



Evaluación Comparativa de Modelos de Aprendizaje Automático para la Clasificación de Tráfico de Red utilizando Redes Neuronales Densas y Regresión Logística Multiclasa sobre CIC-IDS2017

Presentan:

1. Alyson Melissa Sánchez Serratos
2. Miguel Ángel Pérez Ávila

18 de septiembre, 2025
Ciudad de México, México



Contenidos

- Introducción
- Problemática
- Marco Teórico
- Metodología
- Resultados
- Conclusiones



Introducción

- El objetivo principal es analizar el rendimiento de ambos enfoques en la clasificación multiclas de tráfico de red, empleando métricas como precisión, recall, F1-score y exactitud. Los resultados buscan aportar evidencia empírica sobre la capacidad predictiva de los modelos, ofreciendo una base para el diseño de IDS (Sistema de Detección de Intrusiones)basados en ML que sean precisos y factibles de implementar en entornos reales.



Problemática

- La ciberseguridad en redes busca proteger activos de información asegurando confidencialidad, integridad y disponibilidad. El espectro de amenazas incluye ataques de denegación de servicio (DoS/DDoS), infiltraciones, fuerza bruta y la propagación de malware.
- La creciente sofisticación y el volumen de las ciberamenazas representan un desafío permanente, exigiendo mecanismos de defensa proactivos y adaptativos para la protección de sistemas interconectados.



Marco Teórico - ML en Ciberseguridad

- Los Sistemas de Detección de Intrusos (IDS) se clasifican en:
 - Basados en firmas: operan mediante la comparación de la actividad observada con una base de datos de patrones de ataques conocidos.
 - Basados en anomalías: construyen un modelo del comportamiento normal esperado del sistema o red. Cualquier desviación significativa de este perfil basal es señalada como potencialmente maliciosa
- La investigación actual se centra en el empleo de técnicas de ML para refinar la precisión de los modelos de detección de anomalías.



Marco Teórico - Regresión Logística Multinomial

- La Regresión Logística Multinomial generaliza la regresión logística binaria a clases mediante la función softmax, que proyecta los datos de entrada en un espacio de probabilidades, donde cada componente indica la probabilidad asociada a una categoría.



Marco Teórico - Redes Neuronales Densas

- Las Redes Neuronales Densas son modelos de aprendizaje no lineales conformados por múltiples capas de neuronas completamente conectadas. Estas permiten transformar las características de entrada en representaciones internas de alta complejidad, capaces de capturar patrones y relaciones no lineales que facilitan la separación de clases complejas en problemas de clasificación multiclas.



Marco Teórico - Conjunto de Datos CIC-IDS2017

- Desarrollado por el Canadian Institute for Cybersecurity.
- Conjunto de datos diseñado para reflejar de manera fiel el tráfico de red contemporáneo y los ataques más comunes. Este conjunto contiene tanto tráfico benigno como ataques reales capturados durante cinco días de operación continua (3–7 de julio de 2017).
- Un conjunto de datos altamente estructurado de 2,830,743 flujos con 78 características.



Marco Teórico - Conjunto de Datos CIC-IDS2017

- Para el presente estudio, se emplea la versión preprocesada y curada disponible en Kaggle.
- Dicha versión incorpora un conjunto de transformaciones sistemáticas orientadas a mejorar la integridad, la representatividad y la utilidad analítica del dataset original.
- El conjunto de datos resultante comprende 52 características de ingeniería de tráfico que encapsulan propiedades fundamentales del comportamiento de red.



Metodología - Exploración de los Datos

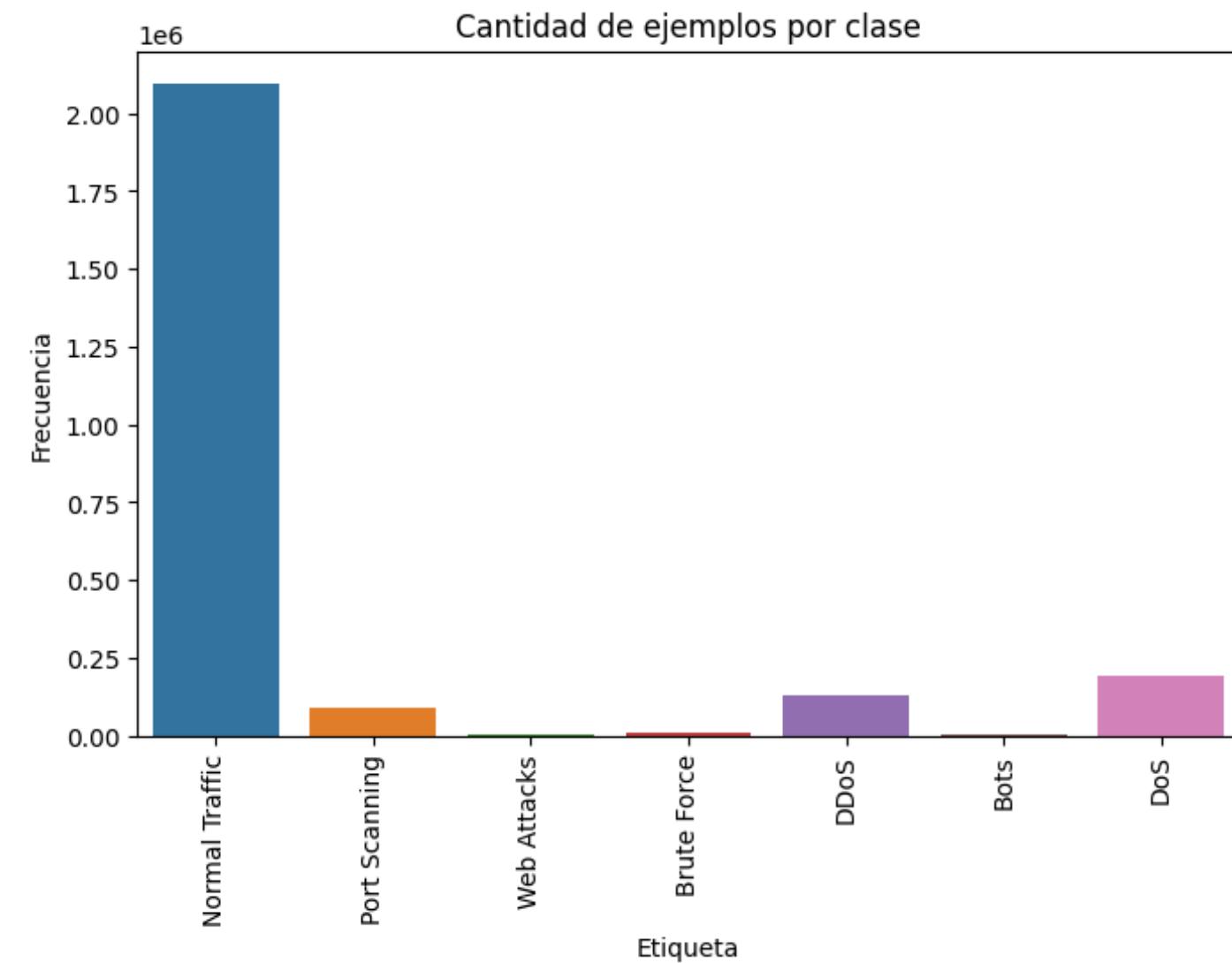
- Distribución de clases:

TABLA I

DISTRIBUCIÓN DE CLASES CIC-IDS2017 PREPROCESADO

Tipo de Tráfico	Instancias
Normal Traffic	2,095,057
DoS	193,745
DDoS	128,014
Port Scanning	90,694
Brute Force	9,150
Web Attacks	2,143
Bots	1,948

Fig. 1 Distribución de Clases CIC-IDS2017 Preprocesado





Metodología - Preparación de Datos

Para abordar la disparidad y favorecer la generalización de los modelos, se aplicó un muestreo aleatorio de cada categoría, ajustando todas las clases al tamaño de la categoría con menor número de muestras.

TABLA II
DISTRIBUCIÓN DE CLASES CIC-IDS2017 BALANCEADAS

Clase	Número de Muestras
Bots	1,948
Brute Force	1,948
DDoS	1,948
DoS	1,948
Normal Traffic	1,948
Port Scanning	1,948
Web Attacks	1,948



Metodología - Preparación de Datos

- Se construyó una matriz de correlación absoluta entre las 52 variables numéricas para examinar de manera cuantitativa las relaciones lineales existentes entre los atributos.
- Esta matriz permitió identificar grupos de variables altamente correlacionadas, evidenciando redundancias potenciales y señalando oportunidades de reducción de dimensionalidad.

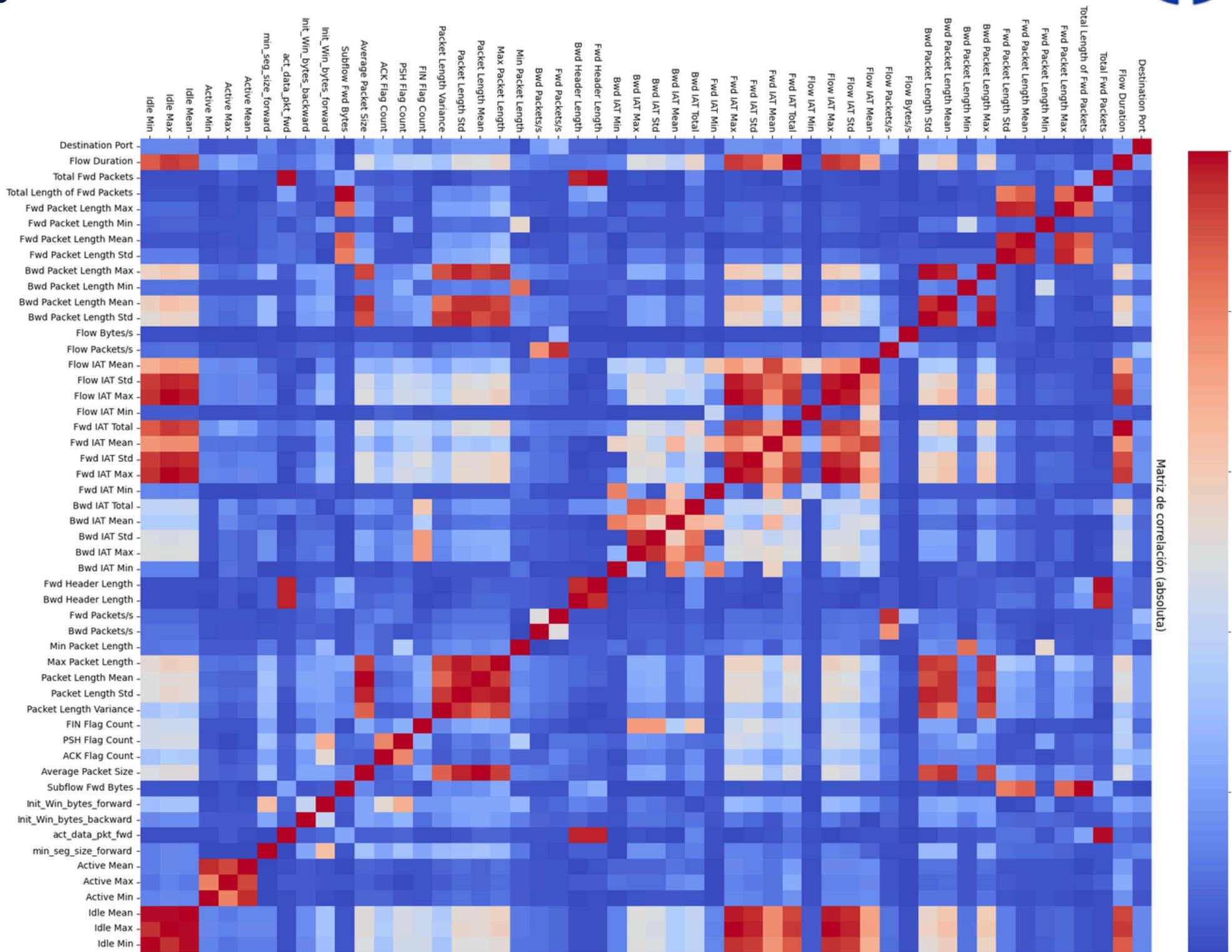
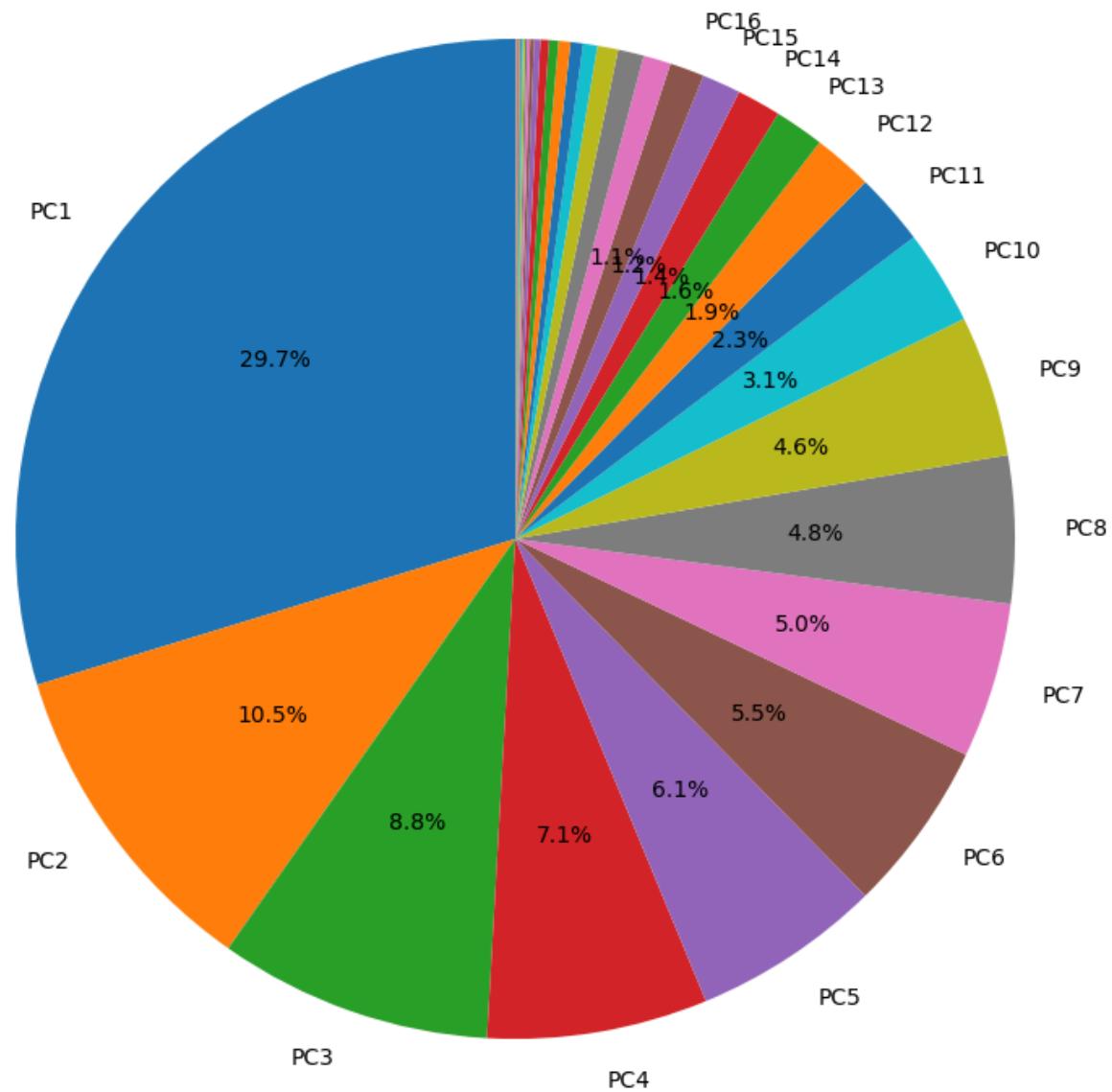


Fig. 2 Matriz de Correlación entre las 52 Características del Flujo de Red

Metodología - Preparación de Datos

Fig. 3 Proporción de Varianza Explicada (explained variance ratio) por cada Componente Principal



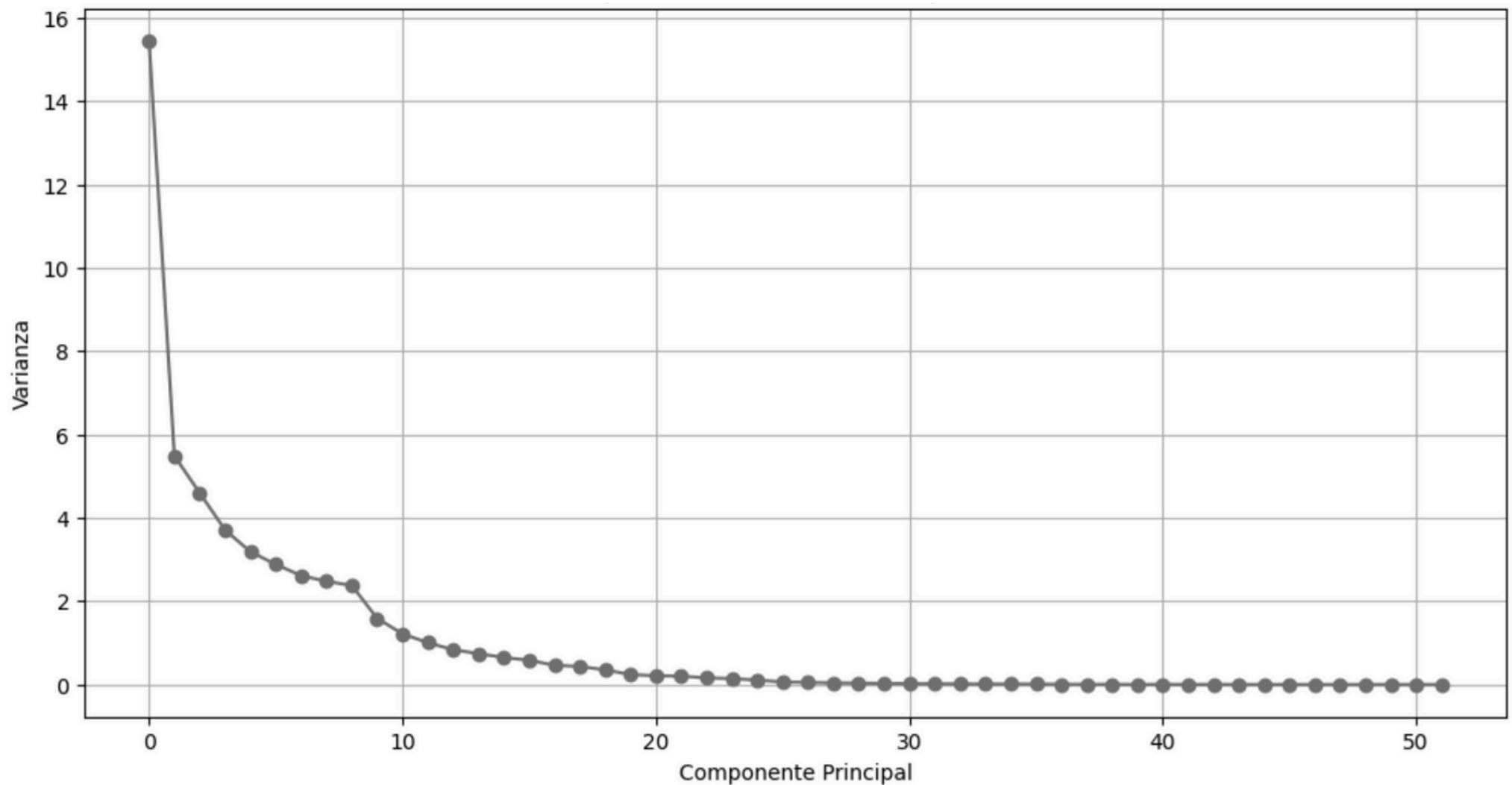
- Para abordar la reducción de dimensionalidad, se empleó PCA para transformar el conjunto original de 52 variables correlacionadas en componentes linealmente independientes, priorizando aquellos que capturan la mayor varianza del conjunto de datos.



Metodología - Preparación de Datos

- A partir de este análisis se seleccionaron los 10 primeros componentes principales, los cuales explican aproximadamente el 85 % de la varianza total.

Fig. 4 Varianza Explicada por Componente Principal (valor absoluto)





Metodología - Preparación de Datos

- Finalmente, se procedió a la separación de los datos en conjuntos de entrenamiento y prueba, reservando un 20 % de las muestras para evaluación y manteniendo la proporción de clases mediante estratificación. Los conjuntos resultantes mostraron un balance consistente, con 10,908 muestras en entrenamiento y 2,728 en prueba, asegurando una representación equitativa de las siete clases en ambas particiones.



Metodología - Modelación Regresión Logística

- La Regresión Logística se implementó empleando la clase LogisticRegression de Scikit-Learn configurada en modo multinomial para modelar simultáneamente todas las clases mediante la activación softmax.
- La búsqueda de hiperparámetros se llevó a cabo con GridSearchCV en una malla diseñada para explorar de forma sistemática los grados de regularización, el método numérico de optimización, la tolerancia a la convergencia y el tratamiento del balance de clases.



Metodología - Modelación Regresión Logística

- La siguiente tabla resume el espacio de búsqueda de hiperparámetros evaluado. En total, se exploraron 60 configuraciones distintas:

TABLA III
HIPERPARÁMETROS EXPLORADOS EN LA OPTIMIZACIÓN DEL MODELO DE REGRESIÓN LOGÍSTICA

Parámetro	Valores considerados
C	{0.01, 0.1, 1, 10, 100}
Penalización	l2
Solver	{newton-cg, saga}
Max_iter	{100, 500, 1000}
Class_weight	{None, balanced}



Metodología - Modelación Redes Neuronales Densas

- La implementación de las Redes Neuronales Densas se realizó utilizando la biblioteca Keras sobre TensorFlow.
- Se definieron tres arquitecturas diferenciadas en profundidad y número de neuronas por capa, con el fin de evaluar el impacto de la complejidad de la red en el rendimiento predictivo.
- Cada red fue definida como un modelo secuencial (Sequential), en el cual las capas se agregan de manera lineal desde la capa de entrada hasta la capa de salida.



Metodología - Modelación Redes Neuronales Densas

- El optimizador seleccionado fue Adam, con una tasa de aprendizaje de 0.01, por su robustez frente a variaciones en la escala de las características y su capacidad para adaptarse de manera dinámica al gradiente durante la optimización.
- La función de pérdida empleada fue categorical crossentropy, adecuada para problemas de clasificación multiclase y consistente con la codificación one-hot de las etiquetas.
- El tamaño del batch se ajustó entre 500 y 1000, evaluando su influencia sobre la estabilidad de la convergencia y la eficiencia del entrenamiento.



Metodología - Modelación Redes Neuronales Densas

- El optimizador seleccionado fue Adam, con una tasa de aprendizaje de 0.01, por su robustez frente a variaciones en la escala de las características y su capacidad para adaptarse de manera dinámica al gradiente durante la optimización.
- La función de pérdida empleada fue categorical crossentropy, adecuada para problemas de clasificación multiclase y consistente con la codificación one-hot de las etiquetas.
- El tamaño del batch se ajustó entre 500 y 1000, evaluando su influencia sobre la estabilidad de la convergencia y la eficiencia del entrenamiento.



Metodología - Modelación Redes Neuronales Densas

TABLA IV
ARQUITECTURAS Y PARÁMETROS DE ENTRENAMIENTO DE LAS
REDES NEURONALES DENSAS EVALUADAS

Modelo	Capas Ocultas	Batch Size	Épocas
RN1	20 – 10	500	50
RN2	20 – 30 – 20 – 10	1000	30
RN3	20	1000	30



Resultados - Regresión Logística

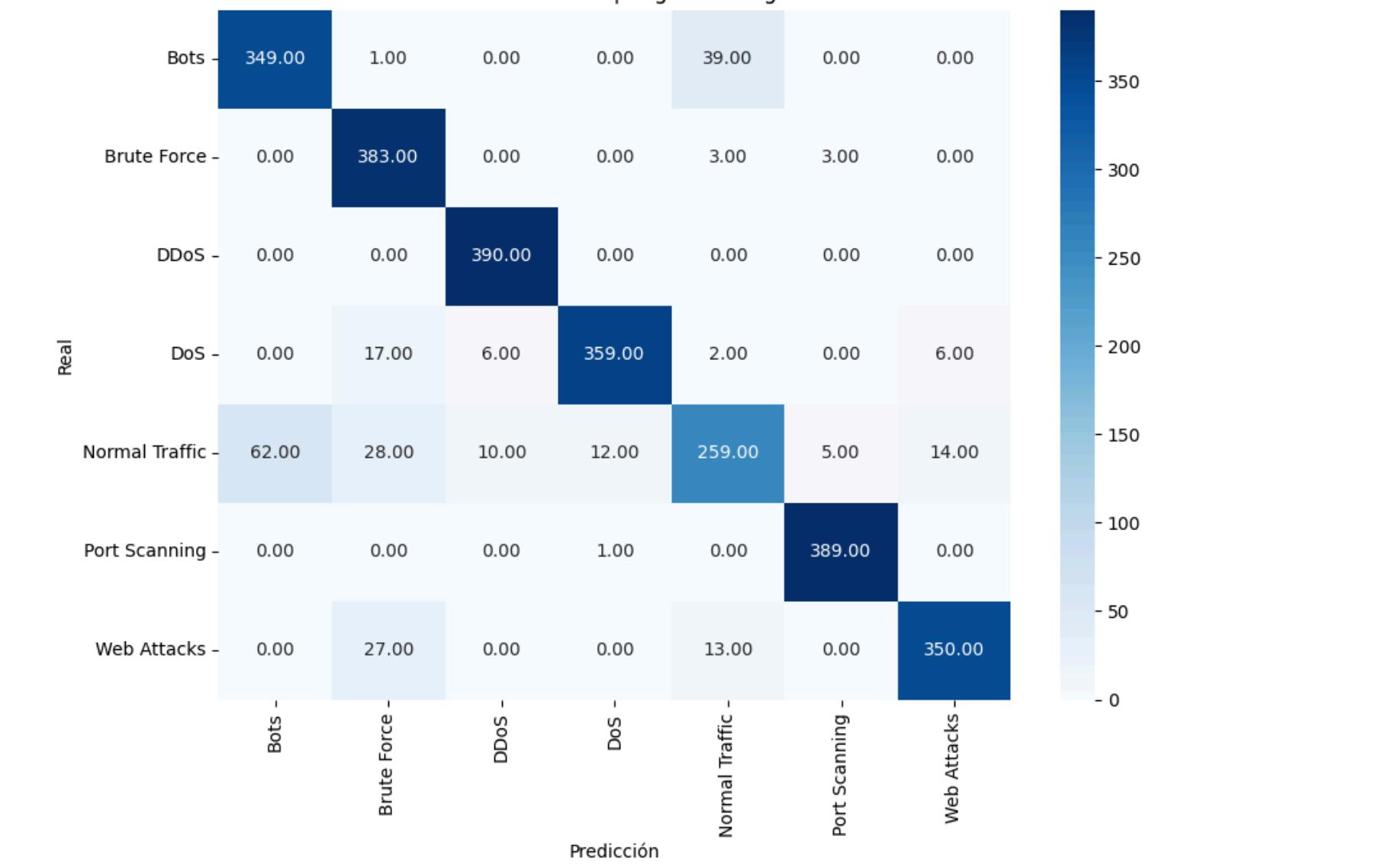
- El modelo de Regresión Logística fue optimizado mediante búsqueda de hiperparámetros, alcanzando los siguientes valores óptimos: un parámetro de regularización C de 100, un peso de clases balanceado para contrarrestar el desbalance en el conjunto de datos, un número máximo de iteraciones de 100, la utilización de la penalización L2 y el solucionador de tipo Newton-CG. La validación cruzada de 5 folds arrojó un score promedio de 0.908, indicando una capacidad de generalización robusta sobre distintas particiones del conjunto de datos.

Resultados - Regresión Logística

TABLA V
MÉTRICAS DE EVALUACIÓN DETALLADAS PARA REGRESIÓN LOGÍSTICA

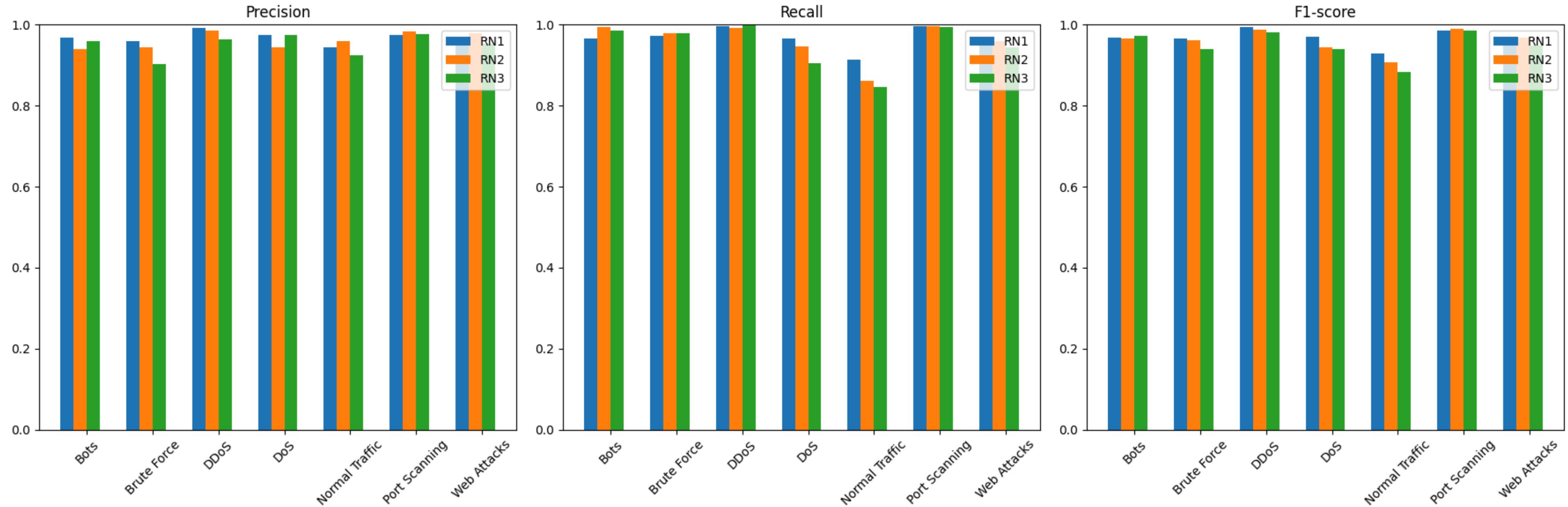
Clase	Precision	Recall	F1-score	Support
Bots	0.849	0.897	0.872	389
Brute Force	0.840	0.985	0.907	389
DDoS	0.961	1.000	0.980	390
DoS	0.965	0.921	0.942	390
Normal Traffic	0.820	0.664	0.734	390
Port Scanning	0.980	0.997	0.989	390
Web Attacks	0.946	0.897	0.921	390
Accuracy	0.909			
Accuracy		0.909		
Macro F1-score		0.906		
Weighted F1-score		0.906		

Fig. 7 Matriz de Confusión para Regresión Logística
Matriz de Confusión | Regresión Logística



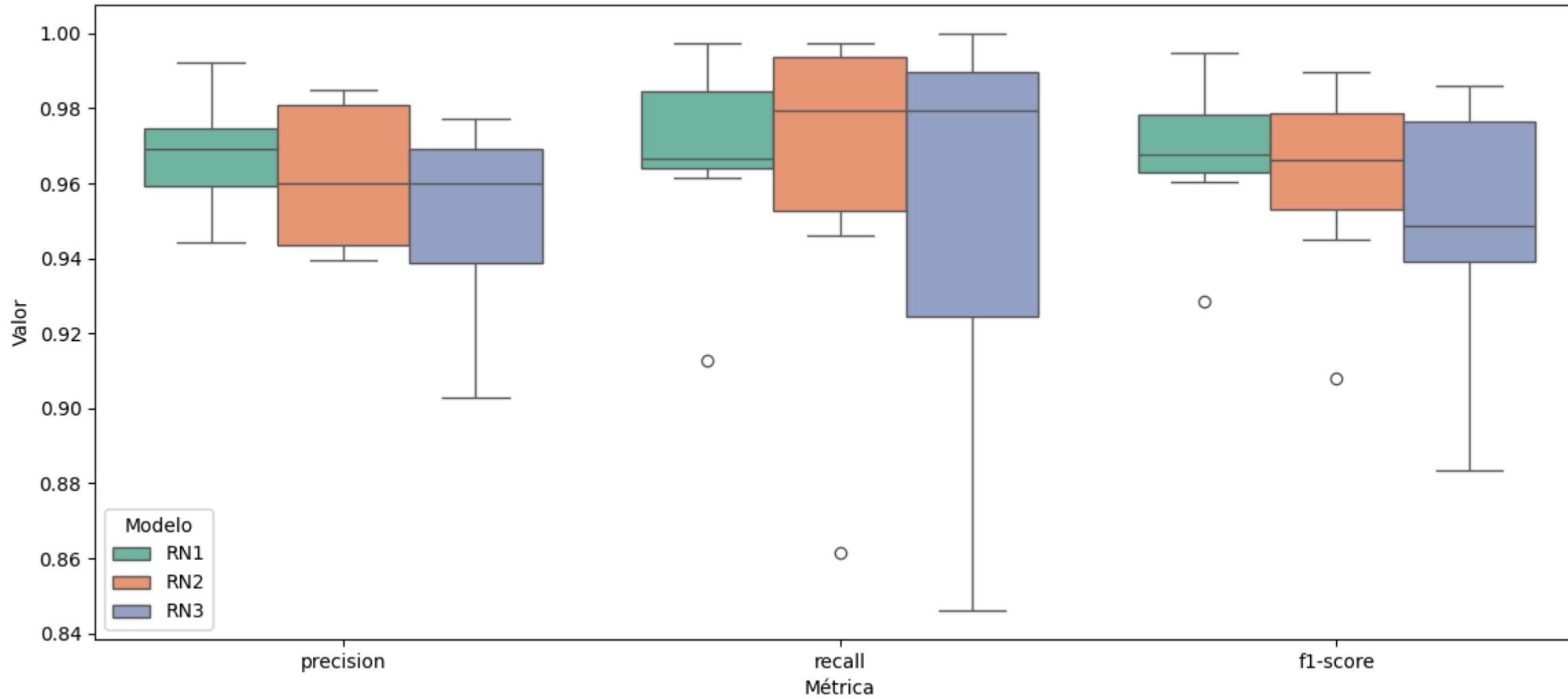
Resultados - R.N. Densas (3 Arquitecturas)

Fig. 8 Comparación de Métricas por Clase entre RN1, RN2 y RN3



Resultados - R.N. Densas (3 Arquitecturas)

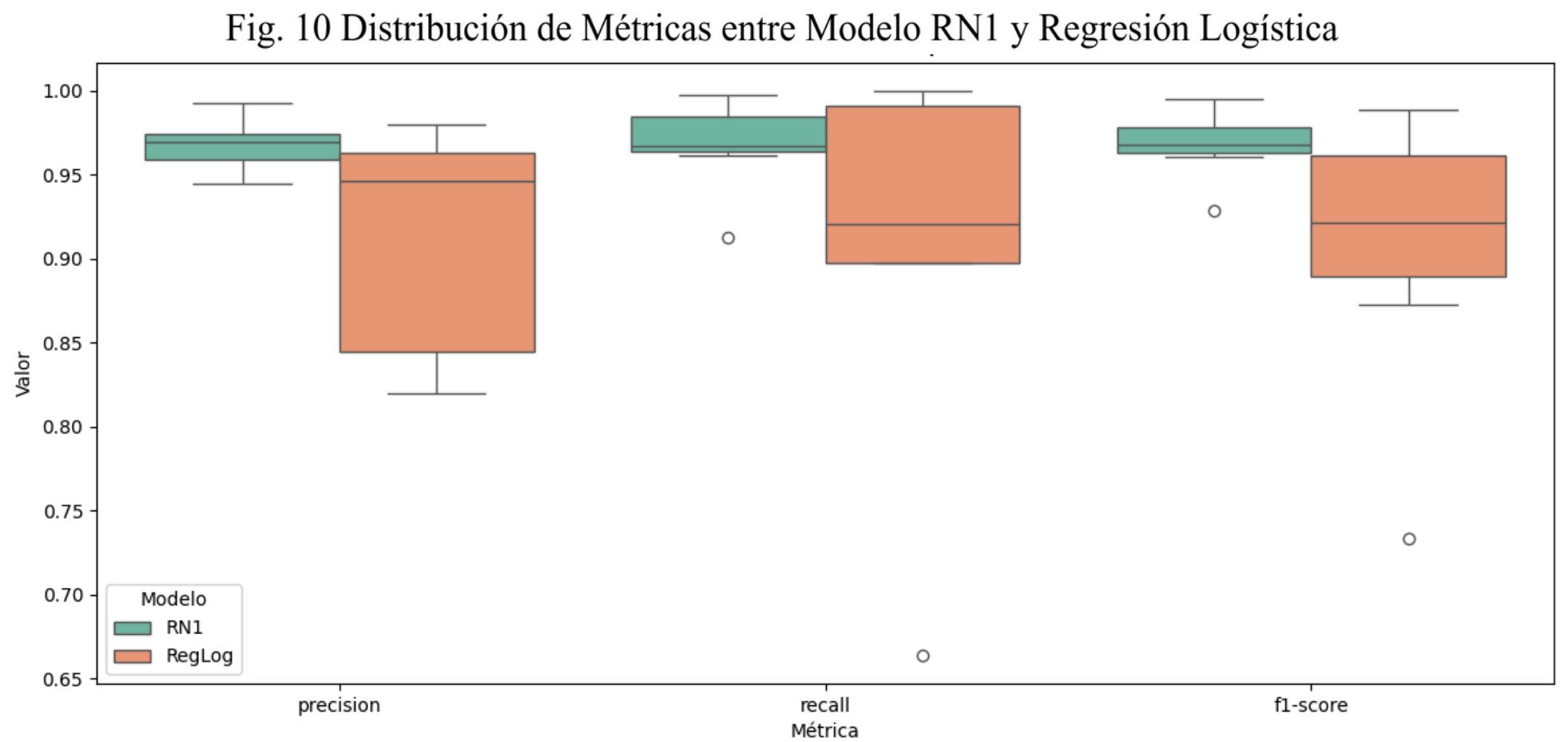
Fig. 9 Distribución de Métricas entre Modelos RN1, RN2 y RN3



Considerando de manera integral todas las métricas de desempeño obtenidas, se evidencia que la arquitectura RN1 alcanza el rendimiento óptimo entre las tres redes evaluadas.

Resultados - Comparación entre RegLog y RN1

En términos generales, RN1 de igual manera superó a la Regresión Logística en todas las métricas de desempeño, con un macro F1-score de 0.968 frente a 0.909 de la regresión logística, lo que representa una mejora absoluta de aproximadamente 5.9 puntos porcentuales.





Resultados - RN1

- RN1 se confirma como el modelo más adecuado para la clasificación de tráfico de red en este conjunto de datos, debido a su superioridad cuantitativa en métricas globales y por clase, a la reducción de confusiones en clases críticas y parcialmente solapadas, y a la mayor certeza en la asignación de probabilidades.
- La evaluación de generalización mediante validación cruzada K-Fold (k=5) para RN1 confirmó la estabilidad del modelo, con un accuracy promedio de 0.942 y macro F1 promedio de 0.940.



Resultados - RN1

TABLA VI

MÉTRICAS DE EVALUACIÓN PARA RN1

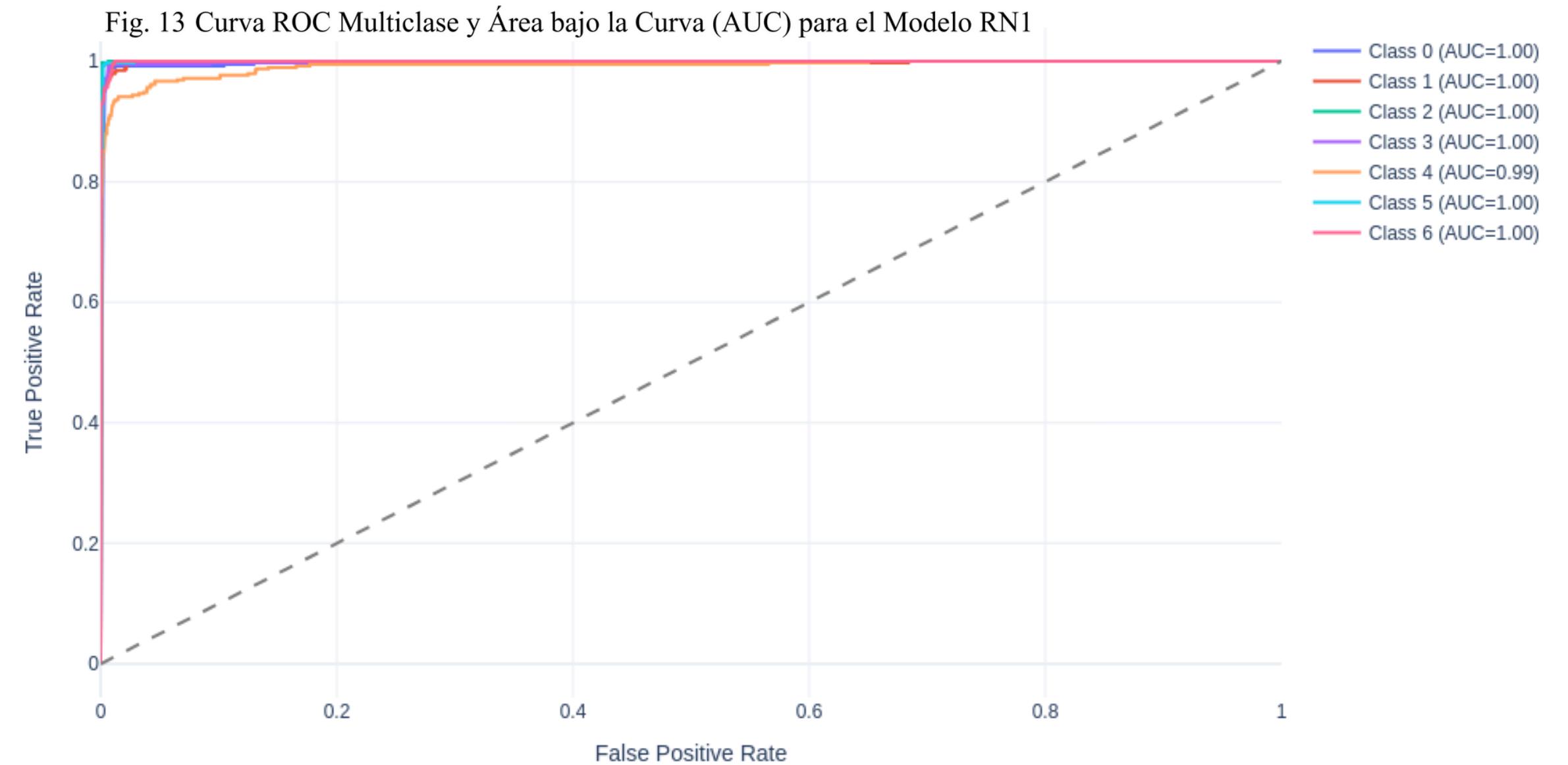
Clase	Precision	Recall	F1-score	Support
Bots	0.969	0.967	0.968	389
Brute Force	0.959	0.972	0.966	389
DDoS	0.992	0.997	0.995	390
DoS	0.974	0.967	0.970	390
Normal Traffic	0.944	0.913	0.928	390
Port Scanning	0.975	0.997	0.986	390
Web Attacks	0.959	0.962	0.960	390
Accuracy			0.968	
Macro F1-score			0.968	
Weighted F1-score			0.968	

Fig. 12 Matriz de Confusión para RN1

Matriz de Confusión | Primer Red Neuronal: 2 Capas Densas (20 y 10 neuronas respectivamente)



Resultados - RN1





Resultados - RN1

Fig. 14 Curva de Pérdida RN1

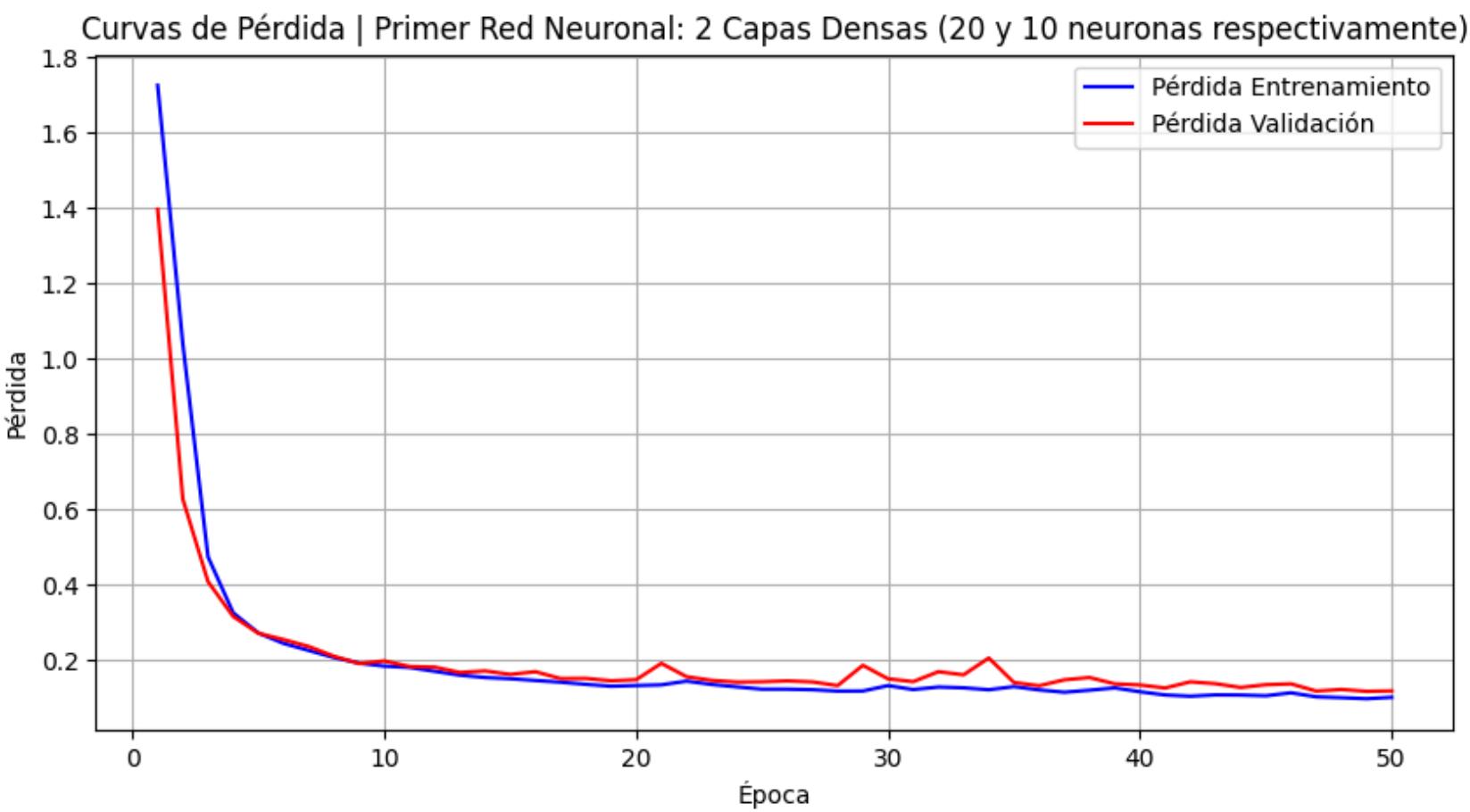
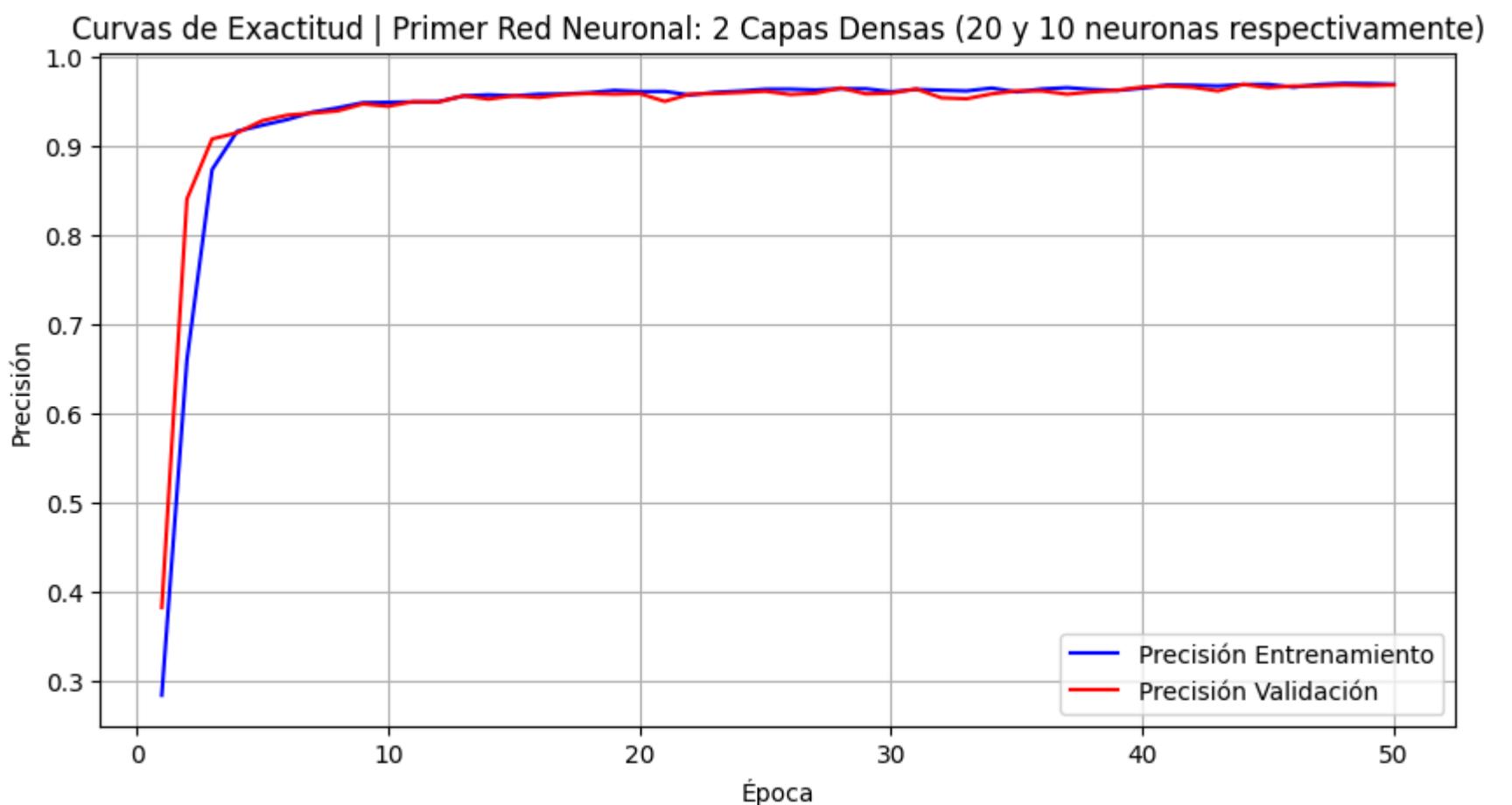


Fig. 15 Curva de Exactitud RN1





Conclusiones

En síntesis, la arquitectura RN1 de dos capas densas se establece como la opción más eficiente y confiable para la clasificación de tráfico de red en este conjunto de datos, combinando alta precisión, robustez frente a confusiones y capacidad de generalización, lo que resalta la ventaja de modelos de aprendizaje profundo de complejidad moderada frente a modelos lineales en escenarios de ciberseguridad de alta criticidad.



Reflexiones

- Alyson Sánchez

Trabajar en este proyecto me permitió comprender de primera mano los retos y sutilezas del aprendizaje automático aplicado a ciberseguridad. Experimentar con la preparación de datos, el balance de clases y la optimización de modelos me enseñó que no basta con aplicar algoritmos: la calidad de los resultados depende del cuidado en cada etapa del proceso. Comparar Regresión Logística con redes neuronales densas me hizo apreciar cómo las arquitecturas profundas pueden capturar patrones complejos que un modelo lineal no detecta, reforzando mi comprensión de la teoría detrás de cada decisión de diseño.



Reflexiones

- Miguel Pérez

Este trabajo me permitió comprender la importancia de comparar distintos enfoques de aprendizaje supervisado en la detección de tráfico de red. Mientras que la regresión logística ofrece rapidez y simplicidad, las redes neuronales resultaron más efectivas al capturar relaciones no lineales propias de los datos.

Además, la implementación de una interfaz gráfica evidenció que la utilidad de un sistema no depende solo de su precisión, sino también de su accesibilidad para los usuarios. En conjunto, esta experiencia fortaleció mi capacidad para diseñar, evaluar y justificar soluciones basadas en inteligencia artificial orientadas a la seguridad informática.



Recursos y Materiales del Proyecto

El informe técnico, el conjunto de datos y el código fuente utilizados en este trabajo están disponibles públicamente en el siguiente repositorio:

TGMAPA, “CIC-IDS2017 ML Analysis,” GitHub repository, 2025. [Online]. Available: https://github.com/TGMAPA/CIC-IDS2017_ML_Analysis



Referencias

- [1] I. Sharafaldin, A. H. Lashkari, y A. A. Ghorbani, "CICIDS2017: Intrusion Detection Evaluation Dataset," Canadian Institute for Cybersecurity, University of New Brunswick, 2017. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>
- [2] "Security Concept," ScienceDirect Topics. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/security-concept>
- [3] M. A. Elhadi y M. A. Ghorbani, "Classification and Importance of Intrusion Detection System," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/340655192_Classification_and_Importance_of_Intrusion_Detection_System
- [4] "Network Forensics: A Short Guide to Digital Evidence Recovery from Computer Networks," Forensic Focus, 15-Mar-2025. [Online]. Available: <https://www.forensicfocus.com/guides/network-forensics-a-short-guide-to-digital-evidence-recovery-from-computer-networks/>
- [5] D. Herzalla, W. T. Lunardi y M. A. Lopez, "TII-SSRC-23 Dataset: Typological Exploration of Diverse Traffic Patterns for Intrusion Detection," arXiv, 14-Sep-2023. [Online]. Available: <https://arxiv.org/abs/2310.10661>
- [6] İ. Yazıcı, "A survey of applications of artificial intelligence and machine learning in cybersecurity," Computers & Security, vol. 114, p. 102525, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2215098623001337>
- [7] T. Saranya, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection Systems," Procedia Computer Science, vol. 171, pp. 1289–1296, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920311121>
- [8] M. Sharafaldin, A. H. Lashkari y A. A. Ghorbani, "CSE-CIC-IDS2018: Intrusion Detection Evaluation Dataset," Canadian Institute for Cybersecurity, University of New Brunswick, 2018. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2018.html>
- [9] E. A. C. Ribeiro, "CICIDS2017: Cleaned & Preprocessed," Kaggle, 2023. [Online]. Available: <https://www.kaggle.com/datasets/ericanacletoribeiro/cicids2017-cleaned-and-preprocessed?resource=download>



Anexo - Entorno Productivo

- Se realizó la serialización de los modelos mediante la librería Pickle.
- La entrada de datos consiste en 52 variables descriptivas del tráfico de red, que pueden suministrarse manualmente mediante un formulario interactivo o mediante la carga de archivos de texto .txt a una interfaz gráfica desarrollada en Tkinter.
- La ejecución de predicciones se realiza mediante botones dedicados que invocan la función de predicción y generan automáticamente los gráficos de salida.



Anexo - Entorno Productivo

Fig. 5 Interfaz gráfica para Ingreso de Variables y Ejecución de Predicciones

Predicción de Tipo de Actividad en Tráfico de Red

Ingrese los valores de las 52 variables:

Destination Port:	
Flow Duration:	
Total Fwd Packets:	
Total Length of Fwd Packets:	
Fwd Packet Length Max:	
Fwd Packet Length Min:	
Fwd Packet Length Mean:	
Fwd Packet Length Std:	
Bwd Packet Length Max:	
Bwd Packet Length Min:	
Bwd Packet Length Mean:	
Bwd Packet Length Std:	
Flow Bytes/s:	
Flow Packets/s:	
Flow IAT Mean:	
Flow IAT Std:	
Flow IAT Max:	
Flow IAT Min:	
Fwd IAT Total:	
Fwd IAT Mean:	
Fwd IAT Std:	
Fwd IAT Max:	
Fwd IAT Min:	
Bwd IAT Total:	
Bwd IAT Mean:	
Bwd IAT Std:	
Bwd IAT Max:	
Bwd IAT Min:	
Fwd Header Length:	
Bwd Header Length:	

Predecir desde formulario Cargar archivo .txt con 52 variables

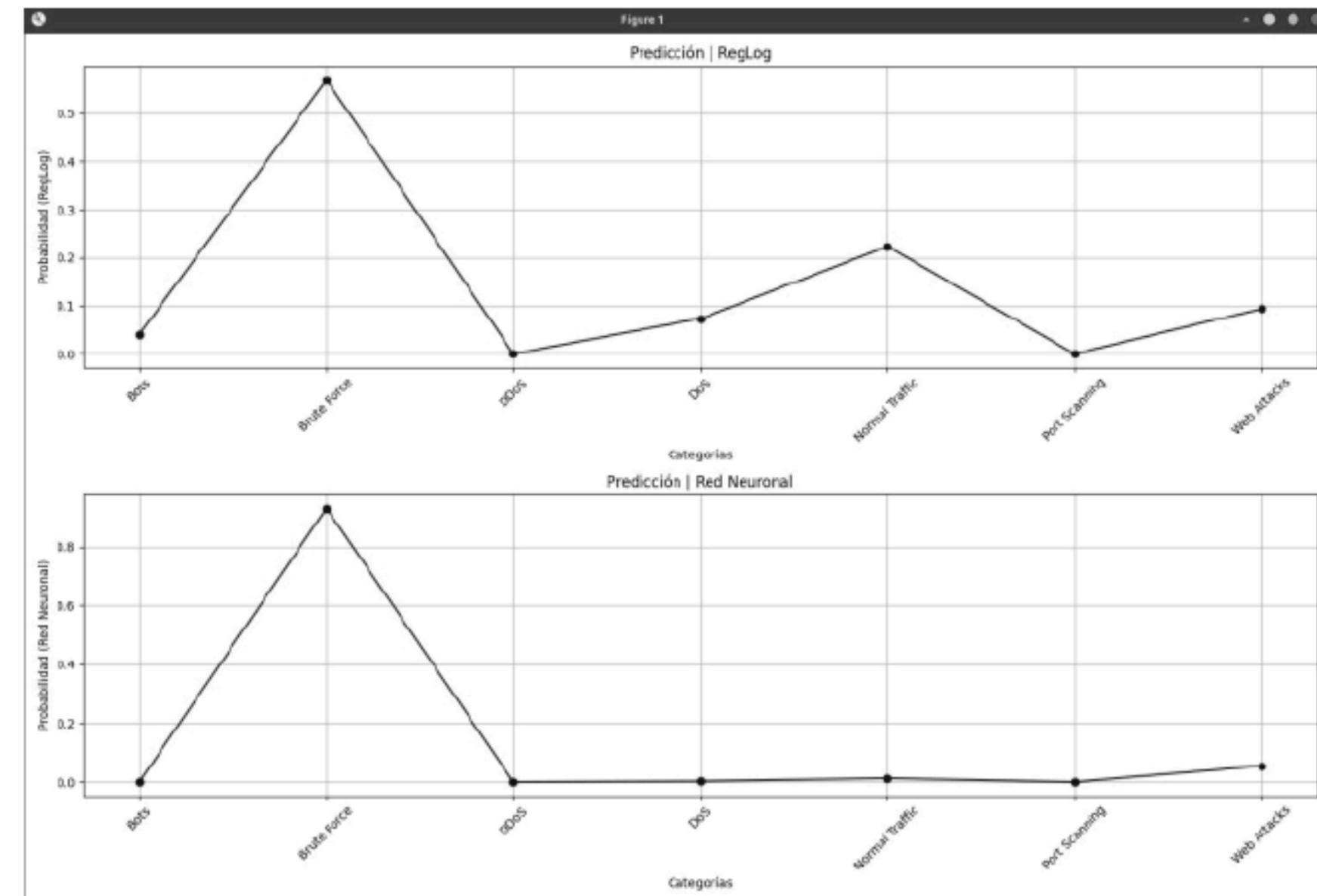


Fig. 6 Gráfico de Probabilidades de Clasificación por cada Modelo