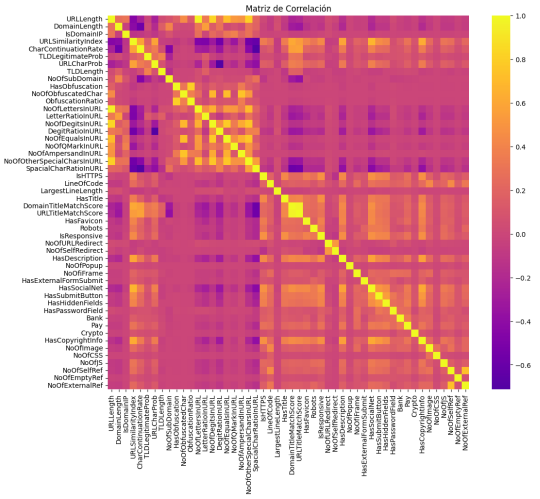

Fig. 7. Matriz de correlación de las variables numéricas del conjunto de datos

De esta matriz se pudo identificar algunos pared de variables con correlaciones absolutas mayores a 0.7, que indican una correlación fuerte, a continuación se describen los patrones más relevantes:

Variable 1	Variable 2
URLLength	NoOfLettersInURL
URLLength	NoOfDegitsInURL
URLLength	NoOfEqualsInURL
URLLength	NoOfOtherSpecialCharsInURL
CharContinuationRate	SpacialCharRatioInURL
URLCharProb	DegitRatioInURL
HasObfuscation	ObfuscationRatio
NoOfObfuscatedChar	NoOfEqualsInURL
NoOfObfuscatedChar	NoOfAmpersandInURL
NoOfLettersInURL	NoOfOtherSpecialCharsInURL
NoOfDegitsInURL	NoOfEqualsInURL
NoOfDegitsInURL	NoOfOtherSpecialCharsInURL
NoOfEqualsInURL	NoOfOtherSpecialCharsInURL
DomainTitleMatchScore	URLTitleMatchScore
NoOfSelfRef	NoOfExternalRef

TABLE VIII
PARES DE VARIABLES CON CORRELACIÓN ELEVADA

Teniendo estas relaciones se quiso complementar el análisis con F-score, Mutual information para así tener una mejor idea de que variables tienen mayor importancia entre las varias que encontramos una fuerte correlación, se desatacó la variable URLSimilarityIndex para ambos métodos, así mismo varias caracterísiticas que describen componentes y/o estructuras de los de las páginas web fueron fueron marcados como muy útiles para identificar sitios legítimos vs maliciosos, los resultados se presentarán a continuación:

TABLE IX
CARACTERÍSTICAS ORDENADAS POR F-SCORE

Característica	F-score
URLSimilarityIndex	535 822.33
HasSocialNet	300 686.22
HasCopyrightInfo	232 887.90
HasDescription	172 562.40
IsHTTPS	111 589.99
DomainTitleMatchScore	98 069.80
HasSubmitButton	94 797.65
IsResponsive	81 166.95
URLTitleMatchScore	77 624.75
SpacialCharRatioInURL	74 792.57
HasHiddenFields	65 502.24
HasFavicon	61 345.86
URLCharProb	52 912.57
CharContinuationRate	52 800.93
HasTitle	50 627.49
DegitRatioInURL	43 127.10
Robots	34 118.41
NoOfOtherSpecialCharsInURL	30 339.34
LetterRatioInURL	29 342.58
Pay	28 199.00

TABLE X
IMPORTANCIA DE VARIABLES SEGÚN MUTUAL INFORMATION

Feature	Mutual Information
URLSimilarityIndex	0.677525
LineOfCode	0.601524
NoOfExternalRef	0.561820
NoOfImage	0.542076
NoOfSelfRef	0.527622
NoOfJS	0.500849
LargestLineLength	0.487839
NoOfCSS	0.446681
HasSocialNet	0.410443
LetterRatioInURL	0.381510
HasCopyrightInfo	0.346477
HasDescription	0.300694
IsHTTPS	0.251048
NoOfOtherSpecialCharsInURL	0.239439
DomainTitleMatchScore	0.211821
HasSubmitButton	0.205996
SpacialCharRatioInURL	0.205360
TLDLegitimateProb	0.194520
URLTitleMatchScore	0.191427
IsResponsive	0.181473

B. PCA

Para la selección de componentes, se contruyó una gráfica para la varianza acumulada, que nos permitiera tener un punto de partida para el cual decidir los criterios por los cuales se eligiría el número, para ello se siguieron las pautas de evaluar los componentes en donde la varianza era de 80%, 90%, 95%, y en donde correspondería el codo, que se tomó en el valor de 25 componentes. Entonces se tomaron los modelos de regresión logística y de árbol de decisión pues fueron los dos que marcaron valores más altos en la presición, y se les hizo el mismo proceso de selección de características, con el fin de buscar resultados similares o mejores a los ya obtenidos, lo cual sucedió por muy poco con el primer modelo, pero igual sigue siendo un muy buen resultado que supera a los obtenidos por otros modelos.

Y con el modelo de árboles de decisión, ya que este obtuvo sin PCA un valor de 1 sólo se podría esperar que igualara su valor, pero en este caso resultó bajando de forma leve, por una diferencia de 0.003 puntos, como se ve a continuación:

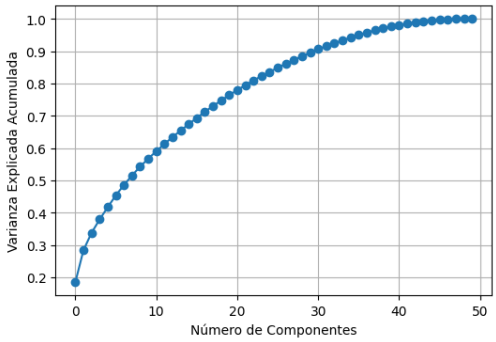

Fig. 8. Varianza Acumulada vs Componentes

Parámetro	Valor
clf__C	10
clf__max_iter	300
clf__penalty	l2
clf__solver	lbfgs
pca__n_components	0.9
Mejor score	0.9997296375833677

TABLE XI
MEJORES HIPERPARÁMETROS DEL MODELO LR + PCA

Parámetro	Valor
clf__criterion	entropy
clf__max_depth	10
clf__min_samples_split	2
pca__n_components	0.95
Mejor score	0.9970419213618358

TABLE XII
MEJORES HIPERPARÁMETROS DEL MODELO DECISION TREE + PCA

C. Discusión

Los resultados obtenidos en los diferentes procesos experimentales obtuvieron unos valores de accuracy bastante altos, sin importar el modelo evaluado, inicialmente pensamos atribuir este comportamiento a un sobreajuste o problemas metodológicos.

Probablemente estos altos valores en la accuracy se deban a algunos factores identificados durante el desarrollo experimental, como son la presencia de múltiples variables categóricas triviales, que indicaban la presencia o la ausencia de elementos en la URL o página a a anlizar, variables que tenían valores muy altos en los scores que determinaban sus capacidades discriminativas.

La aplicación de PCA mostró que la reducción de dimensionalidad no deteriora significativamente el rendimiento; por el contrario, en algunos casos mejora la estabilidad del modelo o reduce el riesgo de redundancia entre características. Esto sugiere que la alta separabilidad del dataset no depende de una sola variable, sino de múltiples grupos de características correlacionadas que capturan propiedades estructurales del sitio web.

Además también es coherente con los hallazgos encontrados en la revisión de la literatura, pero ellos también encontraban

valores muy cercanos al 99%, y concluimos que quizá ese comportamiento es el esperado pues se obtuvieron los datos de distintos sitios web que documentan los hallazgo de sitios web de phishing, por lo tanto, estos ya son altamente caracterizados por ciertos patrones comunes.

REFERENCES

[HH24] Fatma Hendaoui and Saloua Hendaoui. Sentiney: Securing encrypted multi-party computation for enhanced data privacy and phishing detection. *Expert Systems with Applications*, 256:124896, 2024.

[KN25] Sruthi K and Manohar Naik S. A novel framework for effective phishing url detection using an lstm-based siamese network. *Knowledge-Based Systems*, 329:114271, 2025.

[MKPR25] Kartik Manguli, Cheemaladinne Kondaiah, Alwyn Roshan Pais, and Routhu Srinivasa Rao. Graphish: A graph-based approach for phishing detection from encrypted tls traffic. *Journal of Information Security and Applications*, 94:104216, 2025.

[PC24] Arvind Prasad and Shalini Chandra. PhiUSIIL Phishing URL (Website). UCI Machine Learning Repository, 2024. DOI: https://doi.org/10.1016/j.cose.2023.103545.

[VGG24] Vajratiya Vajrobol, Brij B. Gupta, and Akshat Gaurav. Mutual information based logistic regression for phishing url detection. *Cyber Security and Applications*, 2:100044, 2024.