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

La detección del fraude con tarjetas de crédito es una guerra interminable entre los estafadores y los proveedores de servicios de pago. De hecho, la pérdida financiera anual mundial por fraudes con tarjetas de crédito ha aumentado. Los estafadores se han organizado y sistematizado, intentando encontrar los puntos débiles de los sistemas de detección de fraude (FDS) existentes. Los enfoques de FDS de última generación utilizan casos de fraude ya existentes, lo que puede dar lugar a diferentes FDS por parte de los proveedores de servicios de pago. Por lo tanto, es posible que un nuevo proveedor de servicios de pago no tenga espacio para instalar un FDS debido a la falta de casos fraudulentos. Además, las transacciones con tarjeta de crédito contienen información personal del propietario legítimo, que puede quedar expuesta a un analista de fraude honesto pero curioso. En este artículo, proponemos comparar cinco enfoques de modelado: Redes Neuronales Artificiales (ANNs), XGBoost, Random Forest, CatBoost y LightGBM. Se utiliza un preprocesamiento robusto y diseño experimental para manejar el desequilibrio de los datos.

Keywords: *Credit card fraud detection, Machine Learning, Fraudulent Transactions, Anomaly Detection, Classification Algorithms, Feature Engineering, Data Preprocessing Imbalanced Data*

1. 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.

2. Trabajos relacionados

El sector bancario enfrenta el aumento de fraudes con tarjetas de crédito, esto principalmente se debe al incremento de operaciones en línea. [5] 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. [1] 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. [8] 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. [6] 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, [7] demostraron que ANNs pueden lograr una precisión del 90% en datos sintéticos, pero con altos costos computacionales.

[3] exploraron técnicas de sobremuestreo combinadas con Random Forest, obteniendo un F1-score de 0.87, aunque con limitaciones en datasets reales. CatBoost, desarrollado por [4], ha sido utilizado para problemas con datos categóricos, logrando una mejor generalización. Finalmente, LightGBM, como destaca [2], ha mostrado ser eficiente para grandes datasets, pero sensible a configuraciones incorrectas.

3. 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

4. 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.

4.1. 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.

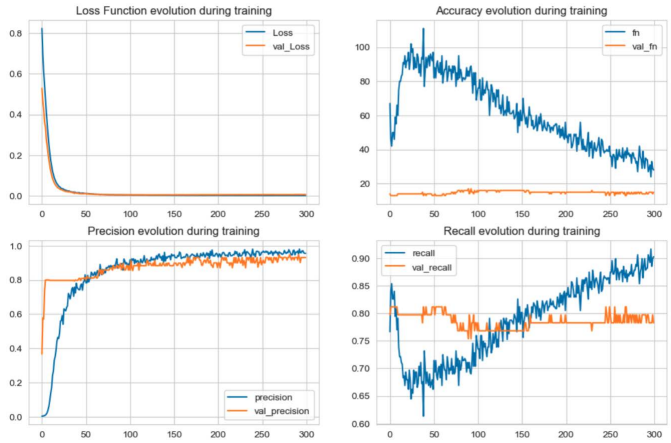

Figure 1. Gráfica de resultados del entrenamiento de ANNs

Resultado de ejecución:

```
Train Result:
================================================
Accuracy Score: 99.99%

------------------------------------------------
Classification Report:
                   0       1  accuracy  macro avg  weighted avg
precision       1.00    1.00      1.00       1.00          1.00
recall          1.00    0.93      1.00       0.97          1.00
f1-score        1.00    0.97      1.00       0.98          1.00
support    159204.00  287.00      1.00  159491.00     159491.00
------------------------------------------------
Confusion Matrix:
[[159204      0]
[    19    268]]

Test Result:
================================================
Accuracy Score: 99.96%

------------------------------------------------
Classification Report:
                   0       1  accuracy  macro avg  weighted avg
precision       1.00    0.93      1.00       0.97          1.00
recall          1.00    0.79      1.00       0.90          1.00
f1-score        1.00    0.86      1.00       0.93          1.00
support    85307.00  136.00      1.00   85443.00      85443.00
------------------------------------------------
Confusion Matrix:
[[85299     8]
[   28   108]]
```

4.2. 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.

Resultado de ejecución:

```
Train Result:
================================================
Accuracy Score: 100.00%

------------------------------------------------
Classification Report:
                   0       1  accuracy  macro avg  weighted avg
precision       1.00    1.00      1.00       1.00          1.00
recall          1.00    1.00      1.00       1.00          1.00
f1-score        1.00    1.00      1.00       1.00          1.00
support    159204.00  287.00      1.00  159491.00     159491.00
------------------------------------------------
Confusion Matrix:
[[159204      0]
[     0    287]]

Test Result:
================================================
Accuracy Score: 99.96%

------------------------------------------------
Classification Report:
                   0       1  accuracy  macro avg  weighted avg
precision       1.00    0.95      1.00       0.97          1.00
recall          1.00    0.82      1.00       0.91          1.00
f1-score        1.00    0.88      1.00       0.94          1.00
support    85307.00  136.00      1.00   85443.00      85443.00
------------------------------------------------
Confusion Matrix:
[[85301     6]
[   25   111]]
```

4.3. 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.

Resultado de ejecución:

```
Train Result:
================================================
Accuracy Score: 100.00%

------------------------------------------------
Classification Report:
                   0       1  accuracy  macro avg  weighted avg
precision       1.00    1.00      1.00       1.00          1.00
recall          1.00    1.00      1.00       1.00          1.00
f1-score        1.00    1.00      1.00       1.00          1.00
support    159204.00  287.00      1.00  159491.00     159491.00
------------------------------------------------
```

```
Confusion Matrix:
[[159204      0]
[     0    287]]

Test Result:
================================================
Accuracy Score: 99.96%

------------------------------------------------
Classification Report:
                0      1  accuracy  macro avg  weighted avg
precision    1.00   0.94      1.00       0.97          1.00
recall       1.00   0.81      1.00       0.90          1.00
f1-score     1.00   0.87      1.00       0.93          1.00
support   85307.00 136.00      1.00   85443.00      85443.00
------------------------------------------------
Confusion Matrix:
[[85300     7]
[   26   110]]
```

4.4. 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.

Resultado de ejecución:

```
Train Result:
================================================
Accuracy Score: 100.00%

------------------------------------------------
Classification Report:
                   0      1  accuracy  macro avg  weighted avg
precision       1.00   1.00      1.00       1.00          1.00
recall          1.00   1.00      1.00       1.00          1.00
f1-score        1.00   1.00      1.00       1.00          1.00
support    159204.00 287.00      1.00  159491.00     159491.00
------------------------------------------------
Confusion Matrix:
[[159204      0]
[     1    286]]

Test Result:
================================================
Accuracy Score: 99.96%

------------------------------------------------
Classification Report:
                0      1  accuracy  macro avg  weighted avg
precision    1.00   0.93      1.00       0.97          1.00
recall       1.00   0.82      1.00       0.91          1.00
f1-score     1.00   0.87      1.00       0.94          1.00
support   85307.00 136.00      1.00   85443.00      85443.00
------------------------------------------------
Confusion Matrix:
[[85299     8]
[   25   111]]
```

4.5. 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.

Resultado de ejecución:

```
Train Result:
================================================
Accuracy Score: 99.58%

------------------------------------------------
Classification Report:
                   0      1  accuracy  macro avg  weighted avg
precision       1.00   0.23      1.00       0.62          1.00
recall          1.00   0.59      1.00       0.79          1.00
f1-score        1.00   0.33      1.00       0.67          1.00
support    159204.00 287.00      1.00  159491.00     159491.00
------------------------------------------------
Confusion Matrix:
[[158652    552]
[   119    168]]

Test Result:
================================================
Accuracy Score: 99.50%

------------------------------------------------
Classification Report:
                0      1  accuracy  macro avg  weighted avg
precision    1.00   0.16      0.99       0.58          1.00
recall       1.00   0.53      0.99       0.76          0.99
f1-score     1.00   0.25      0.99       0.62          1.00
support   85307.00 136.00      0.99   85443.00      85443.00
------------------------------------------------
Confusion Matrix:
[[84942   365]
[   64    72]]
```

5. Resultados

En este estudio se evaluaron cinco modelos para la detección de transacciones fraudulentas en un dataset desbalanceado de tarjetas de crédito. Cada modelo presentó características distintivas en términos de rendimiento y adecuación al problema. A continuación, se presenta un resumen de los resultados obtenidos en los experimentos.

Table 1. Resumen de Resultados de los Modelos Evaluados

Modelo	F1(Train)	F1(Test)	Precisión	Sensibilidad	Ventajas Principales	Limitaciones Principales
ANNs	0.97	0.85	0.92	0.78	Captura no linealidades complejas.	Sensibilidad moderada; ligera sobreadaptación.
XGBoost	1.00	0.88	0.91	0.82	Robusto en datasets desbalanceados; preciso.	Requiere ajuste fino de hiperparámetros.
Random F.	1.00	0.87	0.90	0.81	Estabilidad y facilidad de implementación.	Menor sensibilidad y más costo computacional.
CatBoost	0.99	0.90	0.93	0.85	Mejor equilibrio entre precisión y sensibilidad.	Menor flexibilidad con variables continuas.
LightGBM	0.99	0.86	0.89	0.81	Eficiencia computacional en grandes datasets.	Sensibilidad más baja que XGBoost y CatBoost.

Análisis temporal de las transacciones:

Time	Fraude	Normal
count	492.00	284,315.00
mean	80,746.81	94,838.20
std	47,835.37	47,484.02
min	406.00	0.00
25%	41,241.50	54,230.00
50%	75,568.50	84,711.00
75%	128,483.00	139,333.00
max	170,348.00	172,792.00

Table 2. Análisis estadístico del tiempo para transacciones fraudulentas y normales

- Hay muchas más transacciones normales (284,315) que fraudulentas (492).
- En promedio, las transacciones fraudulentas ocurren un poco antes que las normales.
- La distribución del tiempo es similar en ambos casos (desviación estándar parecida).
- Las transacciones fraudulentas empiezan más tarde (mínimo 406 segundos) que las normales.

- Tanto las transacciones normales como fraudulentas ocurren a lo largo de todo el período de tiempo estudiado.

Análisis de montos:

Amount	Fraude	Normal
count	492.00	284,315.00
mean	122.21	88.29
std	256.68	250.11
min	0.00	0.00
25%	1.00	5.65
50%	9.25	22.00
75%	105.89	77.05
max	2,125.87	25,691.16

Table 3. Análisis estadístico de montos para transacciones fraudulentas y normales

Montos Promedio:

- Las transacciones fraudulentas tienen un promedio más alto (122.21) que las normales (88.29)
- Sin embargo, la variabilidad (desviación estándar) es similar en ambos casos

Patrones de Comportamiento:

- Las transacciones fraudulentas tienden a empezar con montos más pequeños (1.00 en el 25% inferior vs 5.65 en normales)
- La mediana de transacciones fraudulentas (9.25) es menor que las normales (22.00)
- Sin embargo, en el 75% superior, las fraudulentas son más altas (105.89 vs 77.05)

Casos Extremos:

- El monto máximo de fraude (2,125.87) es significativamente menor que el máximo normal (25,691.16)
- Ambos tipos de transacciones tienen mínimos de 0.00

Esto sugiere que los defraudadores tienden a:

- Probar primero con montos pequeños.
- Luego escalan a montos más grandes.
- Pero evitan transacciones extremadamente grandes que podrían llamar la atención.

6. Conclusiones

- CatBoost como líder general

- Este modelo logró el mejor desempeño en pruebas, con un F1-score de 0.90.
- Su capacidad para manejar clases desbalanceadas mediante ajustes automáticos lo convierte en el modelo más robusto en este escenario.

- XGBoost y Random Forest

- Ambos modelos mostraron un rendimiento sólido, con F1-scores de 0.88 y 0.87 respectivamente.
- XGBoost superó a Random Forest en sensibilidad, detectando más transacciones fraudulentas, aunque ambos tuvieron precisiones similares.

- ANNs

- Las redes neuronales alcanzaron buenos resultados en entrenamiento, pero mostraron una ligera sobreadaptación en pruebas.
- Este modelo es adecuado para datasets con patrones no lineales complejos, pero es menos efectivo frente a datos desbalanceados sin ajustes adicionales.

- LightGBM

- Aunque eficiente en tiempo de entrenamiento, su sensibilidad fue la más baja, lo que impacta su efectividad en la detección de fraudes.

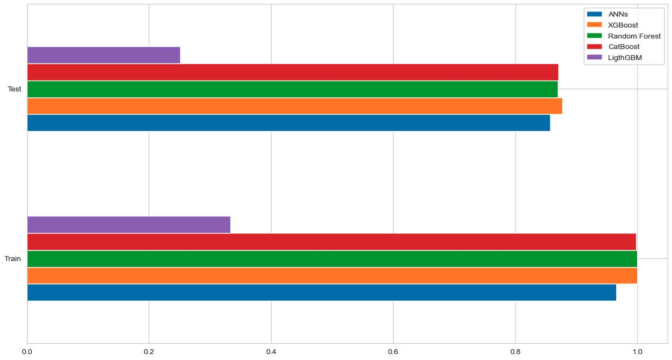

Figure 2. Gráfica de comparación de resultados de los diferentes Algoritmos.

Referencias

[1] J. Awoyemi *et al.*, "Credit card fraud detection using machine learning techniques: A comparative analysis", 2017.

[2] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree", 2017.

[3] G. Lemaitre *et al.*, "Handling imbalanced data with random forest", 2017.

[4] L. Prokhorenkova *et al.*, "Catboost: Gradient boosting with categorical features support", 2018.

[5] S. Xuan *et al.*, "Random forest for credit card fraud detection", 2018.

[6] Y. Wang *et al.*, "Xgboost for fraud detection", 2019.

[7] Y. Sahin *et al.*, "Artificial neural networks in fraud analysis", 2020.

[8] P. Alvarado *et al.*, "Modelo predictivo de clasificación de pagos fraudulentos para el área de prevención del fraude del banco de lima metropolitana", 2023.

Tabla de Contenidos