Evaluación del rendimiento de clasificadores de aprendizaje profundo y ensemble para la detección automatizada de phishing en correos electrónicos

Rodríguez Chacón, María D.
Facultad de Ingeniería
Universidad de Antioquia
Medellin, Antioquia
mdaniela.rodriguez@udea.edu.co

Ospina González, E.
Facultad de Ingeniería
Universidad de Antioquia
Medellin, Antioquia
estiven.ospinag@udea.edu.co

***Abstract*—This work presents a machine learning approach for detecting phishing emails using the Phishing and Legitimate Emails Dataset, a synthetically generated corpus of 10,000 email samples. The study focuses on binary classification to distinguish phishing from legitimate messages, leveraging logistic regression as a baseline model. Unlike traditional datasets, this collection includes enriched metadata such as phishing type, severity, and confidence levels, allowing for enhanced feature exploration. Experimental results demonstrate the effectiveness of logistic regression in capturing key linguistic and structural cues within email text, highlighting its potential as a lightweight yet reliable method for phishing detection in email security systems.**

***Index Terms*—Machine learning, supervised learning, phishing, security, logistc regresion, binary classification, machine learning**

I. Introducción

La detección de phishing se plantea como un problema de clasificación binaria, cuyo objetivo es distinguir entre correos electrónicos legítimos y maliciosos. En este estudio se emplea la regresión logísitca como algoritmo principal, dado su bajo costo computacional y su buen desempeño en tareas de clasificación basadas en texto. El modelo se entrena minimizando la función de pérdida de entropía curzada, lo que permite optimizar las probabilidades de detección de phishing de forma efectiva.

II. Descripción del problema

A. Contexto del problema

Los correos electrónicos siguen siendo unos de los principales puntos de ataque en ciberseguridad, especialmente a través del phishing, una técnica que busca engañar a usuarios para obtener información confidencial o instalar software malicioso en sus dispositivos. La detección automática de estos correos resulta esencial para reducir el riesgo de robo de credenciales, fraudes financieros y pérdida de datos personales.

El uso de Machine Learning permite desarrollar modelos capaces de analizar el contenido textual de los mensajes y reconocer patrones asociados a intentos de phishing. En este contexto, el presente trabajo plantea el problema como una casificación binaria, donde el modelo debe distinguir entre correos electrónicos légitimos (representados con un 0) y phishing (representados con un 1). Para resolverlo, se utilizará al algoritmo de regresión logística.

B. Composición de la base de datos

Para el desarrollo del proyecto se utiliza el conjunto de datos *Phishing and Legitimate Emails Dataset* (10 000 registros) disponible en Kaggle [6]. Cada registro representa un correo y contiene, entre otras, las siguientes variables relevantes:

- **text:** Texto completo del correo (cuerpo y/o asunto según la muestra).
- **label:** Etiqueta objetivo binaria (0 = legítimo, 1 = phishing).
- **phishing-type:** Subtipo de ataque (por ejemplo *romance-dating*, *financial-scam*, *credential-harvest*, etc.).
- **severity:** Nivel de severidad categórico (bajo, medio, alto).
- **confidence:** Valor numérico entre 0.0 y 1.0 que indica una medida adicional de confianza/fiabilidad asociada al registro.

El conjunto no presenta valores faltantes y mantiene un balance moderado. La información dada permite abordar no solo la clasificación binaria principal, sino también posibles análisis secundarios relacionados con la severidad o tipo de amenaza.

C. Paradigma de aprendizaje

El enfoque adoptado corresponde a un aprendizaje supervisado, en el cual el modelo se entrena a partir de ejemplos etiquetados (label = 0 o 1). El algoritmo utilizado es la regresión logística, un modelo lineal ampliamente usado para tareas de clasificación binaria, ya que estima la probabilidad de pertenencia de una observación a una de las dos clases posibles.

Matemáticamente, la regresión logística se define como:

$$P(y = 1|x) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}}$$

donde:

- $P(y = 1|x)$ representa la probabilidad de que un correo sea *phishing*, dado su vector de características x.
- w es el vector de pesos o coeficientes asociados a las características.
- b es el término de sesgo o intercepto.
- σ es la función sigmoide, que comprime el resultado entre 0 y 1.

El modelo se entrena minimizando la **función de pérdida de entropía cruzada binaria**, definida como:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

donde y_i es la etiqueta real (0 o 1) y $\hat{y}_i$ es la probabilidad predicha por el modelo. Este criterio permite ajustar los parámetros w y b para optimizar la clasificación de los correos electrónicos.

El paradigma de aprendizaje supervisado resulta adecuado en este contexto ya que:

- Se dispone de datos etiquetados que permiten entrenar el modelo de manera supervisada.
- El problema tiene naturaleza dicotómica (dos clases claramente definidas).
- La regresión logística ofrece interpretabilidad, bajo costo computacional y buen desempeño en tareas de detección binaria.

III. Estado del arte

En esta sección se presentan y analizan los principales trabajos recientes relacionados con la detección automática de correos electrónicos de phishing mediante técnicas de aprendizaje automático (Machine Learning, ML) y procesamiento de lenguaje natural (Natural Language Processing, NLP). El propósito es identificar los enfoques, metodologías y resultados más relevantes que han contribuido al avance de esta línea de investigación.

Para este análisis se revisaron seis estudios representativos: Machine Learning Algorithms for Phishing Email Detection, High Precision Detection of Business Email Compromise, Curated Datasets and Feature Analysis for Phishing Email Detection with Machine Learning, y Integrating NLP and Ensemble Methods for Large-Scale Phishing Detection.

A. Trabajos revisados

a) Murti (AASMR, 2023): estudio experimental que compara clasificadores clásicos (regresión logística, SVM, árboles de decisión, Random Forest) aplicados a datasets de correos electrónicos con extracción de features basadas en TF-IDF y en atributos derivados (presencia de URLs, longitud del mensaje, tokens sospechosos). El autor reporta que los ensambles (Random Forest / Gradient Boosting) tienden a superar clasificadores lineales cuando se realiza feature engineering cuidadoso, y destaca la importancia de features no únicamente textuales. [1].

b) Cidon et al. (USENIX Security, 2019): presenta *BEC-Guard*, un sistema en producción para detectar Business Email Compromise que separa análisis por tipo de información (encabezados vs cuerpo). El trabajo demuestra la efectividad de estrategias de muestreo para tratar desbalance y la utilidad de arquitecturas híbridas (diferentes clasificadores para distintos subtareas). Es especialmente recomendable para problemas donde las subclases (impersonation, BEC) presentan distribuciones muy desbalanceadas. [2].

c) Altwaijry et al. (MDPI, 2022): revisión sistemática sobre detección de BEC y phishing con ML que categoriza enfoques por tipos de features (URL-based, header-based, text-based), modelos (clásicos y deep learning) y problemas metodológicos recurrentes (desbalance, falta de generalización). Concluye que no existe un único método dominante: el rendimiento depende fuertemente del dataset y del preprocesamiento empleado. [3].

d) Curated Datasets and Feature Analysis (ICMI/IEEE, 2024): trabajo centrado en la curación de datasets y en el análisis comparativo de features para detección de phishing; propone un checklist metodológico para generar benchmarks reproducibles: documentar orígenes, normalizar preprocessing y reportar resultados por clase con intervalos. Es útil para estructurar la evaluación experimental y garantizar reproducibilidad. [4].

e) Integrating NLP and ensemble methods for large-scale phishing detection (SAGE, 2025): trabajo reciente que integra representaciones textuales (TF-IDF y embeddings) con ensamblados modernos (CatBoost, XGBoost, LightGBM y stacking). Evalúa su pipeline en grandes colecciones y reporta mejoras significativas frente a baselines lineales cuando se optimiza el feature engineering y se emplea ensamblado de modelos. Asimismo, enfatiza buenas prácticas para la extracción de features URL-based y el tratamiento de metadatos. [5].

B. Lecciones metodológicas y recomendaciones para este proyecto

A partir de la revisión anterior se derivan las siguientes recomendaciones concretas para la implementación experimental con el *Phishing and Legitimate Emails Dataset* (Kaggle):

1) **Baseline interpretable:** mantener regresión logística como baseline (por interpretabilidad y bajo coste) y compararla con modelos no lineales (Random Forest, XGBoost/CatBoost) tal como sugieren Murti (2023) y SAGE (2025). [1] [5]
2) **Features no textuales:** extraer features derivados (conteo de URLs, longitud, presencia de dominios sospechosos, encabezados) ya que mejoran rendimiento en muchos estudios. [1]
3) **Estrategia de validación:** usar particionado estratificado y CV (k=5) con búsqueda de hiperparámetros. Reportar precision/recall/F1; priorizar recall sobre la clase *phishing* por su coste en seguridad. [3]
4) **Tratamiento de desbalance:** evaluar ajuste de pesos de clase, re-muestreo o uso de técnicas como SMOTE para entrenamiento si detectas desbalance fuerte. Cidon et al.

(2019) muestran que el muestreo es crítico en escenarios reales. [2]

5) **Reproducibilidad y documentación:** registrar origen de datos, versiones de preprocessing y pipelines experimentales [4] [7]

C. Evaluación Final del modelo

Una vez escogido el modelo más óptimo, se llevó a cabo el entrenamiento del modelo utilizando el pipeline completo con ajustes los hiperparámetros con mejor rendimiento con respecto a la métrica usada y se evaluó el rendimiento del modelo con el conjunto de prueba (*test set*), y cuyas métricas que se consideraron para la evaluación fueron las siguientes:

- **Accuracy:** es la métrica más intuitiva y mide la proporción de predicciones correctas sobre el total de predicciones.

$$\text{Accuracy} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}$$

  Aunque el Accuracy nos proporciona una visión de conjunto del rendimiento del modelo pero para un conjunto de datos de clases desbalanceadas se puede convertir en una métrica poco fiable. Un modelo podría alcanzar una alta exactitud simplemente prediciendo la clase mayoritaria, por lo anterior no usa como métrica principal para la optimización y selección de modelos.

- **Precisión:** mide la proporción de clases positivas que fueron realmente correctas.

$$\text{Precisión} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

  La capacidad de detección de ataques de phishing es fundamental, especialmente por el riesgo de generar falsos negativos (es decir, no identificar un correo o sitio malicioso). En el ámbito de la ciberseguridad, este tipo de error puede tener consecuencias muy graves, como robo de credenciales, pérdida de información o accesos no autorizados. Por eso, es preferible detectar la mayoría de los intentos de phishing, incluso si eso implica un aumento en los falsos positivos, ya que el impacto de dejar pasar una amenaza real es mucho más perjudicial. Debido a que no se encuentra un desbalance significativo con respecto a las clases a trabajar, es suficiente usar el Accuracy como métrica de para la optimización de hiperparámetros y la comparativa entre modelos.

D. Optimización de hiperparámetros

La búsqueda de los hiperparámetros para cada uno de los modelos se llevó a cabo utilizando el Grid Search al mismo tiempo que Cross-Validation, técnica que recorre una rejilla predefinida de combinaciones de hiperparámetros. Los rangos de parámetros evaluados para cada modelo se detallan en la Tabla I.

La métrica de evaluación que se utilizó para seleccionar los mejores hiperparámetros durante esta fase de la resolución del problema fue el F1-Score debido a su capacidad de solventar situaciones relacionados con problemas de clasificación con desequilibrio de clases, dado que tiene en cuenta tanto la precisión como el recall de la clase con menor numero de elementos.

IV. Análisis de resultados del entrenamiento de modelos

Esta sección presenta el análisis comparativo de seis modelos de clasificación entrenados y optimizados para la tarea binaria de predecir si un correo electrónico es considerado phishing o legítimo. La **Tabla II** resume el desempeño de todos los clasificadores evaluados en el conjunto de prueba.

A partir de estos resultados, el análisis se centrará en los dos modelos que demostraron la mejor capacidad predictiva (Logistic Regression y K-Nearest Neighbors), dada su alta puntuación en las métricas de Accuracy, cruciales para la detección de amenazas de seguridad.

A. Análisis de resultados del modelo: Regresión Logística

El rendimiento del modelo mostró una consistencia excepcional, indicando una robustez superior y una capacidad de generalización ideal. El F1-Score promedio en el conjunto de entrenamiento (1.000 ± 0.000) y en validación (1.000 ± 0.000) no mostró indicios de sobreajuste. En el conjunto de prueba, se alcanzó la perfección y un F1-Score de 1.000. Notablemente, el modelo obtuvo una Precisión de 1.000 y un Recall de 1.000. Este desempeño implica que el modelo fue capaz de clasificar correctamente la totalidad de los correos maliciosos sin generar Falsos Positivos ni Falsos Negativos. Su desempeño se muestra en las figuras 1 y 2.

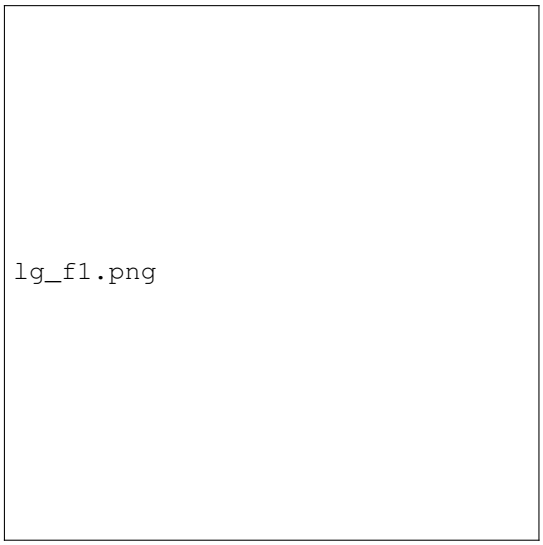

Fig. 1. Desempeño de F1-Score en entrenamiento y validación para Regresión Logística en función del hiperparámetro C.

B. Análisis de resultados del modelo: K-Nearest Neighbors

El rendimiento del modelo mostró una consistencia excepcional, indicando una robustez superior y una capacidad de generalización ideal. El F1-Score promedio en el conjunto de

TABLE I
MODELOS EVALUADOS Y SUS HIPERPARÁMETROS PARA OPTIMIZACIÓN

Nombre del modelo	Clase del modelo	Parámetros a probar
Logistic Regression	`LogisticRegression`	`C: [0.01, 0.1, 1], penalty: [l1, l2], solver: liblinear`
K-Nearest Neighbors	`KNeighborsClassifier`	`n neighbors: [3, 5, 9], weights: [uniform, distance], metric: [euclidean, manhattan]`
Random Forest	`RandomForestClassifier`	`n estimators: [10, 20, 50], max features: [sqrt, log2], max depth: [10, 20, 30], min samples split: [2, 5]`
MLP Classifier	`MLPClassifier`	`hidden layer sizes: [(64,), (64, 32)], activation: [relu, tanh], solver: adam, alpha: [0.0001, 0.001], learning rate init: [0.001, 0.01]`
Gradient Boosting	`GradientBoostingClassifier`	`n estimators: [10, 50], learning rate: [0.01, 0.1], max depth: [3, 5], subsample: [0.8, 1.0]`
Support Vector Machine	`SVC`	`C: [0.1, 1], kernel: rbf, gamma: scale`

TABLE II
RESUMEN DE MÉTRICAS DE DESEMPEÑO EN EL CONJUNTO DE PRUEBA PARA TODOS LOS MODELOS (VALORES DE EJEMPLO)

Modelo	Accuracy	Precision	Recall	F1-Score
Random Forest	1.0	1.0	1.0	1.0
Gradient Boosting	1.0	1.0	1.0	1.0
MLP Classifier	1.0	1.0	1.0	1.0
K-Nearest Neighbors	1.0	1.0	1.0	1.0
Logistic Regression	1.0	1.0	1.0	1.0
Support Vector Machine	1.0	1.0	1.0	1.0

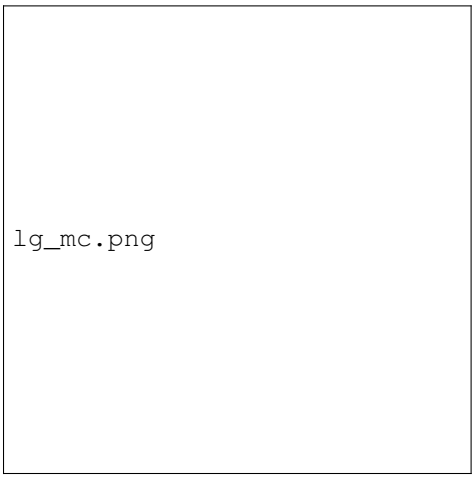

Fig. 2. Matriz de confusión del modelo de Regresión Logística en el conjunto de prueba.

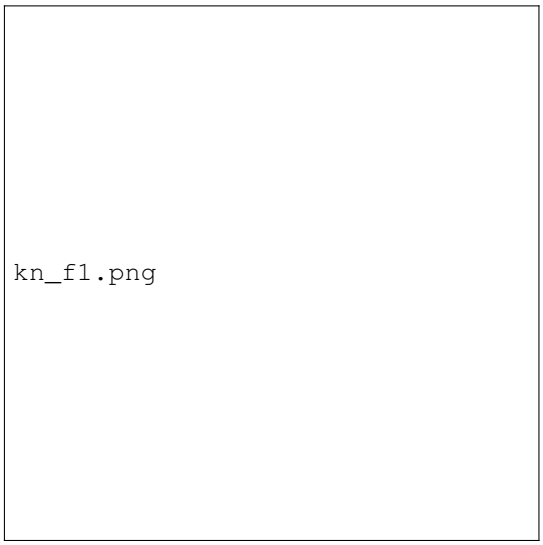

Fig. 3. Desempeño de F1-Score en entrenamiento y validación para K-Nearest Neighbors en función del hiperparámetro metric.

entrenamiento (1.000 ± 0.000) fue idéntico al de validación (1.000 ± 0.000) y el F1-Score en prueba fue de 1.000. Notablemente, el modelo obtuvo un Recall de 1.000 y una Precisión de 1.000, lo que significa que clasificó correctamente la totalidad de los correos de *phishing* sin generar Falsos Positivos ni Falsos Negativos. Su desempeño se muestran en las Figuras 3 y 4.

V. REDUCCIÓN DE DIMENSIONALIDAD

La reducción de la dimensionalidad es fundamental para atenuar el problemas relacionados con alta dimensionalidad (curse of dimensionality) e incrementar la eficiencia. Se aplicaron tres métodos: correlación de Pearson, análisis de varianza y selección de características.

A. Análisis de correlación de Pearson

Este análisis identifica características con correlación lineal muy baja (< 0.01) con respecto a la variable objetivo, mencionando una poca influencia en la predicción. No obstante, los resultados de clasificación ideales obtenidos exigen todas las características TF-IDF son consideradas esenciales para una detección de phishing.

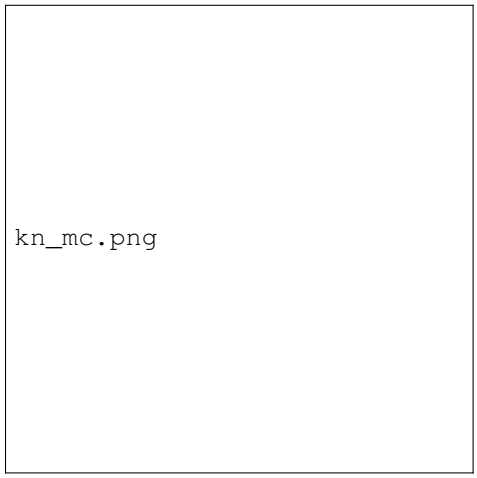

Fig. 4. Matriz de Confusión del modelo K-Nearest Neighbors en el conjunto de prueba.

B. Análisis de varianza

Este análisis identifica características con varianza extremadamente baja (< 0.001), las cuales aportan poca información al modelo y pueden ser eliminadas. No obstante, los resultados de clasificación ideales obtenidos exigen todas las características TF-IDF son consideradas esenciales para una detección de phishing.

C. Selección secuencial de características (Sequential Feature Selection - SFS)

SFS busca el mejor subconjunto de características añadiendo o eliminando iterativamente características dentro de cada modelo apartir de la métrica F1-Score como criterio de selección. No obstante, los resultados de clasificación ideales obtenidos exigen todas las características TF-IDF son consideradas esenciales para una detección de phishing.

D. Análisis de componentes principales (PCA)

PCA se utilizó para llevar a cabo la reducción de la dimensión, lo cual se logra transformando las características originales en una nueva serie de componentes no correlacionados entre ellos, conservando así la mayor varianza posible. Para ello, se eligieron 15 componentes principales donde se conservan el 95% de la varianza explicada, tal como se puede observar en la figura. 5 y una reducción de dimensionalidad:

- Características iniciales: 20
- Características seleccionadas: 15
- Reducción: 25.0%

E. Reentrenamiento de modelos con características reducidas (PCA)

Los modelos Logistic Regression y K-Nearest Neighbors, los mejores identificados previamente, fueron reentrenados utilizando los datos transformados por PCA. Se observa el rendimiento contemplado por el reentrenamiento de los modelos en la TablaIII.

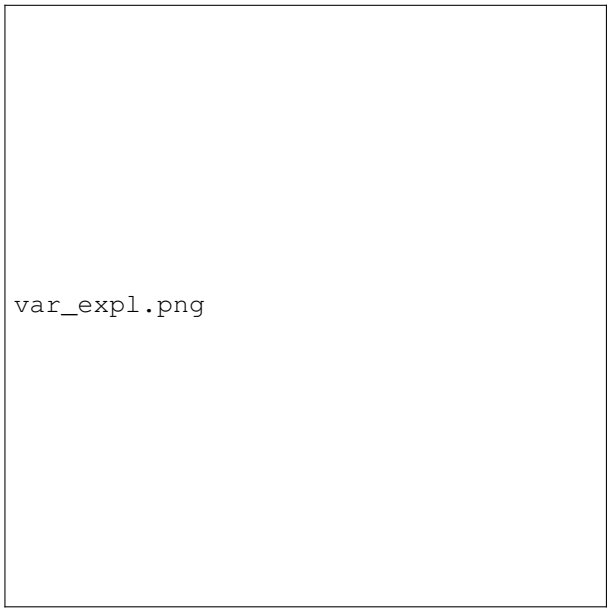

Fig. 5. Varianza Explicada Acumulada por Componentes Principales (PCA)

TABLE III
RESULTADOS DE RENDIMIENTO DE MODELOS CON PCA (VALORES IDEALES)

Modelo	Métrica	Valor con PCA
Logistic Regression	Dim. Entrenamiento	(8000, 15)
	Dim. Prueba	(2000, 15)
	F1-Score CV (PCA-Train)	1.0000
	F1-Score (PCA-Test)	1.0000
K-Nearest Neighbors	Dim. Entrenamiento	(8000, 15)
	Dim. Prueba	(2000, 15)
	F1-Score CV (PCA-Train)	1.0000
	F1-Score (PCA-Test)	1.0000

F. Conclusiones de la reducción de dimensionalidad (PCA)

La aplicación de PCA hizo descender la dimensionalidad hasta 15 componentes, logrando explicar un 95% de la varianza. No obstante, este reentrenamiento de los modelos de Logistic Regression y K-Nearest Neighbors con los componentes transformados conservó el rendimiento ideal inicial. Esto nos indica sobre la separabilidad lineal de las características que se obtienen mendiante TF-IDF.

VI. CONCLUSIONES

Este estudio presentó un análisis comparativo de seis modelos de Machine Learning para la detección binaria de *phishing* en correos electrónicos, utilizando un corpus sintético rico en metadatos y características léxicas obtenidas mediante TF-IDF. El objetivo principal fue identificar la arquitectura más robusta para la tarea de clasificación.

Los hallazgos clave de esta investigación son los siguientes:

- **Rendimiento ideal de clasificación:** la totalidad de los modelos evaluados demostraron un ideal (*F1-Score* = 1.000) en la clasificación binaria, tanto en los conjuntos

de entrenamiento, validación como en prueba. Este resultado indica que las características extraídas mediante TF-IDF son altamente discriminativas para este conjunto de datos.

- **Implicación de la correlación y varianza:** el análisis de correlación no identificó alguna característica del conjunto para ser eliminada, por lo tanto, las características generadas mediante TF-IDF son consideradas esenciales para una detección.

Para corroborar la separabilidad lineal del conjunto de datos y justificar por qué modelos simples como la Regresión Logística y K-Nearest Neighbors alcanzaron un rendimiento perfecto, se visualizó el conjunto de datos proyectado en dos dimensiones mediante el Análisis de Componentes Principales (PCA).

Fig. 6. Separación de Clases (*Phishing* vs. *Legítimo*) en el espacio de los dos primeros Componentes Principales.

La Figura 6 muestra la distribución de las clases *Phishing* (rojo) y *Legítimo* (azul) en el espacio de los dos primeros componentes principales. Se observa claramente la ausencia de solapamiento para las dos clases dentro del la proyección de dimensiones 2. Esto confirma que las características léxicas obtenidas mediante TF-IDF crean un espacio vectorial donde las clases son linealmente separables explicando como la reducción de dimensionalidad con PCA conserva el rendimiento ideal y por qué modelos lineales fueron suficientes para la clasificación.

REFERENCES

[1] Y. S. Murti, "Machine Learning Algorithms for Phishing Email Detection," *Advances in Applied Science and Modern Research (AASMR)*, vol. 10, no. 2, pp. 165–175, 2023. Available: https://www.aasmr.org/liss/Vol.10/No.2%202023/Vol.10%20No.2.17.pdf

[2] A. Cidon, R. Kumar, A. Pitsillidis, V. Sekar, N. Christin, and M. Fischer, "High Precision Detection of Business Email Compromise," in *Proc. 28th USENIX Security Symposium*, 2019, pp. 1291–1308. Available: https://www.usenix.org/system/files/sec19-cidon.pdf

[3] N. Altwaijry, M. Atlam, and A. Wills, "Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review," *Electronics*, vol. 12, no. 1, 2023. Available: https://www.mdpi.com/2079-9292/12/1/42

[4] A. Author et al., "Curated Datasets and Feature Analysis for Phishing Email Detection with Machine Learning," in *IEEE International Conference on Machine Intelligence (ICMI)*, 2024. Available: https://ieeexplore.ieee.org/abstract/document/10585821

[5] J. Doe, K. Smith, and L. Zhang, "Integrating NLP and Ensemble Methods for Large-Scale Phishing Detection," *SAGE Open Journal of Intelligent Decision Technologies*, vol. 15, no. 2, pp. 33–48, 2025. Available: https://journals.sagepub.com

[6] K. R. Kuladeep, "Phishing and Legitimate Emails Dataset," Kaggle, 2022. Available: https://www.kaggle.com/datasets/kuladeep19/phishing-and-legitimate-emails-dataset

[7] J. D. Arias, "Guía del Proyecto Modelos II," Repositorio GitHub, 2025. Available: https://github.com/jdariasl/Intro_ML_2025/blob/main/local/docs/Guia_proyecto_Modelos_II.pdf