**Trabajo Final**

# Detección de Fraude en Operaciones con Tarjetas

**Alberto Valentín Velásquez Santos**[,†], **Rodolfo Morocho Caballero**[,†], **Max Houston Ramirez Martel**[,†] **and Harold Mondragon Tavara**[,†]

[†]*Estos autores contribuyeron igualmente a este trabajo.*

Este archivo fue compilado el 11 de Diciembre del 2024

**Abstract**

Este artículo explora técnicas avanzadas de optimización y heurísticas aplicadas a problemas NP-hard, abordando el Max-Cut, el problema del vendedor viajero y la gestión de portafolios financieros. Se examinan algoritmos metaheurísticos como los algoritmos genéticos, el recocido simulado y las colonias de hormigas, y se introduce el método Johnson y enfoques greedy para resolver problemas de gran escala. En aplicaciones prácticas, se destaca el uso de algoritmos heurísticos en la detección de fraudes en plataformas de streaming y en la predicción de archivos en servicios de almacenamiento. Los casos de estudio demuestran la efectividad de estas técnicas para encontrar soluciones eficientes y escalables, resaltando su valor en la adaptación a entornos cambiantes y la optimización de procesos complejos.

**Keywords:** *optimización, heurísticas, greedy, knapsack, Machine Learning, reconocimiento heurístico, np-hard*

## 1. Resumen

## 2. Introducción

El fraude en transacciones con tarjetas de crédito o débito se ha vuelto un problema crítico en Perú, donde, de acuerdo con datos del INEI (2024), estos ciberdelitos vienen presentando un aumento significativo con respecto a años anteriores. Esta problemática no solo genera pérdidas económicas significativas, sino también una pérdida de confianza por parte del cliente hacia el sistema bancario nacional.

El país y el mundo entero han experimentado una ola de digitalización a raíz de la pandemia; esto provocó un incremento de los cibercriminales y, en el contexto de este estudio, un aumento de los fraudes en las transacciones con tarjetas. Esto resalta la necesidad por parte del sistema bancario de implementar soluciones robustas que ayuden a mitigar, reducir y detectar las transacciones fraudulentas en tiempo real.

La detección de fraude en transacciones con tarjetas de crédito plantea desafíos significativos. Este problema se caracteriza por la presencia de datos desbalanceados, donde las transacciones fraudulentas representan menos del 1% del total. Esto genera que los modelos de IA tiendan a favorecer la clase mayoritaria (transacciones normales), comprometiendo su capacidad para identificar patrones fraudulentos. Además, las transacciones fraudulentas evolucionan constantemente, adaptándose a nuevas medidas de seguridad, lo que exige modelos dinámicos y altamente generalizables.

Este estudio propone evaluar cinco enfoques de modelado avanzado: Redes Neuronales Artificiales (ANNs), XGBoost, Random Forest, CatBoost y LightGBM, aplicando técnicas de preprocesamiento para manejar datos desbalanceados y mejorar la detección de transacciones fraudulentas. Estos modelos se analizan bajo métricas clave como el F1-score, precisión y sensibilidad, con el objetivo de identificar las metodologías más efectivas para enfrentar el fraude en el contexto peruano, caracterizado por su creciente digitalización y exposición a delitos financieros.

Los resultados obtenidos destacan que los modelos basados en árboles, como CatBoost y XGBoost, fueron los más eficaces, logrando F1-scores de 0.90 y 0.88, respectivamente. Estas técnicas demostraron un excelente equilibrio entre precisión y sensibilidad, lo que las posiciona como soluciones robustas para abordar el problema del fraude en transacciones electrónicas en Perú. Su implementación en los sistemas financieros locales podría ofrecer una herramienta efectiva para detectar patrones de fraude en tiempo real, contribuyendo a mitigar el impacto económico y fortalecer la confianza en las plataformas digitales.

## 3. Trabajos relacionados

El sector bancario enfrenta el aumento de fraudes con tarjetas de crédito, esto principalmente se debe al incremento de operaciones en línea. Xuan, et al. (2018) separan este problema en dos formas de fraude: el primero consiste en conseguir una tarjeta de crédito suplantando la identidad de otra persona y otro es robando la información sensible como contraseña, número de tarjeta o CVV. Awoyemi, et al. (2017) argumentan que el aumento de este tipo de fraude se debe a la digitalización del dinero y el modo de pago en línea está tomando mayor protagonismo en comparación a las transacciones en efectivo, lo cual incrementa la probabilidad de un fraude cibernético. Por estas razones, con ayuda del aprendizaje máquina se busca identificar potenciales casos de fraude en las transacciones bancarias que se realicen mediante tarjetas de crédito o débito. Alvarado, et al. (2023) implementan un modelo predictivo que permite clasificar aquellas transacciones anómalas; como parte de su investigación se presentaron ciertas dificultades con respecto al conjunto de datos, por lo que emplearon un muestreo aleatorio con reemplazo para incrementar los datos del grupo inferior.

La detección de fraude es ampliamente estudiada utilizando modelos supervisados y no supervisados. Wang, et al. (2019) implementaron XGBoost para datos desbalanceados, alcanzando un F1-score de 0.85, aunque con limitaciones en la sensibilidad hacia la clase minoritaria. Por otro lado, Sahin, et al. (2020) demostraron que ANNs pueden lograr una precisión del 90% en datos sintéticos, pero con altos costos computacionales.

Lemaitre, et al. (2017) exploraron técnicas de sobremuestreo combinadas con Random Forest, obteniendo un F1-score de 0.87, aunque con limitaciones en datasets reales. CatBoost, desarrollado por Prokhorenkova, et al. (2018), ha sido utilizado para problemas con datos categóricos, logrando una mejor generalización. Finalmente, LightGBM, como destaca Ke, et al. (2017), ha mostrado ser eficiente para grandes datasets, pero sensible a configuraciones incorrectas.

## 4. Propuesta

**Objetivo general:**

- Evaluar modelos que puedan identificar transacciones fraudulentas en tiempo real.

**Objetivo específico:**

- Detectar el 100% de los fraudes minimizando las clasificaciones incorrectas.

**Principales desafíos:**

- Los datos están muy desbalanceados: menos del 1% de las transacciones son fraudulentas (492 fraudes de 284,807 transacciones)

- Es difícil establecer una definición clara de lo que constituye "fraude"
- La mayoría de los comerciantes no son expertos en evaluar el impacto del fraude

## 5. Experimentos

Para evaluar la efectividad de los modelos y sus configuraciones en el problema de detección de fraude con tarjetas de crédito, se llevaron a cabo cinco experimentos principales. Cada experimento incluyó un diseño cuidadoso para manejar el desequilibrio de clases, optimizar hiperparámetros y garantizar una comparación justa entre modelos. A continuación, se describen en detalle los experimentos realizados.

### 5.1. Redes Neuronales Artificiales (ANNs)

El primer experimento utilizó Redes Neuronales Artificiales (ANNs) para abordar el problema. Se diseñó una arquitectura con tres capas ocultas, configuradas con 64, 32 y 16 neuronas, respectivamente, y funciones de activación ReLU para capturar no linealidades. La capa de salida utilizó una activación sigmoide para clasificaciones binarias. El modelo fue entrenado durante 50 épocas, utilizando el optimizador Adam y una tasa de aprendizaje inicial de $10^{-3}$. Además, se implementó early stopping basado en el F1-score en validación, para evitar el sobreajuste. Aunque las ANNs alcanzaron un excelente F1-score de 0.97 en entrenamiento, su rendimiento en pruebas cayó a 0.85, revelando una ligera sobreadaptación. Aunque lograron una buena precisión, la sensibilidad fue moderada, identificando correctamente el 78% de las transacciones fraudulentas.

### 5.2. XGBoost

El segundo experimento evaluó XGBoost, un modelo de boosting basado en árboles de decisión, conocido por su capacidad para manejar datos desbalanceados. Se configuraron 100 estimadores, una profundidad máxima de 6, y una tasa de aprendizaje de 0.1, junto con regularización L1 y L2 para prevenir sobreajuste. Para mitigar el desequilibrio de clases, se ajustó el parámetro scale_pos_weight a la proporción entre clases. XGBoost alcanzó un rendimiento notable con un F1-score de 0.88 en pruebas, equilibrando precisión y sensibilidad. Su robustez frente al desequilibrio de clases lo destacó como uno de los modelos más efectivos, aunque su optimización requiere ajustes detallados.

### 5.3. Random Forest

En el tercer experimento, se evaluó Random Forest, un modelo de ensamble robusto y versátil. Se construyeron 100 árboles con una profundidad máxima de 10, utilizando el criterio Gini para medir la impureza en cada división. Para manejar el desequilibrio de clases, se empleó el parámetro class_weight = 'balanced', otorgando mayor peso a las transacciones fraudulentas. Este modelo alcanzó un F1-score de 0.87 en pruebas, logrando un equilibrio sólido entre precisión y sensibilidad. Aunque su rendimiento fue competitivo, su sensibilidad fue ligeramente menor en comparación con XGBoost y CatBoost. Además, Random Forest presentó un mayor costo computacional debido a la construcción de múltiples árboles completos.

### 5.4. CatBoost

El cuarto experimento empleó CatBoost, un modelo diseñado para manejar directamente datos categóricos, aunque este dataset contenía principalmente variables continuas. Se configuraron 200 estimadores con una profundidad máxima de 8 y una tasa de aprendizaje de 0.05. El modelo ajustó automáticamente los pesos de las clases para tratar el desequilibrio de manera eficiente. CatBoost demostró ser el modelo más robusto, alcanzando un F1-score de 0.90 en pruebas. Su capacidad para equilibrar precisión y sensibilidad lo posicionó como la opción más efectiva, especialmente en contextos con datos desbalanceados.

### 5.5. LightGBM

Finalmente, el quinto experimento exploró el desempeño de LightGBM, un modelo basado en boosting con histogramas, optimizado para grandes datasets. Se configuraron 150 árboles con una profundidad máxima de 31 hojas y una tasa de aprendizaje de 0.1. Además, se ajustaron los pesos de clase para minimizar el impacto del desequilibrio. LightGBM alcanzó un F1-score de 0.86 en pruebas, mostrando un buen rendimiento, aunque inferior al de CatBoost y XGBoost. Su ventaja principal radicó en su eficiencia computacional, siendo adecuado para escenarios con limitaciones de tiempo o recursos.

## 6. Resultados

- Los resultados

## 7. Conclusiones

- Las conclusiones son:

## 8. Referencias

- Referencias

## Tabla de Contenidos