# Edición de Conjuntos de Entrenamiento no Balanceados, haciendo uso de Operadores Genéticos y la Teoría de los Conjuntos Aproximados

Enislay Ramentol Martínez[1], Francisco Herrera[2], Rafael Bello Pérez[3] , Yailé Caballero Mota[1], Yanet Sánchez López[4]

*Resumen*—**Numerosas técnicas se han desarrollado para hacer frente al problema de las clases no balanceadas en el aprendizaje automático. Estas técnicas han sido divididas en 2 grandes grupos: las que están al nivel de los algoritmos y las que están al nivel de los datos, dentro de las que se destacan las que se centran en intentar balancear los conjuntos, reduciendo la clase con mayor cantidad de ejemplos, o ampliando la de menor cantidad, conocidas como under-sampling y over-sampling respectivamente. En este trabajo se hace una nueva propuesta para la edición de éste tipo de conjuntos, a través de la construcción de nuevas instancias, usando para ello técnicas basadas en operadores genéticos y la teoría de los Conjuntos Aproximados. El método propuesto ha sido validado experimentalmente haciendo uso de bases de datos Internacionales.**

*Palabras clave*-- **No balanceados, teoría de conjuntos aproximados, aprendizaje automatizado.**

## I. INTRODUCCIÓN

El aprendizaje a partir de datos no balanceados es uno de los desafíos que actualmente está enfrentando el aprendizaje automático, debido al mal funcionamiento de los algoritmos frente a conjuntos de éste tipo. La ocurrencia de sucesos poco frecuentes ha dado lugar a que exista una desproporción considerable entre el número de ejemplos en cada clase, lo que se conoce como *clases no balanceadas o desbalanceadas*. En numerosas situaciones aparece desbalance entre las clases, dentro de las que sobresalen: diagnóstico de enfermedades con condiciones médicas poco frecuentes como tiroides [1], detección de llamadas telefónicas fraudulentas [2], detección de derrames de petróleo a partir de imágenes de radar [3], recuperación y filtrado de información [4], monitoreo de fallas en la caja de velocidades de helicópteros durante el vuelo [5], entre otras.

[1]Departamento de Computación, Universidad de Camagüey, Circunvalación Norte Km 5 ½ (Cuba) (enislay.ramentol, yaile.caballero){@reduc.edu.cu}

[2] Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S. Ing. Informática, Avda. Andalucía, 18, 18071 Granada (España) herrera@decsai.ugr.es

[3] Centro de Estudios de Informática, Universidad Central de Las Villas Carretera a Camajuaní Km 5 ½ (Cuba) rbellop@uclv.edu.cu

[4] División de RadioCuba Camagüey, Calle Martí entre San Pablo y Apodaca (Cuba) yanetsl@gmail.com

Los clasificadores logran muy buenas precisiones con la clase más representada (mayoritaria), mientras que en la menos representada (minoritaria) ocurre todo lo contrario. En los no balanceados el conocimiento más novedoso suele residir en los datos menos representados, sin embargo muchos clasificadores pueden considerarlos como rarezas o ruido, pues los mismos no tienen en cuenta la distribución de los datos, únicamente se centran en los resultados de las medidas globales. Es por ello que muchas de las medidas que tradicionalmente son usadas para medir la calidad de un clasificador son consideradas inadecuadas en el contexto de las desbalanceadas. Actualmente se ha logrado encontrar medidas para medir el desempeño de los clasificadores en el contexto de las desbalanceadas, tales como la curva ROC.

Como resultado de las investigaciones, han surgido numerosas técnicas, las mismas han sido agrupadas en 2 categorías:

1. Las que están al nivel de los algoritmos de aprendizaje, las cuales no modifican la distribución de los datos, se centran en el ajuste de coste por clase, ajuste de la estimación de probabilidad en las hojas del árbol de decisión, aprender de una sola clase, entre otras.
2. Las que modifican la distribución de los datos, conocidas también como remuestreo, es decir, pueden reducir la clase mayoritaria eliminando ejemplos, a través de la técnica conocida como under-sampling, o pueden aumentar la clase minoritaria creando nuevos ejemplos con la técnica conocida como over-sampling.

En éste trabajo se realiza una nueva propuesta para realizar over-sampling, construyendo nuevos ejemplos haciendo uso de algoritmos genéticos, y garantizando la calidad de estas nuevas instancias a través de técnicas de edición basadas en la Teoría de los Conjuntos Aproximados, (Rough Set Theory RST).

La contribución ha sido estructurada de la siguiente forma, la sección 2 realiza una revisión de las principales técnicas para tratar con el desbalance al nivel de los datos, la sección 3 referida a la propuesta, la sección 4 describe los operadores de

cruce utilizados en la propuesta, la sección 5 describe el algoritmo que se utilizó para evaluar la calidad de los nuevos individuos basado en la RST, la sección 4 dedicada a la experimentación y comparación con otros algoritmos seguida por las conclusiones, y las secciones 5 y 6 están dedicadas a las conclusiones y trabajos futuros.

## II. MÉTODOS DE REMUESTREO

Los métodos de remuestreo, también conocidos como métodos de preprocesado de conjuntos de entrenamiento, pueden ser divididos en 3 grandes grupos: los que eliminan instancias de la clase mayoritaria (under-sampling), los que generan nuevas instancias de la clase minoritaria (over-sampling) o la hibridación de ambas. A continuación son descritos algunos de los métodos más conocidos.

### *A. Métodos de Over-Sampling*

Entre las estrategias más conocidas para la generación de nuevas instancias con el fin de balancear conjuntos de entrenamiento se encuentra SMOTE, Synthetic Minority Over- Sampling TEchnique, propuesto en 2002 por Chawla y colaboradores [6], éste algoritmo para cada ejemplo de la clase minoritaria introduce ejemplos sintéticos en la línea que une al elemento con sus 5 vecinos más cercanos. Sin embargo esta estrategia presenta el problema de que puede introducir ejemplos de la clase minoritaria en el área de la mayoritaria, es decir, crear malos ejemplos que posteriormente pudieran confundir a los clasificadores.

En el 2005 son realizadas 2 nuevas propuestas de SMOTE [7], se presenta borderline-SMOTE1 y el borderline-SMOTE2, ambos generan instancias en la frontera entre las clases, es decir son etiquetados los elementos de la minoritaria situados muy cercanos a la mayoritaria como *Peligrosos* y a partir de ellos y sus vecinos se comienzan a generar las nuevas instancias, lográndose muy buenos resultados.

En el 2006 Cohen y colaboradores [8] proponen el AHC, Agglomerative Hierarchical Clustering Based, donde a partir de la creación de grupos enlazados usando un algoritmo jerárquico aglomerativo de agrupamiento, se seleccionan los centroides de cada grupo como un nuevo elemento sintético y finalmente son insertados en el conjunto original.

### *B. Métodos de Under-Sampling*

Dentro de los métodos más clásicos para realizar under-sampling se encuentra: el RU (Random Under-Sampling), que selecciona de manera aleatoria instancias de la clase mayoritaria para ser eliminados sin reemplazamiento hasta que ambas clases queden balanceadas y el BU (Bootstrap Under- Sampling) que funciona de manera muy parecida al RU pero con reemplazamiento.

En el NCR (Neighborhood Cleaning Rule), propuesto en [9] para cada elemento del conjunto de entrenamiento se buscan sus 3 vecinos más cercanos, si el elemento seleccionado es de la clase mayoritaria y los 3 vecinos son de la minoritaria, entonces se elimina el elemento; si el elemento pertenece a la clase minoritaria entonces se eliminan las vecinos que sean de la mayoritaria

Otros métodos muy usados son también Tomek Link [10], el One-Sided Selection propuesto en [11] que es una hibridación entre el Tomek Link y el CNN, el ENN [12] el cual elimina los elementos ruidosos del conjunto original, es decir, un elemento es eliminado si es mal clasificado por su 3 vecinos más cercanos

### *C. Métodos híbridos.*

A pesar de que tanto el Over-Sampling como el Under-Sampling logran buenos resultados por separado muchos investigadores del área han obtenido magníficos resultados hibridando ambos métodos, de éstos pudiéramos citar:

- SMOTE-Bootstrap Hybrid: éste método inicialmente genera nuevas instancias haciendo uso de SMOTE y luego reduce la clase mayoritaria a través de Bootstrap, hasta lograr que las clases queden con similar número de instancias, [13].
- AHC-KM Hybrid: éste método es una combinación de estas 2 técnicas, primeramente se generan nuevas instancias de la clase minoritaria con el AHC-based y luego se eliminan instancias de la mayoritaria con el KM- based [8].
- SMOTE- Tomek Hybrid: inicialmente se realiza el over-sampling con la clase minoritaria y luego se aplica el Tomek Link a ambas clases[14].

## III. ALGORITMO DE EDICIÓN EDITOSRS

En éste trabajo se realiza una nueva propuesta para balancear conjuntos de entrenamiento no balanceados, creando nuevos ejemplos y garantizando a la vez que éstos pertenezcan con certeza a su clase.

**Algoritmo EditOSRS**

**P1**. Para cada instancia de la clase minoritaria, se elimina el atributo de decisión y se conforma un cromosoma con los n rasgos.

**P2**.Se fija una posición (comenzando por cero) y se comienza a cruzar con el resto (de dos en dos) hasta que la cantidad de instancias generadas sumadas con la cantidad de las originales logre balancear el conjunto, si llega a la última instancia y aún no está balanceado el conjunto, se incrementa la posición y se repite el proceso. Debe guardarse la

posición de la última instancia que fue cruzada, y para cada descendiente la posición de sus padres.

Los operadores de cruce que se aplican son:

**p2.1**. Para rasgos con valores reales se propone el uso del operador de cruce para codificación real centrado en los padres, PBX-α presentado en[15].

**p2.2**. Para rasgos con valores discretos se propone el operador HUX.

*Nota: con cada unos de éstos operadores serán generados 2 nuevos individuos*.

**P3**. Evaluar la calidad de los nuevos individuos a través de la extensión del algoritmo Edit3RS propuesto en [16], de la siguiente forma.

**p3.1** Para la clase minoritaria calcular su aproximación inferior e insertarla en el conjunto solución.

**p3.2** Hallar la frontera y para cada elemento de esta calcular la función de pertenencia aproximada para cada una de las clases:

- si el valor de la función es mayor para la clase mayoritaria entonces esa instancia es desechada,
- si por el contrario es mayor para la clase minoritaria se inserta en la solución,
- las instancias cuyo valor de la función de pertenencia a ambas clases es parecido son eliminadas, porque no existe certeza de su pertenencia a ninguna de las 2 clases.

**P4**. Si al concluir el paso 3, el conjunto no quedara balanceado, se vuelve al paso 1, y se comienza a cruzar a partir de la posición en la que se quedó

Algunos conjuntos presentan un índice de desbalance tan elevado que es imposible balancearlo aplicando éste algoritmo solamente sobre los ejemplos originales en la clase minoritaria, es decir:

Siendo *N* la cantidad de ejemplos en la clase minoritaria, la cantidad máxima de ejemplos que podrán crearse será *N*(N-1)*, en conjuntos donde la proporción sea, por ejemplo de 3313 y 26 (Abalone 19), no será posible crear las 3287 instancias necesarias para balancear el conjunto. Es por ello que se propone para éstos casos, una vez cruzados todos los ejemplos del conjunto original, comenzar a cruzar descendientes que no tengan los mismos padres, garantizando la diversidad del conjunto resultante, éste proceso se repetiría hasta 3 veces.

### *A. Algoritmo de Evaluación de las nuevas instancias basado en RST.*

La Teoría de Conjuntos Aproximados (Rough Sets Theory) fue introducida por Z. Pawlak en 1982 [17]. Se basa en aproximar cualquier concepto, un subconjunto duro del dominio como por ejemplo, una clase en un problema de clasificación supervisada, por un par de conjuntos exactos, llamados aproximación inferior y aproximación superior del concepto. Con esta teoría es posible tratar tanto datos cuantitativos como cualitativos, y no se requiere eliminar las inconsistencias previas al análisis; respecto a la información de salida puede ser usada para determinar la relevancia de los atributos, generar las relaciones entre ellos (en forma de reglas), entre otras [18-27]. La inconsistencia describe una situación en la cual hay dos o más valores en conflicto para ser asignados a una variable.

La aproximación de un conjunto X⊆U, usando una relación de inseparabilidad R, ha sido inducida como un par de conjuntos llamados aproximaciones R-inferior y R-superior de X. Se considera en esta investigación una definición de aproximaciones más general, la cual maneja cualquier relación reflexiva R'. Las aproximaciones R'-inferior ($R'_*(X)$) y R'-superior ($R'^*(X)$) de X están definidas respectivamente como se muestra en las expresiones (1) y (2).

$$R'_*(X) = \{x \in X : R'(x) \subseteq X\} \quad (1)$$

$$R'^*(X) = \bigcup_{x \in X} R'(x) \quad (2)$$

Teniendo en cuenta las expresiones definidas en 1 y 2 se define la región límite de X para la relación R' [28]:

$$BN_B(X) = R'^*(X) - R'_*(X) \quad (3)$$

Si el conjunto BNB es vacío entonces el conjunto X es exacto respecto a la relación R'. En caso contrario, $BN_B(X) \neq \phi$, el conjunto X es inexacto o aproximado con respecto a R'.

El uso de relaciones de similitud ofrece mayores posibilidades para la construcción de las aproximaciones; sin embargo, se tiene que pagar por esta mayor flexibilidad, pues es más difícil desde el punto de vista computacional buscar las aproximaciones relevantes en este espacio mayor [29].

Dentro de la Teoría de los Conjuntos Aproximados el significado de la Aproximación Inferior de un sistema de decisión es de gran interés para la edición de conjuntos de entrenamiento, en esta se agrupan aquellos objetos que con absoluta certeza pertenecen a su clase lo que garantiza que los objetos contenidos en la Aproximación Inferior están exentos de ruidos[16].

El algoritmo Edit3RS presentado en [16], realiza una propuesta para dar tratamiento a los objetos ubicados en la frontera, a través del cálculo de la función de pertenencia aproximada, la cual cuantifica el grado de solapamiento relativo entre *R'*(*x*) (clase de similitud de *x*) y la clase a la cual el objeto *x* pertenece.

Inicialmente se determinan ambas aproximaciones (inferior $B_*$ y superior $B^*$), se calcula la frontera ($B^* - B_*$) y luego a cada elemento de la frontera se le calcula la función de pertenencia aproximada para cada una de las clases, descrita por la expresión:

$$\mu_X(x) = \frac{|X \cap R'(x)|}{|R'(x)|} \quad (4)$$

Donde $X$ son los objetos de la clase y $R'(x)$ los objetos parecidos a él, es decir los que tienen valores de atributos parecidos. La función de pertenencia aproximada puede ser interpretada como una estimación basada en frecuencias, es decir, la probabilidad condicional de que el objeto $x$ pertenezca al conjunto $X$ [30].

Teniendo en cuenta el valor de la función de pertenencia aproximada, la instancia va a ser asignada a la clase a la que pertenezca con mayor grado. El conjunto solución será la unión de los elementos en la aproximación inferior de la clase y los objetos de la frontera que hayan sido reetiquetados para dicha clase.

*B. Sobre el uso de componentes evolutivas en la generación de instancias: Operadores de cruce*

La presente propuesta está basada en la generación de instancias haciendo uso de los operadores de cruce de los algoritmos genéticos, considerando el proceso como una única iteración del algoritmo genético sin operador de mutación. En esta línea, una idea similar es la generación de nuevas reglas en los modelos Michigan de aprendizaje de reglas, donde se genera una nueva regla a partir de reglas de calidad del modelo para sustituir a las reglas que han ido perdiendo confianza [31].

El algoritmo propuesto ha sido diseñado para operar con valores de los rasgos tanto discretos como continuos, es por ello que han sido utilizados 2 operadores de cruce, uno para valores discretos y otro para valores continuos, los cuales se detallan a continuación.

***Operador de cruce HUX para valores discretos***

Éste operador genera por cada cruce 2 nuevos hijos los cuales heredan de sus padres los rasgos que éstos tienen iguales, luego cada hijo hereda la mitad de los rasgos en que sus padres difieren.

***Operador de cruce Parent-Centric para valores continuos***

El operador de cruce Parent-Centric (PX-α) o centrado en los padres, propuesto en [15] es una extensión del operador BLX-α [32]. Éste operador tiene entre sus principales ventajas: que existe mayor probabilidad de crear los descendientes muy cerca de sus padres que en cualquier otro lugar del espacio de búsqueda, que el grado de diversidad que puede inducir es fácilmente ajustado a través de la variación del parámetro α asociado y que asigna a los hijos soluciones proporcionales al espacio de las soluciones de los padres, por lo que le brinda la posibilidad mostrar autoadaptación a los algoritmos genéticos con codificación real que lo utilicen. El PX-α opera de la siguiente forma:

Sean $X = (x_1 ... x_n)$ y $Y = (y_1 ... y_n)(x_i, y_i \in [a_i, b_i] \subset \Re, i = 1...n)$ dos cromosomas con codificación real que han sido seleccionados para aplicar un operador de cruce sobre ellos. El operador PX-α genera de manera aleatoria dos descendientes $Z = (z_1^1 ... z_n^1)$ y $Z = (z_1^2 ... z_n^2)$ donde $z_i^1$ es aleatoriamente (uniformemente) escogido en el intervalo $[l_i^1, u_i^1]$ con $l_i = \max\{a_i, x_i - I \cdot \alpha\}$ y $u_i = \min\{b_i, x_i + I \cdot \alpha\}$ y $z_i^2$ escogido en el intervalo $[l_i^2, u_i^2]$ con $l_i^2 = \max\{a_i, y_i - I \cdot \alpha\}$ y $u_i^2 = \min\{b_i, y_i + I \cdot \alpha\}$ donde $I = |x_i - y_i|$.

## IV. RESULTADOS EXPERIMENTALES

Para probar nuestra propuesta utilizamos 7 bases de datos, con un alto nivel de desbalance, mayor de 10. Los conjuntos fueron divididos para realizar un 5 folds cross validation, las participaciones fueron realizadas de manera que la distribución de la cantidad de elementos en cada clase quedara uniforme. Luego se procedió a balancear los conjuntos de entrenamiento usando el algoritmo EditOSRS, las muestras para test no fueron balanceadas.

Una vez balaceados los conjuntos de entrenamiento, se realizó el estudio experimental con variantes que se describen a continuación:

1. La que hemos denominado test, que consiste en balancear la muestra para conformar el modelo, mientras que la muestra de control continua desbalanceada.
2. La que hemos denominado training que consiste en usar como muestra de control el conjunto de entreamiento.

Las medidas seleccionadas para evaluar la edición de los conjuntos fueron la precisión obtenida en la clase minoritaria y el área ROC para 3 clasificadores: Red Neuronal del tipo MLP, K vecino más cercano con K=1 y árbol de decisión C4.5 (J48 de Weka), para ello se utilizó la herramienta para experimentación Weka.

El área ROC [33] resulta una medida de suma importancia en el contexto de los no balanceados [34-36], la razón por la que se utiliza es porque tiene un valor estadístico muy importante, es equivalente a un test de Wilconxon [37], que es dos veces la zona comprendida entre la diagonal y la curva de ROC.

Concluidas las ejecuciones con las bases de datos balanceados con el algoritmo EditOSRS, se realizaron las mismas ejecuciones pero con las bases de datos balanceadas con Smote, con el objetivo de comparar los resultados.

En las tablas I, II y III se muestran los resultados alcanzados en cuanto al área de ROC por MLP, KNN y C4.5 sobre los conjuntos con alto nivel de desbalance, balanceados con ambos algoritmos (Smote y EditOSRS). En las columnas oscuras aparecen los resultados de los test, cómo se explicó anteriormente se balancea la muestra para conformar el modelo y la muestra de control sigue estando desbalanceada; en las columnas claras aparecen los resultados para entrenamiento, es decir, usando el conjunto de entrenamiento también como muestra de control. En las figuras 1, 2 y 3 se muestran las gráficas correspondientes al área ROC para los test.

TABLA I

RESULTADOS DEL AREA ROC PARA MLP

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS /test* | *EditOSRS /train* |
|---|---|---|---|---|
| **EcoliM5** | 0,931 | 0.998 | 0,981 | **0.999** |
| **Glass** | 0,881 | 0.964 | 0,828 | **0.994** |
| **GlassCont** | 0,970 | 0.976 | 0,983 | **0.999** |
| **GlassTB** | 0,938 | 0.999 | 0,981 | **1** |
| **Vowel0** | 1 | 1 | 0,993 | **1** |
| **yeastCYT-POX** | 0,829 | 0.994 | 1 | **0.996** |
| **yeastMe1** | 0,991 | 0.994 | 0,809 | **0.999** |
| **Media** | **0.924** | **0.989** | **0,932** | **0.998** |

TABLA II

RESULTADOS DEL AREA ROC PARA KNN

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS /test* | *EditOSRS /train* |
|---|---|---|---|---|
| **EcoliM5** | 0,915 | 1 | 0,882 | **1** |
| **Glass** | 0,601 | 1 | 0,570 | **1** |
| **GlassCont** | 0,892 | 1 | 0,821 | **1** |
| **GlassTB** | 0,883 | 1 | 0,893 | **1** |
| **Vowel0** | 1 | 1 | 1 | **1** |
| **yeastCYT-POX** | 0,830 | 1 | 0,810 | **1** |
| **yeastMe1** | 0,933 | 1 | 0,910 | **1** |
| **Media** | **0.844** | **1** | **0.834** | **1** |

TABLA III

RESULTADOS DEL AREA DE ROC PARA C4.5

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS /test* | *EditOSRS /train* |
|---|---|---|---|---|
| **EcoliM5** | 0,754 | 0.996 | 0,746 | **0.994** |
| **Glass** | 0,563 | 0.987 | 0,559 | **0.991** |
| **GlassCont** | 0,853 | 0.996 | 0,809 | **0.994** |
| **GlassTB** | 0,882 | 0.997 | 0,944 | **1** |
| **Vowel0** | 0,962 | 0.999 | 0,903 | **0.999** |
| **yeastCYT-POX** | 0,799 | 0.994 | 0,737 | **0.989** |
| **yeastMe1** | 0,952 | 0.995 | 0,934 | **0.997** |
| **Media** | **0.802** | **0.994** | **0.783** | **0.995** |

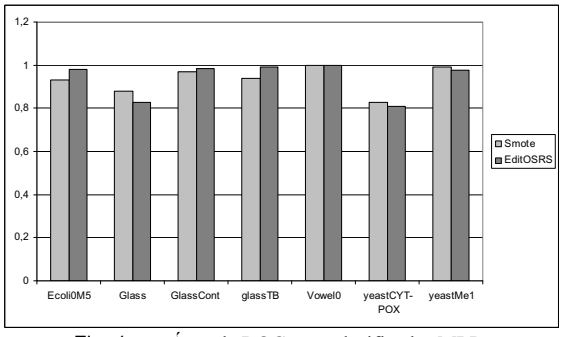


Fig. 1. Área de ROC para clasificador MLP.

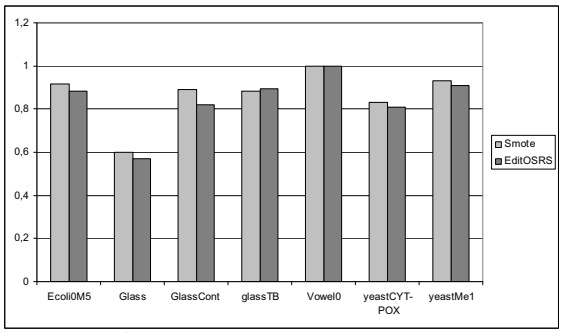


Fig. 2. Área de ROC para un clasificador KNN con k=1.

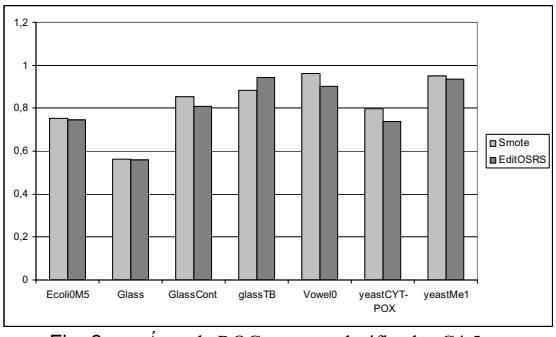


Fig. 3. Área de ROC para un clasificador C4.5

Como se puede observar en los resultados alcanzados en esta medida el comportamiento de ambos algoritmos es comparable, aunque el EditRSOS se comportó superior en todos los casos para entrenamiento y en un caso ligeramente mejor en test.

Otra de las medidas seleccionadas para la comparación es la precisión por clases, centrándonos en la clase minoritaria que tiene una gran importancia para éste tipo de problemas. En las tablas IV, V, y VI se muestran los resultados de la precisión alcanzada en la clase minoritaria, para

entrenamiento en las columnas claras y para test en las oscuras.

TABLA IV
PRECISIÓN DE LA CLASE MINORITARIA PARA MLP

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS/test* | *EditOSRS/train* |
|---|---|---|---|---|
| **EcoliM5** | 0,769 | 0.989 | 0,668 | **0.994** |
| **Glass** | 0,316 | 0.906 | 0,183 | **0.981** |
| **GlassCont** | 0,547 | 0.97 | 0,87 | **0.998** |
| **GlassTB** | 0,9 | 0.996 | 0,8 | **1** |
| **Vowel0** | 1 | 1 | 1 | **1** |
| **yeastCYT-POX** | 0,308 | 0.963 | 0,391 | **0.973** |
| **yeastMe1** | 0,574 | 0.984 | 0,578 | **0.992** |
| **Media** | **0.607** | **0.972** | **0.637** | **0.991** |

TABLA V
PRECISIÓN DE LA CLASE MINORITARIA PARA KNN

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS/test* | *EditOSRS/train* |
|---|---|---|---|---|
| **EcoliM5** | 0,79 | 1 | 0,675 | **1** |
| **Glass** | 0,239 | 1 | 0,246 | **1** |
| **GlassCont** | 0,533 | 1 | 0,763 | **1** |
| **GlassTB** | 0,466 | 1 | 0,68 | **1** |
| **Vowel0** | 1 | 1 | 1 | **1** |
| **yeastCYT-POX** | 0,403 | 1 | 0,5 | **1** |
| **yeastMe1** | 0,61 | 1 | 0,48 | **1** |
| **Media** | **0.544** | **1** | **0.611** | **1** |

TABLA VI
PRECISIÓN DE LA CLASE MINORITARIA PARA C4.5

| *Dataset* | *Smote/test* | *Smote/train* | *EditOSRS/test* | *EditOSRS/train* |
|---|---|---|---|---|
| **EcoliM5** | 0,487 | 0.992 | 0,439 | **0.988** |
| **Glass** | 0,256 | 0.958 | 0,253 | **0.987** |
| **GlassCont** | 0,537 | 0.996 | 0,65 | **0.992** |
| **GlassTB** | 0,55 | 0.997 | 0,867 | **0.997** |
| **Vowel0** | 0,872 | 0.997 | 0,915 | **0.999** |
| **yeastCYT-POX** | 0,557 | 0.987 | 0,33 | **0.989** |
| **yeastMe1** | 0,593 | 0.989 | 0,632 | **0.997** |
| **Media** | **0.560** | **0.988** | **0.607** | **0.992** |

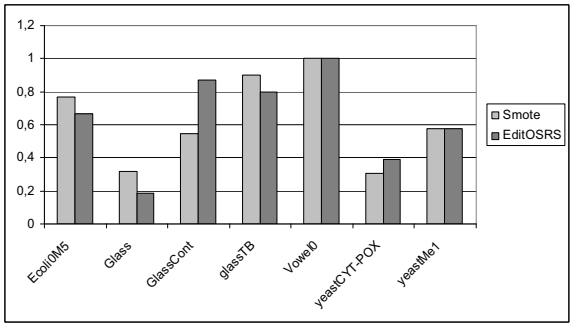


Fig. 4. Precisión de la clase minoritaria para el MLP

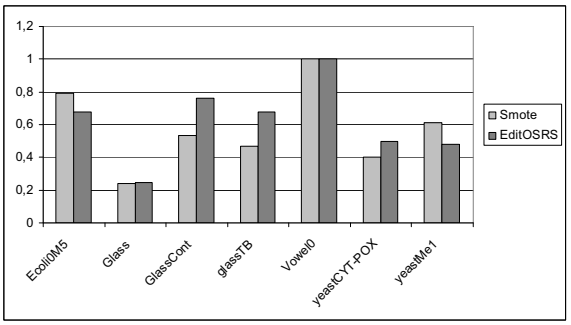


Fig. 5. Precisión de la clase minoritaria para KNN

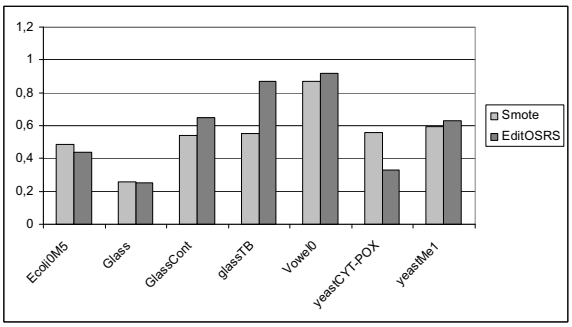


Fig. 6. Precisión de la clase minoritaria para C4.5

Aunque no existen grandes diferencias significativas, los conjuntos editados por EditOSRS alcanzan resultados ligeramente mejores cuando analizamos la clase minoritaria con los diferentes clasificadores.

Éste trabajo constituye el inicio de un estudio más amplio que debe incluir 44 bases de datos, todas con diferentes niveles de desbalance, con vistas a analizar en profundidad los resultados alcanzados por EditOSRS.

## V. CONCLUSIONES

En esta contribución se realiza una nueva propuesta para editar conjuntos de entrenamiento que presenten desbalance en sus clases (2 clases). La propuesta realizada forma parte de las técnicas conocidas como Over-Sampling, al generar nuevas instancias de la clase minoritaria hasta igualar en número a la mayoritaria. Lo novedoso de la propuesta radica en que no solo son generadas nuevas instancias haciendo uso de algoritmos genéticos con eficientes operadores de cruce, sino que también es evaluada la calidad de las nuevas instancias mediante la Teoría de los Conjuntos Aproximados.

Esta evaluación permite que solamente se inserten en el conjunto solución las instancias que sean de la aproximación inferior de la clase, o que estén en la frontera pero que pertenezcan con un elevado grado a la clase.

Los resultados experimentales muestran cómo se logra mejorar por encima de SMOTE la precisión para la clase minoritaria, mientras que el área ROC se comporta de manera comparable.

## VI. TRABAJOS FUTUROS

A partir de éstos resultados alcanzados en la experimentación surgen dos ideas fundamentales: la primera es: además de editar la clase minoritaria, editar también la mayoritaria, es decir, hacer una hibridación entre Over-Sampling y Under-Sampling, básicamente sería eliminar las instancias ruidosas de la clase mayoritaria, haciendo uso de la teoría de Conjuntos Aproximados (Esto puede reducir el número de instancias), luego se procedería con el mismo algoritmo que se propone en éste artículo.

La segunda idea sería utilizar algoritmos genéticos para la selección de los conjuntos de la clase minoritaria y mayoritaria una vez generados los conjuntos de la clase minoritaria, tal como se hace en los trabajos [38] y [39].

## REFERENCIAS


1. Murphy, P.M., Aha, D.W.: UCI Machine Learning Repository. Machine Learning Databases. University of California at Irvine (1994)
2. Fawcett, T.E., Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery **3** (1997) 291-316
3. Kubat, M., R.Holte, Matwin, S.: Machine Learning for the Oil Spills in Satellite Radar Images. Machine Learning **30** (1998) 195-215
4. Lewis, D., Catlett, J.: Heterogeneous Uncertainty Sampling for Supervised Learning. Eleventh International Conferences of Machine Learning (1994) 148-156
5. Japkowicz, N., Myers, C., Gluck, M.: A Novelty Detection Appoach to Clasification. Fourteenth Joint Conferences on Artificial Intelligence (1995) 518-523
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling TEchnique. Articial Intelligence Research (2002) 341-378
7. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. Springer-Verlag (2005) 878-887
8. Cohen, G., Hilario, M., Sax, H., Hogonnet, S., Geissbuhler, A.: Learning from imbalanced data in surveillance of nosocomial infection. Articial Intelligence in Medicine (2006) 7-18
9. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Report A 2001-2 (2001)
10. Tomek, I.: Two Modifications to CNN. IEEE Transactions Systems, Man, and Communications (1976) 769-772
11. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. 14th International Conference on Machine Learning (1997) 179-186
12. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions Systems, Man, and Communications (1972) 408-421
13. Liu, Y., Chawla, N.V., Harper, M.P., Shrilberg, E., A.Stolke: A Study in Machine Learning from Imbalanced dta for Sentence Boundary Detection in Speech. Computer Speech and Language (2006) 468-494
14. Batista, G.E.A.P.A., Monard, M.C., R.C.Prati: A study of several methods for balancing machine learninng training data. Sigkdd Exploations (2004) 20-29
15. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-Coded Memetic Algorithms with Crossover Hill-Climbing. Evolutionary Computation (2004) 273-302
16. Caballero, Y.: Aplicación de la Teoría de los Conjuntos Aproximados en el Prepocesamiento de los Conjuntos de Entrenamiento para los Algoritmos de Aprendizaje Automatizado. Ciencia de la Computación. Universidad Central Martha Abreu de Las Villas, Santa Clara, Cuba (2007)
17. Pawlak, Z.: Rough Sets. International journal of Computer and Information Sciences **11** (1982) 341-356
18. Choubey, S.K.: A comparison of feature selection algorithms in the context of rough classifiers Fifth IEEE International Conference on Fuzzy Systems, Vol. 2 (1996) 1122-1128
19. Chouchoulas, A., Shen, Q.: A rough set-based approach to text classification Lectures Notes in Artificial Intelligence **11** (1999) 118-127
20. Grzymala-Busse, J.W., Siddhaye, S.: Rough set approaches to rule induction from incomplete data. 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases systems IPMU2004, Vol. 2, Perugia, Italy (2004) 923-930
21. Piñero, P., Arco, L., García, M.M., Caballero, Y.: Two New Metrics for Feature Selection in Pattern Recognition. Lectures Notes in computer Science (LNCS 2905)Springer, Verlag, Berlin Heidelberg. New York (2003) 488-497
22. Tsumoto, S.: Automated extraction of hierarchical decision rules from clinical databases using rough set model. Expert systems with Applications **24** (2003) 189-197
23. Sugihara, K., Tanaka, H.: Rough Sets approach to information systems with interval decision values in evaluation problems. The International Symposium on Fuzzy and Rough Sets. ISFUROS2006, Santa Clara, Cuba (2006)
24. Midelfart, H., Komorowski, J., Ñorsett, K., Yadetie, F., Sandvik, A., Laegreid, A.: Learning rough set classifiers from gene expression and clinical data. Fundamenta Informaticae **53** (2003) 155-183

25. Miao, D., Hou, L.: An Application of Rough Sets to Monk's Problems Solving. In: 26392003, L. (ed.): Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China (2003)
26. Zhao, Y., Zhang, H., Pan, Q.: Classification Using the Variable Precision Rough Set. In: 26392003, L. (ed.): Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, RSFDGRC2003, Chongqing, China (2003)
27. Greco, S., Inuiguchi, M., Slowinski, R.: Rough Sets and Gradual Decision Rules. In: 26392003, L. (ed.): Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China (2003)
28. Deogun, J.S.: Exploiting upper approximations in the rough set methodology. In: Fayyad, U.Y.U. (ed.): First International Conference on Knowledge Discovery and Data Mining, Canada (1995) 69-74
29. Pal, S.K., Skowron, A.: Rough Fuzzy Hybridization: A New Trend in Decision-Making (1999)
30. Grabowski, A.: Basic Properties of Rough Sets and Rough Membership Function. Journal of Formalized Mathematics **15** (2003)
31. Butz, M.: Rule-Based Evolutionary Online Learning Systems. Springer (2006)
32. Eshelman, L.J., Schaffer, J.D.: Real -coded genetic algorithms and interval-schemata. Foundations of Genetic Algorithms (1993) 187-202
33. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30** (1997) 1145-1159
34. Xie, J., Qiu, Z.: The effect of imbalanced data sets on LDA:Atheoretical and empirical analysis. Pattern Recognition **40** (2007) 557-562
35. Chawla, N.V., Japkowicz, N., Kolcz, A.: Learning from Imbalanced Data Sets. Proceedings of the ICML'2003Workshop (2003)
36. Chawla, N.V., Japkowicz, N., Kolcz, A.: Special issue on learning from imbalanced data sets. ACM SIGKDD Explorations **6** (2004)
37. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a
receiver operating characteristic (ROC) curve. Radiology **143** (1982) 29-36
38. García, S., Herrera, F.: Evolutionary Under-Sampling for Classification with Imbalanced Data Sets. Proposals and Taxonomy. Evolutionary Computation (2008)
39. Salvador. García, Herrera, F.: Evolutionary Training Set Selection to Optimize C4.5 in Imbalanced Problems. Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS08) (September 2008) 567-572