INGENIUM ET POTENTIA
Revista Electrónica Multidisciplinaria de Ciencias Básicas, Ingeniería y Arquitectura
Año VIII. Vol VIII. N°14. Enero-Junio. 2026
Hecho el depósito de Ley: FA2019000052
FUNDACIÓN KOINONIA (F.K).
Santa Ana de Coro, Venezuela.

https://doi.org/10.35381/i.p.v8i14.4814

Detección de fraudes en transacciones financieras usando algoritmos de Machine Learning

Fraud detection in financial transactions using Machine Learning algorithms

Juan Darío Intriago-Montalván
jintriago3@utmachala.ed.ec
Universidad Técnica de Machala, Machala, El Oro
Ecuador
https://orcid.org/0009-0003-6266-9375

Dowsan Miguel Vásquez-Bermeo
dvasquez5@utmachala.edu.ec
Universidad Técnica de Machala, Machala, El Oro
Ecuador
https://orcid.org/0000-0002-9659-9120

Bertha Eugenia Mazón
bmazon@utmachala.edu.ec
Universidad Técnica de Machala, Machala, El Oro
Ecuador
https://orcid.org/0000-0002-2749-8561

Eduardo Tusa
etusa@utmachala.edu.ec
Universidad Técnica de Machala, Machala, El Oro
Ecuador
https://orcid.org/0000-0002-9408-5134

Recibido: 02 de agosto 2025
Revisado: 03 de octubre 2025
Aprobado: 15 de diciembre 2025
Publicado: 01 de enero 2026

INGENIUM ET POTENTIA
Revista Electrónica Multidisciplinaria de Ciencias Básicas, Ingeniería y Arquitectura
Año VIII. Vol VIII. N°14. Enero-Junio. 2026
Hecho el depósito de Ley: FA2019000052
FUNDACIÓN KOINONIA (F.K).
Santa Ana de Coro, Venezuela.

RESUMEN

El trabajo aborda el fraude financiero en transacciones con tarjetas de crédito. El objetivo es evaluar modelos de machine learning que identifiquen el algoritmo más robusto para la detección de fraudes financieros. Como solución se implementó una comparativa entre Random Forest, K-Nearest Neighbors y Árbol de Decisión aplicando la técnica SMOTEEN para balancear las clases. Se desarrollaron dos aplicaciones una web funcional para la evaluación y visualización de datos, y una de escritorio para la anonimización de los datos. Los resultados importantes demostraron que el modelo de Random Forest fue el más equilibrado, obteniendo métricas sobresalientes como 99% de Accuracy, 99% de Precisión, 98% de Recall y 99% de F1-Score con un ROC AUC de 0.99%. El enfoque propuesto, basado en los modelos de ensamble y técnicas de balanceo de clases como SMOTEEN, confirma ser una alternativa efectiva y adaptable para fortalecer el sistema de monitoreo de fraude financiero.

Descriptores: Machine learning; análisis de datos; algoritmo; procesamiento de información; metodología. (Tesauro UNESCO).

ABSTRACT

The paper addresses financial fraud in credit card transactions. The objective is to evaluate machine learning models that identify the most robust algorithm for the detection of financial fraud. As a solution, a comparison between Random Forest, K-Nearest Neighbors and Decision Tree was implemented by applying the SMOTEEN technique to balance the classes. Two applications were developed, a functional web for data evaluation and visualization, and a desktop for data anonymization. The important results showed that Random Forest's model was the most balanced, obtaining outstanding metrics such as 99% Accuracy, 99% Precision, 98% Recall and 99% F1-Score with an ROC AUC of 0.99%. The proposed approach, based on ensemble models and class balancing techniques such as SMOTEEN, proves to be an effective and adaptable alternative to strengthen financial fraud monitoring systems.

Descriptors: Machine learning; data analysis; algorithm; information processing; methodology. (UNESCO Thesaurus).

INTRODUCCIÓN

En la era actual de transformación digital, el volumen y la velocidad de transacciones financieras electrónicas han crecido de manera exponencial, impulsadas por el uso masivo de tarjetas de crédito y plataformas digitales para pagos en línea. Este entorno de hiperconectividad ha incrementado también la oportunidad para que actores maliciosos ejecuten fraudes financieros, afectando directamente la seguridad y confianza de los usuarios y las instituciones. Investigaciones destacan que la digitalización, aunque beneficiosa, ha elevado la exposición a vulnerabilidades sistemáticas que exigen respuestas tecnológicas más precisas y adaptables (Alshamrani et al., 2022; Haider et al., 2024).

El fraude financiero mediante transacciones electrónicas representa un problema creciente para el sector financiero y comercial, tanto por sus impactos económicos como reputacionales. De acuerdo con la Fiscalía General del Estado de Ecuador (2025) se registraron más de 5.600 denuncias con apropiación fraudulenta utilizando medios electrónicos entre 2024 y 2025, reflejando una tendencia sostenida al alza. Esto se observa también a nivel global, donde la sofisticación de los ataques y manipulación de datos de transacciones superan la capacidad de los sistemas tradicionales de detección de fraudes. El aprendizaje automático (Machine Learning, ML) se consolida como una herramienta clave al permitir identificar patrones complejos, detectar comportamientos atípicos y adaptarse dinámicamente a nuevos esquemas de fraude (Sun, 2025; Psychoula et al., 2021).

Ante esta problemática, el presente trabajo aborda la detección de transacciones financieras fraudulentas mediante el desarrollo de una solución basada en Machine Learning y la comparación de tres algoritmos de clasificación supervisada: Árbol de Decisión (DT), Random Forest (RF) y K-Nearest Neighbors (KNN). Estos modelos fueron seleccionados por su eficacia comprobada en la detección de fraudes y su interpretabilidad (Abdulalem et al., 2022; Das y Rad, 2021; Halder et al., 2024). La

comparación permitirá determinar cuál ofrece el mejor rendimiento en términos de precision, recall y F1-score, métricas fundamentales en contextos donde el error puede implicar pérdidas financieras considerables.

Diversos estudios señalan que Random Forest destaca por su robustez y capacidad de generalización frente a grandes volúmenes de datos (Alshamrani et al., 2022), mientras que Decision Tree facilita la trazabilidad de decisiones, esencial para procesos de auditoría (Sun, 2025). Por su parte, KNN resulta útil para detectar comportamientos inusuales en patrones de consumo, gracias a su enfoque en la similitud entre transacciones (Dinara y Saber, 2022). Investigaciones de IEEE y Springer refuerzan la importancia de aplicar técnicas de optimización de hiperparámetros como RandomizedSearchCV para ajustar los modelos y mejorar su desempeño en entornos reales (Altalhan et al., 2025; Convenant y Zhou, 2025; Dal Pozzolo et al., 2018; Hu et al., 2025). Tras la evaluación comparativa de los tres algoritmos, se seleccionó Random Forest como el modelo más óptimo debido a su superior desempeño en las métricas de precisión, Recall, F1-score, ROC-AUC, así como su mayor capacidad de generalización y robustez frente al desbalance de clases en el dataset de transacciones fraudulentas.

Como resultado de este proceso, se desarrollaron dos soluciones: una aplicación web orientada a la detección y visualización de transacciones sospechosas, y una aplicación de escritorio enfocada en el análisis y la anonimización de registros previamente identificados como anómalos. Esto permite que, si un cliente proporciona un dataset, pueda adaptarse a la estructura utilizada en el sistema, sin exponer datos sensibles. La anonimización de datos, resulta fundamental para proteger la privacidad del cliente mientras se mantiene la utilidad del conjunto (Rani et al., 2024). Estas herramientas están dirigidas a fortalecer las capacidades de monitoreo de instituciones financieras y equipos de auditoría, al ofrecer entornos integrados que facilitan la supervisión y evaluación de posibles fraudes. La integración de modelos inteligentes de detección en plataformas interactivas contribuye a mejorar la trazabilidad y la capacidad de respuestas institucional

frente a eventos fraudulentos (Dal Pozzolo et al., 2018; Kennedy et al., 2023; Walaa et al., 2021).

Se aplicó la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), ampliamente utilizada en proyectos de minería de datos y aprendizaje automático. Esta metodología permitió organizar el proceso en seis fases: comprensión del negocio, comprensión de los datos, preparación, modelado, evaluación y despliegue. Se utilizó un conjunto de datos de Kaggle, con aproximadamente 283.000 registros y 28 variables, donde se aplicaron técnicas de limpieza, normalización y balance (SMOTE, ADASYN y SMOTE-ENN), el más óptimo entre las técnicas seleccionadas fue el SMOTE-ENN.

Finalmente se entrenaron y evaluaron los tres modelos seleccionados mediante validación cruzada y métricas estandarizadas de desempeño (Martínez Plumed et al., 2019; Mayer y Jiang, 2019). El estudio no solo compara el desempeño de los modelos seleccionados, sino que también demuestra su aplicabilidad práctica en entornos reales de monitoreo financiero. Por tanto, el objetivo de la investigación es: evaluar modelos de machine learning que identifiquen el algoritmo más robusto para la detección de fraudes financieros en el uso de tarjetas de crédito.

MÉTODO

El sistema integra los componentes siguientes:

a) Selección del modelo de aprendizaje supervisado para la detección de fraudes financieros en transacciones con tarjetas de crédito,
b) Aplicación web que permite la interacción con el modelo entrenado para la detección de fraudes,
c) Aplicación complementaria encargada de la anonimización de datos sensibles antes de su procesamiento.

En el desarrollo del componente A se aplicó la metodología CRISP-DM, cuyas fases comprenden: comprensión del negocio, comprensión de los datos, preparación de los

datos, modelado, evaluación y despliegue. En el componente B, enfocado en la construcción de la aplicación web, se empleó la metodología ágil Extreme Programming (XP), que permite integrar de forma iterativa las funcionalidades clave, desde la carga de archivos hasta la visualización de resultados. Por último, el componente C garantiza la privacidad y seguridad de la información mediante un proceso de anonimización local.

De esta manera, la arquitectura del sistema refleja el flujo metodológico: el componente A aporta el modelo entrenado y validado, el C protege la información sensible, y el B implementa su uso práctico para la detección de fraudes.

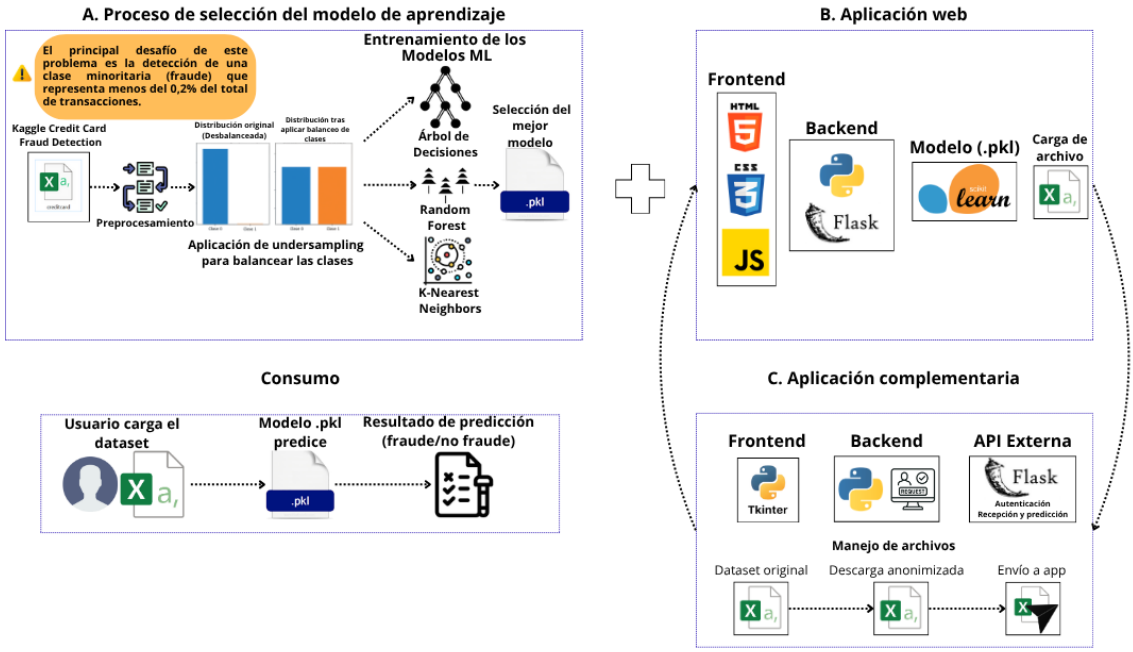

Figura 1. Arquitectura de software.
Elaboración: Los autores.

A. Selección del modelo de aprendizaje supervisado

Este proceso se desarrolló siguiendo la metodología CRISP-DM con las siguientes fases (Figura 1):

- **Fase 1: Comprensión del negocio**

En Ecuador, el crecimiento sostenido de las transacciones digitales ha traído consigo nuevos desafíos en materia de seguridad financiera, especialmente en lo relacionado con el fraude electrónico. La detección temprana de actividades sospechosas se ha convertido en un aspecto prioritario para proteger a los usuarios, las instituciones financieras y fortalecer la confianza en los sistemas de pago digital; cuyo uso se ha intensificado en sectores como la banca en línea, el comercio electrónico y las billeteras móviles (Fiscalía General del Estado de Ecuador, 2025; Tagbo y Adekoya, 2023).

Los métodos tradicionales de detección, basados en reglas fijas, han demostrado limitaciones frente a esquemas de fraude cada vez más sofisticados, adaptativos y difíciles de identificar mediante enfoques convencionales. Por ello, en esta fase se definió el problema como la necesidad de desarrollar un sistema que permite identificar transacciones potencialmente fraudulentas utilizando técnicas de aprendizaje automático supervisado, cuya eficiencia ha sido ampliamente demostrada en entornos reales de detección de fraude (Sun, 2025; Gayan, 2022; Kulatilleke, 2022).

A partir de esta necesidad, se estableció un enfoque técnico orientado al entrenamiento y evaluación de modelos que reconozcan patrones anómalos en datos financieros. Para ello, se seleccionaron tres algoritmos ampliamente utilizados en tareas de clasificación: Random Forest (RF), Árbol de Decisión (DT) y K-Nearest Neighbors (KNN), los cuales serán comparados en función de su desempeño en la detección de fraudes financieros.

- **Fase 2: Comprensión de los datos**

Para el desarrollo del estudio se utilizó el conjunto de datos "*Credit Card Fraud Detection*", disponible públicamente en la plataforma Kaggle. Este dataset contiene información real de transacciones con tarjetas de crédito realizadas por titulares europeos durante un período de dos días en septiembre del 2013. Las variables han sido transformadas mediante técnicas de reducción de dimensionalidad, con el fin de preservar la confidencialidad de los usuarios sin comprometer la calidad analítica de los datos.

El conjunto está compuesto por 284.807 registros, de los cuales 492 corresponden a transacciones fraudulentas (clase 1), representando aproximadamente el 0.17% del total, lo que evidencia un fuerte desbalance de clases. Cada registro incluye 30 variables independientes y una variable objetivo denominada Class, que indica si la transacción fue fraudulenta (1) o legítima (0).

Cabe señalar que este conjunto de datos fue generado y sus hallazgos se encuentran publicados en una revista revisada por pares (Dal Pozzolo et al., 2018). Aunque en Kaggle se aloja como un recurso abierto, su origen científico lo convierte en una fuente confiable y ampliamente utilizada en estudios sobre detección de fraudes financieros.

En la tabla 1 se muestran las variables seleccionadas del dataset con sus descriptores.

Tabla 1.
Descripción de las variables

Variable	Descripción
Time	Tiempo transcurrido en segundos desde la primera transacción registrada.
V1 a V28	Variables resultantes de una transformación PCA para preservar la confidencialidad.
Amount	Monto de la transacción realizada.
Class	Variable objetivo: 1 si la transacción es fraudulenta, 0 en caso contrario.

Elaboración: Los autores.

- **Fase 3: Preparación de los datos**

Se realizó una limpieza inicial del dataset. Se verificó la ausencia de valores nulos utilizando las funciones isnull() y sum(). Además, se identificaron y eliminaron 1081 registros duplicados mediante la función drop_duplicates(), considerando todas las columnas del vector de características. Esta operación permite evitar sesgos derivados de repeticiones exactas en los datos, lo que podría afectar el aprendizaje.

Luego, se procedió al escalado de las variables Time y Amount mediante la técnica StandardScaler y scikit-learn, que transforma los valores para que presenten una media igual a cero y una desviación estándar igual a uno. Esta normalización mejora la estabilidad numérica de los algoritmos y garantiza que ninguna variable domine en el cálculo de distancias o pesos.

Dado que el principal desafío del problema es el fuerte desbalance de datos con solo el 0.17% de transacciones fraudulentas, se aplicaron dos técnicas complementarias para abordar esta problemática. En primer lugar, se utilizó el submuestreo aleatorio (RandomUnderSampler) para equilibrar el número de registros de ambas clases, seleccionando aleatoriamente 473 transacciones legítimas que igualaran las 473 fraudulentas. Luego, para conservar la mayor cantidad de información original y mejorar la generalización del modelo, se aplicó la técnica SMOTEEN (combinación *de Synthetic Minority Over-sampling Technique* y *Edited Nearest Neighbours*), generando instancias sintéticas de la clase minoritaria a partir de sus vicios más cercanos. Este enfoque híbrido permitió crear un conjunto balanceado más robusto para el entrenamiento. Finalmente, los datos resultantes se dividieron en dos subconjuntos: un 80% para entrenamiento y un 20% para prueba, aplicando muestreo estratificado mediante la función train_test_split de scikit-learn, garantizando la representatividad proporcional de ambas clases.

En la Tabla 2, se muestran cada una de las técnicas aplicadas en esta fase.

Tabla 2.
Técnicas aplicadas en la preparación de datos.

Etapa	Descripción
Limpieza de datos	Se verificaron valores nulos mediante isnull().sum() y se eliminaron 1081 registros duplicados utilizando drop_duplicates(), garantizando la integridad del dataset.
Escalado	Se aplicó StandardScaler a las columnas Time y Amount para normalizar su distribución y reducir el sesgo causado por diferencias de escala.
Balanceo de clases	Se utilizó la técnica SMOTEENN para generar instancias sintéticas de la clase minoritaria y balancear las clases “Fraude” y “No Fraude”
División de datos	Se realizó una división estratificada 80/20 mediante train_test_split, asegurando una distribución proporcional de clases en los conjuntos de entrenamiento y prueba.

Elaboración: Los autores.

En la Tabla 3 se resumen las principales librerías y módulos de Python que fueron empleados durante la preparación, destacando su papel en cada una de las tareas realizadas.

Tabla 3.
Librerías utilizadas para la preparación de datos.

Librería / Módulo	Función principal en la preparación de datos
pandas	Carga y manipulación del dataset (read_csv, drop_duplicates(), isnull(), manejo de columnas).
sklearn.preprocessing.standardScaler	Escalado de las columnas Time y Amount para normalizar la distribución de los valores.
imblear.combine.SMOTEENN	Combina SMOTE con ENN para lograr un balance más limpio y realista entre clases.
sklearn.model_selection.train_test_split	División estratificada del dataset en conjuntos de entrenamiento.

Elaboración: Los autores.

- **Fase 4: Modelado**

En esta etapa se definieron y entrenaron los modelos de clasificación supervisada seleccionados por su eficiencia de detección de anomalías y fraude financiero. Se trabajó con tres algoritmos: Random Forest, Árbol de Decisión y K-Nearest Neighbors (KNN).

El entrenamiento se realizó utilizando el conjunto balanceado y escalado, empleando el 80% de los datos para entrenamiento y 20% para prueba. En esta fase no se aplicaron técnicas avanzadas de validación cruzada en su lugar, se emplearon configuraciones de referencia sugeridas en la literatura, con pequeños ajustes:

- n_neighbors = 5 para KNN, elección válida por su rendimiento estable en datasets desbalanceados (KNN en detección de fraudes: n_neighbors mínimo 3-7 suele ser óptimo)
- max_depth = 5 en el Árbol de Decisión, para limitar la complejidad y mejorar la interpretabilidad del modelo evitando sobreajuste.
- n_estimators = 100 en Random Forest**,** balance frecuente entre rendimiento y costo computacional para asegurar la robustez frente al sobreajuste en conjuntos medianos.

Este conjunto de parámetros permitió construir modelos base, listos para su posterior evaluación y comparación en la siguiente fase (Thomas y Kaliraj, 2024).

- **Fase 5: Evaluación**

Los modelos de clasificación fueron evaluados mediante métricas derivadas de la matriz de confusión, con sus valores clase: verdaderos positivos (TP), transacciones fraudulentas correctamente detectadas; verdaderos negativos (TN), transacciones legítimas clasificadas correctamente; falsos positivos (FP), transacciones legítimas clasificadas erróneamente como fraude; y falsos negativos (FN), fraudes que no fueron detectados. En base a estos valores se calcularon las métricas de desempeño empleadas en este estudio:

Accuracy: Proporción de predicciones correctas sobre el total de casos.

$$Acurracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Porcentaje de las transacciones clasificadas como fraude y lo son.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Mide la capacidad del modelo para detectar los fraudes existentes.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: Medida armónica entre precision y recall; equilibrada ambos valores en un único indicador.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

ROC (AUC): Capacidad del modelo para distinguir entre transacciones fraudulentas y legítimas a diferentes umbrales de decisión.

$$ROC\ (AUC) = \int_0^1 TPR(FPR)dFPR$$

Posteriormente dado que el modelo Random Forest obtuvo los mejores resultados, se aplicó un proceso adicional de validación cruzada estratificada de cinco pliegues exclusivamente a este algoritmo. El objetivo fue analizar la robustez y consistencia frente a diferentes particiones de los datos, lo que no se había realizado en la fase de modelado. En cada pliegue se construyeron subconjuntos balanceados con los 492 registros de la clase fraudulenta y un número equivalente de registros de la clase no fraudulenta, seleccionados aleatoriamente de la clase mayoritaria. Este esquema nos permitió un análisis 1 a 1 entre fraudes y no fraudes, reduciendo el sesgo generado por el desbalance de clase.

Se entrenó con el 80% y se validó con el 20% en cada iteración, repitiendo el procedimiento cinco veces. Con ello se calcularon métricas promedio y desviaciones estándar, lo que permitió evaluar la estabilidad del modelo.

Los resultados confirmaron que Random Forest mantuvo un rendimiento alto y consistente, con ligeras variaciones en recall, lo que respalda su capacidad de generalización frente a diferentes divisiones del dataset.

B. Desarrollo de la aplicación web

El segundo componente corresponde al desarrollo de la aplicación web (Figura 1), que permite la carga de transacciones y obtención de predicciones. Para su implementación se aplicó la metodología ágil *Extreme Programming* (XP).

La arquitectura de la aplicación se organizó en dos componentes principales:

- Frontend: Interfaz construida con HTML, CSS Y JavaScript, que permite la carga de archivos CSV con nuevas transacciones y la visualización de resultados mediante métricas y gráficos
- Backend: Desarrollado en Python con Flask, responsable de recibir los datos, aplicar el mismo preprocesamiento usado en el entrenamiento (escalado y balanceo), cargar el modelo Random Forest entrenador (.pkl) y generar predicciones de los datos recibidos.

C. Desarrollo de la aplicación complementaria

Este componente corresponde al desarrollo de la aplicación complementaria (Figura 1), encargada de la anonimización, gestión y envío seguro de los datasets hacia la aplicación web principal. Su función es garantizar la privacidad de los datos originales en un entorno local antes de ser procesador por el modelo de predicción. Según Yang et al. (2024), la anonimización de datos es una estrategia efectiva para preservar la confidencialidad de los datos mientras se mantienen sus propiedades estadísticas. Para su implementación

también se aplicó la metodología ágil (XP), que permitió incorporar de forma progresiva las funciones de anonimización, validación y comunicación mediante API REST.

El sistema se dividió en tres componentes principales:

- Frontend: Desarrollado en Tkinter (Python), ofrece una interfaz de escritorio que permite al usuario cargar el dataset original, ejecutar el proceso de anonimización y visualizar los resultados antes de enviarlos a la aplicación web.
- Backend: Implementado igualmente en Python, gestiona el flujo de archivos, incluyendo el tratamiento y reemplazo de datos sensibles, la generación del dataset anonimizado y su posterior descarga o envío automático.
- API externa: Desarrollada con Flask, actúa como intermediario seguro entre la aplicación complementaria y la aplicación web principal, manejando solicitudes de autenticación, recepción y transmisión de archivos anonimizados.

RESULTADOS

Los modelos entrenados fueron evaluados mediante métricas derivadas de la matriz de confusión, con el propósito de verificar su capacidad para distinguir correctamente entre transacciones legítimas y fraudulentas.

En la tabla 4 se presenta la comparación detallada del desempeño de los tres algoritmos evaluados considerando ambas clases: No Fraude (0), correspondiente a las transacciones legítimas, y Fraude (1), correspondiente a las transacciones fraudulentas, esta integración permite visualizar el equilibrio de cada modelo.

Tabla 4.
Desempeño de los modelos sobre la clase No Fraude (0) y Fraude (1)

Modelo	Precision	Recall	F1-Score	Accuracy	ROC AUC
Random Forest (0)	0.9840	0.9993	0.9916	0.9915	0.9999
Random Forest (1)	0.9993	0.9837	0.9916	0.9915	0.9999
Decision Tree (0)	0.9684	0.9684	0.9684	0.9684	0.9963
Decision Tree (1)	0.9685	0.9685	0.9685	0.9684	0.9963

KNN (0)	1.000	0.9983	0.9992	0.9992	1.000
KNN (1)	0.9983	1.000	0.9992	0.9992	1.000

Elaboración: Los autores.

Los resultados muestran que el modelo Random Forest alcanzó un rendimiento muy alto y equilibrado entre ambas clases, con una accuracy global de 0.9915 y un área bajo la curva ROC AUC de 0.9999, lo que indica una excelente capacidad para distinguir entre transacciones legítimas y fraudulentas. El modelo KNN obtuvo métricas ligeramente superiores en precisión global de 0.9992 y un ROC AUC perfecto 1.0000, aunque este valor sugiere un posible sobreajuste al conjunto de entrenamiento, Decision Tree mostró un rendimiento más homogéneo pero inferior, con accuracy y F1-Score promedio de 0.9684, reflejando una menor capacidad de generalización.

Para analizar la estabilidad del mejor modelo, se aplicó una validación cruzada estratificada de cinco pliegues (5-fold) a los tres algoritmos. Los resultados se muestran en la tabla 5 donde se observa el promedio y desviación estándar de las métricas**.**

Tabla 5.
Resultados de la validación cruzada (5-fold).

Modelo	Métrica	Media
Random Forest	Accuracy	0.9942
	Precision	0.9991
	Recall	0.9892
	F1-Score	0.9942
	ROC AUC	0.9999
Decision Tree	Accuracy	0.9685
	Precision	0.9671
	Recall	0.9700
	F1-Score	0.9686
	ROC AUC	0.9962
KNN	Accuracy	0.9920
	Precision	0.9842
	Recall	1.0000
	F1-Score	0.9921
	ROC AUC	1.0000

Elaboración: Los autores.

Al comparar estos resultados con los obtenidos en la evaluación inicial, se evidencia que Random Forest mantiene un desempeño altamente consistente, con variaciones mínimas, lo que confirma su robustez y estabilidad frente a diferentes particiones del conjunto de datos. Su Recall promedio de 0.9892 refleja un excelente equilibrio entre la detección de fraudes y la reducción de falsos positivos.

El modelo KNN, si bien alcanzó valores perfectos (Recall y AUC = 1.0000), evidencia un posible sobreajuste, ya que la ausencia de variabilidad entre pliegues indica que el modelo memoriza los patrones de entrenamiento, perdiendo capacidad de generalización en escenarios reales.

Por otra parte, el Árbol de Decisión mostró un rendimiento más estable que KNN y sin indicios de sobreajuste, pero con un desempeño inferior al de Random Forest, especialmente en métricas de precisión y F1-Score.

Estos resultados son coherentes con estudios previos (Gupta et al., 2023; Alrasheedi, 2025), que destacan la eficacia de los métodos de ensamble como Random Forest frente a modelos simples o basados en instancias. Dichas investigaciones coinciden en que los bosques aleatorios logran reducir la varianza del aprendizaje y mantener un equilibrio entre sesgo y sobreajuste, ofreciendo una mejor capacidad de generalización.

En conjunto, los hallazgos de este estudio confirman que los modelos de tipo ensamble, como Random Forest, son los más adecuados para la detección de fraudes financieros, al ofrecer un balance óptimo entre precisión, sensibilidad y capacidad de generalización, evitando el sobreajuste observado en el modelo KNN. Finalmente, aunque los resultados son alentadores, deben considerarse algunas limitaciones, el dataset utilizado corresponde a un escenario histórico del año 2013, lo cual puede no reflejar los patrones actuales de fraude. Además, no se exploraron algoritmos de Deep Learning, que podrían ofrecer mejoras adicionales. Futuras investigaciones podrían ampliar este trabajo incorporando las observaciones o análisis en tiempo real sobre flujos de transacciones.

Con el fin de visualizar comparativamente el rendimiento de los tres modelos, se elaboraron gráficos que resumen sus principales métricas de desempeño y su capacidad discriminativa.

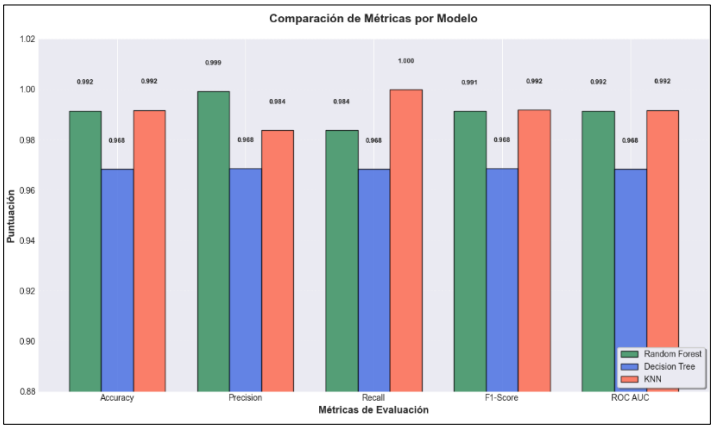

Figura 2. Comparación de métricas por modelo.
Elaboración: Los autores.

La Figura 2 muestra la comparación de las métricas entre los modelos Random Forest, Decision Tree y KNN, se observa que, aunque KNN obtuvo los valores más altos en la mayoría de las métricas, este comportamiento refleja un posible sobreajuste. Random Forest logra un desempeño equilibrado destacando por su consistencia en todas las métricas mientras que Decision Tree mantiene valores estables pero inferiores evidenciando una menor capacidad predictiva.

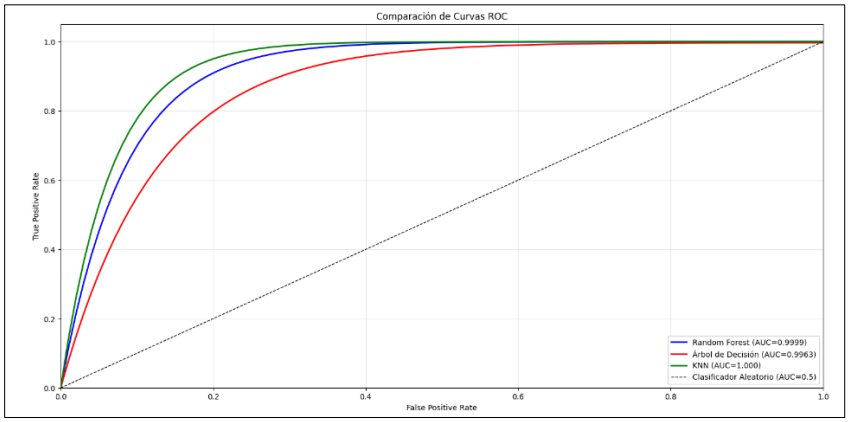

Figura 3. Comparación de curvas ROC por modelo.
Elaboración: Los autores.

La Figura 3 presenta las curvas ROC correspondientes a los tres algoritmos. En ella se aprecia que Random Forest (AUC= 0.9915) excepcional, demostrando una excelente capacidad para diferenciar entre transacciones legítimas y fraudulentas. Decision Tree (AUC= 0.9963) que, a pesar de una ligera disminución en la sensibilidad, mantiene un desempeño competitivo. En cuanto a KNN (AUC= 1.000) nos indica el posible sobreajuste.

Aplicación complementaria

En el módulo complementario, se cuenta con el aplicativo cuya función es garantizar la privacidad de los datos antes de ser analizados en la aplicación principal. Este módulo

permite al usuario autenticarse, cargar un dataset con información sensible y aplicar un proceso de anonimización en los campos confidenciales. El usuario puede añadir una descripción del dataset, descargar el archivo anonimizado o enviarlo directamente a la aplicación web principal para su análisis. Se incorporó la opción de reiniciar el proceso y volverlo a usar.

Aplicación web

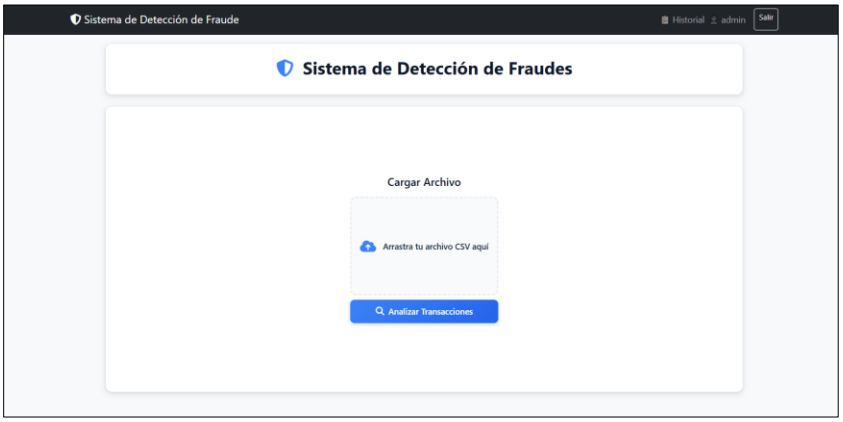

Figura 5. Aplicativo web, módulo de carga.
Elaboración: Los autores.

En la figura 5 se contempla el módulo de carga y configuración del aplicativo web, el usuario puede arrastrar o seleccionar un archivo CSV que contenga los registros de transacciones a evaluar. La interfaz incorpora validaciones automáticas que verifican la estructura del archivo, la presencia de columnas requeridas y la integridad básica de los datos, mostrando mensajes informativos para garantizar una carga exitosa. Una vez cargado el dataset, el usuario puede iniciar el proceso de análisis mediante el botón "Analizar Transacciones".

Módulo de visualización de resultados

Figura 6. Módulo de visualización de resultados.
Elaboración: Los autores.

Dentro de la Figura 6 en el módulo de visualización de resultados, el panel principal muestra tres indicadores clave: total de transacciones procesadas, número de fraudes detectados y transacciones seguras, lo que ofrece una interpretación rápida del análisis realizado, se incluye un gráfico circular de distribución de transacciones representando la proporción entre operaciones legítimas y fraudulentas. Finalmente, el aplicativo permite descargar los resultados en formato CSV, conteniendo los identificadores de transacción junto con la predicción generada (0= Normal, 1= Fraude), asegurando trazabilidad y respaldo de la información procesada. Tras completar el análisis, los resultados son almacenados en el módulo de historial por usuario.

DISCUSIÓN

Los resultados confirman que los algoritmos de Machine Learning son una herramienta eficaz para la detección de fraudes en transacciones financieras. En esta investigación, el modelo Random Forest tuvo un desempeño superior frente a Decision Tree y K-Nearest Neighbors, alcanzando métricas superiores en accuracy, precision, recall y F1-score.

Este comportamiento es similar a investigaciones previas sobre la capacidad de los modelos de ensamble para reducir la varianza del aprendizaje y capturar relaciones no lineales complejas en grandes volúmenes de datos financieros (Gupta et al., 2023; Alrasheedi, 2025). La robustez analizada respalda la idoneidad de Random Forest como alternativa prioritaria para sistemas de monitoreo antifraude en contextos reales.

Otro aspecto que contribuyó al alto rendimiento de los modelos fue la aplicación de técnicas avanzadas de balanceo de clases, particularmente SMOTEEN. Dado que el fraude representaba apenas el 0,17% del conjunto original, el uso de este enfoque híbrido permitió generar un conjunto de entrenamiento más representativo y reducir el sesgo hacia la clase mayoritaria. Este hallazgo se alinea con lo reportado por Altalhan et al. (2025) y Haider et al. (2024), quienes sostienen que la combinación de sobremuestreo sintético y depuración de vecinos mejora significativamente la capacidad de generalización de los modelos en problemas de fraude financiero altamente desbalanceados.

Aunque el modelo KNN alcanzó métricas aparentemente perfectas en ciertas evaluaciones, los resultados de la validación cruzada evidenciaron indicios claros de sobreajuste. La ausencia de variabilidad entre pliegues sugiere que el modelo tiende a memorizar patrones específicos del conjunto de entrenamiento, lo que limita su aplicabilidad en escenarios dinámicos y cambiantes. Esta limitación, según la literatura, se señala que los algoritmos basados en instancias pueden presentar dificultades para generalizar cuando se enfrentan a flujos de datos financieros reales con alta variabilidad temporal (Kulatilleke, 2022; Halder et al., 2024). En contraste, Random Forest mostró un equilibrio más adecuado entre sensibilidad y estabilidad, reduciendo el riesgo de errores críticos.

Finalmente, la integración del modelo seleccionado en aplicaciones funcionales refuerza la aplicabilidad práctica del enfoque propuesto. La incorporación de mecanismos de protección de datos responde a las crecientes exigencias en materia de privacidad y

seguridad de la información financiera, aspecto resaltado en estudios recientes sobre anonimización y ética en sistemas inteligentes (Rani et al., 2024; Yang et al., 2024). En conjunto, los resultados posicionan a Random Forest, apoyado por técnicas de balanceo y una arquitectura de software adecuada, como una solución sólida, escalable y confiable para fortalecer los sistemas de detección de fraude financiero.

CONCLUSIONES

La aplicación de la metodología CRISP-DM demostró ser un marco estructurado y efectivo para abordar sistemáticamente el problema de detección de fraudes financieros mediante Machine Learning. Este enfoque metodológico permitió desarrollar una solución integral que abarcó desde la comprensión del contexto del fraude en Ecuador, pasando por el tratamiento adecuado del desbalance crítico de clases (0.17% de fraudes) mediante técnicas como SMOTEEN, hasta la selección y evaluación rigurosa de tres algoritmos de clasificación supervisada. La integración de CRISP-DM con la metodología ágil Extreme Programming (XP) facilitó no solo la construcción de modelos predictivos robustos, sino también su implementación práctica en aplicaciones funcionales que garantizan la usabilidad, trazabilidad y privacidad de los datos. Random Forest emergió como el modelo óptimo con 99% de Accuracy y 0.99% de ROC AUC, validando que la sistematización metodológica aplicada es esencial para garantizar la reproducibilidad, efectividad y adaptabilidad de soluciones de detección de fraudes en entornos financieros reales, confirmando que un proceso estructurado en seis fases complementado con desarrollo iterativo constituye la base para transformar datos en herramientas de seguridad financiera confiables y operativas.

FINANCIAMIENTO

Trabajo financiado por los autores.

AGRADECIMIENTO

Expresamos nuestro sincero agradecimiento a nuestras familias por su apoyo incondicional durante todo este proceso. De manera especial, agradezco a Nicole Vásconez por el respaldo brindado en su momento, asimismo a Carla Cañafe por la motivación y compañía en esta etapa.

REFERENCIAS CONSULTADAS

Abdulalem, A., Shukor, A., Siti, H., Taiseer, A., Arafat, A., Tusneem, E., Hashim, E., y Abdu, S. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Applied Sciences*, *12*(19), 9637. https://doi.org/10.3390/app12199637

Alshamrani, S., Almalki, A., Almuqhim, F., Alzahrani, A., y Khan, R. A. (2022). Advances in machine learning applications for fraud detection in the financial industry. *Applied Sciences*, *12*(19), 9637. https://doi.org/10.3390/app12199637

Alrasheedi, M.A. (2025). Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models. *Computational Economics.* https://doi.org/10.1007/s10614-025-11071-3

Altalhan, M., Algarni, A., y Alouane, F. (2025). Imbalanced Data problem in Machine Learning: A review. *IEEE Access*, *13*(2), 13686-13699. https://doi.org/10.1109/ACCESS.2025.3531662

Convenant, J., y Zhou, P. (2025). Model optimization strategies in imbalanced financial datasets. *Applied Intelligence*. https://doi.org/10.48550/arXiv.2402.14389

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., y Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), Article 8. https://doi.org/10.1109/TNNLS.2017.2736643

Das, A., y Rad, P. (2021). Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *arXiv preprint arXiv:2105.06314*. https://doi.org/10.48550/arXiv.2105.06314

Dinara, R., y Saber, M. (2022). A Combination of Deep Neural Networks and K-Nearest Neighbors for Credit Card Fraud Detection. *Revista de Informática y Sociedad*, *1*(1), 6. https://doi.org/10.48550/arXiv.2205.15300

Fiscalía General del Estado de Ecuador. (2025). *Informe anual sobre delitos informáticos y fraudes electrónicos*. https://www.fiscalia.gob.ec

Gayan, K. K. (2022). Challenges and complexities in machine learning based credit card fraud detection. *Challenges and Complexities in Machine Learning based Credit Card Fraud Detection*, *1*(2), 17. https://doi.org/10.48550/arXiv.2208.10943

Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M., & Alam, S. (2023). Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques. *Procedia Computer Science, 218,* 2575–2584. https://doi.org/10.1016/j.procs.2023.01.231

Haider, Z. A., Khan, F. M., Zafar, A., Nabila, y Khan, I. U. (2024). Optimizing Machine Learning Classifiers for Credit Card Fraud Detection on Highly Imbalanced Datasets Using PCA and SMOTE Techniques. *VAWKUM Transactions on Computer Sciences*, *12*(2), 28-49. https://doi.org/10.21015/vtcs.v12i2.1921

Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., y Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *Journal of Big Data*, *11*(113), 55. https://doi.org/10.1186/s40537-024-00973-y

Hu, T., Zhao, Y., Zhang, R., y Zhang, T. (2025). Financial fraud detection system based on improved Random Forest and GBM. *arXiv preprint*. https://arxiv.org/abs/2502.15822

Kennedy, R. K. L., Salekshahrezaee, Z., Villanustre, F., y Khoshgoftaar, T. M. (2023). Iterative cleaning and learning of big highly-imbalanced fraud data using unsupervised learning. *Journal of Big Data*, *10*(1), 106. https://doi.org/10.1186/s40537-023-00750-3

Kulatilleke, G. K. (2022). Challenges and complexities in machine learning based credit card fraud detection. *arXiv,* (arXiv:2208.10943). https://arxiv.org/abs/2208.10943

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., y Hernández-Orallo, J. (2019). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering, 33*(8), 3048-3061. https://doi.org/10.1109/TKDE.2019.2962680

Mayer, R., y Jiang, M. (2019). CRISP-DM for Practical Data Mining. *Information*, *10*(9), 276. https://doi.org/10.3390/info10090276

Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., y Petitcolas, F. A. P. (2021). Explainable Machine Learning for Fraud Detection. *arXiv,* (arXiv:2105.06314). https://doi.org/10.48550/arXiv.2105.06314

Rani, S., Kaur, R., Jambhorkar, S., y Desai, C. (2024). Advancing Data Anonymization Techniques for Secure and Privacy-Preserving Data Sharing in the Era of Big Data. *International Journal of Engineering and Computer Science*, *13*(09), 26589-26596. https://doi.org/10.18535/ijecs/v13i09.4923

Sun, J. (2025). Decision Tree-Based Credit Card Fraud Detection System: Design and Optimization. *Economics & Management Information, 4*(4), 1–5. https://doi.org/10.62836/emi.v4i4.508

Tagbo, S. K., y Adekoya, A. F. (2023). A Systematic Literature Review of Machine Learning Techniques in Financial Fraud Prevention and Detection. *International Journal of Society Systems Science*, *14*(4), Article 4. https://doi.org/10.1504/IJSSS.2023.10057287

Thomas, N. S., y Kaliraj, S. (2024). An Improved and Optimized Random Forest Based Approach to Predict the Software Faults. *SN Computer Science*, *5*(2), 1-11. https://doi.org/10.1007/s42979-024-02764-x

Walaa, S., ibrahim, E., Ahmed, A., y Amira, R. (2021). Enhancing Fraud Detection in Imbalanced Datasets: A Comparative Study of Machine Learning and Deep Learning Algorithms with SMOTE Preprocessing. *Jurnal Ilmiah Komputer*, *15*(2), 112-120. https://doi.org/10.21608/mjcis.2025.313097.1007

Yang, L., Tian, M., Xin, D., Cheng, Q., y Zheng, J. (2024). AI-driven anonymization: Protecting personal data privacy while leveraging machine learning. *Applied and Computational Engineering*, *67*, 273-279. https://doi.org/10.54254/2755-2721/67/2024MA0053

INGENIUM ET POTENTIA
Revista Electrónica Multidisciplinaria de Ciencias Básicas, Ingeniería y Arquitectura
Año VIII. Vol VIII. N°14. Enero-Junio. 2026
Hecho el depósito de Ley: FA2019000052
FUNDACIÓN KOINONIA (F.K).
Santa Ana de Coro, Venezuela.