UNIVERSIDAD DEL VALLE DE GUATEMALA
Security Data Science
Sección 10
Ing. Jorge Yass

DETECCIÓN DE PHISHING

Laboratorio 1

Michelle Angel de María Mejía Villela, 22596
Silvia Alejandra Illescas Fernández, 22376

1 Objetivos

- Realizar una revisión de literatura para identificar características potenciales en las URLs de Phishing
- Implementar un modelo de ML para clasificar si un dominio es legítimo o es Phishing.

2 Preámbulo

Phishing

Se basa en la ingeniería social (manipulación de emociones, aprovechamiento de atajos mentales y sesgos cognitivos) para engañar a las víctimas y lograr que estas den información (normalmente credenciales). Los atacantes envían mensajes haciéndose pasar por una entidad legitima a través de correos y SMS bajo diversos "motivos urgentes" que requieren que la persona tome acción inmediatamente, para lo cual incluyen un enlace que redirige al "sitio web" de la entidad.

Estos sitios son literalmente copias de los sitios legítimos que intentan imitar, en muchas ocasiones son muy difíciles de detectar. El usuario, temeroso de un evento negativo ingresa con sus credenciales, las cuales son robadas y utilizadas por los atacantes para acceder a los verdaderos sitios legítimos ocasionando pérdidas económicas (entre otros).

Sin embargo, los dominios web no pueden copiarse al 100%, aunque existen técnicas avanzadas que los hacen parecer similares al ojo humano. Además, las URLs de phishing poseen características que las diferencian de las URLs legítimas, y que un modelo de ML puede utilizar para detectarlas y proteger a los usuarios de estos ataques.

3 Desarrollo

El laboratorio será desarrollado en parejas. Se debe entregar un enlace a un repositorio de Github con el reporte y el código fuente de los modelos. El reporte debe incluir la respuesta a las preguntas y la explicación de las métricas de evaluación. Se proporcionará un dataset con URLs legítimas y de pishing. El lenguaje de programación a utilizar será libre.

Parte 1 – Ingeniería de características

Exploración de datos

1. Cargue el dataset en un dataframe de pandas, muestre un ejemplo de cinco observaciones.

	url	status
0	http://www.crestonwood.com/router.php	legitimate
1	http://shadetreetechnology.com/V4/validation/a...	phishing
2	https://support-appleId.com.secureupdate.duila...	phishing
3	http://rgipt.ac.in	legitimate
4	http://www.iracing.com/tracks/gateway-motorspo...	legitimate

2. Muestre la cantidad de observaciones etiquetadas en la columna *status* como “legit” y como “pishing”. ¿Está balanceado el dataset?

Sí, se encuentra totalmente balanceado, con un 50% de datos legítimos y un 50% de datos categorizados como phishing (lo que equivale a 5715 registros de cada tipo). Podemos decir que, es favorable en el sentido de que se evita un sesgo hacia una clase mayoritaria.

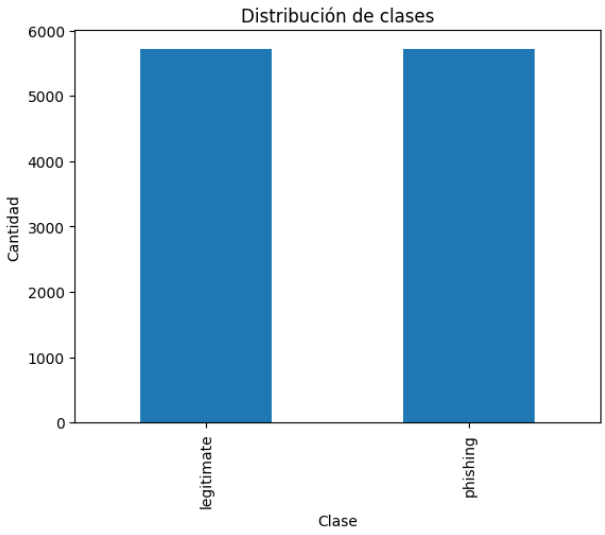
```
Distribución de clases:
status
legitimate    5715
phishing      5715
Name: count, dtype: int64
```

```
# Calcular porcentaje
porcentaje = df['status'].value_counts(normalize=True) * 100

print("\nDistribución porcentual:")
print(porcentaje)
```

✓ 0.0s

```
Distribución porcentual:
status
legitimate    50.0
phishing      50.0
Name: proportion, dtype: float64
```



Derivación de características

Revise los artículos proporcionados, especialmente en el análisis de las URLs. En base a su análisis responda las siguientes preguntas:

1. ¿Qué ventajas tiene el análisis de una URL contra el análisis de otros datos, cómo el tiempo de vida del dominio, o las características de la página Web?

   El análisis de URLs presenta ventajas significativas frente al análisis del contenido web o de características externas como el tiempo de vida del dominio. En primer lugar, permite detección temprana (zero-hour detection), ya que no requiere descargar ni ejecutar la página web, lo que podría ser un riesgo. Aung y Yamana (2019) señalan que muchos sitios phishing contienen información mínima o contenido oculto, lo que dificulta los enfoques basados en análisis visual o de contenido, por lo que proponen centrarse exclusivamente en la estructura de la URL. Además, destacan que el enfoque basado únicamente en URL es el único que permite detectar ataques nuevos sin necesidad de acceso al contenido del sitio.

   Asimismo, las características extraídas de la URL son más rápidas de obtener y adecuadas para sistemas en tiempo real. Calzarossa et al. (2023) explican que las características léxicas de las URLs no requieren descarga de la página, lo que reduce riesgos de seguridad y latencia en la detección. Esto resulta especialmente relevante considerando que los sitios phishing son de vida corta y cambian constantemente, lo que dificulta enfoques dependientes del contenido o del historial del dominio (Hannousse & Yahiouche, 2020)

2. ¿Qué características de una URL son más prometedoras para la detección de phishing?

   Diversos estudios coinciden en que ciertas características estructurales y estadísticas de la URL son altamente discriminativas. Entre ellas se encuentran la longitud de la URL, el número de subdominios, el uso de direcciones IP en lugar de nombres de dominio y la frecuencia de caracteres especiales, características ampliamente reportadas en la literatura. Estas propiedades reflejan intentos de manipulación estructural que buscan ocultar el dominio real o generar confusión en el usuario.

   En particular, Aung y Yamana (2019) proponen medir la entropía de los caracteres no alfanuméricos (como “-”, “//”, “_” y “.”), argumentando que su distribución difiere entre URLs legítimas y phishing. Además, advierten que la presencia de HTTPS ya no es un indicador confiable de legitimidad, dado el aumento de sitios phishing que utilizan certificados SSL para generar confianza. En conjunto, estas características demuestran que el análisis estructural de la URL puede ofrecer señales sólidas y eficientes para la detección automática de phishing.

En base a la respuesta anterior escriba al menos **quince** funciones basadas en los artículos, para derivar características que un modelo pueda utilizar y añada dichas características al dataset original. Incluya dentro de las quince funciones, la entropía de Shanon y relativa.

Se implementaron quince funciones de ingeniería de características basadas en el análisis estructural y estadístico de las URLs, con el objetivo de enriquecer el dataset original y proporcionar variables relevantes para el modelo de clasificación. Entre las funciones incorporadas se incluyen métricas básicas como: la longitud total de la URL, longitud del dominio, número de puntos, guiones, subdominios, barras y parámetros, así como el conteo y proporción de dígitos y caracteres especiales. Asimismo, se añadieron indicadores binarios como la presencia de dirección IP en lugar de nombre de dominio y el uso de HTTPS. Desde una perspectiva más avanzada, se integraron medidas de complejidad como la entropía de Shannon de la URL completa, la entropía relativa (divergencia respecto a una distribución uniforme) y la entropía de caracteres no alfanuméricos, esta última inspirada en la propuesta de Aung y Yamana (2019).

Estas funciones permiten capturar patrones estructurales, irregularidades léxicas y niveles de desorden en la cadena de la URL, aspectos que la literatura identifica como altamente discriminativos para la detección automática de sitios phishing.

Preprocesamiento

Realice las modificaciones necesarias para convertir la variable categórica *status* a una variable binaria. Elimine la columna del dominio. Realice el pre-procesamiento necesario en las demás columnas.

Se convirtió la variable categórica status en una variable binaria (status_bin), donde 1 representa phishing y 0 legítimo. Posteriormente, se eliminó la columna url, ya que corresponde a texto crudo y las características relevantes ya fueron derivadas previamente. Finalmente, se verificaron los tipos de datos, se reemplazaron valores infinitos y nulos, y se realizó imputación utilizando la mediana para garantizar que el dataset estuviera completamente numérico y listo para el entrenamiento de modelos de clasificación.

Selección de Características

En la exploración de datos, determine las columnas que son constantes, o que no tienen una varianza alta con la columna *status*. Elimine las características repetidas o irrelevantes para la clasificación de un sitio de phishing. Verifique que no posee observaciones repetidas. Apóyese con la visualización de características y correlación para seleccionar las características más importantes para clasificar una URL legítima de una URL de phishing.

> Tras realizar el análisis de calidad de datos, se identificaron 547 observaciones duplicadas, reduciendo el conjunto de datos de 11,430 a 10,883 registros únicos. La eliminación de duplicados fue necesaria para evitar sesgos en el entrenamiento del modelo, ya que registros repetidos pueden provocar sobreajuste y una estimación artificialmente optimista del

desempeño. Este paso mejora la capacidad de generalización del modelo al garantizar que cada observación represente una URL única.

Posteriormente, se verificó la presencia de columnas constantes, sin encontrarse variables con varianza nula. El análisis de correlación entre características no evidenció multicolinealidad alta (correlaciones superiores a 0.9), por lo que no fue necesario eliminar variables por redundancia estructural. En cuanto a la relación con la variable objetivo (status), las características con mayor correlación absoluta fueron digit_ratio, shannon_entropy, parameter_count, digit_count, url_length, domain_length y relative_entropy, las cuales capturan patrones estructurales y niveles de complejidad típicos en URLs de phishing. Estas variables fueron priorizadas debido a su mayor capacidad discriminativa en el conjunto de datos analizado.

3. ¿Qué columnas o características fueron seleccionadas y por qué?

Se seleccionaron principalmente las características que mostraron mayor correlación absoluta con la variable objetivo (status) y que además tienen respaldo en la literatura sobre detección de phishing. Entre ellas destacan digit_ratio, shannon_entropy, parameter_count, digit_count, url_length, domain_length, relative_entropy y slash_count, ya que presentaron mayor capacidad discriminativa en el análisis exploratorio. Estas variables capturan patrones estructurales relevantes, como la proporción de números en la URL, la complejidad o desorden de la cadena (medido mediante entropía) y la presencia de parámetros adicionales, características comúnmente asociadas con URLs maliciosas.

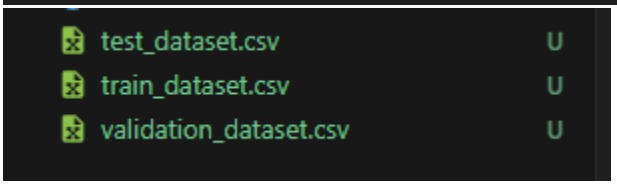
Asimismo, se conservaron variables que, aunque mostraron correlación moderada, aportan información estructural complementaria, como dot_count, special_char_count y nan_entropy. No fue necesario eliminar características por alta multicolinealidad, ya que ninguna superó el umbral establecido de correlación entre variables. En conjunto, las características seleccionadas permiten modelar tanto la estructura léxica como la complejidad estadística de la URL, lo que mejora la capacidad del modelo para diferenciar entre sitios legítimos y phishing.

Parte 2 – Implementación Separación de datos

- Datos de entrenamiento: 55%
- Datos de validación: 15%
- Datos de prueba: 30%
- Almacene cada dataset como un archivo .csv

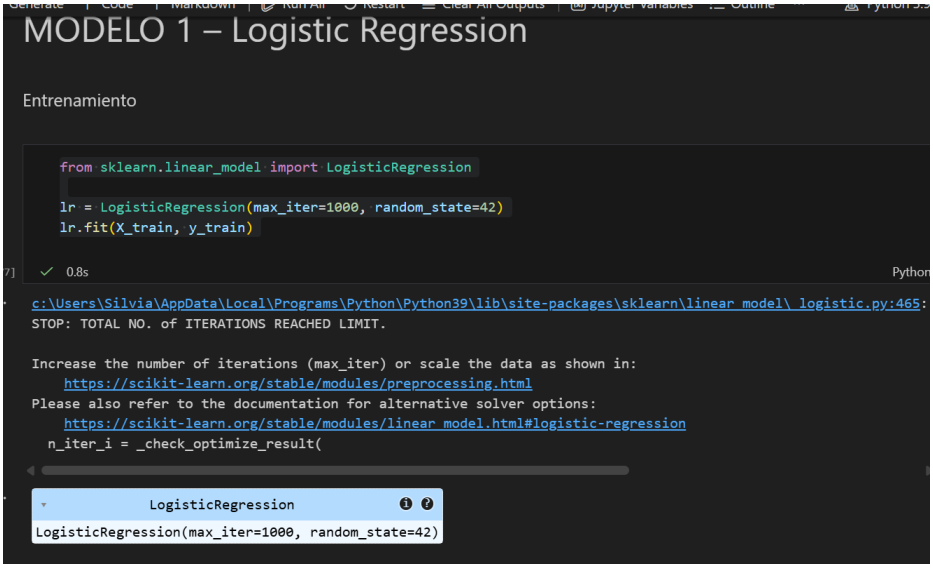
```
Train: 6286
Validation: 1714
Test: 3430

Proporciones:
Train %: 0.5499562554680665
Validation %: 0.14995625546806648
Test %: 0.300087489063867
```



Implementación

Implemente dos modelos de Machine Learning (a su discreción) para la clasificación de phishing. Muestre y explique los valores obtenidos de las siguientes métricas para los datos de validación y pruebas, para cada modelo, en base al contexto del problema (detección de Pishing).

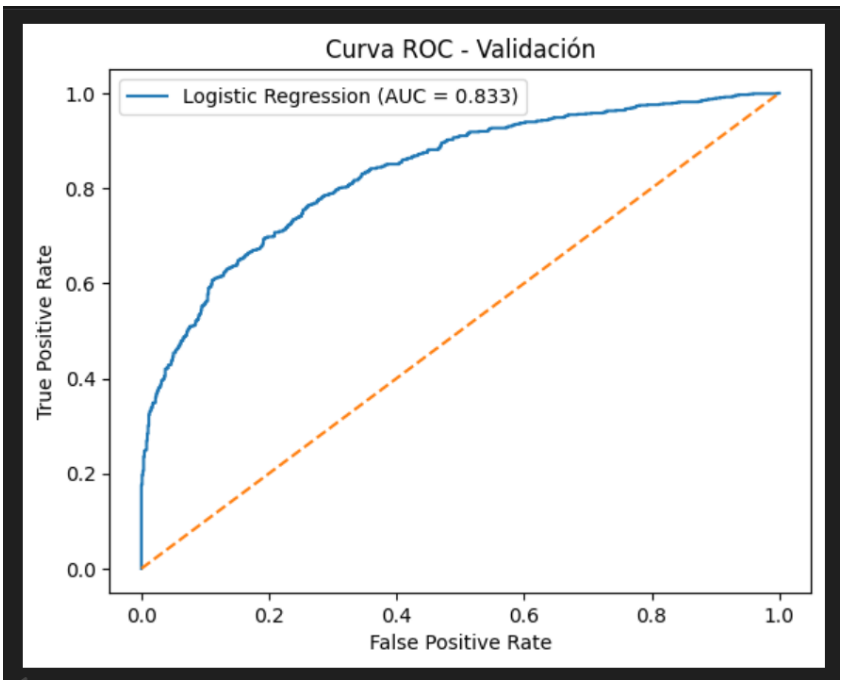


```
Matriz de confusión - Validación (LR)
[[688 169]
 [259 598]]

Reporte clasificación - Validación (LR)
              precision    recall  f1-score   support

           0       0.73      0.80      0.76       857
           1       0.78      0.70      0.74       857

    accuracy                           0.75      1714
   macro avg       0.75      0.75      0.75      1714
weighted avg       0.75      0.75      0.75      1714
```



MODELO 2 – Random Forest

Entrenamiento

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)
```

✓ 3.4s

RandomForestClassifier

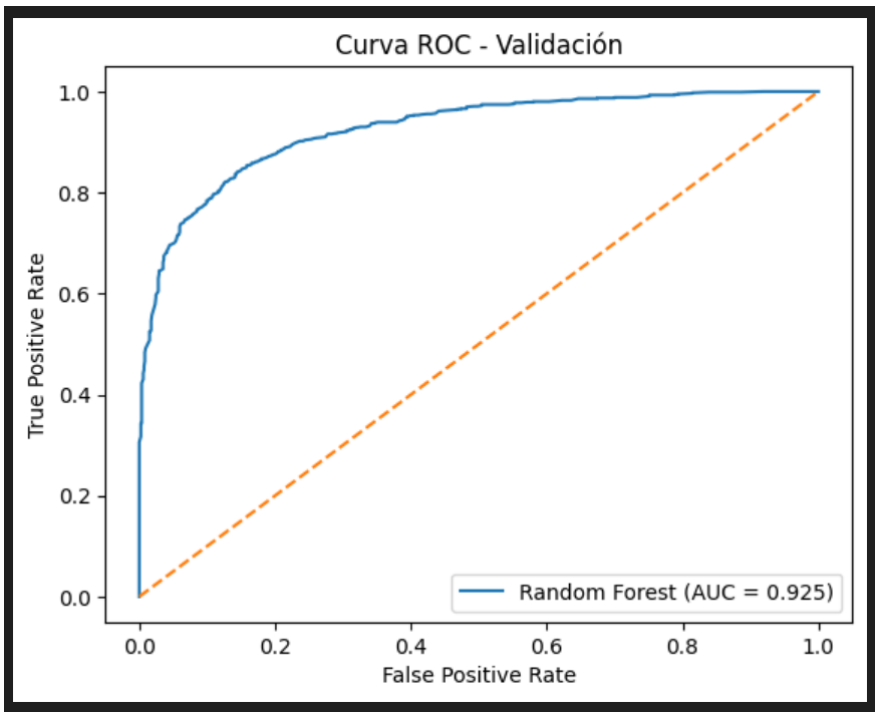
RandomForestClassifier(n_estimators=200, random_state=42)

```
Matriz de confusión - Validación (RF)
[[712 145]
 [121 736]]

Reporte clasificación - Validación (RF)
              precision    recall  f1-score   support

           0       0.85      0.83      0.84       857
           1       0.84      0.86      0.85       857

    accuracy                           0.84      1714
   macro avg       0.85      0.84      0.84      1714
weighted avg       0.85      0.84      0.84      1714
```



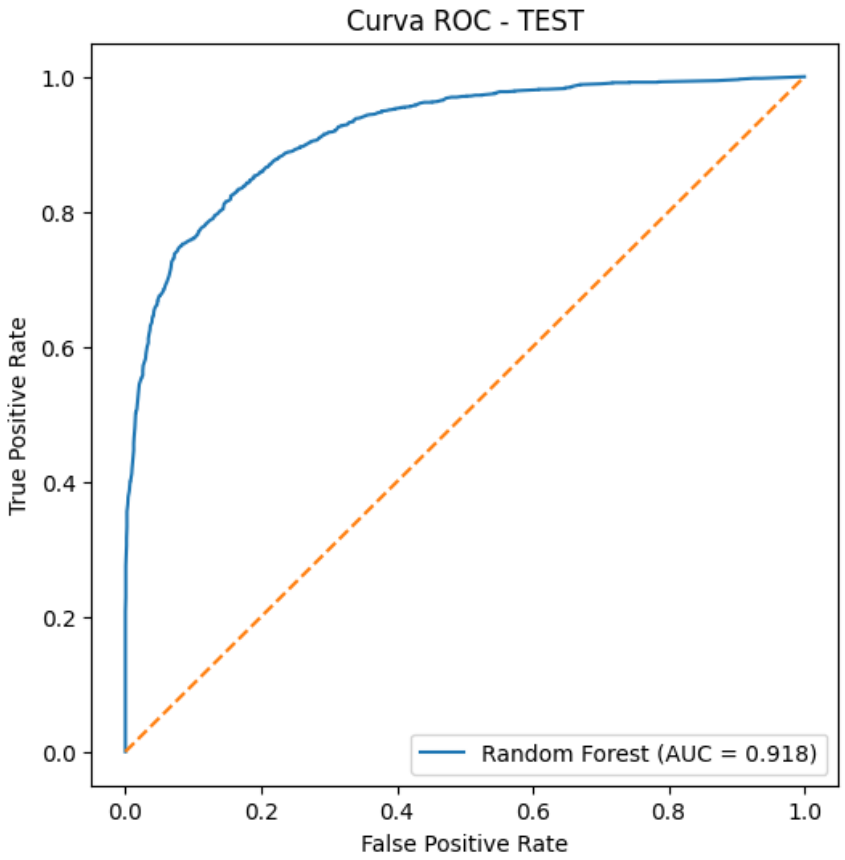
Rando Forest – Test

```
Matriz de confusión - TEST
[[1420  295]
 [ 283 1432]]

Reporte clasificación - TEST
              precision    recall  f1-score   support

           0       0.83      0.83      0.83      1715
           1       0.83      0.83      0.83      1715

    accuracy                           0.83      3430
   macro avg       0.83      0.83      0.83      3430
weighted avg       0.83      0.83      0.83      3430
```



Discusión

En el contexto de la detección de phishing, los errores de clasificación no tienen el mismo impacto. Clasificar un sitio legítimo como phishing corresponde a un falso positivo. En términos prácticos, esto implica bloquear un recurso que no representa una amenaza real. Aunque desde la perspectiva de seguridad puede parecer un error tolerable, en un entorno empresarial puede generar molestias operativas, pérdida de productividad y desconfianza hacia el sistema de protección si ocurre con demasiada frecuencia. Un número elevado de falsas alarmas puede provocar que los usuarios comiencen a ignorar las alertas, disminuyendo la efectividad del sistema en el largo plazo.

Por otro lado, clasificar un sitio de phishing como legítimo (falso negativo) representa un riesgo mucho mayor. En este caso, el modelo permite que un ataque pase desapercibido, lo que podría derivar en robo de credenciales, acceso no autorizado a sistemas internos o pérdidas económicas. Desde una perspectiva de gestión de riesgos, este tipo de error es más crítico, ya que el impacto potencial es directo y puede comprometer la seguridad de la organización. Por esta razón, en problemas de ciberseguridad suele priorizarse la capacidad del modelo para detectar la mayor cantidad posible de ataques reales.

Considerando lo anterior, la métrica más relevante para comparar modelos en este problema es el recall de la clase phishing, ya que mide la proporción de ataques correctamente detectados. Sin embargo, también es importante evaluar el AUC, dado que resume la capacidad general del modelo para separar correctamente ambas clases a distintos umbrales. En este laboratorio, el modelo Random Forest mostró un desempeño superior al de la Regresión Logística, tanto en validación como en pruebas. En particular, presentó un AUC cercano a 0.92 en el conjunto de prueba y métricas equilibradas de precisión y recall (~0.83), lo que indica una buena capacidad discriminativa y estabilidad en datos no vistos.

Si se traslada este modelo a un escenario real donde una empresa recibe 50,000 correos electrónicos y aproximadamente el 15% son phishing (7,500 correos), el modelo detectaría correctamente alrededor del 83% de esos ataques, es decir, aproximadamente 6,225 correos maliciosos. Sin embargo, cerca de 1,275 ataques podrían pasar como legítimos. Asimismo, considerando su precisión, el modelo generaría aproximadamente 1,275 falsas alarmas sobre correos legítimos. En total, el sistema activaría cerca de 7,500 alertas, de las cuales una parte correspondería a verdaderos ataques y otra a correos legítimos mal clasificados.

En este sentido, el modelo es funcional y representa una mejora significativa frente a no tener ningún mecanismo de detección, pero todavía implica un riesgo residual importante. Para reducir la cantidad de falsas alarmas o mejorar la detección, podrían ajustarse los umbrales de decisión, incorporar técnicas adicionales como modelos ensemble más complejos, o combinar el modelo con filtros heurísticos y sistemas de verificación en múltiples capas. De esta manera, se podría lograr un equilibrio más adecuado entre seguridad y eficiencia operativa dentro de la organización.

Referencias

Aung, E. S., & Yamana, H. (2019). URL-based phishing detection using the entropy of non-alphanumeric characters. Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019). https://doi.org/10.1145/3366030.3366064

Calzarossa, M. C., Giudici, P., & Zieni, R. (2024). Explainable machine learning for phishing feature detection. Quality and Reliability Engineering International, 40, 362–373. https://doi.org/10.1002/qre.3411

Hannousse, A., & Yahiouche, S. (2020). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. arXiv preprint arXiv:2010.12847.