Uso de redes neuronales para detección de phishing en URL

Iván Gabriel Salinas Castillo, Matricula: 1856735

Profesor: Dr. José de Jesús Rocha Salazar

Universidad Autónoma de Nuevo León, Maestría en Ciencia de Datos

Aprendizaje Profundo Grupo 001

Fecha: 25 de Noviembre del 2024

Resumen

Este trabajo presenta un enfoque para la detección de sitios web de phishing mediante el análisis de URLs utilizando redes neuronales. Se utilizó un dataset específico de URLs, con etiquetas que clasifican cada muestra como legítima o phishing. Inicialmente, las características del dataset relacionadas con el análisis de URLs fueron seleccionadas y preprocesadas. Las variables categóricas, como el dominio de nivel superior (TLD), se codificaron numéricamente, y las características numéricas fueron normalizadas. Para reducir la dimensionalidad y mantener las características más relevantes, se aplicó el método de Análisis de Componentes Principales (PCA), seleccionando las principales ocho componentes. Posteriormente, se construyó una red neuronal profunda con dos capas densas y regularización mediante *dropout* para evitar el sobreajuste. El modelo fue entrenado utilizando una partición del dataset y evaluado mediante una matriz de confusión y la precisión del modelo. Los resultados demostraron que este enfoque basado en redes neuronales es capaz de identificar de manera efectiva URLs maliciosas, ofreciendo una solución prometedora para combatir ataques de phishing en línea. Además, el uso de PCA redujo la complejidad computacional, haciendo el modelo más eficiente sin comprometer significativamente el rendimiento.

Introducción

El **phishing** es un delito cibernético en el que actores malintencionados, haciéndose pasar por instituciones legítimas, se comunican con individuos a través de llamadas telefónicas, correos electrónicos o mensajes de texto. El objetivo de estas comunicaciones engañosas es inducir a las personas a revelar información sensible, como contraseñas personales, datos de tarjetas de crédito, autorizaciones bancarias y otros datos privados. Una vez que se obtienen estos detalles, los perpetradores

pueden acceder a cuentas cruciales, lo que puede derivar en robos de identidad y pérdidas financieras, además en el mundo cibernético, los ataques de phishing han experimentado un aumento significativo en los últimos años (Bouijij, H., & Berqia, A. 2024).

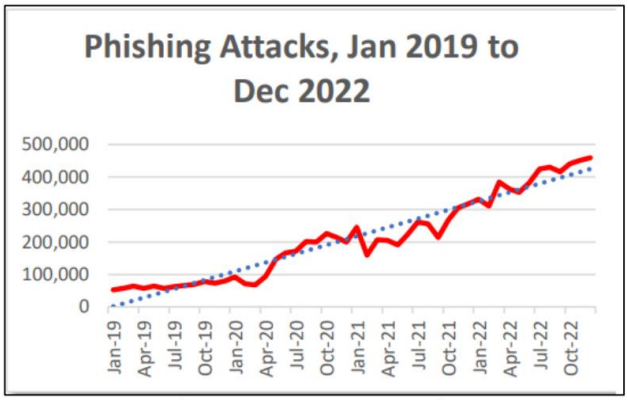

Figura 1. Incremento de phishing del 2019 al 2022 (Bouijij, H., Berqia, A., & Saliah-Hassan, H. 2022, June).

Las redes neuronales profundas son modelos computacionales inspirados en la estructura del cerebro humano. Estas redes están compuestas por capas de neuronas conectadas entre sí, donde cada conexión tiene un peso ajustable que se optimiza durante el entrenamiento. En este proyecto, se utiliza una red neuronal densa, o fully connected, que es adecuada para problemas de clasificación binaria, como distinguir entre URLs legítimas y maliciosas. El modelo implementado contiene múltiples capas densas con funciones de activación no lineales (ReLU y sigmoid) que permiten al modelo aprender relaciones complejas entre las características de las URLs y sus etiquetas (Goodfellow, I., Bengio, Y., & Courville, A. 2016).

Uno de los pasos fundamentales para garantizar el rendimiento de las redes neuronales es el preprocesamiento de los datos. **La normalización** es una técnica utilizada para escalar los valores de las características de manera que tengan una media cercana a 0 y una desviación estándar de 1. Este paso es esencial para que las redes neuronales converjan más rápido durante el entrenamiento, ya que evita que las características con escalas diferentes dominen el proceso de optimización (Géron, A. 2019).

Además, debido a la alta dimensionalidad de las características del conjunto de datos, se aplicó el **Análisis de Componentes Principales (PCA)**, una técnica de reducción de dimensionalidad que transforma las características originales en un conjunto de componentes principales ortogonales. Estos componentes están ordenados según la varianza que explican en los datos, lo que permite retener la mayor cantidad de

información relevante utilizando un menor número de dimensiones. PCA no solo ayuda a reducir el ruido y la redundancia en los datos, sino que también mejora la eficiencia computacional del modelo y puede evitar el sobreajuste al eliminar características irrelevantes (Jolliffe, I. T., & Cadima, J. 2016).

Combinando estas técnicas con la capacidad de las redes neuronales para identificar patrones no lineales en los datos, este trabajo propone una solución efectiva para la detección de phishing en URLs. El enfoque integra normalización, PCA y el diseño de una red neuronal bien estructurada para abordar este desafío con alta precisión y robustez.

Metodología

1. Recolección y comprensión de los datos

- Se utilizó el conjunto de datos *PhiUSIIL Phishing URL Dataset*, que contiene información sobre URLs categorizadas como legítimas o phishing (Prasad, A., & Chandra, S. 2024).
- El conjunto de datos incluye diversas características relacionadas con la estructura de los URLs, como la longitud del dominio, presencia de caracteres especiales, y el uso de HTTPS, entre otros.
- Se identificó la variable objetivo label que clasifica cada URL como phishing (1) o legítima (0).

2. Preprocesamiento de los datos

- **Transformación de datos categóricos:**
  La columna TLD (Top Level Domain) fue transformada a valores numéricos utilizando *Label Encoding*. Esto permitió representar de manera eficiente los dominios categóricos en el modelo.
- **Selección de características relevantes:**
  Se eliminaron características irrelevantes para el análisis, como la cantidad de imágenes en la página web, ya que no están relacionadas directamente con la estructura de las URLs.
- **Normalización:**
  Todas las características numéricas se escalaron utilizando el método *StandardScaler*. Esto ajusta los datos a una distribución con media cero y desviación estándar uno, lo que mejora el rendimiento de los modelos basados en redes neuronales.

3. Reducción de dimensionalidad con PCA

- Se aplicó el *Análisis de Componentes Principales* (PCA) para reducir la dimensionalidad de los datos.
- El número de componentes principales se estableció en 8, seleccionados con base en el análisis de la varianza explicada acumulada. Esto permitió reducir el ruido y optimizar el rendimiento del modelo.

4. División de los datos

- Los datos preprocesados se dividieron en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%) utilizando *train_test_split*.
- Además, dentro del conjunto de entrenamiento, se reservó el 20% para validación cruzada durante el entrenamiento del modelo.

5. Construcción del modelo de red neuronal

- Se diseñó una red neuronal utilizando la biblioteca Keras. La arquitectura incluyó:
    - Una capa densa de entrada con 64 neuronas y función de activación ReLU.
    - Una capa densa oculta con 32 neuronas y función de activación ReLU.
    - Una capa de salida con una única neurona y función de activación sigmoide, adecuada para clasificación binaria.
    - Capas *Dropout* con una tasa de 0.5 para prevenir el sobreajuste.
- La red se compiló utilizando el optimizador *Adam*, con la función de pérdida *binary_crossentropy* y la métrica de precisión.

6. Entrenamiento del modelo

- El modelo fue entrenado durante 12 épocas con un tamaño de lote de 32, utilizando el conjunto de entrenamiento y validación.

7. Evaluación del modelo

- El modelo se evaluó en el conjunto de prueba, utilizando la precisión como métrica principal.
- Se generó una matriz de confusión para analizar la distribución de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

8. Visualización de resultados

- Se realizó un *Scree Plot* para visualizar la varianza explicada por los componentes principales seleccionados en PCA.
- Se utilizó un mapa de calor para explorar la matriz de correlación entre las características.
- Se graficó la evolución de la precisión y error en el conjunto de entrenamiento y prueba a lo largo de las épocas.
- Finalmente, se representó gráficamente la matriz de confusión obtenida durante la evaluación del modelo.

Datos

El Conjunto de Datos de URLs de Phishing PhiUSIIL comprende 134,850 URLs legítimas y 100,945 URLs de phishing. Las características se extraen del código fuente de la página web y de la URL. A continuación, se muestran las descripciones de nuestras 55 características del conjunto de datos.

1. URL: La URL completa.
2. URLLength: La longitud de la URL en número de caracteres.
3. Domain: El dominio de la URL.
4. DomainLength: La longitud del dominio en número de caracteres.
5. IsDomainIP: Indica si el dominio es una dirección IP (1 = sí, 0 = no).
6. TLD: El dominio de nivel superior (Top-Level Domain) de la URL (por ejemplo, .com, .org).
7. URLSimilarityIndex: Un índice que mide la similitud de la URL con URLs conocidas.
8. CharContinuationRate: La tasa de continuación de caracteres en la URL.
9. TLDLegitimateProb: La probabilidad de que el TLD sea legítimo.
10. URLCharProb: La probabilidad de caracteres específicos en la URL.
11. TLDLength: La longitud del TLD en número de caracteres.
12. NoOfSubDomain: El número de subdominios en la URL.
13. HasObfuscation: Indica si la URL tiene ofuscación (1 = sí, 0 = no).
14. NoOfObfuscatedChar: El número de caracteres ofuscados en la URL.
15. ObfuscationRatio: La proporción de caracteres ofuscados en la URL.
16. NoOfLettersInURL: El número de letras en la URL.
17. LetterRatioInURL: La proporción de letras en la URL.
18. NoOfDegitsInURL: El número de dígitos en la URL.
19. DegitRatioInURL: La proporción de dígitos en la URL.
20. NoOfEqualsInURL: El número de signos de igualdad (=) en la URL.
21. NoOfQMarkInURL: El número de signos de interrogación (?) en la URL.
22. NoOfAmpersandInURL: El número de signos de ampersand (&) en la URL.

23. NoOfOtherSpecialCharsInURL: El número de otros caracteres especiales en la URL.
24. SpacialCharRatioInURL: La proporción de caracteres especiales en la URL.
25. IsHTTPS: Indica si la URL usa HTTPS (1 = sí, 0 = no).
26. LineOfCode: Número de líneas de código en la página web correspondiente a la URL.
27. LargestLineLength: La longitud de la línea más larga de código.
28. HasTitle: Indica si la página web tiene un título (1 = sí, 0 = no).
29. Title: El título de la página web.
30. DomainTitleMatchScore: Puntaje que mide la coincidencia entre el dominio y el título de la página.
31. URLTitleMatchScore: Puntaje que mide la coincidencia entre la URL y el título de la página.
32. HasFavicon: Indica si la página web tiene un favicon (1 = sí, 0 = no).
33. Robots: Indica si hay un archivo robots.txt (1 = sí, 0 = no).
34. IsResponsive: Indica si la página web es responsiva (1 = sí, 0 = no).
35. NoOfURLRedirect: El número de redirecciones URL.
36. NoOfSelfRedirect: El número de redirecciones internas.
37. HasDescription: Indica si la página web tiene una meta descripción (1 = sí, 0 = no).
38. NoOfPopup: El número de ventanas emergentes.
39. NoOfiFrame: El número de elementos iFrame.
40. HasExternalFormSubmit: Indica si la página web tiene un formulario que envía datos externamente (1 = sí, 0 = no).
41. HasSocialNet: Indica si la página web tiene enlaces a redes sociales (1 = sí, 0 = no).
42. HasSubmitButton: Indica si la página web tiene un botón de envío (1 = sí, 0 = no).
43. HasHiddenFields: Indica si la página web tiene campos ocultos (1 = sí, 0 = no).
44. HasPasswordField: Indica si la página web tiene un campo de contraseña (1 = sí, 0 = no).
45. Bank: Indica si la URL está relacionada con banca (1 = sí, 0 = no).
46. Pay: Indica si la URL está relacionada con pagos (1 = sí, 0 = no).
47. Crypto: Indica si la URL está relacionada con criptomonedas (1 = sí, 0 = no).
48. HasCopyrightInfo: Indica si la página web tiene información de derechos de autor (1 = sí, 0 = no).
49. NoOfImage: El número de imágenes en la página web.
50. NoOfCSS: El número de hojas de estilo CSS en la página web.
51. NoOfJS: El número de archivos JavaScript en la página web.
52. NoOfSelfRef: El número de referencias internas.
53. NoOfEmptyRef: El número de referencias vacías.
54. NoOfExternalRef: El número de referencias externas.
55. label: La etiqueta de la URL (1 = phishing, 0 = legítima).

	URL	URLLength	Domain	NoOfExternalRef	label	TLD_encoded
0	https://www.southbankmosaics.com	31	www.southbankmosaics.com	124	1	231
1	https://www.uni-mainz.de	23	www.uni-mainz.de	217	1	254
2	https://www.voicefmradio.co.uk	29	www.voicefmradio.co.uk	5	1	647
3	https://www.sfnmjournal.com	26	www.sfnmjournal.com	31	1	231
4	https://www.rewildingargentina.org	33	www.rewildingargentina.org	85	1	503
...	...	...	...	...	...	...
235790	https://www.skincareliving.com	29	www.skincareliving.com	191	1	231
235791	https://www.winchester.gov.uk	28	www.winchester.gov.uk	31	1	647
235792	https://www.nononsensedesign.be	30	www.nononsensedesign.be	67	1	157
235793	https://patient-cell-40f5.updatedlogmylogin.wo...	55	patient-cell-40f5.updatedlogmylogin.workers.dev	0	0	258
235794	https://www.alternativefinland.com	33	www.alternativefinland.com	261	1	231

Figura 2. Primeras columnas del conjunto de datos.

En el presente estudio, hemos seleccionado un conjunto de 18 características específicas que serán utilizadas en el modelo de detección de phishing en URLs. Estas características, que se pueden obtener directamente del URL, incluyen DomainLength, IsDomainIP, CharContinuationRate, TLDLength, NoOfSubDomain, HasObfuscation, NoOfObfuscatedChar, ObfuscationRatio, NoOfLettersInURL, LetterRatioInURL, NoOfDegitsInURL, DegitRatioInURL, NoOfQMarkInURL, NoOfAmpersandInURL, NoOfOtherSpecialCharsInURL, SpacialCharRatioInURL, IsHTTPS y TLD_encoded. Estas características han sido seleccionadas debido a su relevancia y su capacidad para describir adecuadamente las propiedades del URL, además de haber superado un proceso de selección de características basado en su correlación, asegurando así que se consideren las más informativas y menos redundantes para el modelo.

Resultados

Análisis de correlación

Se realizó un análisis de correlación entre las variables para identificar relaciones lineales fuertes que podrían redundar en el modelo. Las variables con una correlación superior a 0.8 incluyen:

- NoOfLettersInURL y URLLength con una correlación de 0.96.
- NoOfDegitsInURL y URLLength con una correlación de 0.84.
- NoOfEqualsInURL y NoOfDegitsInURL con una correlación de 0.81.

Dado que estas relaciones implican redundancia, se seleccionaron las siguientes características para el análisis final: DomainLength, IsDomainIP, CharContinuationRate, TLDLength, NoOfSubDomain, HasObfuscation, NoOfObfuscatedChar, ObfuscationRatio, NoOfLettersInURL,

LetterRatioInURL, NoOfDegitsInURL, DegitRatioInURL, NoOfQMarkInURL, NoOfAmpersandInURL, NoOfOtherSpecialCharsInURL, SpacialCharRatioInURL, IsHTTPS, label, y TLD_encoded.

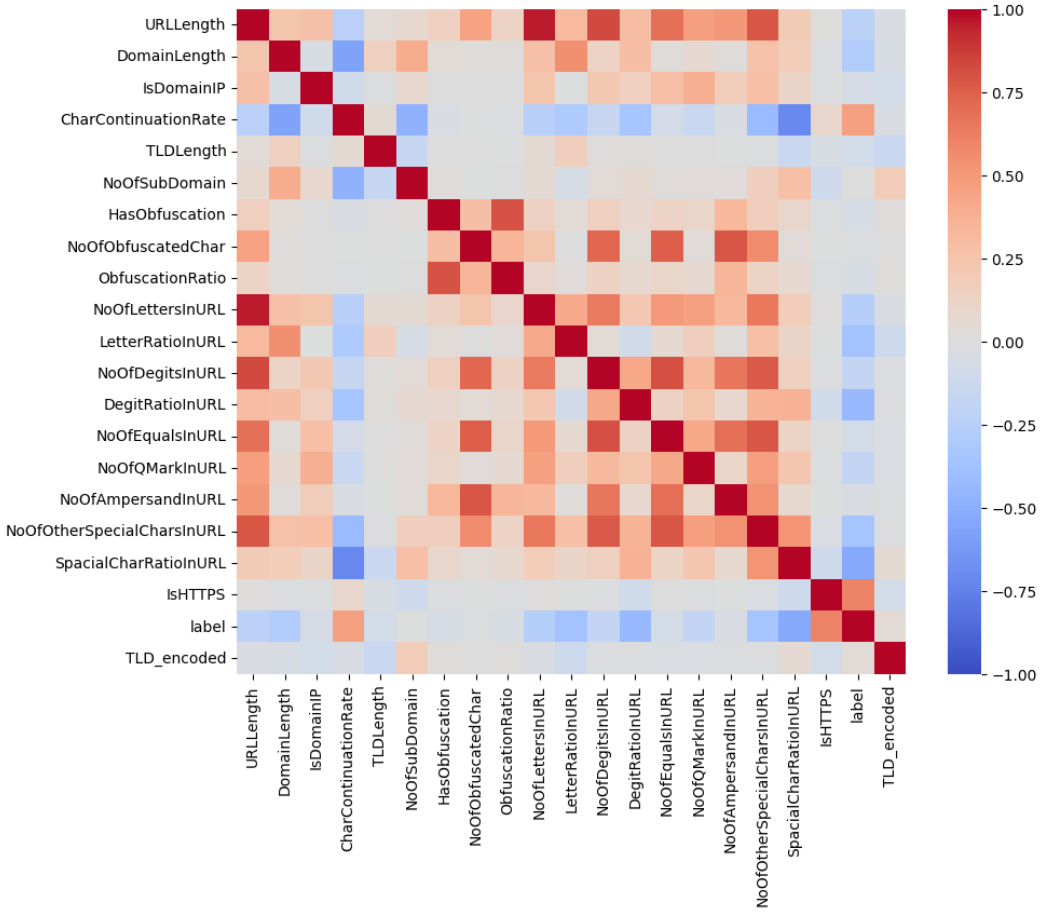

Figura 3. Matriz de correlación de características que solo se encuentran en la URL.

Reducción de dimensionalidad con PCA

Se aplicó Análisis de Componentes Principales (PCA) con 8 componentes principales, obteniéndose los siguientes resultados:

- Varianza explicada por cada componente principal:
  - PC1: 24.67%
  - PC2: 13.88%
  - PC3: 9.57%
  - PC4: 8.86%
  - PC5: 7.22%
  - PC6: 5.96%
  - PC7: 5.53%
  - PC8: 4.90%

- Varianza acumulada explicada:
  La suma de las varianzas explicadas por los 8 componentes principales fue de 80.59%, lo que indica que esta cantidad de componentes es suficiente para preservar una gran parte de la información del conjunto de datos original.

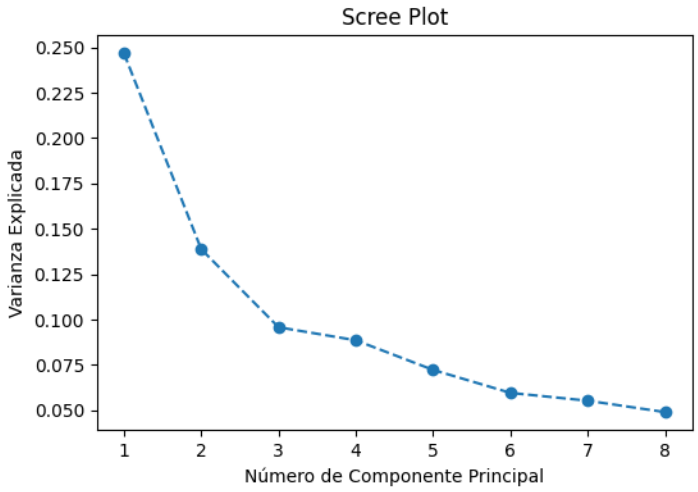

Figura 4. Scree plot del PCA realizado.

Resultados de la red neuronal

La red neuronal fue entrenada durante 12 épocas, mostrando los siguientes resultados:

- Precisión y pérdida en el conjunto de entrenamiento:
  - Precisión: 95.56%.
  - Pérdida: 0.0218.
- Precisión y pérdida en el conjunto de validación:
  - Precisión: 99.67%.
  - Pérdida: 0.0162.

El entrenamiento de la red neuronal mostró un rendimiento consistente en precisión y pérdida, con una clara tendencia hacia la mejora en ambas métricas a medida que avanzaban las épocas.

Evolución de métricas durante el entrenamiento

A continuación, se presentan gráficas que ilustran el comportamiento de la precisión y la pérdida a lo largo de las épocas:

1. Cambio de precisión respecto a las épocas:
   La gráfica muestra un aumento constante en la precisión tanto para el conjunto de entrenamiento como para el de validación, lo que sugiere una mejora general en la capacidad predictiva del modelo.
2. Cambio del error conforme avanzan las épocas:
   Se observó una disminución continua en la pérdida para ambos conjuntos, lo que refleja que el modelo converge adecuadamente hacia una solución óptima sin signos evidentes de sobreajuste.

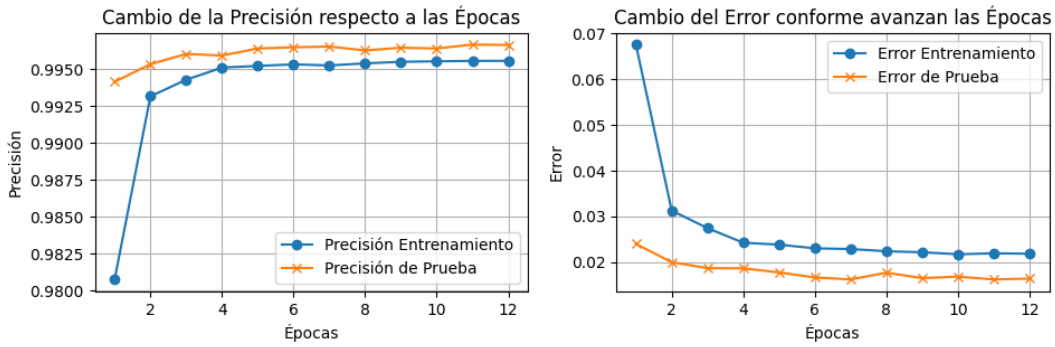

Figura 5. Cambio de la precisión y error respecto a las épocas.

Ambas gráficas indican que el modelo entrenado es robusto y tiene un alto rendimiento en la clasificación de URLs legítimas y de phishing.

La matriz de confusión ilustra la capacidad del modelo para distinguir entre URLs legítimas (clase 0) y URLs de phishing (clase 1). Los resultados se desglosan a continuación:

- **Verdaderos positivos (VP):** El modelo identificó correctamente 27,010 URLs de phishing.
- **Verdaderos negativos (VN):** Se identificaron correctamente 19,998 URLs legítimas.
- **Falsos positivos (FP):** Se clasificaron erróneamente 126 URLs legítimas como phishing.
- **Falsos negativos (FN):** Solo 25 URLs de phishing fueron clasificadas incorrectamente como legítimas.

Estos resultados reflejan un excelente desempeño del modelo, con una alta sensibilidad (capacidad de identificar correctamente los casos positivos) y precisión.

Esto se refuerza con la baja cantidad de errores tanto de falsos positivos como de falsos negativos.

Para cuantificar aún más el desempeño, se pueden calcular las siguientes métricas:

Precisión (Accuracy): Mide el porcentaje total de predicciones correctas.

$$Precisión = \frac{(VP + VN)}{Total} = \frac{(27010 + 19998)}{47159} \approx 99.68\%$$

Sensibilidad (Recall): Indica la proporción de URLs de phishing correctamente identificadas.

$$Sensibilidad = \frac{VP}{VP + FN} = \frac{27010}{27010 + 25} \approx 99.91\%$$

Especificidad: Evalúa la proporción de URLs legítimas correctamente clasificadas.

$$Especificidad = \frac{VN}{VN + FP} = \frac{19998}{19998 + 126} \approx 99.37\%$$

Precisión Positiva (Precision): Representa la proporción de predicciones positivas que realmente eran phishing.

$$Precisión\ Positiva = \frac{VP}{VP + FP} = \frac{27010}{27010 + 126} \approx 99.54\%$$

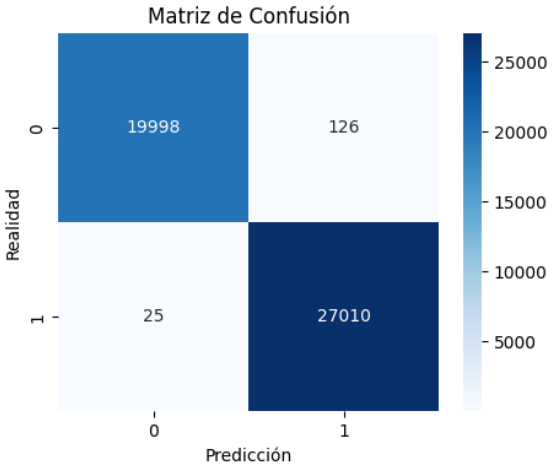

Figura 6. Matriz de confusión.

Conclusiones

El presente estudio demostró la eficacia de una red neuronal para la detección de URLs de phishing, utilizando características específicas extraídas directamente de los enlaces. Mediante un análisis exhaustivo de correlaciones, se seleccionaron variables

relevantes que minimizan la redundancia y maximizan la información útil para el modelo. El uso de PCA permitió una reducción eficiente de dimensionalidad, preservando más del 80% de la varianza original en solo 8 componentes principales, lo que contribuyó a simplificar el modelo y mejorar su rendimiento. Los resultados del modelo de red neuronal mostraron una precisión sobresaliente, alcanzando un 99.67% en el conjunto de validación. Además, la evolución de la precisión y la pérdida a lo largo de las épocas indicó un entrenamiento estable y una capacidad de generalización robusta. En conclusión, el enfoque propuesto combina técnicas de preprocesamiento, reducción de dimensionalidad y redes neuronales para abordar de manera eficiente el problema del phishing, ofreciendo una solución viable y de alto rendimiento para sistemas de seguridad informática.

Referencias

Dataset: https://data.mendeley.com/datasets/shwpxscxy2/2

Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Security*, *136*, 103545.

Bouijij, H., & Berqia, A. (2024). Enhancing IOT security: Proactive phishing website detection using Deep Neural Networks case study: smart home. Journal of Telecommunications and the Digital Economy, 12(1), 446-462.

Bouijij, H., Berqia, A., & Saliah-Hassan, H. (2022, June). Phishing URL classification using Extra-Tree and DNN. In 2022 10th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202