Un primer acercamiento a un modelo predictivo ajustable por umbrales para detección de fraudes financieros. *

Fabián Frola[1], Carlos Alvez[1], and Carlos Chesñevar[2]

[1] Facultad de Ciencias de la Administración – Universidad Nacional de Entre Ríos
Av. Tavella 1424 (E3202KAC), Concordia, Entre Ríos, Argentina
fabian.frola@uner.edu.ar, carlos.alvez@uner.edu.ar
[2] Instituto de Ciencias e Ingeniería de la Computación (ICIC CONICET UNS)
Universidad Nacional del Sur, San Andrés 800, 8000 Bahía Blanca, Argentina.
cic@cs.uns.edu.ar

Resumen El fraude en el sector financiero en transacciones con tarjetas de crédito y débito es un fenómeno que ha recibido el estudio de la comunidad científica por su impacto económico, tanto en individuos como en instituciones. Analizar este problema desde la perspectiva de machine learning es un gran desafío por la poca disponibilidad de transacciones etiquetadas y el desbalanceo en la proporción de clases. En este trabajo exploramos un enfoque alternativo basado en el ajuste del umbral de probabilidad del algoritmo de fraude. A través de experimentaciones mostramos que este abordaje es eficiente y constituye una alternativa válida para detectar fraude de forma efectiva.

Keywords: Fraude financiero · Inteligencia Artificial · Machine Learning

1. Introducción y motivaciones

Desde hace varios años el fraude en el sector financiero en transacciones con tarjetas de crédito y débito ha tenido un crecimiento exponencial, afectando la economía de los individuos, instituciones financieras y la banca en general [17,10,8]. Defenderse de este problema es particularmente desafiante en este dominio, dada la poca disponibilidad de transacciones etiquetadas disponibles (situaciones de fraude comprobables por investigadores o analistas de datos) y la clasificación altamente desequilibrada asociada al problema. Cabe mencionar que en este escenario la tasa (o *ratio*) de eventos que hacen posible detectar fraude suele ser del 0.17 % (lo que equivale usualmente a una transacción fraudulenta por cada 1000 transacciones genuinas).

* Este trabajo fue realizado con el apoyo de la Universidad Nacional de Entre Ríos (UNER), la Universidad Nacional del Sur (UNS) y el CONICET.

La construcción de modelos para fraude financiero a través de Machine Learning (ML) presenta desafíos poco frecuentes en otros dominios. Cuando disponemos de datos que no están equilibrados en relación al balance de clase, los algoritmos estándares de ML tienden a maximizar la precisión general (exactitud) tendiendo a clasificar todas las observaciones como instancias de la clase mayoritaria [15]. Esto se traduce en una baja capacidad de predicción (*recall*) para la clase de nuestro interés, que en el caso concreto de fraudes financieros corresponde a la clase minoritaria (esto es, aquellas transacciones que son fraudulentas).

En este trabajo analizamos un enfoque específico para abordar este problema utilizando el cómputo de un umbral para probabilidades de ocurrencia. Este enfoque lo encuadraremos como un componente específico dentro de una arquitectura genérica para detección de fraudes financieros, mostrando empíricamente que el acercamiento presentado resulta competitivo.

El resto del artículo está organizado de forma que sea auto-contenido y se estructura como sigue: primeramente en la Sección 2 se presenta una visión general de la arquitectura propuesta para detección de fraudes financieros, identificando sus principales componentes y el rol de los mismos. Se brinda un análisis de las herramientas disponibles para caracterizar el desempeño de las tareas de clasificación en este contexto. En la Sección 3 se presenta el abordaje propuesto para caracterizar el modelo predictivo, así como el algoritmo en pseudocódigo para construir el mismo. La Sección 4 sintetiza los resultados empíricos obtenidos que muestran la performance resultante para nuestro modelo, comparada con otros abordajes alternativos. La Sección 5 presenta un análisis del trabajo relacionado y su conexión con nuestra propuesta. Finalmente, la sección 6 resume las conclusiones obtenidas y presenta algunas líneas de trabajo futuro.

2. Arquitectura del Framework SDF y conceptualización del modelo predictivo

En un trabajo preliminar [6] hemos abordado la problemática de los sistemas de detección de fraude en transacciones financieras desde una perspectiva de alto nivel. Esto resultó en la definición de un framework denominado SDF (Sistema de Detección de Fraude, Fig. 1), siendo de interés abordar en este trabajo los aspectos concernientes a la construcción del modelo predictivo y sus desafíos.

El framework SDF apunta a identificar los distintos componentes distinguidos y su relación cuando se trata de detectar transacciones fraudulentas. Como punto de partida tenemos datos de distintas transacciones financieras (ej. registros correspondientes a transacciones de tarjetas de crédito o débito de diferentes clientes de un banco). Dichos registros están conformados por atributos tales como fecha y hora de la operación, identificador del comercio, identificación de pago en cuotas, monto y número de cuotas, etc. (ver Figura 2 y 3) . Dentro de dichos datos transaccionales se asume que se cuenta con una cierta cantidad de transacciones que han sido identificadas como *fraudulentas*, y que proveen la base para poder realizar una clasificación supervisada. Típicamente el número de

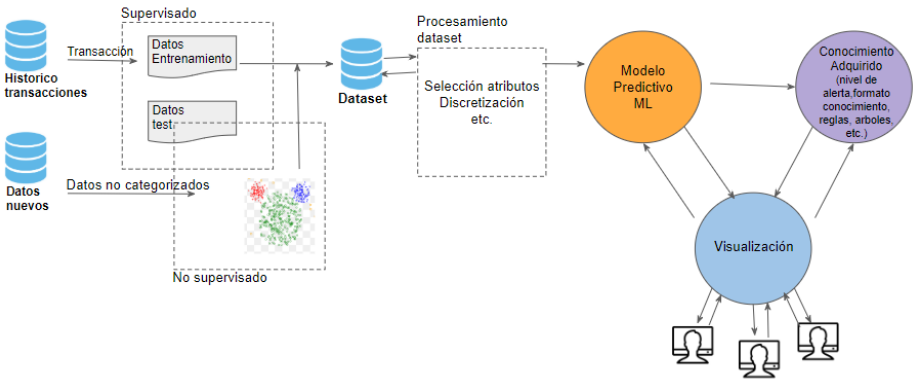

Figura 1: Arquitectura general del Framework SDF y principales componentes

estas transacciones es extremadamente bajo en relación al total de transacciones disponibles (lo que provee un fuerte desbalance de clases, como se discutirá más adelante). Estos datos transaccionales son los que permiten configurar un dataset con diferentes atributos que debe luego procesarse adecuadamente a través de las distintas técnicas existentes en aprendizaje automatizado (discretización, eliminación de atributos no significativos, etc.) para poder construir un *modelo predictivo* para capturar situaciones de fraude. El modelo resultante puede tener diferentes formalizaciones (conjuntos de reglas, árboles de decisión, etc.) y será provisto al tomador de decisión en el ámbito financiero a cargo de determinar si hay una situación de fraude o no. Dicho modelo debe poder visualizarse adecuadamente a fin de comunicar de manera conceptualmente clara el criterio utilizado para definir el espacio de decisiones.

En la arquitectura de framework propuesta se observan tres componentes principales identificables como nodos sobre la derecha de la figura: el *modelo predictivo*, el *conocimiento adquirido* y la *visualización* de los resultados vinculados a ambos nodos. En nuestro caso, focalizaremos nuestro análisis en el modelo predictivo, dado que la contribución central de este artículo se centra en ese aspecto.

Construcción del Modelo predictivo. Comenzaremos detallando las características de uno de los elementos centrales: el conjunto de datos (dataset). Para su construcción, es preciso un abordaje desde dos perspectivas, datos relevantes asociados al conjunto de transacciones históricas y la información de la transacción actual. El conjunto histórico de transacciones posee distintos elementos relevantes (ver análisis en [25]). Particularmente podemos identificar los atributos indicados en la Figura 2.

Al analizar la transacción actual (esto es, aquella para la cual queremos determinar si se trata de un fraude o no), también encontramos atributos relevantes (ver análisis en [18,25,26]). En nuestro caso estos atributos se sintetizan en la Fig.

Nombre	Tipo	Descripción
Id	Alfanumérico	Hash sobre número de cuenta primario (PAN)
Marca Tiempo	Numérico	Fecha-Hora de última transacción de la tarjeta
Ultimo Valor	Float	Valor de la última transacción de la tarjeta
Media Valor Transacción	Float	Media de valor de transacción del último mes de la tarjeta
Media de Tiempo	Float	Media de tiempo en segundos entre transacciones consecutivas de la misma tarjeta
Comercio Conocido	Boolean	1 si corresponde a un comercio ya utilizado por la tarjeta, 0 en otro caso
Media Fraude	Float	Media de fraude, para todas las tarjetas, en las últimas 50.000 transacciones

Figura 2: Atributos relevantes (archivo histórico de transacciones)

3. Debe notarse que las transacciones relevantes para nuestro análisis se pueden identificar con el campo MTI 0200 (asociado al protocolo de comunicaciones de mensajes financieros según el estándar ISO 8583-1:2003 (ISO 8583) [7]).

Campo	Tipo	Descripción
Campo 2	LLVAR (19)	Número de cuenta primario (PAN)
Campo 4	NUMERICO (12)	Monto de la transacción
Campo 11	NUMERICO (6)	Número de auditoría
Campo 12	NUMERICO (6)	Hora local terminal
Campo 13	NUMERICO (MMDD)	Fecha local terminal
Campo 22	NUMERICO (4)	Modo entrada terminal
Campo 25	NUMERICO (2)	Código condición de POS
Campo 35	LLVAR	Track 2 (dato sensible)
Campo 38	ALFA (6)	Identificación de autorización
Campo 39	ALFA (2)	Código de respuesta
Campo 41	ALFA (8)	Identificación de terminal
Campo 42	ALFA (15)	Identificación de comercio
Campo 49	ALFA (3)	Moneda de la transacción
Fraude	BOOLEAN	1 si ha sido reconocida como fradue, 0 caso contrario

Figura 3: Atributos significativos de la transacción actual para su consideración como posible fraude

Metodología

Selección de atributos. En este proceso se identifican los atributos relevantes a utilizar para la construcción de modelos. En ML el abordaje se realiza conforme a dos categorías principales de técnicas: métodos de envoltura y métodos de filtro [11]. Los métodos de envoltura agregan o eliminan atributos basados en el rendimiento de clasificación de un atributo elegido, seleccionando aquella combinación que logra la menor tasa de error. Alternativamente los modelos de filtro

no utilizan ningún modelo de predicción para evaluar los atributos; la relevancia de éstos se determina evaluando únicamente los mismos fuera de cualquier clasificador. En caso de contar con atributos ruidosos (*noisy*) se puede utilizar un método de envoltura (para eliminación de atributos en forma recursiva) y de filtro (para atributos basados en correlación). Como mencionamos anteriormente, uno de los principales desafíos en la construcción del modelo predictivo es el abordaje adecuado del problema del fuerte desbalance de clases (un análisis más pormenorizado sobre este tema puede verse en [10,15,1]). A continuación analizaremos algunas alternativas en este sentido:

Alternativas para manejar el desequilibrio de clases en predicción de fraudes. Diversos enfoques han sido propuestos para lograr balancear el conjunto de datos. El muestreo podría ser un posible enfoque para superar este problema; el objetivo es alterar la distribución de la clase minoritaria o la clase mayoritaria a fin de lograr aproximadamente una distribución uniforme entre ellas [22]. Esto sin embargo puede conducir al clasificador a otra situación no deseable como ser el sobreajuste (*"overfitting"*) o subajuste (*"underfitting"*), lo que redunda en una falencia de nuestro modelo a la hora de generalizar el comportamiento que pretendemos que adquiera.

Algunas de las variantes posibles para abordar el desequilibrio de clases es la combinación de técnicas de muestreo, las técnicas de *Synthetic Minority Oversampling Technique (SMOTE)* (acrónimo de *Synthetic Minority Oversampling Technique*), así como otras variantes tales como la incorporación de la noción de costo (*Cost-Sensitive Cosine Similarity K-Nearest Neighbors (CoSKNN)*) y la variante *K-modes Imbalance Classification Hybrid Approach (K-MICHA)*)[10], entre otros. Todos estos enfoques propuestos tienen diversas ventajas y desventajas, y en esencia modifican la distribución real de los datos.

Como veremos existe otro enfoque que no afecta la distribución y logra un buen desempeño ajustando la parametrización de los algoritmos en el umbral de probabilidad de decisión de una clase u otra, es decir se ajusta más a la realidad sugiriendo que la probabilidad de ocurrencia en un entorno de clasificación binaria (fraude o no fraude) no tiene la misma probabilidad de ocurrencia.

2.1. Evaluación de los problemas de clasificación

En un problema de clasificación se evalúa la capacidad de los algoritmos de predecir la clase correcta de una nueva observación y generalmente se evalúa en términos del error medio de clasificación errónea o *Mean Misclassification Error (MME)*. La evaluación para la mínima expresión de un problema de clasificación se refleja en una matriz de confusión (ver Tabla 1).

Indicadores de Performance

Los indicadores de rendimiento se utilizan para representar con la mayor precisión posible cómo el modelo se comportaría cuando esté en uso [9]. Algunos

Tabla 1: Matriz de Confusión.

	Positivo (Fraude predicho)	Negativo (Normal predicho)
Positivo (Fraude real)	Verdaderos Positivos (TP)	Falsos Negativos (FN)
Negativo (Normal)	Falsos Positivos (FP)	Verdaderos Negativos (TN)

de los indicadores más populares son *recall* y *precision*. Ambos reflejan aspectos claves que son importantes para medir el rendimiento de un modelo.

El indicador *recall* mide la cantidad de casos que se predijeron como positivos que de hecho deberían ser positivo, se describe en (fórmula(1)). La *precision* da el porcentaje de verdaderos positivos como una proporción sobre todos los casos que deberían haber sido verdaderos (fórmula (2)). Por otro lado, *accuracy* describe el porcentaje general de muestras que reciben predicción correcta (fórmula (3)). El valor predictivo negativo (NPV) corresponde al porcentaje de cuántos verdaderos negativos (TN) hay en el conjunto del total de negativos que se lograron en la predicción (fórmula (4)).

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$NPV = \frac{TN}{TN + FN} \tag{4}$$

El *F-Score* es la media armónica entre la *precision* y el *recall*, definido como se indica en (fórmula (5)) y tiene un valor simbólico especial en nuestro dominio, dado que mide los errores que se puede cometer al considerar la clase positiva (fraude). Al maximizar la *precision* minimizamos los falsos positivos (FP); al maximizar el *recall* se minimizan los falsos negativos (FN).

$$F\text{-}Score = \frac{(2 * Precision * Recall)}{Precision + Recall} \tag{5}$$

$$F\beta\text{-}Score = \frac{\left(\left(1 + \beta^2\right) * Precision * Recall\right)}{(\beta^2 * Precision + Recall)} \tag{6}$$

En síntesis las métricas de mayor aporte para la evaluación en escenarios desbalanceados que muestran de mejor forma el desempeño de clasificación en la clase minoritaria (fraude) son (*recall*, *precision*) y las ponderaciones (*F2-score*, *G-means*, *NVP*) [14,20]. La elección de *F2-score*, remplazando $\beta = 2$ en (fórmula (6)) [4], se realiza dado que son más relevantes los casos de falsos negativos, asumiendo el detrimento de generar algunas falsas alertas que podrán ser descartadas por los evaluadores de forma posterior.

3. Ajuste del modelo predictivo: consideraciones y abordaje utilizado

En el contexto de nuestro modelo predictivo, existen algunas consideraciones generales que debemos tener en cuenta: (I) analizar los indicadores de precisión general del modelo predictivo puede no ser lo más conveniente para escenarios desbalanceados; (II) estudiar problemas con datos desequilibrados utilizando los clasificadores producidos por algoritmos de aprendizaje automático estándar (sin ajustar el umbral de probabilidad) puede ser un error crítico, tal como ha sido enunciado tempranamente en [16]; (III) identificar otras características relevantes (algo que también se reconoce en la literatura) de los datos del mundo real que, en conjunto con el desequilibrio de clase, pueden potencialmente degradar el rendimiento de los algoritmos de aprendizaje (complejidad de los datos, subconceptos, disyunciones, escasez, superposición, datos ruidosos, ejemplos límites y datos imperfectos) [5].

En el primero de los escenarios basarnos en la *accuracy* podría explicitar excelentes resultados del modelo, logrando un 99 % de éxito de clasificación [19]. Sin embargo, esto puede significar que ese uno por ciento de fraude que pretendemos encontrar (siendo extremistas) puede que ni siquiera haya sido detectado. Para esto debemos concentrarnos en las métricas que evalúan el desempeño de la clase minoritaria (el fraude), es decir, ***precision*** y ***recall*** o los indicadores que balancean este desempeño o sesgan la prioridad según los intereses.

En lo que hace a la segunda consideración se trata de asignar la "justa" probabilidad de ocurrencia de una clasificación. En escenarios de clasificación binaria como este (fraude vs. no-fraude), un acercamiento simplista podría ser en equiparar el problema a obtener un resultado al arrojar una moneda y calcular la probabilidad de que salga cada uno de los lados (es decir, existe un 50 % de probabilidad de ocurrencia de cada uno de sus lados). En nuestro dominio la probabilidad de ocurrencia claramente no es así. En conclusión, sin importar el algoritmo final que elegimos para clasificar debemos corregir hacia una manera más acorde qué probabilidades reales tiene de ocurrencia el suceso (en un rango $0 < p < 1$, donde cero significa que no hay posibilidad de fraude, y uno que todos los casos son fraudes).

La forma de definir esta probabilidad p de ocurrencia de manera imparcial (como parámetro del algoritmo de detección de fraude) es buscando un *"umbral"* de probabilidad, es decir, localizar el límite en el cual el algoritmo deja de lado el crecimiento en el desempeño de clasificación y comienza a degradarse en la resolución de la clasificación. Muchos algoritmos de ML utilizan una probabilidad para discernir la etiqueta con la cual clasificar.

La contribución de este trabajo aborda tres líneas de interés, cuya sinergia es necesaria para un abordaje adecuado del problema:

- Considerar la detección del umbral p. Para esto debe realizarse una iteración del modelo para cada valor de probabilidad p (ver Fig. 4) utilizando un

umbral de probabilidades en el rango 0 hasta 1 con un paso de crecimiento de $\epsilon = 0{,}001$.[3]
- Validar con los algoritmos actualmente existentes según el estado del arte y contrastar con otras propuestas alternativas.
- Seleccionar las métricas correctas para dominios altamente desbalanceados. Como mostramos anteriormente, trabajar con dominios de problemas con un alto desbalance de clases obliga a consideraciones especiales que no se aplican en problemas de clasificación de ML en general.

Clasificadores para el modelo predictivo.

Seguidamente detallamos las tres técnicas de ML que utilizamos para construir modelos predictivos, encontrando en cada caso distintos valores de umbral p y un valor $\epsilon = 0{,}001$.

Clasificadores Random Forests (RF): Los árboles de decisión son muy populares en la minería de datos, especialmente por la simplificación del algoritmo y la flexibilidad de los tipos de datos de los atributos. Su deficiencia es que son potencialmente sensibles al conjunto de datos de entrenamiento y pueden sufrir sobreajuste [24]. Los *Random Forest* (RF) buscan paliar esta situación, combinando múltiples predictores, donde cada predictor es un árbol de decisión. Cada árbol depende de un conjunto de datos independientes y todos los árboles del bosque respetan la misma distribución. Los RF han sido muy utilizados para tratar con conjuntos de datos desequilibrados [12,22]. Es importante señalar que aumentar el tamaño del bosque mejora el resultado en líneas generales, pero si la cantidad de árboles es alta (ej. más de 400) el algoritmo tiende a sobreajustar [21]. A nivel de tiempo computacional es rápido (inclusive en tiempo real) y escalable para trabajar con datasets masivos [12].

Regresión logística Logistic Regression (LR): La regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.

Clasificadores Boosting: Boosting es un método que pretende mejorar el desempeño de cualquier algoritmo de aprendizaje supervisado mediante la combinación de los resultados de varios clasificadores débiles o de base para obtener un clasificador final robusto. El clasificador de árbol *Gradient Boosted Tree (GBT)* es una colección de modelos de clasificación y regresión, buscando mejorar la precisión del árbol. En particular, *Extreme Gradient Boosting* (XGBoost) funciona con todo tipo de conjunto de datos [3].

[3] El valor de ϵ es uno de los parámetros del algoritmo. El valor 0.001 fue usado para las experimentaciones realizadas.

Pseudocódigo del Algoritmo para la construcción del modelo predictivo.

Entrada: Dataset con atributos de transacciones individuales y del histórico de la tarjeta.
Salida: Modelo entrenado con el umbral óptimo de probabilidad ajustado.

Preparación de datos:
Sea: X dataset de $\{x_1, x_2, \ldots x_{n+m}\}$
$a_1, a_2, \ldots, a_k$ atributos en x_i y c_i etiqueta de clase [1 - fraude | 0 -no fraude] para X_i

Dividir el conjunto de datos

X_train $\{x_1, x_2, \ldots x_n\}$: Datos de entrenamiento: 70 %

X_test $\{x_1, x_2, \ldots x_m\}$: Datos para testing del modelo: 30 %

//Reemplazo de valores nulos por la media aritmética:

$\forall a_i$ = null entonces $a_i = 1/n \sum_{Xi}^{Xn} a_i$

Entrenamiento del modelo:

1. Ajustar el modelo en el conjunto de datos de entrenamiento.
   1.1-Seleccionar Algoritmo de ML a utilizar
   1.2-Configurar meta parámetros
   1.3-Entrenar el modelo.

2. Algoritmo 1: Predecir las probabilidades

```
prob[]  //vector tipo real
//Para toda instancia en X_test, calcular probabilidad que sea fraude
Para i ← 1 Hasta m Hacer
      //probabilidad de fraude
      p = probabilidad que c_i sea fraude en x_i
      prob[i] = p
Fin Para
```

3. Algoritmo 2: Obtener el umbral de probabilidad

```
score[]  //vector tipo real
umbral_optimo =0
max_score=0
Para t ← 0 Hasta 1 con Paso 0.001 Hacer
      Para i ← 1 Hasta m // X_test
            Si prob[i] >= t Entonces
                  c_i en x_i = 1 //fraude
            Sino
                  c_i en x_i = 0 //no fraude
            Fin Si
      Fin Para
      score[t] = f2-score en X_test
      Si score[t]> max_score Entonces
            max_score =  score[t]
            umbral_optimo = t
      End Si
Fin Para
```

4. Utilizar el umbral adoptado (variable: umbral_optimo) para predecir la clase en datos nuevos.

Figura 4: Algoritmos para la construcción del modelo predictivo utilizando un umbral p. Dado un modelo de ML se obtiene un umbral de probabilidad óptimo para realizar posteriormente la predicción definitiva

Implementación del modelo predictivo. En este trabajo utilizamos scikit-learn [13] como librerías de alto nivel (escritas en Python) para validar nuestra propuesta. En la configuración de RF se eligió una parametrización con 53 árboles para la estimación, considerando acelerar el proceso de aprendizaje en las distintas iteraciones. Para el caso de *Gradient boosting o Potenciación del gradiente* (Gradient boosting) y XGBoost luego de varias pruebas de laboratorio quedaron configurados con 300 árboles de estimación, la profundidad máxima de 4 para los estimadores de regresión individual, un mínimo de dos para la división interna de nodos y un índice de aprendizaje de 0.01 para la contribución de cada árbol.

4. Resultados obtenidos

Configuración del experimento: El conjunto de datos que utilizamos ha sido recopilado durante una investigación colaborativa entre Worldline y Machine Learning Group (http://mlg.ulb.ac.be) de la ULB (Université Libre de Bruxelles) [23]. A efectos de la evaluación de nuestra propuesta, este conjunto de datos presenta un fuerte desequilibrio de clases, con 492 fraudes sobre un total de 284.807 transacciones. Los resultados de los distintos modelos se muestran en la Fig. 7. Se puede observar que las tasas de *accuracy* (precisión) son altas, generalmente alrededor del 99.95 %. Sin embargo, este no es un resultado totalmente representativo, ya que la tasa de la detección de fraude varía desde un piso de 71 % para modelos con bajo desempeño como LR o más precisamente desde 82 % para XGBoost, logrando un desempeño hasta de 93 % para RF.

En el análisis de la curva ROC (ver Figura 5(a)), demuestra que se puede obtener el umbral de probabilidad para la mejor opción (esto es, *accuracy*) con el significado magro que hemos discutido en secciones anteriores. En el caso del fraude es preferible analizar la curva ROC para una mejor *precisión* y *recall* (ver Figura 5(b)). Se identifica en ambas gráficas un círculo en color negro que corresponde con el punto en el que se maximiza el valor para el indicador *F2-Score*; se aprecia una abrupta caída en *precisión* en este punto, determinando así el valor para el umbral de probabilidad que maximiza la identificación de casos de fraude (*recall*).

5. Trabajos Relacionados

En [10] se brinda una importante revisión de la literatura y el estado del arte en el desarrollo de algoritmos para detección de fraude. Se destaca en tal sentido una exhaustiva exploración de algoritmos basados en distintas técnicas de ML tales como *Artificial Neural Networks* (ANN) (redes neuronales), *Support Vector Machines* (SVM) (máquinas de soporte vectorial), etc. No obstante, el desempeño logrado en estos algoritmos (en lo que respecta a su precisión) es limitado. En resumen, los enfoques de clasificación desequilibrados y el número de falsas alarmas que generan suele ser mayor que el número de fraudes que se detectan. Como alternativas se proponen distintas variantes basadas en la

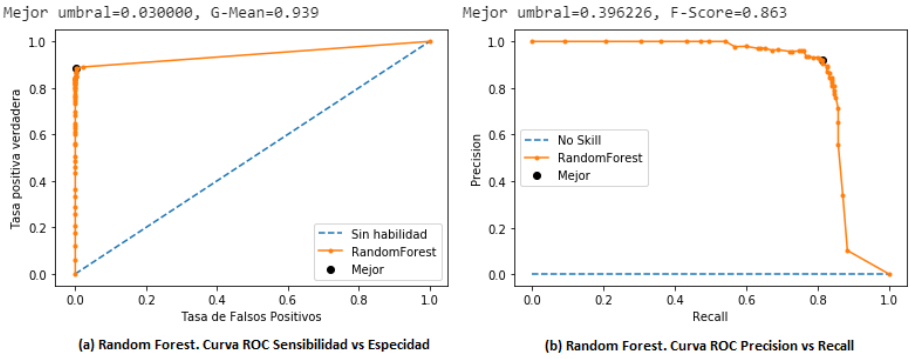

Figura 5: Random Forest. Curva ROC

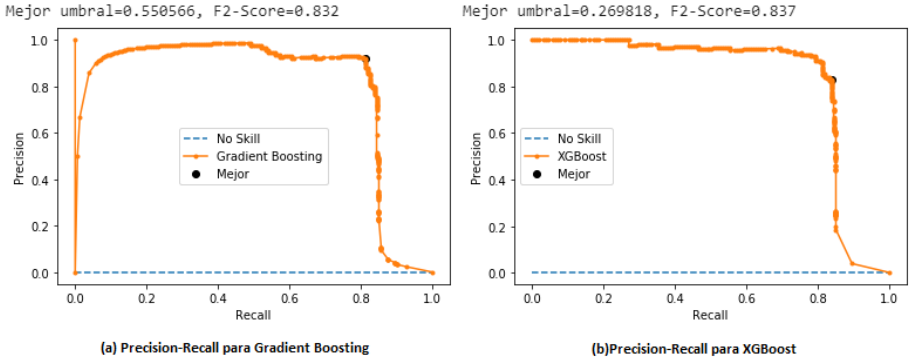

Figura 6: Gradient Boosting y XGBoost. Curva ROC (*Precision* vs *Recall*)

incorporación de nociones de costos (*CoSKNN* y la variante *K-MICHA*) que no mejoran sustancialmente las métricas (*accuracy:* 95.3, *recall:* 63 % y *F1Score:* 0.62) en un escenario similar. También se identifica la importancia de elegir el umbral de probabilidad como crucial en problemas de fraude con tarjetas de crédito, pero no se propone un algoritmo concreto para obtener el mismo como en este trabajo.

En lo referido a la construcción de agrupamientos (*clusters*) para caracterizar la noción de fraude podemos mencionar el trabajo realizado en [2], donde se trabaja en la conjunción de algoritmos supervisados y no supervisados. Este trabajo se centra en el análisis de patrones contextuales para detectar valores atípicos, sin mención del problema del desbalanceo. Adicionalmente, los enfoques que se basan en el balanceo de clases con técnicas tales como SMOTE, *Random Under Sampling* (RUS) y *Balanced Bagging Ensemble* (BBE) muestran también pobres resultados de precisión (en torno al 51 %) [2,1], introduciendo sesgos propios de datos sintéticos.

Nro.	Algortimo	Umbral	*Accuracy*	*Precision*	*Recall*	F2-score
$\lambda 1$	Regresión Logística	0.5	99.89	71.63	65.16	-
$\lambda 2$	Gradient Boosting	0.5	99.95	90.65	81.29	-
$\lambda 3$	Random Forest	0,5	99.95	93.13	78.71	-
$\lambda 4$	Random Forest	0.396226	99.95	92.59	80.65	-
$\lambda 5$	Gradient Boosting	0.550566	99.95	91.97	81.29	0.832
$\lambda 6$	XGBoost	0.269818	99.94	82.80	**83.87**	0.837

Figura 7: Evaluación de Algoritmos

6. Conclusiones y trabajo futuro

En este trabajo hemos analizado una técnica que ha resultado particularmente atractiva para poder identificar con mayor nivel de certeza a las transacciones fraudulentas a partir de ajustar el umbral de probabilidad del algoritmo de detección de fraude. Esta técnica la hemos evaluado en el marco del framework SDF (Sistema Detección de Fraude). En el contexto de nuestra investigación estos resultados son preliminares pero promisorios en cuanto a su aplicación efectiva en el contexto de su aplicación en el mundo real.

Es necesario realizar un estudio más profundo sobre la elección de los parámetros de los algoritmos y aplicar mecanismos de optimización, a fin de mejorar los resultados en general para la clasificación. Se presentó una forma de elegir el umbral de probabilidad que proporcionó mejores resultados en escenarios con fuerte desbalance de clases, validado en conjunto de datos públicos estudiados exhaustivamente desde múltiples propuestas. Las evaluaciones empíricas realizadas nos muestran resultados satisfactorios a partir del enfoque utilizado.

Como parte de nuestro trabajo futuro avanzaremos en una clasificación taxonómica detallada de los distintos abordajes posibles que pueden realizarse, identificando aquellos aspectos de cada abordaje que resulten más atractivos para modelar el identificado de fraudes.

Referencias

1. Beigi, S., Aminnaseri, M.: Credit card fraud detection using data mining and statistical methods. Journal of AI and Data Mining (2019)
2. Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Oblé, F., Bontempi, G.: Combining unsupervised and supervised learning in credit card fraud detection. Information sciences (2019)
3. Dhankhad, S., Mohammed, E.A., Far, B.: Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study. In: 2018 IEEE International Conference on Information Reuse and Integration, IRI 2018, Salt Lake City, UT, USA, July 6-9, 2018. pp. 122–125 (2018). https://doi.org/10.1109/IRI.2018.00025, https://doi.org/10.1109/IRI.2018.00025
4. Eban, E.E., Schain, M., Mackey, A., Gordon, A., Saurous, R.A., Elidan, G.: Scalable learning of non-decomposable objectives. arXiv preprint arXiv:1608.04802 (2016)

5. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Data intrinsic characteristics. In: Learning from Imbalanced Data Sets, pp. 253–277. Springer (2018)
6. Frola, F., Chesñevar, C.I., Alvez, C.E., Etchart, G., Miranda, E., Ruiz, S., Aguirre, J.J., Teze, J.C.: Framework sdf machine learning en transacciones financieras y detección temprana de fraudes. In: XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan). (2019)
7. International Organization for Standardization (ISO): Financial transaction card originated messages — interchange message specifications — part 1: Messages, data elements and code values. https://www.iso.org/obp/ui/iso:std:iso:8583:-1:ed-1:v1:en (1998)
8. KPMG: Global banking fraud survey - the multi-faceted threat of fraud: Are banks up to the challenge ? https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/05/global-banking-fraud-survey.pdf (2019)
9. M., R., T., H.: Fraud detection on unlabeled data with unsupervised machine learning (June 2018)
10. Makki, S.: An Efficient Classification Model for Analyzing Skewed Data to Detect Frauds in the Financial Sector. Ph.D. thesis, Université de Lyon; Université libanaise (2019)
11. Moepya, S.O., Nelwamondo, F.V., Twala, B.: Increasing the detection of minority class instances in financial statement fraud. In: Intelligent Information and Database Systems - 9th Asian Conference, ACIIDS 2017, Kanazawa, Japan, April 3-5, 2017, Proceedings, Part II. pp. 33–43 (2017). https://doi.org/10.1007/978-3-319-54430-4_4, https://doi.org/10.1007/978-3-319-54430-4_4
12. Mohammed, R.A., Wong, K.W., Shiratuddin, M.F., Wang, X.: Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study. In: PRICAI 2018: Trends in Artificial Intelligence - 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28-31, 2018, Proceedings, Part II. pp. 237–246 (2018). https://doi.org/10.1007/978-3-319-97310-4_27, https://doi.org/10.1007/978-3-319-97310-4_27
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011)
15. Pozzolo, A.D.: Adaptive Machine Learning for Credit Card Fraud Detection. Ph.D. thesis, Machine Learning Group, Computer Science Department. Université Libre de Bruxelles (December 2015)
16. Provost, F.: Machine learning from imbalanced data sets 101. In: Proceedings of the AAAI'2000 workshop on imbalanced data sets. vol. 68, pp. 1–3. AAAI Press (2000)
17. PwC's Global Economic Crime and Fraud Survey: 2020 fighting fraud: A never-ending battle. https://www.pwc.com/gx/en/forensics/gecs-2020/pdf/global-economic-crime-and-fraud-survey-2020.pdf (2020)
18. Randhawa, K., Loo, C.K., Seera, M., Lim, C.P., Nandi, A.K.: Credit card fraud detection using adaboost and majority voting. IEEE Access **6**, 14277–14284 (2018). https://doi.org/10.1109/ACCESS.2018.2806420, https://doi.org/10.1109/ACCESS.2018.2806420

19. de Sá, A.G.C., Pereira, A.C.M., Pappa, G.L.: A customized classification algorithm for credit card fraud detection. Eng. Appl. of AI **72**, 21–29 (2018). https://doi.org/10.1016/j.engappai.2018.03.011, https://doi.org/10.1016/j.engappai.2018.03.011
20. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PloS one **10**(3) (2015)
21. Soh, W.W., Yusuf, R.M.: Predicting credit card fraud on a imbalanced data. International Journal of Data Science and Advanced Analytics **1**(1), 12–17 (2019)
22. Sohony, I., Pratap, R., Nambiar, U.: Ensemble learning for credit card fraud detection. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2018, Goa, India, January 11-13, 2018. pp. 289–294 (2018). https://doi.org/10.1145/3152494.3156815, http://doi.acm.org/10.1145/3152494.3156815
23. U.M.L. Group, Editor, U.M.L.G.: European credit card dataset. https://www.kaggle.com/mlg-ulb/creditcardfraud (2013)
24. Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., Jiang, C.: Random forest for credit card fraud detection. In: 15th IEEE International Conference on Networking, Sensing and Control, ICNSC 2018, Zhuhai, China, March 27-29, 2018. pp. 1–6 (2018). https://doi.org/10.1109/ICNSC.2018.8361343, https://doi.org/10.1109/ICNSC.2018.8361343
25. Zanin, M., Romance, M., Moral, S., Criado, R.: Credit card fraud detection through parenclitic network analysis. CoRR **abs/1706.01953** (2017), http://arxiv.org/abs/1706.01953
26. Zanin, M., Romance, M., Moral, S., Criado, R.: Credit card fraud detection through parenclitic network analysis. Complexity **2018**, 5764370:1–5764370:9 (2018). https://doi.org/10.1155/2018/5764370, https://doi.org/10.1155/2018/5764370