Artículo Original

Redes Neuronales Densas y Redes Neuronales Convolucionales para la detección de fraudes en movimientos transaccionales

Dense Neural Networks and Convolutional Neural Networks for Fraud Detection in Transactional Movements

Cristian Guerrero Balber[1], Camilo Andrés Pulzara Mora[2], Juan David Losada Losada[3]

[1,2] Universidad Internacional de Valencia. Valencia, España.
[3] Facultad de Ciencias e Ingeniería. Universidad de Manizales, Manizales, Colombia.
*Autor de correspondencia: capulzaram@unal.edu.co

Resumen

En la actualidad, el fraude bancario afecta a empresas y usuarios de instituciones financieras, generando pérdidas económicas considerables que afectan la confianza en los sistemas de pago y en las transacciones electrónicas. Los métodos tradicionalmente utilizados para la detección de actividades fraudulentas, que se basan en reglas predefinidas y análisis manuales, son insuficientes para hacerle frente a un gran volumen de datos, que cada vez es más complejo. En el presente estudio se aplican técnicas y modelos de Deep Learning (DL) como redes neuronales densas (DNN) y redes neuronales convolucionales (CNN), con el objetivo de detectar transacciones de carácter fraudulento en el ámbito financiero, aportando así, a la seguridad de los sistemas bancarios en línea. La implementación en conjunto de los modelos DNN y CNN muestra un resultado positivo, con un AUC-ROC superior a 0,8 y una sensibilidad mayor al 80%. Lo cual implica que, los modelos permiten detectar un número considerable de estafas sin comprometer la exactitud global del sistema. Por lo tanto, la implementación de redes neuronales profundas, y su combinación con diferentes arquitecturas, puede ser una excelente alternativa para la detección de estafas y fraudes en el sistema bancario.
Palabras clave: Aprendizaje profundo; redes neuronales artificiales; redes neuronales densas; redes neuronales convolucionales; delitos informáticos.

Abstract

Currently, banking fraud affects both businesses and users of financial institutions, leading to significant economic losses that undermine trust in payment systems and electronic transactions. Traditional methods for detecting fraudulent activities, which rely on predefined rules and manual analysis, are insufficient to handle the increasing volume and complexity of data. This study applies Deep Learning (DL) techniques and models, such as Dense Neural Networks (DNN) and Convolutional Neural Networks (CNN), with the objective of detecting fraudulent transactions in the financial sector, thereby contributing to the security of online banking systems. The combined implementation of DNN and CNN models produced positive results, achieving an AUC-ROC greater than 0.8 and a sensitivity exceeding 80%. These results indicate that the models can detect a significant number of fraudulent activities without compromising overall system accuracy. Therefore, the implementation of deep neural networks, along with their integration with different architectures, represents a promising approach for fraud detection in the banking system.

Keywords: Deep learning; artificial neural networks; dense neural networks; convolutional neural networks; cybercrime.

1. Introducción

El fraude en los sistemas bancarios genera un temor cada vez mayor a nivel internacional impulsado constantemente por la expansión sostenida de los servicios digitales y la creciente globalización de las transacciones. En 2024, la asamblea general de la ONU adoptó la convención de las naciones unidas contra la ciberdelincuencia (A/RES/79/243). Este instrumento establece un marco jurídico internacional para prevenir, investigar y sancionar delitos informáticos, fomentando la cooperación entre Estados, la protección de los derechos humanos y la asistencia técnica a países en desarrollo. La convención busca armonizar legislaciones nacionales, fortalecer capacidades institucionales y promover un ciberespacio seguro, equilibrando la seguridad digital con el respeto a las libertades fundamentales.

En este contexto, el incremento en el número y valor de las transacciones en línea ha sido aprovechado por ciberdelincuentes que desarrollan estrategias sofisticadas para vulnerar los sistemas de seguridad de las plataformas digitales, así como las medidas de prevención adoptadas por los usuarios. Las actividades fraudulentas han proliferado en un entorno favorecido por el auge del comercio electrónico, la banca digital y las billeteras digitales. Entre las principales amenazas identificadas se encuentran el phishing, los esquemas Ponzi y piramidales, las estafas con criptomonedas, los ataques de tipo ransomware, el compromiso de correos electrónicos corporativos (BEC), los fraudes en compras o subastas en línea, y las estafas asociadas a servicios de soporte técnico [1,2].

Frente a este panorama, las entidades financieras han decidido incorporar nuevas tecnologías en sus estrategias para mitigar el riesgo de fraude. En particular, la inteligencia artificial (IA) y el aprendizaje automático (Machine Learning, ML) han demostrado ser herramientas eficaces para la detección temprana de actividades sospechosas. Este conjunto de técnicas permite procesar grandes volúmenes de datos en tiempo real, identificar patrones inusuales y predecir comportamientos asociados al fraude, con una precisión superior a los métodos tradicionales.

Una de las aplicaciones con mayor proyección en este ámbito es el aprendizaje profundo (Deep Learning, DL), una subdisciplina del ML que emplea redes neuronales profundas para incrementar la capacidad de detección de anomalías en entornos complejos. La evidencia empírica muestra que los modelos de DL resultan especialmente útiles en el análisis de transacciones financieras, al identificar con mayor precisión anomalías que podrían pasar inadvertidas mediante enfoques estadísticos convencionales [3-8]. Su aplicación en la detección de fraude presenta particularidades que exigen una atención especial, como el desbalance significativo entre clases (transacciones legítimas frente a fraudulentas) y la necesidad de minimizar tanto falsos positivos como negativos.

En este sentido, el uso de redes neuronales profundas en la detección de fraudes bancarios se justifica por la capacidad de estos modelos para aprender representaciones complejas de los datos y adaptarse a patrones emergentes. Su implementación contribuye no solo a mejorar la capacidad predictiva del sistema, sino también a reducir posibles pérdidas económicas resultado de actos delictivos. Por tanto, el presente estudio se centra en el diseño, entrenamiento y validación de modelos de DL orientados a la detección automatizada de fraudes en transacciones bancarias, con el fin de aportar soluciones efectivas y escalables en el contexto de la ciberseguridad financiera.

2. Materiales y Métodos

En este estudio se diseñaron y compararon dos arquitecturas de redes neuronales de distinta naturaleza: una Red Neuronal Profunda (DNN) y una Red Neuronal Convolucional (CNN), ambas orientadas a resolver el mismo problema. En primera instancia, se configuró la arquitectura de la

DNN y se entrenaron múltiples modelos variando parámetros como el dropout y el tipo de datos empleados (originales y aumentados). Asimismo, se evaluaron tres estrategias de tratamiento del desbalance de clases:

- Entrenamiento sin balanceo, manteniendo la distribución original y aplicando un class_weight con proporción 1:1 para simular la ausencia de ajuste.

- Entrenamiento con balanceo a nivel de red, mediante la asignación de pesos calculados con compute_class_weight de scikit-learn durante el entrenamiento.

- Entrenamiento con balanceo en la red y en el conjunto de datos, utilizando dos técnicas para generar instancias sintéticas de la clase minoritaria: i) CTGAN (Conditional Tabular Generative Adversarial Network), basada en redes generativas adversariales. ii) SMOTE (Synthetic Minority Over-sampling Technique), para sobre-muestreo sintético.

Para cada estrategia se entrenaron tres modelos con diferentes valores de dropout con el fin de reducir el sobreajuste, lo que resultó en un total de nueve modelos DNN. Estos se evaluaron mediante métricas clásicas (accuracy, recall, AUC) y el análisis de las curvas de pérdida en los conjuntos de entrenamiento, validación y prueba. [9]

Posteriormente, la configuración con mejor desempeño se replicó empleando una CNN, previa transformación de los datos tabulares en representaciones gráficas. Finalmente, se construyó un modelo híbrido DNN+CNN, combinando las salidas de ambas redes para mejorar la precisión predictiva. Los resultados obtenidos permitieron realizar un análisis comparativo y extraer conclusiones sobre el impacto del balanceo, la arquitectura y las técnicas de aumento en el rendimiento del modelo.

A. Descripción del dataset

El dataset utilizado fue seleccionado del conjunto de datos de fraude de cuentas bancarias (BAF) publicado en NeurIPS 2022. De acuerdo con Jesus et al [10], BAF es un banco de pruebas realista, completo y robusto que dispone un total de 6 conjuntos de datos ideales para evaluar métodos innovadores en aprendizaje automático justo (ML). Debido al volumen de datos de cada uno de los dataset, y teniendo en cuenta los objetivos de este estudio, solamente se seleccionó el dataset denominado Base.csv. Este dataset está conformado por un total de 32 variables entre numéricas, categóricas y binarias, y un millón de instancias que corresponden cada una a una solicitud financiera en línea. De las 32 variables, es de interés particular para la investigación, la variable fraud_bool, de carácter binario que permite identificar si una transacción es fraudulenta (1) o legítima (0). Los datos para fraud_bool son 988.971 para la clase 0 y 11.029 para la clase 1, lo que corresponde al 98,9% y 1,10%, respectivamente. Finalmente, el tratamiento de los datos, la visualización y el aprendizaje automático fueron realizados en librerías de Python, tales como Pandas, NumPy, Seaborn, Matplotlib y Scikit-learn.

Por otro lado, se identificó que el dataset no distingue entre variables numéricas y binarias, lo cual es relevante para aplicar tratamientos diferenciados. Por ello, se construyó un diccionario que asigna a cada variable su tipo correspondiente. Luego, se implementó una función para tratar los valores faltantes (representados por números negativos) según el tipo de variable:
Variables numéricas: Los valores negativos se reemplazan por cero, para evitar distorsiones al normalizar y preservar la ausencia de datos como posible indicador de fraude.
Variables binarias o categóricas: Los valores negativos se asignan a la clase menos representada, con el fin de no sesgar el análisis ni perder señales relevantes.

B. Distribución de densidad de los datos

La visualización de estimaciones de densidad por núcleo (KDE) en variables numéricas, diferenciadas por clase (0 y 1), permite evaluar su capacidad para discriminar entre categorías en un problema de clasificación. Si las distribuciones están bien separadas, se considera que la variable es un buen predictor. Por el contrario, si son similares su utilidad es limitada o requiere técnicas más complejas para extraer patrones relevantes. Este análisis se aplica únicamente a variables numéricas, ya que se basa en distribuciones continuas de probabilidad.

Por lo tanto, en la figura 1 se muestra la distribución de algunas variables numéricas en función de la variable de interés fraud_bool. Las transacciones legítimas se indican con color azul (0) y las fraudulentas de color verde (1).

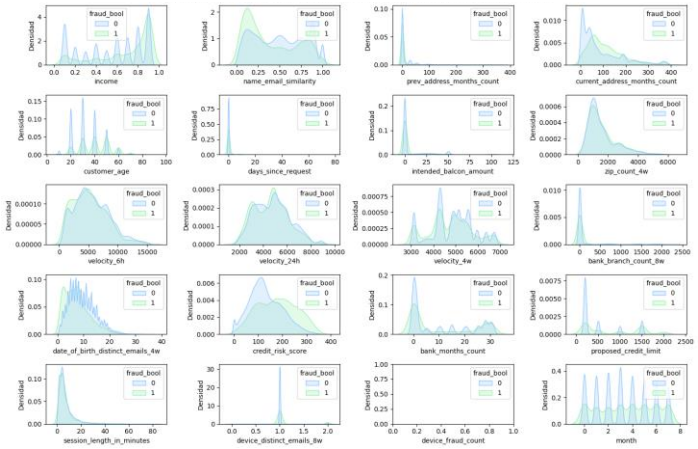

Fig.1 Densidades KDE por variable numérica diferenciada por clase 0 (legítima) y 1 (fraudulenta).

En la Figura 1 se observa una superposición entre la mayoría de las clases, lo que dificulta establecer un umbral claro para su separación. Sin embargo, en variables como name_email_similarity y credit_risk_score, el contraste entre clases es mayor, lo que sugiere que podrían ser más relevantes para la clasificación. Por el contrario, la similitud en las distribuciones de las clases para los campos session_length_in_minutes y customer_age sugiere que estas variables carecen de relevancia predictiva. Adicionalmente, se observan ciertos patrones de comportamiento multimodal en las distribuciones, que podrían ser beneficiosos para la identificación de fraude. Por otro lado, variables como proposed_credit_limit y zip_count_4w presentan valores extremos que pueden influir en la clasificación. Modelos más avanzados como redes neuronales pueden manejar mejor estos casos que modelos lineales. Finalmente, la variable device_fraud_count no aporta información relevante para el modelo, por lo tanto, es eliminada del dataset.

C. Entrenamiento y validación

De acuerdo con Jesus et al [10], no existe dependencia entre las instancias debido a que no son datos secuenciales, por lo tanto, es posible dividir la información con la variable month de la siguiente manera: Para entrenamiento se utilizaron las instancias anteriores al sexto mes y para la validación se utilizaron instancias iguales y posteriores al sexto mes, y se dividieron los datos en dos partes equitativas para el testeo (50%) y validación (50%).

D. Submuestreo

El proceso de submuestreo de las instancias se realizó teniendo en cuenta dos razones principales: reducir el costo computacional al manejar un millón de datos y disminuir el desbalanceo de clases para evitar sesgos hacia la clase mayoritaria. Aunque esta técnica puede reducir la variabilidad de los datos, se tuvieron en cuenta ciertas precauciones específicas: i) El submuestreo se hace solo sobre la clase 0 (transacciones legítimas) en el conjunto de entrenamiento. ii) Los conjuntos de validación y prueba no deben tener un submuestreo en solo una clase, sino de forma equilibrada en ambas, para mantener una representación realista. iii) La misma ratio de reducción (0,7) se aplica a todos los conjuntos, asegurando coherencia y manteniendo la proporción típica del 70–80% para entrenamiento. Este enfoque permite reducir el desequilibrio entre clases en el entrenamiento sin distorsionar la evaluación en validación y prueba.

E. Escalado de datos

El uso de MinMaxScaler es común en el preprocesamiento de datos debido a su capacidad para transformar las características del conjunto de datos a un rango definido, generalmente [0, 1], lo cual favorece el entrenamiento de redes neuronales. Esta transformación garantiza que las variables tengan escalas comparables, evitando que aquellas con mayores magnitudes influyan desproporcionadamente en el proceso de aprendizaje. Además, contribuye a una convergencia más rápida y estable en los algoritmos de optimización basados en gradientes, que operan de forma más eficiente con datos normalizados. El método preserva la distribución original de las variables, lo que resulta beneficioso para mantener la interpretabilidad del modelo. Asimismo, el escalado es un requisito previo para la aplicación de técnicas de sobremuestreo basadas en distancias, como SMOTE (Synthetic Minority Over-sampling Technique), asegurando que todas las características aporten de manera equitativa al cálculo de dichas distancias [11]. Por estas razones, el escalador se entrena sobre el conjunto de datos completo antes de la partición en subconjuntos de entrenamiento, validación y prueba, con el fin de capturar la máxima variabilidad posible en los valores mínimos y máximos utilizados para la transformación.

F. Clustering

En este estudio se implementó un proceso de clusterización de instancias con el objetivo de identificar patrones subyacentes en las transacciones, agrupar observaciones con características similares y detectar posibles segmentos asociados a actividades fraudulentas. Se empleó el algoritmo Density-Based Spatial Clustering of Applications with Noise (DBSCAN) debido a su capacidad para determinar automáticamente el número de clústeres y manejar eficazmente ruido y outliers [12]. No obstante, se reconocen sus limitaciones ante distribuciones no globulares o densidades heterogéneas, así como su sensibilidad a los hiperparámetros ε (epsilon) y min_samples. Para optimizar dichos hiperparámetros, se diseñó una función iterativa que evalúa combinaciones predefinidas mediante la métrica de silueta (silhouette score), la cual cuantifica la coherencia interna y separación entre clústeres, con valores en el rango [-1, 1]. De esta forma, valores próximos a 1 indican asignaciones correctas, cercanos a 0 sugieren proximidad entre clústeres, y negativos evidencian asignaciones incorrectas [13].

En la determinación de los valores posibles de ε, se consideró que en un espacio N-dimensional con variables escaladas en [0, 1], la distancia máxima entre dos puntos es $\sqrt{N}$. Tras el preprocesamiento y codificación One-Hot, se obtuvieron 45 dimensiones, estableciendo así una distancia máxima teórica de 6,71. Este enfoque permitió explorar configuraciones óptimas para mejorar la calidad del

agrupamiento y, por ende, la capacidad de detección de patrones anómalos. De esta manera, se definió la siguiente lista de valores: [0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5].

El hiperparámetro min_samples define un número mínimo de instancias requeridas para considerar que una región del espacio de características constituye un clúster. No existe una forma de saber cuál es un número óptimo, ya que depende de la densidad y del tamaño del conjunto de datos. Por esta razón, se optó por explorar un rango amplio de valores: [3, 5, 10, 100, 1000, 50000, 100000]. Los valores pequeños se incluyeron para evaluar si la formación de clústeres ocurre con un número reducido de puntos, y de esta forma detectar patrones locales o microestructuras en el conjunto de datos. Además, los valores intermedios (100,1000) se consideraron para incrementar la restricción de densidad en conjuntos de datos de gran tamaño, evitando la generación de agrupamientos falseados. Finalmente, los valores elevados (50 000 y 100000) se seleccionaron para probar configuraciones que solo identifican clústeres muy densos y robustos.
Esta estrategia de búsqueda sistemática permite evaluar la sensibilidad del algoritmo frente a diferentes escalas de densidad, asegurando que la elección final del hiperparámetro maximice la coherencia y relevancia de los clústeres obtenidos. Tras combinar cada posibilidad y evaluarla, se obtiene la gráfica en 3D que se muestra en la figura 2.

Además, es posible hacer una representación visual sobre un plano 2D tanto de los clusters como de la distribución de las clases 0 y 1. Para pasar de 45 dimensiones a 2 dimensiones, se utilizaron dos técnicas de reducción de dimensionalidad, PCA y t-SNE, el resultado se muestra en la figura 3.

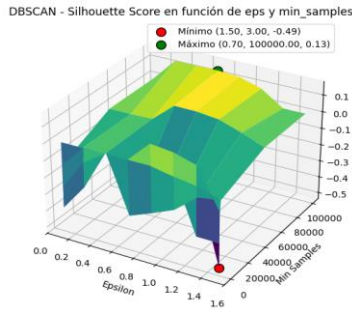

Fig.2 Silhouette Score en función de épsilon y min_samples

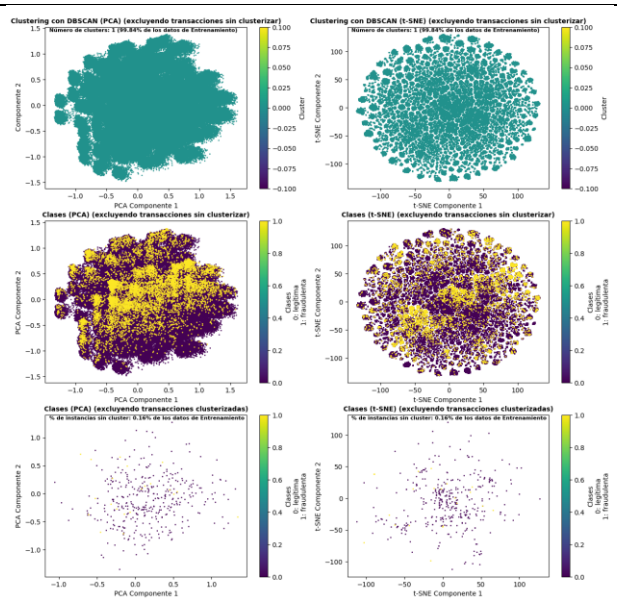

Fig.3 Reducción de hiperdimensiones: Visualización en 2D mediante PCA y t-SNE.

De acuerdo con lo anterior, la figura 3 muestra la proyección bidimensional de los datos utilizando PCA (columna izquierda) y t-SNE (columna derecha). El análisis revela que, según el algoritmo DBSCAN, el número óptimo de clústeres, definido tras evaluar 30 combinaciones de parámetros mediante la métrica de silueta, corresponde a una única agrupación que concentra el 99,84 % de los datos de entrenamiento, mientras que el 0,16 % restante se clasifica como ruido. La dispersión y superposición presentes en ambas proyecciones indican la ausencia de una separación clara entre transacciones fraudulentas y no fraudulentas, lo que sugiere que las características empleadas no resultan suficientes para una discriminación efectiva tras la reducción de dimensionalidad.

Aunque t-SNE proporciona una distribución más esférica y PCA una estructura más lineal, en ambos casos los fraudes permanecen dispersos. La falta de diversidad en los clústeres y el bajo valor obtenido en la métrica de silueta confirman la ineficacia de DBSCAN para la clasificación no supervisada en este contexto. Si bien podrían explorarse otros métodos de agrupamiento, los resultados respaldan la necesidad de aplicar modelos más complejos, como redes neuronales, para la detección de fraudes en el conjunto de datos analizado.

G. Desarrollo de la arquitectura DNN

Se diseñó una red neuronal profunda (Deep Neural Network, DNN) para abordar un problema de clasificación binaria, orientado a predecir una salida en el rango [0, 1]. La arquitectura propuesta consta de capas densas secuenciales con 512, 256, 128, 64 y 32 neuronas, seguidas de una capa de salida con activación sigmoide. Las capas ocultas emplean la función de activación ReLU, mientras que la normalización por lotes (Batch Normalization) y el abandono aleatorio (Dropout) se aplican tras cada capa densa para mitigar el sobreajuste y mejorar la capacidad de generalización. [14],[15].

El entrenamiento se realizó con el optimizador Adam, parametrizado por la tasa de aprendizaje (learning rate), utilizando como función de pérdida binary crossentropy [16]. La evaluación del modelo se basó en métricas de precisión (accuracy), sensibilidad (recall) y Área Bajo la Curva (AUC), esta última incorporada para valorar la capacidad discriminativa entre las dos clases en escenarios de datos desbalanceados.

H. Red Neuronal Convolucional

Se propone un enfoque alternativo a las redes neuronales densas (DNN) para la clasificación binaria de transacciones bancarias, basado en la utilización de redes neuronales convolucionales (CNN) mediante la transformación de datos tabulares en representaciones visuales RGB. Cada instancia se escala al rango [0, 1] y se reorganiza en una matriz cuadrada cuya diagonal contiene los valores originales, mientras que las posiciones restantes se completan simétricamente mediante operaciones entre pares de características.

La representación final consta de tres canales: (i) media entre pares de características, que captura la tendencia central de sus relaciones; (ii) diferencia entre pares, que resalta contrastes; y (iii) producto entre pares, que refleja interacciones multiplicativas. Esta conversión a imagen permite a la CNN extraer características espaciales y jerárquicas, potenciando la detección de patrones complejos que podrían pasar inadvertidos en una DNN.

Se desarrolla una metodología para transformar datos tabulares en imágenes RGB con el fin de optimizar la detección de transacciones fraudulentas mediante redes neuronales convolucionales (CNN). El proceso parte de instancias vectoriales cuyas características, normalizadas en [0, 1], se colocan en la diagonal principal de una matriz base. Los elementos restantes se completan de forma simétrica utilizando tres operaciones distintas para cada canal: (i) promedio de pares de características (canal rojo), (ii) valor absoluto de sus diferencias (canal verde) y (iii) producto de pares (canal azul).

La superposición de estas tres matrices genera una seudoimagen tridimensional por instancia, lo que permite a la CNN capturar patrones espaciales y jerárquicos no evidentes en formato tabular. Este enfoque, además de mejorar el rendimiento en la clasificación, ofrece ventajas adicionales: preservación de la privacidad mediante anonimización de datos sensibles, integración con sistemas de visión por computadora ya desplegados, compresión eficiente de la información mediante formatos de imagen y posibilidad de análisis visual directo por expertos. La metodología constituye

así una alternativa versátil y segura frente a redes densas (DNN) para aplicaciones críticas en el sector financiero.

I. Desarrollo de la arquitectura CNN

La arquitectura propuesta para la red convolucional (CNN) está compuesta por bloques convolucionales y capas densas, diseñados para procesar imágenes como entradas y realizar clasificación binaria. El modelo inicia definiendo las entradas, correspondientes a imágenes con dimensiones (resol, resol, channels), donde resol indica la resolución (cantidad de características del vector original) y channels representa el número de canales de color, siendo tres en el caso de imágenes RGB.

El primer bloque convolucional incorpora dos capas con filtros 3×3 y función de activación ReLU, seguidas por una capa de normalización por lotes (BatchNormalization) que estabiliza el entrenamiento y mejora la velocidad de convergencia. Posteriormente, se aplica MaxPooling2D para reducir la dimensionalidad y Dropout para mitigar el sobreajuste. Este patrón se repite en los bloques siguientes, incrementando progresivamente el número de filtros (32, 64 y 128), lo que permite extraer representaciones cada vez más complejas y abstractas.

Tras la fase convolucional, las características aprendidas se transforman en un vector unidimensional mediante Flatten. A continuación, el flujo pasa por varias capas densas con activación ReLU y normalización por lotes, complementadas con Dropout para reforzar la capacidad de generalización. Esta combinación facilita la captura de patrones complejos y mejora el rendimiento en datos no vistos.

La capa de salida consta de una única neurona con activación sigmoide, apropiada para clasificación binaria. El modelo emplea la función de pérdida binary_crossentropy para predecir la etiqueta correspondiente (fraude o no fraude). El entrenamiento se realiza con el optimizador Adam, ajustando la tasa de aprendizaje (learning rate) e incorporando métricas como precisión, sensibilidad (recall) y AUC. Asimismo, se aplica early stopping para interrumpir el proceso en caso de ausencia de mejora en la validación, evitando así el sobreajuste. Finalmente, se utiliza class_weight para equilibrar las clases, manteniendo la coherencia con la estrategia aplicada en la DNN.

3. Resultados y Discusión

En la tabla 1 se presenta la comparación entre los modelos basados en DNN, en ella se observa que, entre los valores de dropout evaluados, 0,5 ofrece la mejor relación entre la pérdida en entrenamiento y validación. Este valor permite alcanzar una estabilidad adecuada en la pérdida de validación, manteniéndose por debajo de la pérdida de entrenamiento, lo que indica una mayor capacidad para mitigar el sobreajuste y, en consecuencia, una mayor robustez frente a la variabilidad de los datos de entrada. Asimismo, las métricas asociadas al entrenamiento y prueba —exactitud, precisión y sensibilidad— se mantienen estables conforme avanzan las épocas de entrenamiento.

Con el objetivo de facilitar la comparación entre modelos, se utilizó un esquema visual basado en un degradado de color, donde el tono verde indica un mejor rendimiento y el rojo un desempeño menos favorable. Es relevante que el umbral se mantenga lo más cercano posible a 0,5, ya que esto refleja curvas más suaves y estables, lo que implica mayor robustez frente a variaciones en los datos externos.

Tabla 1. Comparación de los modelos de tipo DNN

Modelo	Tipo	Aj. Pesos	Bal. Dataset	Alg. Bal.	Dropout	Max ROC-AUC	Umbral mejor AUC-ROC	Prec. En umbral	Recall. En Umbral	Acc. En umbral	VP(1)	FP(0)
1	DNN	No	No	-	0	0,796	0,04	0,054	0,787	0,803	339	5909
2	DNN	No	No	-	0,33	0,799	0,04	0,056	0,789	0,809	340	5771
3	DNN	No	No	-	0,5	0,802	0,02	0,049	0,833	0,772	359	6948
4	DNN	Si	No	-	0	0,771	0,2	0,047	0,763	0,778	329	6728
5	DNN	Si	No	-	0,33	0,807	0,42	0,055	0,814	0,8	351	6082
6	DNN	Si	No	-	0,5	0,811	0,43	0,055	0,824	0,8	355	6089
7	DNN	Si	Si	CTGAN	0	0,787	0,19	0,052	0,755	0,798	334	6109
8	DNN	Si	Si	CTGAN	0,33	0,794	0,37	0,048	0,821	0,768	354	7050
9	DNN	Si	Si	CTGAN	0,5	0,804	0,43	0,061	0,78	0,827	336	5211
10	DNN	Si	Si	SMOTE	0	0,766	0,03	0,049	0,733	0,799	316	6071
11	DNN	Si	Si	SMOTE	0,33	0,804	0,41	0,053	0,817	0,792	352	6310
12	DNN	Si	Si	SMOTE	0,5	0,807	0,48	0,06	0,789	0,825	340	5298

En términos generales, todos los modelos presentan métricas de prueba similares, con un valor máximo de la curva ROC-AUC cercano a 0,8, aunque el umbral y la forma de la curva difieren entre modelos. Este máximo coincide, aproximadamente, con el punto en el que sensibilidad y exactitud se interceptan, alcanzando en dicho umbral valores cercanos al 80 %. Sin embargo, la precisión en ese punto se mantiene alrededor del 5 %. Dado que la curva ROC-AUC refleja la capacidad del modelo para discriminar entre clases positivas y negativas, el umbral correspondiente a su máximo constituye un criterio adecuado para la comparación entre modelos. Por consiguiente, las valoraciones siguientes se fundamentan en dicho umbral, el cual varía según el modelo:

- Modelo 3: Alcanzó 359 verdaderos positivos para la clase fraudulenta (clase 1), equivalentes a una sensibilidad del 83,3 %. Este modelo fue entrenado sin técnicas de balanceo y con un dropout de 0,5. A pesar de que este resultado sugiere un buen desempeño, el umbral asociado a dicho valor es 0,02, muy próximo a cero, con curvas de ROC-AUC, sensibilidad y exactitud altamente pronunciadas. Esto implica una baja estabilidad frente a perturbaciones en los datos de prueba: una variación mínima del umbral (por ejemplo, de 0,01) reduciría la exactitud a aproximadamente 40 %. Por ello, no se considera el mejor modelo.
- Modelo 6: Obtuvo 355 verdaderos positivos, equivalentes a una sensibilidad del 82,4 %. Este modelo aplicó balanceo de pesos durante el entrenamiento, pero no balanceo de clases en el conjunto de datos, manteniendo un dropout de 0,5. A diferencia del modelo anterior, presenta curvas más suaves, lo que indica mayor robustez frente a variaciones en los datos. El umbral correspondiente es 0,43.

Al analizar el valor máximo de la curva ROC-AUC, el Modelo 6 sobresale con 0,811, el más alto entre todos los modelos evaluados. En cuanto a precisión y falsos positivos para la clase 0, se mantiene en rangos medios con 5,5 % y 6.089 respectivamente, logrando una exactitud global cercana al 80 %. En consecuencia, para el diseño de la CNN, no se implementará balanceo de clases en el conjunto de datos; sin embargo, se aplicará balanceo de pesos durante el entrenamiento, dado su impacto positivo en la estabilidad del modelo.

Resultados imagen RGB

La conversión de datos tabulares en representaciones visuales para su procesamiento mediante redes neuronales convolucionales (CNN) constituye una estrategia eficaz para la detección de fraudes en entornos financieros. Este enfoque permite analizar las transacciones como imágenes, lo que facilita la identificación de patrones locales y globales que resultan difíciles de capturar en el formato tabular tradicional. Para lograrlo, se generan matrices que organizan las características de manera estructurada, incorporando operaciones matemáticas específicas que enriquecen la representación y potencian la capacidad del modelo para discriminar entre transacciones legítimas y fraudulentas.

Entre las ventajas de esta metodología destaca la posibilidad de anonimizar datos sensibles mediante la codificación en imágenes, eliminando la necesidad de conservar identificadores personales en su forma original. Esta característica es particularmente relevante en contextos donde la privacidad es crítica, como el sector bancario y financiero. Asimismo, la representación visual habilita la integración con sistemas de visión por computadora ya implementados, lo que permite aplicar herramientas existentes para inspección y análisis visual en tareas como la detección de anomalías o la verificación de patrones.

Otra ventaja importante es la compresión eficiente de la información. Las matrices generadas pueden almacenarse utilizando algoritmos estándar de compresión de imágenes (por ejemplo, JPEG o PNG), reduciendo significativamente el espacio requerido sin pérdida sustancial de información relevante. Este aspecto resulta esencial cuando se gestionan volúmenes masivos de transacciones, optimizando el almacenamiento y la transmisión de datos.

Finalmente, la conversión en imágenes facilita la interpretación visual, permitiendo que expertos humanos colaboren con los modelos automatizados mediante la inspección directa de las representaciones gráficas. Esta sinergia entre análisis automático y validación experta abre la posibilidad de un enfoque híbrido que incremente la robustez y confiabilidad en la detección de fraudes. A continuación, se observan imágenes de transacciones legitimas (figura 4) y transacciones fraudulentas (figura 5) tras la aplicación de la conversión de vectores unidimensionales a imágenes RGB. Además, en la tabla 2 se proponen algunas consideraciones que se deben tener en cuenta en el momento de una inspección directa para la clasificación (legitima o fraudulenta) de las transacciones.

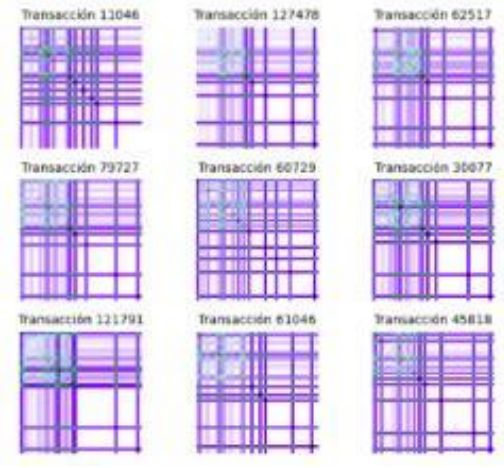

Fig.4 Imágenes sintéticas a partir de vectores de transacciones legítimas.

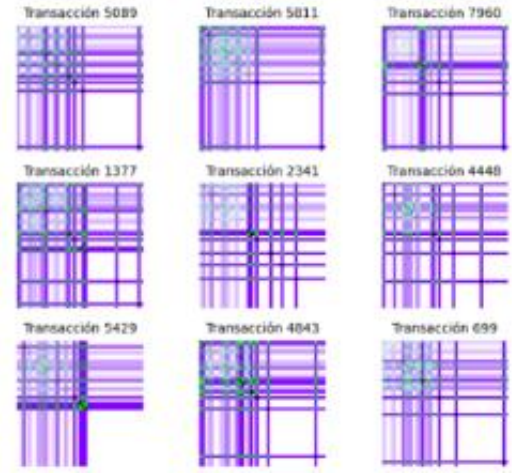

Fig.5 Imágenes sintéticas a partir de vectores de transacciones fraudulentas.

Tabla 2. Criterios para clasificar una transacción fraudulenta o legitima.

Característica	Transacciones no fraudulentas	Transacciones fraudulentas
Distribución de líneas	Estructura homogénea y cuadricula regular	Presencia de zonas densas y regulares
Intensidad de color	Tonos purpura suaves y uniformes	Mayor contraste, áreas con purpura intenso y puntos destacados
Patrones localizados	Sin concentraciones anómalas	Regiones altamente contrastadas, especialmente en la zona superior izquierda
Geometría	Líneas bien distribuidas y simétricas	Irregularidades y secciones compactas
Contraste global	Bajo, predominan áreas uniformes	Alto, con zonas claras y oscuras marcadas
Puntos verdes	Distribuidos de manera regular	Más concentrados en áreas especificas

Resultados prueba CNN

En la figura 6 y en la tabla 3, se observa que el dropout de 0,5 vuelve a demostrar el mejor rendimiento entre los valores evaluados. En particular, el modelo 15 supera a las demás iteraciones en todas las métricas y condiciones analizadas, presentando curvas más suaves y estables tanto durante el entrenamiento de la red neuronal como en las métricas de prueba.

Tabla 3. Comparación de los modelos de tipo CNN

Modelo	Tipo	Aj. Pesos	Bal. Dataset	Alg. Bal.	Dropout	Max ROC-AUC	Umbral mejor AUC-ROC	Prec. En umbral	Recall. En Umbral	Acc. En umbral	VP(1)	FP(0)
13	CNN	Si	No	-	0	0,772	0,26	0,045	0,782	0,762	337	7231
14	CNN	Si	No	-	0,33	0,776	0,33	0,046	0,787	0,708	339	7101
15	CNN	si	No	-	0,5	0,808	0,45	0,056	0,81	0,806	349	5882

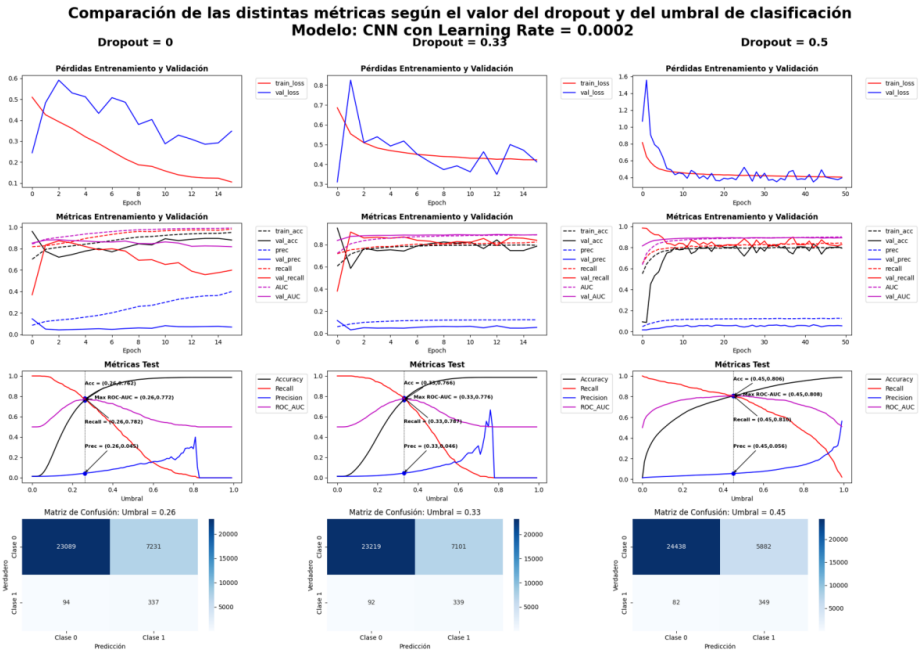

Fig.6 CNN entrenada con balanceo de clases (pesos ajustados por clase) en red neuronal pero no en el dataset de imágenes.

Resultados del modelo combinado DNN + CNN

Según lo presentado en la figura 7 y la tabla 4, correspondientes a la comparación de modelos de tipo CNN combinada, los resultados indican que, si bien el Modelo 16 alcanza la mayor sensibilidad y, en consecuencia, el mayor número de verdaderos positivos (clase 1), el modelo 17 muestra un desempeño superior en términos globales de discriminación entre clases, evidenciado por un valor más elevado en la métrica ROC-AUC.

Tabla 4. Comparación de los modelos de tipo CNN combinada

Modelo	Tipo	Aj. Pesos	Bal. Dataset	Alg. Bal.	Dropout	Max ROC-AUC	Umbral mejor AUC-ROC	Prec. En umbral	Recall. En Umbral	Acc. En umbral	VP(1)	FP(0)
16	CNN + DNN comb	Si	No	-	0,5	0,801	0,6	0,052	0,814	0,788	351	6426
17	CNN + DNN prom	Si	No	-	0,5	0,813	0,46	0,061	0,8	0,824	345	5314

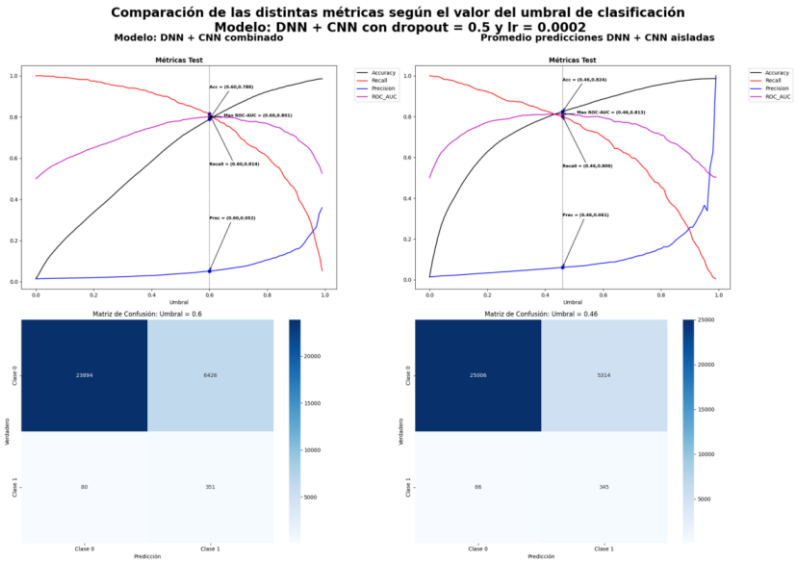

Fig.7 DNN + CNN combinados vs DNN + CNN promediados.

Resultados generales

La tabla 5 resume los resultados obtenidos por diferentes modelos de redes neuronales, incluyendo DNN, CNN y arquitecturas híbridas, evaluados bajo diversas configuraciones que combinan ajustes de pesos, balanceo de datos y valores de dropout. Para la comparación se consideran métricas clave como ROC-AUC, precisión, sensibilidad, exactitud, así como el número de verdaderos positivos (VP) y falsos positivos (FP), lo que permite evaluar su desempeño en la detección de fraudes.

Entre los modelos analizados, destaca el modelo 6 (DNN con dropout de 0,5 y sin balanceo de datos), que alcanza una de las puntuaciones más altas en ROC-AUC (0,811), lo que refleja una capacidad sólida para diferenciar entre clases. Su sensibilidad (0,824) es también de las más elevadas, indicando una detección efectiva de transacciones fraudulentas. Este comportamiento se confirma con el mayor número de verdaderos positivos (355). No obstante, el modelo presenta un volumen considerable de falsos positivos (6.089), lo que implica un aumento en las alertas erróneas

frente a otros modelos. Aun así, cuando la prioridad es maximizar la detección de fraudes, este modelo se perfila como una opción adecuada.

Otro caso relevante es el modelo 17 que obtiene la mayor puntuación ROC-AUC de la tabla (0,813), reflejando un rendimiento global superior. Además, alcanza la precisión más alta (0,061), lo que significa que, cuando clasifica una transacción como fraude, la probabilidad de que sea correcta es mayor. Su exactitud (0,824) también resulta competitiva, manteniendo un buen equilibrio entre sensibilidad y errores. Cabe destacar que reduce el número de falsos positivos a 5.314, lo que representa un avance frente a otros modelos en términos de robustez.

Tabla 5. Resumen de todos los datos obtenidos

Modelo	Tipo	Aj. Pesos	Bal. Dataset	Alg. Bal.	Dropout	Max ROC-AUC	Umbral mejor AUC-ROC	Prec. En umbral	Recall. En Umbral	Acc. En umbral	VP(1)	FP(0)
1	DNN	No	No	-	0	0,796	0,04	0,054	0,787	0,803	339	5909
2	DNN	No	No	-	0,33	0,799	0,04	0,056	0,789	0,809	340	5771
3	DNN	No	No	-	0,5	0,802	0,02	0,049	0,833	0,772	359	6948
4	DNN	Si	No	-	0	0,771	0,2	0,047	0,763	0,778	329	6728
5	DNN	Si	No	-	0,33	0,807	0,42	0,055	0,814	0,8	351	6082
6	DNN	Si	No	-	0,5	0,811	0,43	0,055	0,824	0,8	355	6089
7	DNN	Si	Si	CTGAN	0	0,787	0,19	0,052	0,755	0,798	334	6109
8	DNN	Si	Si	CTGAN	0,33	0,794	0,37	0,048	0,821	0,768	354	7050
9	DNN	Si	Si	CTGAN	0,5	0,804	0,43	0,061	0,78	0,827	336	5211
10	DNN	Si	Si	SMOTE	0	0,766	0,03	0,049	0,733	0,799	316	6071
11	DNN	Si	Si	SMOTE	0,33	0,804	0,41	0,053	0,817	0,792	352	6310
12	DNN	Si	Si	SMOTE	0,5	0,807	0,48	0,06	0,789	0,825	340	5298
13	CNN	Si	No	-	0	0,772	0,26	0,045	0,782	0,762	337	7231
14	CNN	Si	No	-	0,33	0,776	0,33	0,046	0,787	0,708	339	7101
15	CNN	si	No	-	0,5	0,808	0,45	0,056	0,81	0,806	349	5882
16	CNN + DNN comb	Si	No	-	0,5	0,801	0,6	0,052	0,814	0,788	351	6426
17	CNN + DNN prom	Si	No	-	0,5	0,813	0,46	0,061	0,8	0,824	345	5314
Promedio					-	0,795	0,304	0,053	0,795	0,792	343	6208
Desviación típica					-	0,015	0,179	0,005	0,025	0,028	11	618

En términos prácticos, la elección entre ambos modelos depende del objetivo estratégico. Si la meta principal es maximizar la detección de fraudes aun a costa de un mayor número de alertas falsas, el modelo 6 constituye la mejor alternativa. Por el contrario, si se busca un balance entre detección y reducción de falsos positivos, el modelo 17 representa la opción más adecuada.

En síntesis, ambos modelos presentan fortalezas diferenciadas. El modelo 6 resulta adecuado cuando la prioridad es maximizar la detección de fraudes, ya que ofrece una alta sensibilidad y un mayor número de verdaderos positivos, aunque a costa de un incremento en los falsos positivos. En contraste, el modelo 17 proporciona un equilibrio más consistente entre capacidad de detección y reducción de falsas alarmas, lo que lo posiciona como una opción más confiable desde una perspectiva operativa.

Discusiones:

La Tabla 2 efectivamente puede interpretarse como un conjunto de reglas explícitas que orientan la clasificación de transacciones, lo cual abre la posibilidad de considerar a las redes neuronales como un mecanismo para identificar patrones o “síntomas” que dichas reglas emplearían como insumo. En este sentido, resulta factible que un modelo de RN utilice criterios derivados de dichas reglas como variables de entrada, integrándolos en el proceso de aprendizaje y reforzando así la capacidad

predictiva. No obstante, este enfoque plantea un desafío adicional en cuanto a la interpretabilidad de los modelos, dado que las redes neuronales profundas se caracterizan por su opacidad en la interpretación de los procesos internos, lo que dificulta la comprensión y trazabilidad de la lógica que sustenta sus predicciones. En consecuencia, como línea de trabajo futuro se sugiere avanzar en metodologías de explicabilidad e interpretabilidad (por ejemplo, SHAP, LIME o mapas de activación en CNN), así como en la combinación de modelos híbridos que integren reglas explícitas y redes neuronales, con el fin de mejorar la transparencia y legibilidad de los resultados obtenidos.

4. Conclusiones

El estudio permitió comparar distintos enfoques para la detección de fraude bancario mediante redes neuronales profundas (DNN), convolucionales (CNN) y una arquitectura híbrida que combina ambas, evaluando parámetros como dropout, balanceo de clases y ajuste de pesos. Los resultados evidencian que el valor de dropout influye significativamente en el rendimiento, siendo 0,5 el más adecuado en términos de estabilidad, ROC-AUC y sensibilidad, lo que mejora la capacidad de generalización del modelo; no obstante, se sugiere explorar valores superiores en investigaciones futuras. La transformación de datos tabulares en imágenes para su procesamiento con CNN se consolida como un enfoque innovador y prometedor, ya que permite capturar relaciones complejas mediante patrones espaciales y jerárquicos, además de ofrecer ventajas adicionales como anonimización de datos, compresión eficiente e integración con sistemas de visión por computadora. El modelo híbrido DNN+CNN (Modelo 17) destacó por lograr la mayor puntuación ROC-AUC (0,813) y un equilibrio entre detección de fraudes y reducción de falsos positivos, lo que confirma el potencial de las arquitecturas combinadas para mejorar la discriminación entre clases. Sin embargo, el uso de técnicas de balanceo debe gestionarse con cautela, dado que enfoques como SMOTE y CTGAN no resultaron eficaces al introducir ruido y pérdida de información, mientras que el algoritmo DBSCAN mostró un bajo desempeño para segmentar los datos, lo que sugiere la necesidad de explorar métodos más robustos. Finalmente, se reconoce como limitación la capacidad computacional disponible, que restringió el análisis a un solo dataset del conjunto BAF, lo que plantea la conveniencia de escalar futuros experimentos a múltiples bases de datos para fortalecer la capacidad de generalización de los modelos propuestos.

Referencias

1. Kipngetich, A. *A review of online scams and financial frauds in the digital age*. GSC Advanced. Research and Reviews, 2025. 22(1), p. 302–329.
   DOI: https://doi.org/10.30574/gscarr.2025.22.1.0025
2. Díaz Agudelo, V., Osorno Gallego, M. C., Tangarife Gómez, L., & Chamorro González, C. *Componentes que influyen en la ejecución de fraudes financieros: percepción de los profesionales contables*. Semestre Económico, *24*(56), 2021.
   DOI: https://doi.org/10.22395/seec.v24n56a4
3. A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams and P. Beling., *Deep learning detecting fraud in credit card transactions*. Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2018, pp. 129-134.
   DOI: https://doi.org/10.1109/SIEDS.2018.8374722
4. Mienye E, Jere N, Obaido G, Mienye ID, Aruleba K. *Deep learning in finance: A survey of applications and techniques*. AI. 2024;5(4):2066-2091.
   DOI: https://doi.org/10.3390/ai5040101

5. Niño, C., & López, O. *Aproximación a la Detección de Fraude Financiero en Transacciones con Tarjeta de Crédito Empleando Machine Learning*. FACE: Revista De La Facultad De Ciencias Económicas Y Empresariales, 2025. 25(2), 217–225. DOI: https://doi.org/10.24054/face.v25i2.4029
6. Tustón Fuentes, J. B., & Macías Arias, E. J. *Modelos de machine learning para la detección de fraudes financieros: Una revisión de la literatura*. *UNESUM - Ciencias. Revista Científica Multidisciplinaria*, 2025. *9*(2), 220–234. DOI: https://doi.org/10.47230/unesum-ciencias.v9.n2.2025.220-234
7. Hernandez Aros, L., Bustamante Molano, L.X., Gutierrez-Portela, F. *et al. Financial fraud detection through the application of machine learning techniques: a literature review*. *Humanit Soc Sci Commun*, 2024. 11, 1130. https://doi.org/10.1057/s41599-024-03606-0
8. Jin, J., Zhang, Y. *The analysis of fraud detection in financial market under machine learning*. *Sci Rep* **15**, 29959. 2025. DOI: https://doi.org/10.1038/s41598-025-15783-2
9. Obi, J. C. (2023). *A comparative study of several classification metrics and their performances on data*. *World Journal of Advanced Engineering Technology and Sciences, 8*(1), 308–314. https://doi.org/10.30574/wjaets.2023.8.1.0054
10. Jesus, S., Pombal, J., Alves, D., Cruz, A. F., Saleiro, P., Ribeiro, R. P., Gama, J., & Bizarro, P. (s. f.). *BAF Dataset Suite Datasheet.* Advances in Neural Information Processing Systems. 2022. Recuperado de: https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022.
11. V. Asha, M. T. Vasumathi, A. Prasad, Y. V, Y. A. P and M. Sivani, *Evaluation of ML Models using SMOTE and Feature Scaling for Intrusion Detection System (IDS)*, International Conference on Visual Analytics and Data Visualization (ICVADV), Tirunelveli, India, pp. 243-249, 2025. DOI: https://doi.org/10.1109/ICVADV63329.2025.10961506
12. Huang, Z.; Liang, Z.; Zhou, S.; Zhang, S. *An Improved Density-Based Spatial Clustering of Applications with Noise Algorithm with an Adaptive Parameter Based on the Sparrow Search Algorithm.* Algorithms, 18, 273, 2025. *DOI:* https://doi.org/10.3390/a18050273
13. Januzaj, Y., Beqiri, E., & Luma, A. *Determining the optimal number of clusters using Silhouette Score as a data mining technique*, International Journal of Online and Biomedical Engineering (iJOE), *19*(04), 174–182. 2023. DOI: https://doi.org/10.3991/ijoe.v19i04.37059
14. Bai, Y. (2022). *RELU-Function and Derived Function Review*. SHS Web of Conferences, 144, 02006. EDP Sciences. https://doi.org/10.1051/shsconf/202214402006
15. Balestriero, R., & Baraniuk, R. G. (2022). *Batch Normalization Explained*. arXiv. DOI: https://doi.org/10.48550/arXiv.2209.14778
16. Ramos, D., Franco-Pedroso, J., Lozano-Diez, A., & Gonzalez-Rodriguez, J. (2018). Deconstructing Cross-Entropy for Probabilistic Binary Classifiers. *Entropy*, *20*(3), 208. https://doi.org/10.3390/e20030208

Conflicto de Intereses

Los autores declaran la inexistencia de conflicto de interés con institución o asociación comercial de cualquier índole.

Contribución de los autores

Cristian Guerrero Balber. ORCID: https://orcid.org/0009-0006-9494-1179
Participó en el diseño de la investigación, conceptualización, curación de datos y desarrollo metodológico. Redacción y edición del manuscrito original.

Camilo Andrés Pulzara Mora. ORCID: https://orcid.org/0000-0002-5243-309X
Participó en el diseño de la investigación, análisis, supervisión y validación de los resultados. Redacción y edición del manuscrito original.

Juan David Losada Losada. ORCID: https://orcid.org/0000-0001-9935-9977 Participó en la redacción y edición del manuscrito original; Revisión y análisis del contenido.