DETECCIÓN DE LA CALIDAD DEL AGUA EN SISTEMAS LÓTICOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Edwin Ferney Castillo Quintero

Director:
MSc. Ing. David Camilo Corrales Muñoz

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, 2016

DETECCIÓN DE LA CALIDAD DEL AGUA EN SISTEMAS LÓTICOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Edwin Ferney Castillo Quintero

Monografía presentada como requisito para optar por el título de Ingeniero en Electrónica y Telecomunicaciones

Director:
MSc. Ing. David Camilo Corrales Muñoz

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, 2016

Tabla de contenido

Lista de Figuras

Lista de Tablas

Lista de Ecuaciones

Capítulo 1

1 Introducción

1.1 Planteamiento del problema

El agua dulce es considerada uno de los recursos naturales renovables más importantes. En este sentido, Colombia se ubica entre los países con mayor oferta hídrica del mundo [1] con cinco vertientes: Caribe, Orinoco, Amazonas, Pacífico y Catatumbo. En este sentido, es de vital importancia estudiar y evaluar la calidad del agua de sus ríos y/o sistemas lóticos[1], aunque determinar el estado ambiental de estos se convierte en una tarea compleja, particularmente, cuando la condición de referencia de las corrientes se desconoce y han estado sujetas por largo tiempo a perturbaciones antropogénicas [2].

En esta clase de ecosistemas, la comunidad de macro-invertebrados es altamente diversa y, debido a sus límites de tolerancia a diferentes alteraciones del entorno [3], tienen un uso potencial en el monitoreo de sistemas lóticos [4] complementado por el análisis de variables fisicoquímicas (pH, oxígeno disuelto, etc.) y variables ambientales (temperatura, humedad, pluviosidad, radiación solar, etc.). Actualmente, muchos científicos hacen uso de índices biológicos para calcular la calidad del agua [5-7]. En Colombia comúnmente son usados bio-indicadores en sistemas fluviales como: Biological Monitoring Working Party (BMWP) y Average Score Per Taxon (ASPT), los cuales son adaptados para cada región del país, debido a la gran diversidad de climas y relieves [8].

En este sentido, existen nuevas aproximaciones [9-17] que automatizan la evaluación de la calidad del agua a través de algoritmos que detectan el comportamiento de un conjunto de datos mediante técnicas de aprendizaje automático, las cuales se dividen en algoritmos de aprendizaje supervisado y no supervisado. El aprendizaje supervisado predice o clasifica un nuevo dato de entrada tomando como referencia un conjunto de ejemplos (instancias) llamados comúnmente datos de entrenamiento (compuesto de atributos y una variable objetivo) [18], haciendo uso de algoritmos como: Arboles de Decisión (AD), Redes Bayesianas (RB), Redes Neuronales Artificiales (RNA), Vecino más Cercano (K-NN), y Máquinas de Vector de Soporte (MVS). Por otra parte, el aprendizaje no supervisado genera grupos sobre un conjunto de datos no etiquetados (solo atributos) con base en criterios establecidos [19], mediante algoritmos jerárquicos, densos y particionales.

Ahora bien, para obtener un correcto funcionamiento en esta clase de algoritmos, los datos utilizados no deben ser redundantes y el número de instancias debe estar distribuido de manera uniforme según los valores que tome la variable objetivo (balanceo de clases). De esta manera, investigaciones como [20-29] proponen mecanismos para

[1] Sistemas lóticos: ecosistemas de aguas en constante movimiento y en una misma dirección (ríos, estuarios).

mejorar la calidad del conjunto de datos, a través de técnicas de aprendizaje automático y medidas estadísticas. Sin embargo, estos trabajos solo intentan abordar uno de los problemas mencionados anteriormente (redundancia en: instancias, atributos, o desbalanceo de clases).

Teniendo en cuenta todas las consideraciones planteadas, se hace necesario seleccionar las técnicas de aprendizaje automático idóneas, construir el conjunto de datos que evite valores redundantes y además presente una distribución uniforme según los valores que tome la variable objetivo, con la finalidad de detectar la calidad del agua en sistemas lóticos. Con base a esto, se plantea la siguiente pregunta de investigación:

¿Cómo detectar la calidad del agua en ecosistemas lóticos mediante técnicas de aprendizaje automático y la correcta construcción de un conjunto de datos?

1.2 Escenario de motivación

La calidad del agua dulce es un factor relevante para el sostenimiento de cualquier sociedad, ya sea para consumo humano, actividades pecuarias, turismo, como para otras actividades. Existen diversas razones que justifica el estudio de la calidad del agua, aunque el más trascendental está atado a las actividades antropogénicas [2], debido a que hoy en día el recurso hídrico presenta un creciente deterioro como consecuencia del crecimiento de la población, el incremento de las actividades pecuarias, el establecimiento de asentamientos humanos en zonas no adecuadas, tala indiscriminada de bosques, la minería ilegal, entre otras. Esto ha degradado y reducido este recurso lo que traduce a una gran amenaza de la salud humana y al funcionamiento de los ecosistemas acuáticos.

Teniendo en cuenta la problemática expuesta anteriormente, es necesario contar con un mecanismo que permita detectar la calidad del agua con el fin de poner al alcance de las autoridades sanitarias información de una manera ágil y fácil de interpretar para tomar acciones preventivas, o correctivas, según sea el caso.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un mecanismo para la detección de la calidad del agua en sistemas lóticos a través de técnicas de aprendizaje automático.

1.3.2 Objetivos específicos

- Definir técnicas que permitan detectar instancias y atributos redundantes sobre un conjunto de datos de entrenamiento.
- Establecer mecanismos que balanceen las clases de un conjunto de datos de entrenamiento.
- Seleccionar los algoritmos de aprendizaje supervisado que permiten detectar la calidad del agua tomando como entrada el conjunto de datos de entrenamiento refinado.
- Desarrollar y evaluar experimentalmente un prototipo para la detección de la calidad del agua, mediante las técnicas de aprendizaje automático propuestas.

1.4 Contribuciones

Las principales contribuciones de éste proyecto de grado son:

- Dos conjuntos de datos de entrenamiento, uno constituido por variables biológicas (macro-invertebrados) y fisicoquímicas, recolectadas en tres punto de la cuenca Rio las Piedras: Puente Alto, Puente Carretera y la Bocatoma Diviso, en diferentes periodos de precipitación: alta (Octubre - Noviembre), media (Junio - Julio) y baja (Agosto - septiembre). Y el segundo dataset está conformado por variables fisicoquímicas, recolectadas en el Estuario California.

- Un mecanismo para el pre-procesamiento de datos: reducción de la dimensionalidad y balanceo de clases.

- La selección de un conjunto de modelos de clasificación idóneos para la detección de la calidad del agua.

- Un prototipo, conformado por tres niveles de pre-procesamiento de datos distribuidos de la siguiente forma: reducción de atributos, reducción de instancias y balanceo de clases, y además cuenta por un modelo de clasificación con el propósito de detectar la calidad del agua en los sistemas loticos.

- Un artículo expuesto en la revista Sistemas & Telemática, el cual presenta un mecanismo para la generación de clases de los conjunto de datos, que contribuyen a la construcción del conjunto de datos de entrenamiento para detectar la calidad del agua (ANEXO A).
- Un artículo sometido a evaluación en: *The 16th International Conference on Computational Science and Its Applications (ICCSA 2016)*, el cual presenta los resultados del presente trabajo de grado.

1.5 Esquema general de la monografía

La monografía se encuentra organizada en 7 capítulos, los cuales son presentados a continuación:

Capítulo 2. Estado actual del conocimiento

Presenta una visión general sobre los trabajos relacionados y los conceptos que giran en torno al problema de investigación declarado.

Capítulo 3. Reducción de la dimensionalidad

En este capítulo se presentan de manera detallada las técnicas de reducción de la dimensionalidad seleccionadas. Adicionalmente, se exponen los aportes realizados conforme al objetivo específico número uno y se exhibe la experimentación y los resultados obtenidos.

Capítulo 4. Desbalanceo de clases

Presenta de manera detallada el proceso de sobre-muestreo seleccionado para hacer frente al problema de desbalanceo de clases. De igual manera, se expone el aporte llevado a cabo para mejorar el proceso de balanceo de clases conforme al objetivo

específico número dos. Por último, se exhibe la experimentación realizada y los resultados obtenidos.

Capítulo 5. Algoritmos para la detección de la calidad de agua

Este capítulo presenta los algoritmos de clasificación más utilizados en la literatura. Adicionalmente, se realiza la evaluación de los modelos de clasificación y, finalmente, son seleccionados los algoritmos adecuados para dar solución al problema de investigación planteado, conforme al objetivo específico número tres.

Capítulo 6. Prototipo

Este capítulo provee una visión global del enfoque de desarrollo para realizar el prototipo para la detección de la calidad del agua. Además, se presentan los resultados y artefactos generados en el proceso de ingeniería llevado a cabo.

Capítulo 7. Conclusiones y trabajos futuros

Finalmente, se analizan los resultados del trabajo realizado, se detallan las principales contribuciones obtenidas en la ejecución del proyecto y se expone un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

Capítulo 2

2 Estado actual del conocimiento

En este capítulo se presenta los conjuntos de datos (dataset) para la detección de la calidad del agua, seguida de las bases teóricas para comprender la temática del presente trabajo de grado, el cual propone un prototipo para la detección de la calidad del agua en sistemas loticos. Posteriormente, se expone los trabajos relacionados respecto al problema de investigación declarado. Finalmente, se realiza un resumen que presenta los principales aportes de este capítulo.

2.1 Contexto

En este apartado se presentan los conjuntos de datos de (dataset) para la detección de la calidad del agua, seguido de los clasificadores utilizados en la detección de la calidad del agua.

2.1.1 Descripción De Los Datos

Los datos utilizados en el presente estudio provienen de dos sistemas loticos que proveen información acerca de la calidad del agua. El primer conjunto de datos de la Cuenca Rio Piedras en el Departamento del Cauca (Colombia), mientras que el segundo del estuario del Estado de California (Estados Unidos), suministrado por el repositorio USGS (United States Geological Survey). A continuación son explicados.

DATASET RIO LAS PIEDRAS

Estos datos fueron recolectados trimestralmente en la Cuenca Rio Piedras, ubicada en el Departamento del Cauca, Colombia (Nacimiento: 76° 31’ 10” al Oeste de Greenwich y 2° 21’ 45” de latitud Norte, Desembocadura: 76° 23’ 45” longitud Oeste y 2° 25’ 40” de latitud Norte del río Cauca), por el Grupo de Estudios Ambientales (GEA) de la Universidad del Cauca, entre los años 2011 y 2013, teniendo en cuenta la metodología seguida en [30]. Las muestras capturadas contienen variables biológicas (macro-invertebrados) y fisicoquímicas, de tres puntos de la cuenca: Puente Alto, Puente Carretera y la Bocatoma Diviso (Figura 1), en diferentes periodos de precipitación: alta (Octubre - Noviembre), media (Junio - Julio) y baja (Agosto - septiembre).

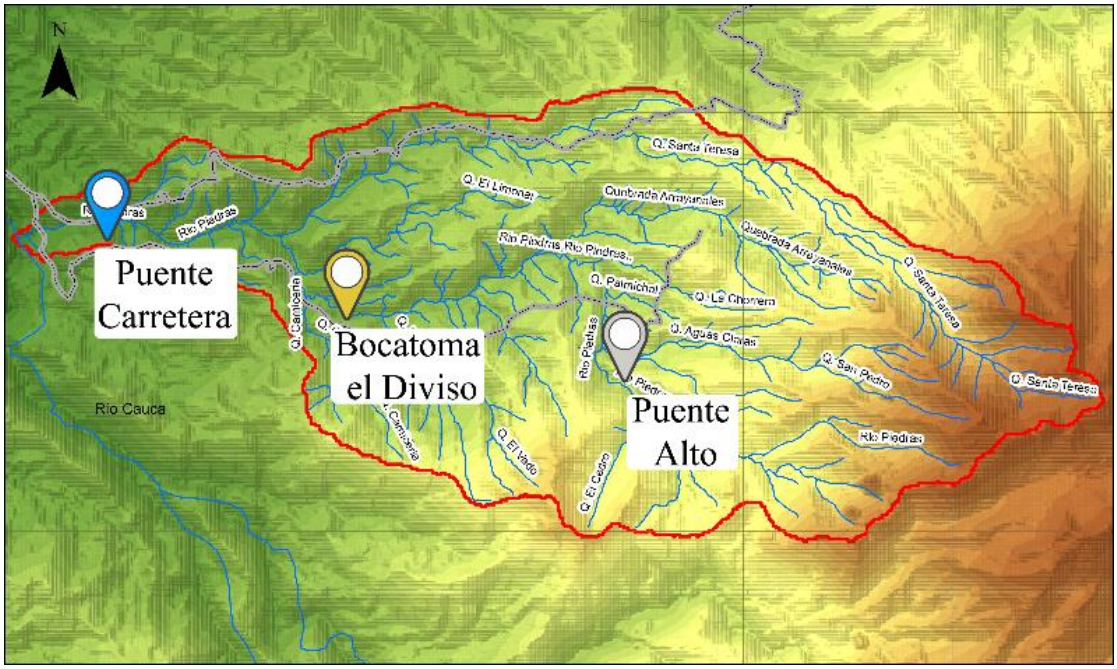


Figura 1. Localización de los puntos de muestreo (fuente propia)

De esta forma fueron capturados 10 indicadores fisicoquímicos, 5 Biológicos y 3 atributos que describen el periodo de precipitación, los cuales constan de 645 registros, y 3 valores a clasificar (clases) [31], tal y como se expone en la Tabla 1.

Categoría	Atributo	Unidad de medida	Rango	Clases
Indicadores fisicoquímicos	Temperatura	°C	13.0 - 17.8	**Calidad del Agua Alta (1)** **Calidad del Agua Buena (2)** **Calidad del Agua Regular (3)**
	Conductividad	µs/cm	35.2 - 89.0	
	Total solidos disueltos	mg/L	16.5-42.1	
	Oxígeno disuelto	mg/L	7.17-8.23	
	pH	mg/L	6.62-8.17	
	Amoniaco	mg/L	0.01-0.04	
	Nitratos	mg/L	0.01-0.09	
	Nitritos	mg/L	0.01-0.06	
	Fosfatos	mg/L	0.08-0.24	
	Turbidez	mg/L	1.0-9.8	
Indicadores biológicos	Clase	-	-	
	Orden	-	-	
	Familia	-	-	
	Taxón	-	-	
	Número de individuos	-	-	
Periodos de precipitación	Mes	-	-	
	Año	-	-	
	Punto muestreo	-	-	

Tabla 1. Atributos dataset Rio Piedras

Según este el último trabajo, los tres (3) valores a clasificar se denotan por los números 1, 2 y 3, los cuales representan una calidad del agua alta (aguas muy limpias), buena (aguas ligeramente contaminadas) y regular (moderadamente contaminadas), respectivamente.
A continuación se describen cada uno de ellos:

a) Indicadores fisicoquímicos

Las variables fisicoquímicas son descritas por los autores [32, 33], y se presentan a continuación.

- **Temperatura (T):** actúa sobre procesos de absorción de oxígeno, actividad biológica, precipitación de compuestos, formación de depósitos y modificación de la solubilidad de sustancias. Unidad de medición: grados Centígrados (°C).
- **Conductividad (C):** utilizada como índice de concentración de solutos (cantidad de sólidos) disueltos en el agua. Unidad de medición: µs/cm.
- **Total de Sólidos Disueltos (TDS):** mide las sustancias orgánicas e inorgánicas, en forma molecular, ionizada o micro-granular del agua. Unidad de medición: mg/L.
- **Oxígeno disuelto (OD):** cantidad de oxígeno disuelto en el agua. Es un indicador que mide la contaminación del agua. Un nivel alto de oxígeno disuelto indica agua de mejor calidad. Unidad de medición: mg/L.
- **PH:** mide la concentración de iones de hidrógeno en el agua. Las aguas naturales (no contaminadas) exhiben un PH de rango 5 – 9.
- **Amoniaco (Am):** formado durante la biodegradación de los compuestos orgánicos nitrogenados. Un nivel alto causa daños en los ríos u estanques. Unidad de medición: mg/L.
- **Nitratos (Nitra):** nutriente requerido por plantas y animales acuáticos para la creación de proteínas. La descomposición de las plantas y animales muertos y el excremento de los animales vivos descargan nitratos en los ecosistemas acuáticos. Unidad de medición: mg/L.
- **Nitritos (Nitra):** se transforman naturalmente a partir de los nitratos, y su presencia en el agua es indicativo de contaminación de carácter fecal. Unidad de medición: mg/L.
- **Fosfatos (F):** son nutrientes esenciales para los organismos acuáticos tanto de aguas naturales como de aguas negras y además son necesarios para reproducción y síntesis de nuevos tejidos celulares. Unidad de medición: mg/L.
- **Turbidez (Tu):** falta de transparencia en el agua debido a materiales insolubles en suspensión o coloides (arcilla, limo, tierra, etc.). Cuanto más materiales haya en el agua (turbidez alta), menor será la concentración de oxígeno en la misma. Unidad de medición: NTU.

b) Indicadores biológicos (Macro-invertebrados)

Las muestras biológicas colectadas en los puntos de muestreo fueron identificadas mediante las claves taxonómicas: clase, orden, familia, taxón y número de individuos [32]. A continuación, se hace una breve descripción de las muestras colectadas por orden.

- **Acari:** viven en aguas continentales (agua dulce), lóticas (aguas en movimiento) y lénticas (aguas estancadas) limpias, altamente oxigenadas.

- **Pelecypoda:** pertenecen a ecosistemas acuáticos marinos y continentales altamente oxigenados. Son muy sensibles a la contaminación, por lo que se consideran excelentes bioindicadores para determinar la calidad de agua.
- **Plecoptera:** viven en aguas continentales lóticas limpias, turbulentas, frías y altamente oxigenadas, utilizados como bioindicadores para determinar la calidad de agua.
- **Lepidoptera:** viven tanto en aguas lénticas como lóticas, sobre fondos pedregosos, vegetación sumergida y muy oxigenadas. Las especies son intolerantes a la eutrofización (contaminación química de las aguas).
- **Coleóptera:** habitan en aguas continentales lóticas y lénticas limpias, de baja profundidad (someras), con concentraciones altas de oxígeno, temperaturas medias y baja velocidad.
- **Díptera:** viven en nichos terrestres como en aguas continentales lóticas y lénticas someras (poco profundas) y profundas. Este orden Incluye parásitos, predadores y degradadores. En virtud a esto, ha adquirido un gran significado sanitario (aguas de calidad mala). Las especies son tolerantes a distintos grados de contaminación.
- **Ephemeroptera:** viven en aguas continentales lóticas claras, limpias y bien oxigenadas con bajo contenido de materia orgánica de desechos, utilizadas como bioindicadores de calidad del agua.
- **Hemíptera:** viven en aguas continentales lóticas de baja velocidad y lénticas. Algunas especies (Neuston) resisten cierto grado de salinidad y a altas temperaturas, usado como bioindicador en aguas superficiales.
- **Odonata:** viven en aguas continentales lóticas de baja velocidad y lénticas, poco profundas, rodeadas de abundante vegetación acuática sumergida o emergente. Algunas especies pueden resistir cierto grado de contaminación
- **Trichoptera:** la mayoría de las especies viven en aguas continentales lóticas (debajo de piedras, troncos y material vegetal) y algunos pocos viven en aguas lénticas, limpias y oxigenadas.
- **Tricladida:** viven en aguas poco profundas tanto lénticas como lóticas. La mayoría viven en aguas bien oxigenadas, pero algunas especies pueden resistir altos grados de contaminación orgánica.
- **Isopoda:** comunes en hábitats marinos, sin embargo, algunas especies son de agua dulce y muchas terrestres. Grandes números de especies de este orden indican enriquecimiento orgánico.
- **Glossiphoniformes:** son ectoparásitos de peces en aguas continentales. Este orden posee un alto grado de tolerancia a la contaminación del agua.
- **Haplotaxida:** la mayoría de los organismos de éste orden viven en aguas lénticas y lóticas eutrofizadas, sobre fondo fangoso y con abundante cantidad de residuos. Las especies son altamente tolerantes a la contaminación orgánica.

c) Periodos de precipitación

Describen el periodo de precipitación en el cual fueron tomadas las muestras por: año, mes y código del punto de muestreo.

DATASET ESTUARIO CALIFORNIA

Los datos recolectados para este dataset fueron suministrados por USGS, el cual se compone de 7 atributos fisicoquímicos, 2505 registros divididos en 4 valores a clasificar

(clases), los cuales fueron generados a partir de la metodología propuesta en [31]. En la Tabla 2 se observa la descripción de los atributos.

Categoría	Atributo	Unidad de medida	Rango	Clases
Indicadores fisicoquímicos	Temperatura	°C	13.0 - 17.8	**Calidad del Agua Alta (0)**
	Conductividad	µs/cm	35.2 - 89.0	
	Total solidos disueltos	mg/L	16.5-42.1	**Calidad del Agua Buena (1)**
	Oxígeno disuelto	mg/L	7.17-8.23	**Calidad del Agua Regular (2)**
	Turbidez	mg/L	1.0-9.8	
Periodos de precipitación	Año	-	-	**Calidad del Agua Mala (3)**
	Punto muestreo	-	-	

Tabla 2. Atributos dataset Estuario California

Los cuatro (4) valores a clasificar se encuentran denotados por los números 0, 1, 2 y 3, donde el valor tres (0) representa una calidad del agua alta, el valor cero (1) indica una calidad del agua buena, el valor uno (2) simboliza aguas de dudosa calidad mientras que el valor dos (3) constituye aguas de calidad crítica (aguas bastante contaminadas).

2.1.2 Aprendizaje Supervisado

Los algoritmos de aprendizaje supervisado requieren un conjunto de datos (como los presentados en el apartado anterior), conocidos comúnmente como datos de entrenamiento, los cuales son utilizados para definir el comportamiento del algoritmo. Los datos de entrenamiento están organizados por registros (instancias) y se componen de un conjunto de atributos y una variable objetivo (también llamada clase), la cual se intenta clasificar o predecir [18].
Es importante tener en cuenta que el resultado del proceso de aprendizaje (entrenamiento) del algoritmo utilizado genera un clasificador (hipótesis o modelo) capaz de predecir, detectar o clasificar el valor correspondiente a cualquier nuevo dato de entrada válido. Para abordar este tipo de aprendizaje, existen diferentes familias de algoritmos, las cuales son presentadas a continuación:

ARBOLES DE DECISIÓN (AD)

Esta familia de algoritmos divide el conjunto de datos de entrenamiento en sub-grupos de manera recursiva, a partir de un conjunto de condiciones definidas en cada rama (nodo) en el árbol. El árbol se compone de un nodo raíz, un conjunto de nodos internos, y un conjunto de nodos terminales (hojas) [34]. El nodo raíz define el caso más general del árbol de decisión, mientras que el nodo interno define una o varias condiciones sobre los atributos del conjunto de datos de entrenamiento. Ahora bien, los nodos terminales contienen la clase que indica el valor que se quiere predecir.

REDES BAYESIANAS (RB)

Las redes bayesianas son utilizadas como técnicas descriptivas y predictivas que modelan, cualitativa y cuantitativamente, las relaciones de dependencia entre los parámetros de interés [35]. Dado este modelo, permite inducir patrones probabilísticos para adquirir conocimiento del problema. Este método permite estimar explícitamente las probabilidades asociadas a cada una de las hipótesis posibles [36]. Las redes bayesianas son un modelo gráfico probabilístico que tiene dos componentes esenciales: un grafo a-

cíclico dirigido que muestra la dependencia e independencia entre las variables y un conjunto de tablas de distribución de probabilidad [37].

RED NEURONAL ARTIFICIAL (RNA)

Una RNA se define como un modelo no lineal y flexible inspirado en ciertas características asociadas al procesamiento de la información en el cerebro humano [38]. Su estructura está compuesta por neuronas interconectadas, que operan al mismo tiempo para resolver un problema específico [39]. De esta manera, las neuronas se organizan a través de tres tipos de capas: entrada, ocultas (intermedias) y de salida [39-41].

VECINO MÁS CERCANO (K-NN)

Por sus siglas en inglés (K-Nearest Neighbors) consiste en asignar a la instancia de entrada, la clase más frecuente de entre los K vecinos más cercanos localizados en el conjunto de entrenamiento [42]. Para identificar a los vecinos, el conjunto de datos de entrenamiento se representa por vectores en un espacio multidimensional [43]. K-NN utiliza frecuentemente estrategias como: distancia euclidiana, distancia Mahalanobis [42] y la distancia de Manhattan [43], las cuales se presentan en su forma vectorial para una distancia entre dos vectores.

MÁQUINAS DE VECTOR DE SOPORTE (MVS)

Una MVS realiza la clasificación de un conjunto de instancias mediante la construcción de un híper-plano N dimensional que separa de manera óptima los datos en dos o más clases [44]. Con base en lo anterior, busca encontrar la mayor distancia (margen máximo) que separe las clases para construir un modelo que sea capaz de predecir si un punto nuevo pertenece a una clase u otra [44]. Este método comúnmente se aplica cuando el conjunto de instancias son linealmente separables. En caso contrario, primero se debe transformar los datos [45], mediante un mecanismo conocido como *kerneltrick* [46].

2.1.3 Problemas de calidad en datos

Para obtener un correcto funcionamiento de algoritmos de aprendizaje supervisado descritos en la sección 2.1.2, los datos utilizados no deben ser redundantes y el número de instancias debe estar distribuido de manera uniforme según los valores que tome la variable objetivo (balanceo de clases). De esta manera, se hace necesario construir el dataset que evite valores redundantes, presente una distribución uniforme según los valores que tome la variable objetivo (clase), y además seleccionar las técnicas de aprendizaje supervisado idóneas, con la finalidad de detectar la calidad del agua en sistemas lóticos. Con base a esto, en este trabajo de grado se propone como primera medida la reducción de la dimensión para abordar el problema de valores redundantes y, en segunda instancia, utilizar el proceso de balanceo de clases para obtener una distribución uniforme de los datos. De igual manera, para seleccionar las técnicas con el mejor desempeño para detectar la calidad del agua, en el estado del arte (sección 2.2) se realiza una revisión sistemática de los algoritmos más importantes en los últimos cinco años.

REDUCCIÓN DE LA DIMENSIÓN

La reducción de dimensión es la transformación de datos de alta dimensión en una representación significativa de menor dimensión. Esta representación reducida debe tener una dimensión intrínseca de los datos, ósea el número mínimo de parámetros necesarios

para tener en cuenta las propiedades observadas de los datos [47, 48]. La reducción de la dimensión está orientada fundamentalmente hacia dos objetivos: técnicas de reducción de atributos y técnicas reducción de instancias (Figura 2).

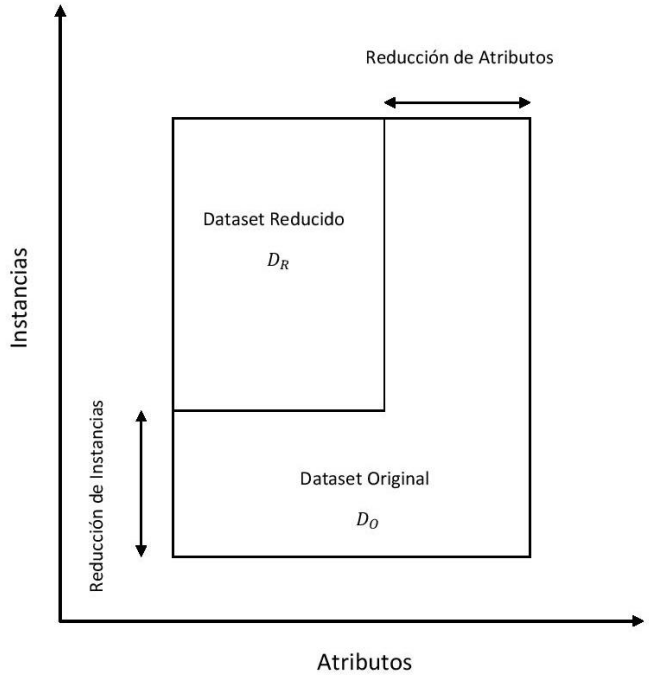
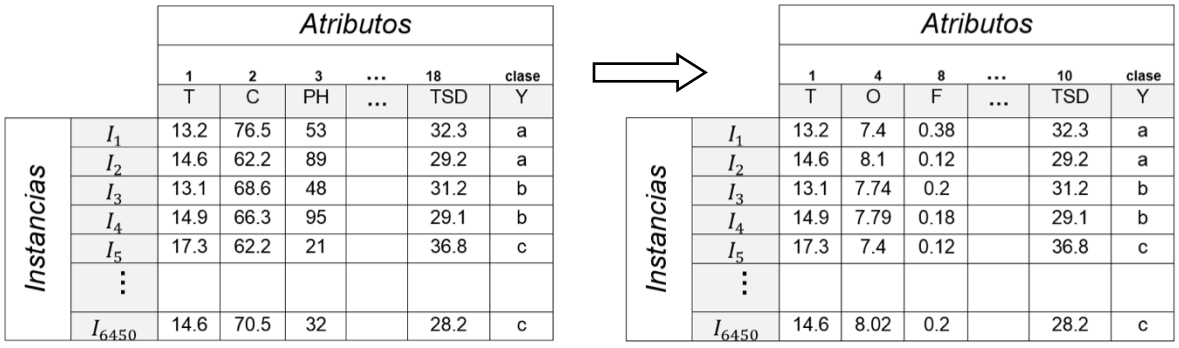


Figura 2. Reducción de la dimensión(fuente propia)

a) Reducción de atributos

La reducción de atributos (RA) disminuye la dimensión de los atributos o características de un dataset [48-51]. En la Figura 3, se puede observar que el dataset original, compuesto por 18 atributos fisicoquímicos (Figura 3(a)), es reducido a 10 características (Figura 3(b)), posterior al proceso de RA. Los métodos de RA se agrupan en dos categorías: selección y extracción de atributos; el primero busca el mejor subconjunto de características de acuerdo a un determinado criterio (elección de atributos y/o número de atributos a seleccionar), descartando atributos redundantes, inconsistentes, e irrelevantes, mientras que el segundo transforma el conjunto de atributos de alta dimensión en un espacio de menor dimensión [52, 53].



		Atributos					
		1	2	3	…	18	clase
		T	C	PH	…	TSD	Y
Instancias	I_1	13.2	76.5	53		32.3	a
	I_2	14.6	62.2	89		29.2	a
	I_3	13.1	68.6	48		31.2	b
	I_4	14.9	66.3	95		29.1	b
	I_5	17.3	62.2	21		36.8	c
	⋮						
	I_{6450}	14.6	70.5	32		28.2	c

a) Dataset original

		Atributos					
		1	4	8	…	10	clase
		T	O	F	…	TSD	Y
Instancias	I_1	13.2	7.4	0.38		32.3	a
	I_2	14.6	8.1	0.12		29.2	a
	I_3	13.1	7.74	0.2		31.2	b
	I_4	14.9	7.79	0.18		29.1	b
	I_5	17.3	7.4	0.12		36.8	c
	⋮						
	I_{6450}	14.6	8.02	0.2		28.2	c

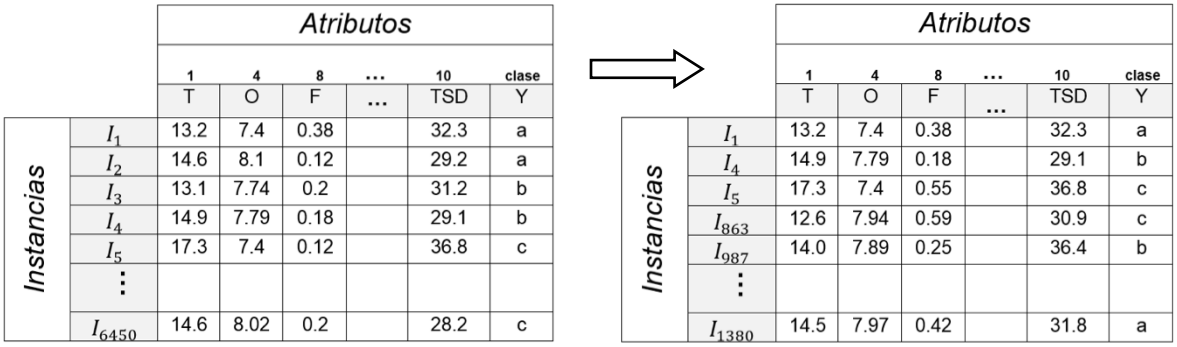
b) Dataset reducido en atributos

Figura 3. Ejemplo reducción de atributos (fuente propia)

Cabe destacar la importancia de los mecanismos para la RA dentro de la optimización de los conjunto de datos. Sin embargo, al aplicar técnicas para la selección de atributos se genera pérdida de información [49], el cual es un problema en dataset pequeños, como es el caso de los sistemas loticos que fueron presentados en la sección anterior. Por tal motivo, este trabajo se enfoca en métodos de extracción de características para dataset en sistemas lóticos. En la sección del estado del arte (sección 2.2) se presentan los algoritmos más utilizados en la última década pertenecientes a este enfoque de reducción.

b) Reducción de instancias o editado

La Reducción de instancias (RI) disminuye el número de las instancias irrelevantes dentro de un dataset [54-56]. En la Figura 4, es presentado un dataset con atributos fisicoquímicos, compuesto por 6450 instancias. Una vez aplicado el proceso de RI se obtiene como resultado un dataset constituido por 1380 ejemplos, descartando 5070 instancias las cuales representan datos inconsistentes, o información redundante.



		Atributos					
		1	4	8	…	10	clase
		T	O	F	…	TSD	Y
Instancias	I_1	13.2	7.4	0.38		32.3	a
	I_2	14.6	8.1	0.12		29.2	a
	I_3	13.1	7.74	0.2		31.2	b
	I_4	14.9	7.79	0.18		29.1	b
	I_5	17.3	7.4	0.12		36.8	c
	⋮						
	I_{6450}	14.6	8.02	0.2		28.2	c

		Atributos					
		1	4	8	…	10	clase
		T	O	F	…	TSD	Y
Instancias	I_1	13.2	7.4	0.38		32.3	a
	I_4	14.9	7.79	0.18		29.1	b
	I_5	17.3	7.4	0.55		36.8	c
	I_{863}	12.6	7.94	0.59		30.9	c
	I_{987}	14.0	7.89	0.25		36.4	b
	⋮						
	I_{1380}	14.5	7.97	0.42		31.8	a

Figura 4. Ejemplo reducción de instancias (fuente propia)

Diversos autores proponen clasificaciones de métodos para RI, como es el caso de [57], en el cual agrupan técnicas de RI en: “Noise Filters”, “Condensation Algorithms” y “Prototype Algorithms”, mientras que en [58] son clasificados en: Wrapper y Filter. Por otro lado, en [59-61] los métodos ensambladores como Cascading, Stacking, Bagging,

Boosting, Random Forest son utilizados para tareas de RI. La Tabla 3 presenta una breve descripción de los modelos para abordar RI.

Método	Descripción
Condensation	Las técnicas de condensación conservan las instancias que se encuentran en los límites de decisión y elimina las instancias interiores (instancias superfluas o no necesarias), debido a que estas últimas tienen poco efecto en el proceso de clasificación. Es decir, los métodos de condensación descartan las instancias que no influyan directamente con el resultado de la clasificación. La capacidad de reducción de los métodos de condensación es normalmente alta debido al hecho de que hay menos instancias en la frontera que instancias internas en la mayoría de los datos.
Prototype	Pretenden encontrar conjuntos de instancias tales que ofrezcan los mayores porcentajes de clasificación empleando la regla del vecino más cercano.
Filter	El método de filtro no hace uso de ningún clasificador, sino que intenta encontrar subconjuntos de instancias $S \subseteq T$, haciendo uso de simples estadísticas calculadas a partir de una distribución empírica de los datos (variable de selección). Es decir, este método evalúa un conjunto de instancias S de acuerdo a las características generales de los datos y decide si una instancia dada debe ser retenida o eliminada.
Envoltura	Los métodos de envoltura utilizan el conjunto entero de entrenamiento T para evaluar un subconjunto de instancias $S \subseteq T$, utilizando la validación cruzada para comparar el rendimiento del clasificador entrenado con cada subconjunto S evaluado. En otras palabras, el método de envoltura consiste en utilizar la predicción del rendimiento de un clasificador para evaluar la utilidad de los subconjuntos de instancias.
Ensamble	Los métodos de ensamble están formados por diferentes métodos de reducción de instancias que trabajan de manera conjunta, aprovechando el rendimiento de cada uno de los modelos de RI individuales y mejorando sustancialmente los resultados del proceso de selección de instancias.

Tabla 3. Modelos de reducción de instancias

En la sección del estado del arte (sección 2.2) se presentan los algoritmos más utilizados en la última década pertenecientes a este enfoque de reducción.

DESBALANCEO DE CLASES

El desbalanceo de clases ocurre cuando el número de instancias de una etiqueta de la clase es mayor (clase mayoritaria o negativa “C-”) respecto al número de instancias que tienen otras etiquetas de la clase (clases minoritarias o positivas “C+”) [62, 63]. En la Figura 5 se presenta un ejemplo de desbalanceo de datos, en el cual se observa mayor número de instancias de clase negativa, respecto al número de ejemplos de clase positiva.

En este escenario los clasificadores presentan una tendencia de clasificación hacia la clase mayoritaria, minimizando de ésta manera el error de clasificación y clasificando correctamente instancias de clase mayoritaria en detrimento de instancias de clase minoritaria. Esto quiere decir que la mayoría de los clasificadores tienden a trabajar con la clase mayoritaria e ignorar a la clase minoritaria [63, 64].

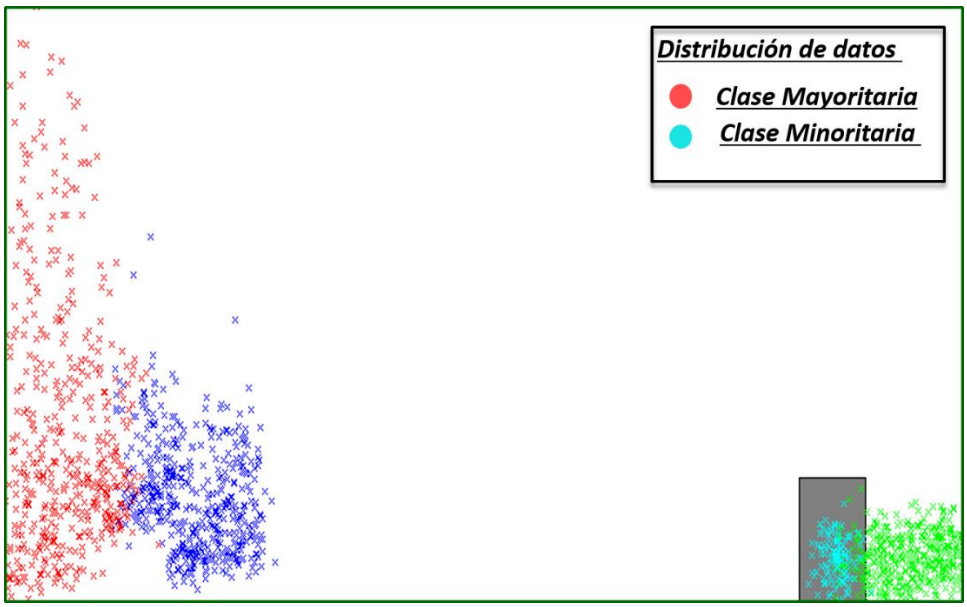


Figura 5. Distribución desbalanceada de datos (fuente propia)

Para medir el grado de desbalanceo de un dataset se utiliza la razón de desbalanceo (IR) [64-66] expuesta en la Ecuación 1. Cuando IR toma valores iguales a la unidad, se está indicando que el dataset se encuentra perfectamente balanceado. Por el contrario, cuando toma valores superiores a la unidad se indica que el dataset se halla desbalanceado (entre mayor sea el valor de esta razón mayor es el grado de desbalance del dataset).

$$IR = \frac{C_M}{C_m} \qquad \text{Ecuación 1}$$

En la expresión anterior, C_M representa el número de instancias que pertenecen a la clase mayoritaria y C_m es el número de ejemplos que pertenecen a la clase minoritaria [66]. Otra manera de expresar el nivel de desbalanceo es con la notación $n{:}\,1$ para $n \geq 1$, que indica la proporción del número de instancias pertenecientes a la clase mayoritaria con respecto a una instancias etiquetada a la clase minoritaria, es decir, que por cada *n* instancias de la clase mayoritaria, existe 1 instancia que pertenece a la clase minoritaria [67, 68]. En la sección del estado del arte (sección 2.2) se presentan los enfoques más utilizados en la última década.

2.2 Estado Del Arte

Con base en las definiciones presentadas anteriormente, en las siguientes sub-secciones, son expuestos los trabajos de investigación más relevantes en tres áreas de estudio del presente trabajo de grado: clasificadores para la evaluación de la calidad del agua, enfoques para la reducción de la dimensionalidad, y mecanismos para el tratamiento del desbalanceo de clases.

2.2.1 Clasificadores Para La Evaluación De La Calidad del Agua

La revisión del estado del arte en clasificadores para la evaluación de la calidad del agua, contiene trabajos de investigación desde el año 2010 hasta el presente, tomando como fuentes de búsqueda: IEEE (2), ScienceDirect (8 artículos) y Google (9 artículos). En la Figura 6 se presentan los trabajos utilizados con mayor frecuencia para la evaluación de la calidad del agua, haciendo uso de algoritmos de aprendizaje supervisado.

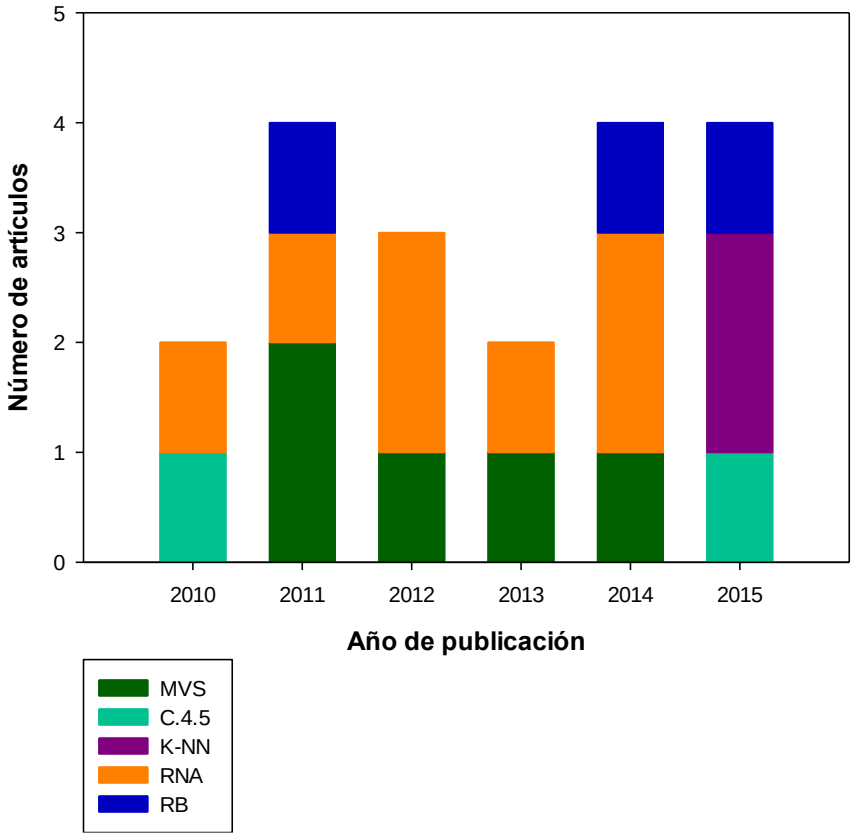


Figura 6. Revisión sistemática clasificadores para el monitoreo de la calidad del agua

En esta revisión se analizaron 19 artículos, en donde los algoritmos MVS (5 artículos) y RNA (7 artículos) son utilizados con mayor frecuencia para predecir y monitorear la calidad del agua, monitorear el comportamiento de organismos acuáticos y percibir sustancias contaminantes o toxicas. Por otra parte, los algoritmos C.4.5 (2 artículos), K-NN (2 artículos) y RB (3 artículos), fueron utilizados con el fin de predecir, evaluar y generar alertas de calidad de agua, reducción de la contaminación y establecer el criterio de nitrógeno de los ríos, teniendo en cuenta variables fisicoquímicas y organismos acuáticos.

Es importante señalar que el presente proyecto busca seleccionar un clasificador que genere un alto grado de precisión en la predicción y un clasificador de fácil interpretación. Para esto, se evalúan y seleccionan en el Capítulo 5 los algoritmos de aprendizaje

supervisado explicados anteriormente para los dataset: rio Las Piedras y estuario de California.

2.2.2 Revisión Sistemática - Reducción De La Dimensionalidad

Para la selección de las técnicas de reducción de la dimensión, se realizó la revisión sistemática de investigaciones que tratan la reducción de la dimensión de manera independiente, en términos de atributos e instancias bajo diferentes contextos.

REVISIÓN SISTEMÁTICA - REDUCCIÓN DE ATRIBUTOS

En la Figura 7 son presentadas las investigaciones más relevantes que utilizan técnicas de extracción de características desde el año 2004 hasta el presente, tomando como fuentes de búsqueda: IEEE Xplore (35 artículos), SienceDirect (9 artículos). En esta revisión se analizaron 44 artículos enfocados diferentes dominios de aplicación como: la detección de intrusos, la medicina, los sistemas biométricos, reconocimiento facial, clasificación de imágenes satelitales, entre otras.

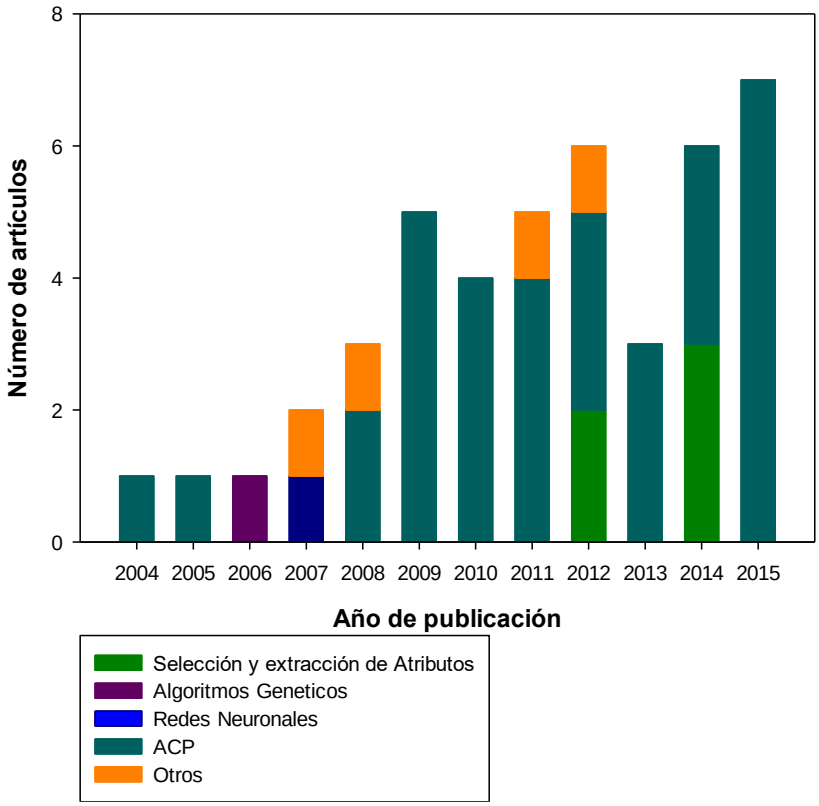


Figura 7. Revisión sistemática técnicas de extracción de características

Así, 5 de los artículos revisados utilizan técnicas de selección y extracción de atributos de manera individual y combinada como: algoritmos tipo filtro, de envoltura, análisis de componentes independientes (ACI), análisis lineal discriminante (ALD) y análisis de componentes principales probabilísticos (ACPP). Otros autores utilizan algoritmos genéticos (1 artículo), redes neuronales (1 artículo), algoritmos basados en la regla del

vecino más cercano (1 artículo), reglas de asociación (1 artículo), probabilidad y estadística (2 artículos). Sin embargo, el algoritmo: Análisis de Componentes Principales (ACP) es el más utilizado (33 artículos) en los últimos 8 años (desde el 2008 hasta el presente).

Debido a que el algoritmo de Análisis de Componentes Principales es el más utilizado según la revisión literaria expuesta anteriormente, este se tomará como punto de partida para la extracción de atributos que permitan evaluar de forma adecuada la calidad del agua, en los dataset detallados en la sección 2.1.

REVISIÓN SISTEMÁTICA REDUCCIÓN DE INSTANCIAS

En la Figura 8 son presentadas las investigaciones más importantes que utilizan métodos para la reducción de instancias desde el año 2006 hasta el presente, tomando como fuentes de búsqueda: IEEE Xplore (12 artículos), SienceDirect (10 artículos), Springer Link (6 artículos) y Google Scholar (6 artículos). En esta revisión se analizaron 34 artículos enfocados en diferentes áreas de aplicación como: detección de intrusiones bancarias, seguridad informática, construcción de clasificadores, series de tiempo, reconocimiento de texto, entre otras.

En este sentido, los métodos "Envoltura" (los métodos de envoltura utilizan la predicción del rendimiento de un clasificador para evaluar la utilidad de los subconjuntos de instancias) y "Ensamble" (están formados por diferentes métodos de reducción de instancias que trabajan de manera conjunta, aprovechando el rendimiento de cada uno de los modelos de RI individuales y mejorando sustancialmente los resultados del proceso de selección de instancias) son los más utilizados en la literatura a lo largo de la última década, con 12 y 16 artículos respectivamente, mientras que los métodos tipo "Filtro" solo son referenciados 6 veces durante este mismo periodo.

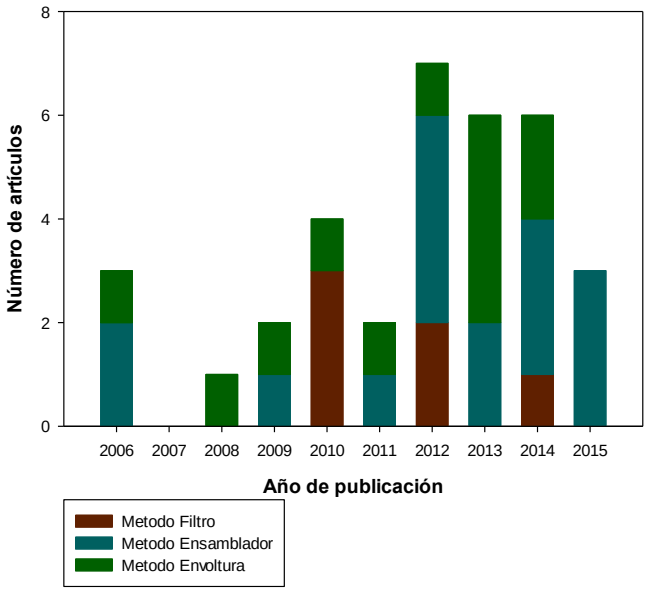


Figura 8. Revisión sistemática técnicas de extracción de instancias

Es importante señalar que los algoritmos tipo Envoltura tienden a ser sobre entrenados (overfiting), debido al uso frecuente de la técnica de evaluación: validación cruzada sobre un único dataset, el cual tiende a ser ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo [69]. Con base en las razones expuestas anteriormente, este trabajo de investigación tomará como punto de partida el método Ensamble, de manera más precisa Boosting su mayor representante [60], para RI debido a la alta frecuencia de uso, el cual es presentado en detalle en la sección 3.2.

2.2.3 Revisión Sistemática - Desbalanceo De Clases

Para dar solución al desbalanceo de clases, se han presentado principalmente dos enfoques [63, 68, 70]: el método externo (a nivel de datos) y el método interno (a nivel de algoritmos de clasificación). El primero consiste en alcanzar un balance entre las clases mediante la eliminación de instancias de la clase mayoritaria (sub-muestreo) o la inclusión de instancias en la clase minoritaria (sobre-muestreo), mientras el método interno ajusta los clasificadores para favorecer la clase minoritaria.

En este orden de ideas, en la Figura 9 son presentados los estudios de investigación que utilizan tanto métodos internos como externos para el tratamiento de datos desbalanceados desde el año 2004 hasta el presente año, tomando como fuentes de búsqueda: IEEE Xplore (1 articulo), SienceDirect (14 artículos), Springer Link (3 artículos), DL ACM (3 artículos) y Google Scholar (9 artículos). En esta revisión fueron analizados 30 artículos enfocados en diferentes dominios de aplicación como: la detección de intrusos, la medicina, reconocimiento facial, clasificación de imágenes satelitales, entre otras.

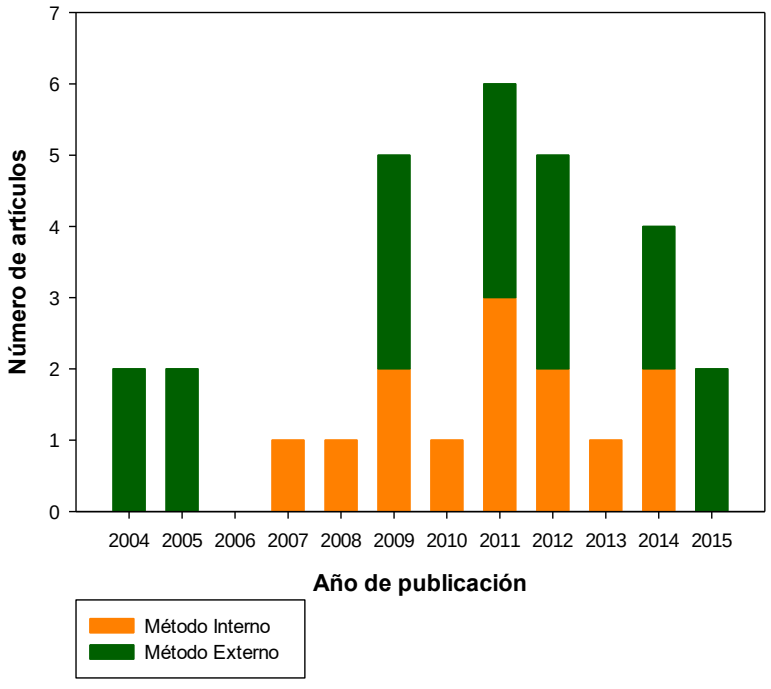


Figura 9. Revisión sistemática modelos para el tratamiento de clases desbalanceadas

Así, 17 de los artículos revisados hacen uso del enfoque para solucionar el problema de desbalanceo de clases, con el fin de mejorar la eficacia de clasificación y las 13 investigaciones restantes utilizan el método interno para el mismo propósito. Adicionalmente se tiene, que los autores en [65, 68, 71] mencionan que el método interno, aunque no altera la distribución de los datos del dataset, tiene el gran inconveniente de depender de la naturaleza del algoritmo y del dominio de aplicación (ajusta un clasificador para resolver un problema específico por medio de un método especifico, es decir, se ajusta el clasificador para que funcione sobre un único dataset desbalanceado), lo que dificulta explotar sus funcionalidades frente a la resolución de otra clase de problemas.

Del mismo modo se encuentra el método externo, el cual modifica la distribución de instancias del dataset, pero tiene la gran ventaja de ser independiente del algoritmo de aprendizaje. Esto indica que tiene la flexibilidad de hacer frente a la resolución de otras clases de problemas, es decir, puede hacer frente a cualquier clase de dataset desbalanceado. De esta forma, el método externo resulta ser más versátil, ya que no requieren ningún cambio en el algoritmo y, además, puede ser utilizado en diferentes dominios de aplicación.

Partiendo de las consideraciones presentadas anteriormente, en donde se expone que el enfoque externo es la técnica más versátil y la más utilizada en la última década, se toma la metodología externa como punto de partida para hacer frente al desbalanceo de clases de los dataset pertenecientes a los sistemas lóticos, la cual se presenta a continuación en mayor detalle:

MÉTODO EXTERNO

El método externo es una de las formas de resolver el problema de clasificación con clases desbalanceadas, cuyo objetivo consiste en realizar un re-muestreo en la distribución de los datos para obtener un mejor balance entre las clases, ya sea agregando casos sintéticos a la clase minoritaria (sobre-muestreo), o eliminando casos o instancias de la clase mayoritaria (sub-muestreo) [63, 70].

El re-muestreo de datos, en sus dos técnicas, presenta tanto ventajas como desventajas: el sub-muestreo puede provocar perdida de la información al eliminar instancias útiles de la clase mayoritaria, pero tiene como ventaja la reducción del tiempo de procesamiento del dataset. Por el contrario, está el sobre-muestreo de datos, el cual tiene la ventaja de no perder información importante. No obstante, esta técnica puede agregar instancias con ruido a la clase minoritaria, ocasionando un sobreajuste (sobre entrenamiento) del clasificador y un aumento en el tiempo de procesamiento del dataset.

En trabajos como [65, 72, 73], los autores exponen que el proceso de sobre-muestreo supera consistentemente el proceso de sub-muestreo cuando el dataset está fuertemente desbalanceado. Por esta razón, y teniendo en cuenta que los dataset con los que se cuenta en este trabajo tienen pocas instancias (no es factible quitarle aún más instancias), en este trabajo de investigación se opta por utilizar el sobre-muestreo como técnica para mitigar el problema del desbalanceo de clases.

2.2.4 Análisis de Componente Principales (ACP)

Esta técnica multivariada procedente del análisis exploratorio de los datos (estadística), cuya finalidad es sintetizar la información (reducir la dimensión), perdiendo con ello la

menor cantidad de información posible. Dicho de otra forma, esta técnica transforma un número grande de atributos correlacionados entre sí, en un número de atributos más pequeño no correlacionados (linealmente independientes u ortogonales), denominados componentes principales (CP). Los componentes principales son una combinación lineal de las características originales, que describen la mayor cantidad de información del dataset y están ordenados según la cantidad de información que contengan (de mayor a menor) [74-76].

Esta técnica busca seleccionar los CP en los cuales se concentre la mayor cantidad información (mayor varianza) y omitir los que no contribuyen significativamente a la variabilidad observada (variables redundantes e irrelevantes). Adicionalmente se basa en los siguientes criterios:

- Los CP obtienen secuencialmente la máxima varianza de *X*, por lo que se garantiza la mínima pérdida de información.
- Los CP obtenidos son ortogonales entre sí, facilitando su interpretación, ya que pueden tratarse de manera independiente.

$$\boldsymbol{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix}$$ Ecuación 2

$$\boldsymbol{C} = \begin{bmatrix} C_{11} & C_{21} & \cdots & C_{n1} \\ C_{12} & C_{22} & \cdots & C_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1m} & C_{2m} & \cdots & C_{nm} \end{bmatrix}$$ Ecuación 3

En este sentido en la Ecuación 2 se presenta la matriz del dataset $\boldsymbol{X_{n*p}}$, donde *p* corresponde al número de variables observadas y *n* al número de individuos o unidades de observación. La técnica en cuestión busca la forma de representar adecuadamente la información, con un número menor de variables que son construidas como combinaciones lineales de las originales. En otras palabras, este método permite representar en un espacio de dimensión pequeña *m*-dimensional las observaciones de un espacio *p*-dimensional (*m*<*p*), tal que al proyectar los puntos sobre el espacio *m*-dimensional, estos conserven su estructura con la menor distorsión posible.

Adicionalmente, el ACP permite transformar los atributos originales (correlacionados entre sí) en nuevas variables no correlacionadas que facilitan la interpretación, debido a su naturaleza independiente. Dada la matriz de datos *X* presentada en la Ecuación 2, se obtiene como resultado de ésta transformación la matriz $\boldsymbol{C_{n*m}}$, con *m* variables de salida (CP) y *n* observaciones, denominada matriz característica (Ecuación 3) y en la cual los primeros *m* componentes $C_1, C_2, \dots, C_m$, donde $1 \leq m \leq p$, son linealmente independientes y denotan la mayor cantidad de varianza.

Generalmente, dichos componentes son representados por los vectores propios asociados a los primeros *m* valores propios (VP) provenientes de la matriz de covarianzas $S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^T(x_i - \bar{x})$, donde, $x_i \in \boldsymbol{X_{n*p}}, \bar{x}$ representa la media y *n* el número de

instancias de la matriz de datos y cada vector de datos original puede ser representado por su vector de componentes principales tal y como se expone en la Ecuación 4:

$$y = C^T x \qquad \text{Ecuación 4}$$

La matriz C^T transforma linealmente el espacio de los datos de entrada en *X* en otra matriz *C* con un número menor de variables. Este proceso se realiza mediante una combinación lineal de las variables originales de modo que se proyecten sobre las direcciones de máxima varianza de los datos, conservando así la máxima cantidad de información.

Actualmente no existe una regla definida sobre el número exacto de CP que se debe utilizar. Sin embargo, para seleccionar el subconjunto óptimo de CP, los autores en [77] exponen diversas técnicas tales como: B1-Backward, B1-Forward, B2 y B4, obteniendo mejores resultados el método B4.

B4 toma los primeros CP representadas por los valores propios más altos, los cuales corresponden a la porción de varianza asociada a cada componente $\frac{\widehat{\lambda_i}}{\sum_{k=1}^{m}\widehat{\lambda_k}}$ y propone tomar los primeros componentes (valores propios más altos), tales que la suma de su varianza acumulada oscile entre $60\% < \frac{\sum_{k=1}^{i}\widehat{\lambda_k}}{\sum_{h=1}^{m}\widehat{\lambda_h}} < 95\%$. En otras palabras, B4 propone utilizar los CP cuya porción de varianza explicada acumulada supere el 60% del total de la información.

Si bien es cierto que B4 es el método que obtiene mejores resultados según [77], el presente trabajo de investigación propone un nuevo enfoque para la selección de CP, el cual es presentado en la sección 3.1.

2.2.5 Selección de Instancias Boosting (SIB)

Boosting es un método iterativo que asigna pesos a cada instancia (inicialmente el mismo para todos). Cada vez que itera, se construye o entrena un modelo de clasificación al que denominan clasificador base, y el cual minimiza la suma de los pesos de aquellas instancias clasificadas erróneamente. De esta forma, los pesos de cada instancia se reajustan en función de este resultado (error de clasificación), incrementando el peso de los mal clasificados y reduciendo el peso en aquellos que han sido correctamente clasificados. La estructura grafica de este proceso se puede observar en la Figura 10.

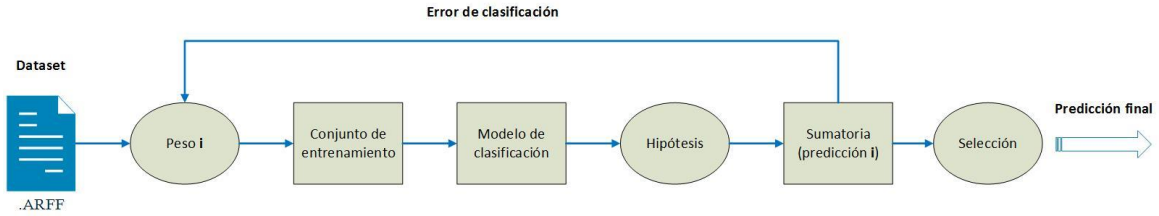


Figura 10. Proceso Boosting (fuente propia)

En este orden de ideas, desde el área de la RI, en [60] toman como punto de referencia el algoritmo AdaBoost (Boosting adaptativo) y reemplazan los modelos de clasificación por algoritmos de RI como: Decremental Reduction Optimization Procedure 3 (Drop3), IB3, Iterative Case Filtering (ICF), Modified Selective Subset (MSS), Reduced Nearest Neighbor (RNN), Condensed Nearest Neighbor Rule (CNN). En la Tabla 4 se realiza una breve descripción de los métodos de RI.

Algoritmo	Descripción
Decremental Reduction Optimization Procedure 3 (Drop3)	Ordena las instancias acuerdo a la distancia de sus vecinos más cercanos y elimina las instancias más alejadas. Se basa en la regla del vecino más cercano K-NN
IB3	Utiliza una prueba de significancia (90%) para determinar qué instancias son bien clasificadas y cuales son instancias ruidosas.
Iterative Case Filtering (ICF)	Define dos conjuntos: cobertura (C) y accesibilidad (R). El algoritmo se basa en aplicar repetidamente la regla de eliminación |R|>|C| para el conjunto de instancias hasta que no existan más casos que cumplan con la regla de eliminación. Se basa en la regla del vecino más cercano K-NN
Modified Selective Subset (MSS)	Consiste en encontrar subconjuntos mínimamente consistentes utilizando la propiedad de selectividad. Un subconjunto selectivo se define como un subconjunto del conjunto de entrenamiento que contiene, para cada instancia x en el conjunto de entrenamiento, el elemento de su vecindario más cercano a una clase distinta de la de x.
Reduced Nearest Neighbor (RNN)	Selecciona la primera instancia y luego procede iterativamente. Cada instancia del conjunto de entrenamiento es clasificada con el conjunto actual de casos seleccionados; si está mal clasificada la instancia, esta se agrega al conjunto seleccionado y se reinicia el proceso. El método termina cuando todos los casos se clasifican correctamente.
Condensed Nearest Neighbor Rule (CNN)	Su paso inicial consiste en incluir aleatoriamente en *S* una instancia de cada clase. Después de este paso cada instancia en *T* se clasifica utilizando *S* como el conjunto de entrenamiento. Si una instancia *p* es mal clasificada, entonces *p* es incluido en *S* para asegurar que los nuevos casos similares a *p* se clasificarán correctamente.

Tabla 4. Descripción algoritmo de RI utilizados

El proceso del modelo SIB itera tantas veces como indique el usuario (el número de iteraciones es denotado por la letra *M*), donde el valor de *M* es proporcional tanto al grado de precisión del algoritmo como de su coste computacional. Lo anterior quiere decir que si el valor de *M* aumenta, el nivel de precisión del algoritmo y el coste computacional también lo harán.

Inicialmente SIB asigna el peso $w_i = \frac{1}{N}$ a todas las *N* instancias del dataset. Posteriormente en cada iteración, es seleccionado un algoritmo de RI de manera aleatoria, el cual identifica el conjunto de instancias más representativas del dataset y construye con ellos un nuevo dataset. De manera seguida, este dataset se transfiere al clasificador K-NN (De hecho, cualquier otro clasificador puede ser usado; Sin embargo, para este trabajo de investigación nos limitaremos al caso de una regla K-NN debido a su alta velocidad de aprendizaje [78]), el cual reajusta los pesos de la siguiente iteración, de acuerdo la precisión y el error de la clasificación.

Es importante mencionar que en cada repetición del proceso de SIB se realiza un proceso de votación, que consiste en asignar un voto a cada instancia seleccionada por una determinada técnica de RI seleccionada se manera aleatoria. Estos votos son contados y, de acuerdo a este conteo, se arma un posible conjunto de umbrales con las instancias

que obtuvieron mayor votación, es decir, a partir del proceso de votación se genera el conjunto de posibles umbrales.

Después de las *M* iteraciones se obtiene como resultado un vector de votos *V*, que registra los votos obtenidos por cada instancia x_i y, a partir de este último se construye un conjunto de umbrales $(\emptyset_1, \emptyset_2, \dots, \emptyset_M)$ con las instancias que obtuvieron mayor cantidad de votos, donde el número máximo de umbrales es el número de iteraciones M (en cada iteración se selecciona un posible umbral). La definición de la votación más alta como umbral $\emptyset$ no logra buenos resultados debido a que al final del proceso son seleccionadas pocas instancias (solo las que superen dicho umbral) y además la precisión de clasificación es bastante baja. Lo contrario ocurre cuando se define como umbral la votación más baja, donde al final del proceso son seleccionadas la mayoría de las instancias, lo que implica gran pérdida de información del dataset. En el mismo sentido, es poco probable que un umbral estático (definido de manera a priori) sea apropiado para cualquier dataset, ya que puede funcionar para algunos dataset y para otros no.

A partir de las anteriores consideraciones, en [60] proponen el uso de un umbral dinámico, que funciona para cualquier dataset. Con este umbral se limita la selección de instancias a aquellas instancias cuyos registros de votación están por encima del mismo, tal y como se expone en la Ecuación 5.

$$v_i > \emptyset \qquad \text{Ecuación 5}$$

Para obtener el mejor umbral definen un criterio $J(\emptyset)$, el cual es representado por un subconjunto de instancias $S_\emptyset$ pertenecientes al conjunto de entrenamiento, de tal forma que cumplan con la (5). Este criterio se evalúa para todos los posibles umbrales $(\emptyset_1, \emptyset_2, \dots, \emptyset_M)$, es decir, por cada posible umbral $\emptyset_i$ se define su respectivo criterio $J(\emptyset_i)$ que es representado por un conjunto de instancias $S_{\emptyset_i}$ que cumplan con la Ecuación 6.

$$\theta: S_\theta = x_i \in T: v_i > \emptyset \qquad \text{Ecuación 6}$$

De esta forma, para evaluar cada criterio $J(\emptyset_i)$ se entrena el clasificador K-NN con el subconjunto $S_{\emptyset_i}$, donde la calidad de cada subconjunto de instancias seleccionadas se encuentra en función tanto del rendimiento de clasificación como de la cantidad de instancias eliminadas, tal y como se observa en la Ecuación 7.

$$J(\theta) = \alpha C + (1 - \alpha) r \qquad \text{Ecuación 7}$$

Donde, C simboliza el rendimiento de clasificación (precisión o AUC), r denota el porcentaje de instancias eliminadas y α es el parámetro utilizado para distinguir la importancia de los factores $C \; y \; r$.

Por otro lado, se tiene que la evaluación de cada umbral implica entrenar el clasificador K-NN con el subconjunto de instancias que lo representa, lo que implica un alto grado de

complejidad del proceso de evaluación $(2M + 2)$ cuando el número de iteraciones M es alto (el máximo número de posibles umbrales es el número de iteraciones M).

2.2.6 Sobre-Muestreo

Los estudios existentes orientan el sobre-muestreo a dos enfoques: el enfoque aleatorio, el cual adiciona ejemplos a la clase minoritaria duplicando de manera aleatoria instancias pertenecientes a esta clase; y el enfoque denominado SMOTE (Synthetic Minority Over-sampling Technique), que agrega ejemplos a la clase minoritaria por medio de la creación de nuevas instancias (a las que denominan datos sintéticos), obtenidas a partir del proceso de interpolación.

Es importante destacar que en el sobre-muestreo aleatorio se presenta con mayor frecuencia el problema de overfiting (sobreajuste), y con ello, resultados más imprecisos que en SMOTE, debido a que en el primero se replican instancias existentes, mientras que en el segundo se crean nuevas instancias [79, 80]. Tomando como referencia la revisión de los trabajos relacionados descritos en la sección 2.2.3, SMOTE es el algoritmo que se tomó como punto de partida para resolver este tipo de problemas. A continuación es presentado:

SMOTE

El algoritmo SMOTE crea datos o instancias sintéticas en la clase minoritaria, a través de la interpolación de una instancia y sus k vecinos más cercanos pertenecientes a dicha clase. En primer lugar, son seleccionados los K vecinos más cercanos pertenecientes a la clase minoritaria y, posteriormente, se elige el porcentaje de instancias sintéticas a crear (denotado por P). Seguidamente, para generar un nuevo dato sintético, se realiza la interpolación entre la línea que une cada instancia de la clase minoritaria con alguno (o todos) de sus k vecinos más cercanos seleccionados previamente. Este cálculo se realiza utilizando la definición de la distancia euclidiana y usando una función de reemplazo o superposición (denominada overlap) que asigna un valor 0 (en caso de que ambos valores sean iguales) o un valor 1 (en el caso de que sean diferentes) [81].

Es importante señalar que SMOTE únicamente se aplica sobre clases binarias (una clase mayoritaria y otra minoritaria), es decir, contempla únicamente una clase mayoritaria y una minoritaria, e ignora el resto de clases [82].

El proceso descrito anteriormente es ilustrado en la Figura 11, donde x_i es la instancia seleccionada y $x_{i1}, x_{i2}, \dots, x_{i6}$ son sus k vecinos más cercanos (en este ejemplo k=6), mientras que $r_1, r_2, \dots, r_6$ son los datos sintéticos creados mediante interpolación.

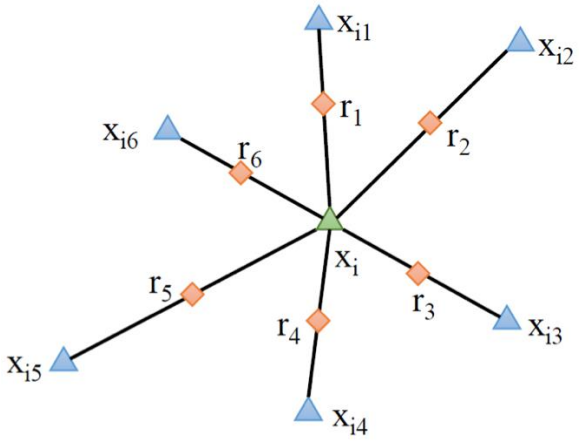


Figura 11. Representación gráfica de la creación de datos sintéticos con SMOTE, con k=6. Tomada de [83]

Aunque esta técnica genera nuevas instancias que permite el balanceo de clases, hasta el momento no se ha encontrado un método que indique el número óptimo de instancias que se deben crear, ya que un número elevado de instancias sintéticas puede sobre-entrenar el clasificador y generar resultados imprecisos [70, 82, 84]. En este sentido, el presente trabajo de investigación propone una estrategia para generar el número adecuado de datos sintéticos de tal forma que minimice el sobre-entrenamiento del clasificador.

2.3 Resumen

Este capítulo presentó los datos de entrenamiento (dataset) para la detección de la calidad del agua, el concepto de aprendizaje supervisado y sus principales técnicas (MVS, RNA, C.4.5, K-NN, y RB), con el fin de comprender la temática del presente trabajo de pregrado.

Posteriormente fueron expuestos los trabajos relacionados respecto al problema de investigación declarado. Las propuestas de estos trabajos fueron orientadas, en primera instancia, el uso de clasificadores para llevar a cabo la detección de la calidad del agua, y en segunda instancia hacia el desarrollo de un mecanismo de pre-procesamiento de los datos (reducción de la dimensión y desbalanceo de clases) en diferentes dominios de aplicación. Finalmente se presenta la definición de cada una de las técnicas utilizadas para el procesamiento de los datos.

Capítulo 3

3 Reducción de la dimensionalidad

En este capítulo se presentan de manera individual y detallada las técnicas de reducción de la dimensionalidad seleccionadas en la sección 2.2.2. Adicionalmente, se exponen los aportes realizados y, por último, se exhibe la experimentación y los resultados obtenidos.

3.1 Selección Automática de Componentes Principales (SAC)

El mecanismo propuesto consiste en construir subconjuntos de datos $\boldsymbol{X'}_{\boldsymbol{n*p'}}$, para $p' = 1,2,\dots,p$ y $p = VP$ (ACP determina tantos valores propios (VP) como atributos *p* haya en el dataset) con p' variables y *n* observaciones, a partir de cada uno de los valores propios y sus respectivos vectores propios.

Supongamos que se tiene un dataset con *m* atributos y *n* instancias al cual se aplica el proceso de ACP, como resultado se obtiene *m* VP organizados de acuerdo a la cantidad de información que representan (de mayor a menor varianza) y sus respectivos vectores propios (vector columna de *n***x**1) asociados. El nuevo dataset está conformado por dichos vectores, donde cada vector está representado por una componente principal (X_i). En la Figura 12 se presenta de manera gráfica este proceso, y se puede observar que cada componente principal presenta un determinado porcentaje de información y se disponen de izquierda a derecha de acuerdo a la cantidad de información que represente (de mayor a menor respectivamente).

		Componentes Principales					
		X_1	X_2	X_3	…	X_m	Y
Nuevas Instancias	I_1	0.47	0.12	0.53		0.41	a
	I_2	0.25	0.47	0.89		0.69	a
	I_3	0.69	0.36	0.48		0.47	b
	I_4	0.77	0.47	0.95		0.85	b
	I_5	0.69	0.78	0.21		0.52	c
	⋮						
	I_n	0.85	0.81	0.32		0.13	c
Valores propios		$\boldsymbol{VP_1}$	$\boldsymbol{VP_2}$	$\boldsymbol{VP_3}$		$\boldsymbol{VP_m}$	
Información		45%	25%	10%		5%	

Figura 12. Matriz de vectores propios

A partir de aquí, se construyen m subconjuntos de datos con base en el acumulado de $\{1\},\{1+2\},\{1+2+3\},\dots,\{1+2+3+m\}$ componentes principales (línea 7 del Algoritmo 1). Posteriormente, se entrena las técnicas supervisadas: MVS, RB, K-NN, AD y RNA con

cada subconjunto de datos, y se calcula el promedio de las precisiones de los clasificadores P_i (línea 13 del Algoritmo 1), como se puede observar en la Tabla 5.

Subconjunto de datos	CP	Precisión promedio
1	CP_1	P_1
2	CP_1, CP_2	P_2
3	CP_1, CP_2, CP_3	P_3
m	$CP_1, CP_2, CP_3, \dots, CP_m$	P_m

Tabla 5. Subconjunto de datos generados

Finalmente, es seleccionado el subconjunto de datos que mejor precisión obtuvo y con ello el número de CP asociado a este (líneas 15 y 17 del Algoritmo 1). El método para la selección automática de componentes principales propuesto, se puede observar en detalle en el Algoritmo 1.

Algoritmo 1. Selección automática de componentes principales

***Datos de entrada**: conjunto de entrenamiento: T con M atributos y N instancias*
***Datos de salida**: conjunto de componentes principales seleccionadas*

1. $S' = T$, *se normaliza*
2. *Calcular la media* $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
3. *Calcular matriz de covarianzas* $S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^T(x_i - \bar{x})$
4. *Calcular M valores propios* $V_M(S)$
5. *Calcular M vectores propios* v_M *(S)*
6. ***for*** $m \rightarrow 1$ ***to*** *M* ***do***
7. *Construir matriz característica* $MC_m(v_m)$ */*organizados de acuerdo al valor de los valores propios de mayor a menor valor*/*
8. */*Se entrena los clasificadores:* C_{RN}, C_{MVS}, C_{K-NN} *y* $C_{C.4.5}$ *con la matriz* MC_m *y se calcula la precisión*/*
9. $p_1 = C_{RN}(MC_m)$
10. $p_2 = C_{MVS}(MC_m)$
11. $p_3 = C_{K-NN}(MC_m)$
12. $p_4 = C_{C.4.5}(MC_m)$
13. *Cálculo de precisión* $P_m = \frac{p_1+p_2+p_3+p_4}{4}$
14. ***end for***
15. *Seleccionar la mayor precisión* P_{Mayor}
16. *Asociar N vectores propios* $Nv_M(P_{Mayor})$ */*cada vector propio representa una componente principal*/*
17. *Construir matriz:* M_s
18. ***Return*** M_s

Cabe resaltar que para el análisis de los componentes principales se debe observar la relación entre los CP y los atributos iniciales a partir de la matriz característica, teniendo en cuenta el signo y la magnitud de las correlaciones.

3.2 Reducción De Instancias Propuesto (RI-E)

Para hacer frente al inconveniente presentado en el ítem anterior, en este trabajo se propone la selección de un umbral óptimo al que se denomina $\emptyset_o$, a partir de un enfoque que funciona en dos niveles. El primer nivel (línea 32 del Algoritmo 2) consiste en construir un subconjunto de umbrales $(\theta_1, \theta_2, \dots, \theta_p)$ a partir del conjunto de posibles umbrales $(\emptyset_1, \emptyset_2, \dots, \emptyset_M)$ para $p \leq M$, descartando los umbrales repetidos o equivalentes de este último, debido a que estos obtienen evaluaciones similares en términos de rendimiento de clasificación y de reducción. De esta manera, se logra un grado de complejidad menor o igual $(2\mathrm{p} + 2)$.

En el segundo nivel (línea 34 del Algoritmo 2) se busca encontrar un valor de umbral que represente tanto a los valores de votación altos como los valores bajos y de este modo contrarrestar los inconvenientes mencionados anteriormente. Para ello, se determina la media aritmética del subconjunto de umbrales resultante del paso anterior, proceso que es representado por la Ecuación 8.

$$\emptyset_o = \frac{1}{p}\sum_{i=1}^{p} \theta_i \qquad \text{Ecuación 8}$$

En el Algoritmo 2 se presenta de manera formal la nueva propuesta del método de reducción de instancias RI-E.

Algoritmo 2. Reducción de instancias RI-E

***Datos de entrada**: conjunto de entrenamiento: T con N instancias, número de iteraciones M*
***Datos de salida**: conjunto de instancias seleccionadas*

1. $S' = T$, se asigna el peso $w_i = \frac{1}{N}$ a todas las instancias
2. **for** $m \rightarrow 1$ **to** M **do**
3. *Selección aleatoria del algoritmo de RI*
4. *Aplicar algoritmo de RI:* $S_m = RI(S')$
5. */*Se entrena el clasificador K-NN con el conjunto de instancias seleccionadas* S_m*, el clasificador se simboliza como* C_{Sm} **/*
6. $\varepsilon_m = \frac{1}{N}\sum_{x_j \in T:\, C_{Sm}(x_j) \neq y_j} w_j$ */*Actualiza el valor del peso*/*
7. **if** $\varepsilon_m > 0.5$ **then**
8. $\alpha_m = 0$
9. *Muestrea a* S' *a partir de T y actualiza el peso w tal que* $\sum w_i = 1$
10. *Elimina las instancias repetidas de* S'
11. **else if** $\varepsilon_m = 0$
12. $\alpha_m = 1$
13. *Muestrea a* S' *a partir de T y actualiza el peso w tal que* $\sum w_i = 1$
14. *Elimina las instancias repetidas de* S'
15. **else**
16. $\alpha_m = \frac{1}{2}\ln\frac{1-\varepsilon_m}{\varepsilon_m}$
17. **foreach** $x_j \in T$ **do**
18. **if** $C_{Sm}(x_j) = y_j$ **then**
19. $w_j = \frac{w_j}{2(1-\varepsilon_m)}$
20. **end if**

21. **end foreach**
22. **end if**
23. *Normalizar w tal que* $\sum w_i = 1$
24. *Obtener* S' *como una muestra de T usando el peso w*
25. *Elimina las instancias repetidas de* S'
26. *Realizar proceso de votación de la instancias* x_j
27. *Almacenar j-esimo voto en el vector* v_j
28. *Realizar conteo de votos /* sirve como posible umbral* $\emptyset_j$ */*
29. *Armar posible conjunto de umbrales* $(\emptyset_1, \emptyset_2, \dots, \emptyset_j)$
30. $m = m + 1$
31. **end for**
32. *Eliminar umbrales equivalentes del conjunto de umbrales* $(\emptyset_1, \emptyset_2, \dots, \emptyset_M)$
33. *Crear nuevo conjunto de umbrales* $(\theta_1, \theta_2, \dots, \theta_p)$
34. $\emptyset_o = \frac{1}{p}\sum_{i=1}^{p} \theta_i$
35. */*Retorna el umbral representativo */*
36. **Return** $\emptyset_o$
37. */*Realiza la selección instancia utilizando el umbral de votos obtenido* $\emptyset_o$ *y retornar el conjunto de instancias seleccionadas*/*
38. $S \subset T, S = x_i \in T: \sum_{m=1}^{M} S_m > \emptyset_o$

Como se observa, el algoritmo recibe un conjunto de entrenamiento y retorna un subconjunto de instancias de este conjunto (dataset reducido).

3.3 Experimentación Y Resultados

Esta sección presenta las evaluaciones y el análisis de los resultados de los algoritmos de reducción de la dimensión explicados anteriormente, aplicados sobre los conjuntos de datos descritos en el apartado 2.1.1. En primera instancia, se expone brevemente las métricas de evaluación utilizadas para valorar el resultado del aprendizaje de los clasificadores. Posteriormente, se presentan los resultados obtenidos con dichos métodos sobre los conjuntos de datos reducidos y se contrastan estos resultados.

3.5.1 Métricas De Evaluación Del Rendimiento

Para la evaluación de los métodos de selección de atributos e instancias se utilizaron 5 clasificadores: Maquina de Vector de Soporte (MVS), Redes Bayesianas (RB), Vecinos más cercanos (K-NN), el árbol de decisión C.4.5 y las Redes Neuronales Artificiales (RNA), para los cuales fue calculada la precisión sobre los cuales se evaluarán tanto los dataset originales como los reducidos (resultado de las técnicas en cuestión).

La precisión está definido como la proporción de instancias verdaderas del conjunto de instancias predichas como positivas (capacidad del clasificador para evitar el ruido) [85] y se calcula con la Ecuación 9:

$$P_p = \frac{VP}{VP + FP} \qquad \text{Ecuación 9}$$

Para explicar los términos de esta expresión se parte de que se tiene la variable objetivo, la cual contiene las clases $C_1, C_2, \ldots, C_n$, de este modo por cada clase se deben aplicar los siguientes conceptos:

- **Falsos Positivos (FP)**: número instancias incorrectamente clasificadas en la clase $C_{x.}$
- **Falsos Negativos (FN)**: instancias de la clase C_x que fueron incorrectamente clasificadas en otra clase.
- **Verdaderos Positivos (VP)**: número instancias correctamente clasificadas en la clase C_x.
- **Verdaderos Negativos (VN)**: todas las instancias restantes correctamente clasificadas diferente a la clase C_x.

Para medir la precisión se utilizó el método de validación cruzada con k=10. Adicionalmente, para medir la cantidad de información se utiliza el porcentaje de varianza explicada o varianza de los datos que se definió matemáticamente en la sección 2.2.4.

3.5.2 Resultados Experimentales

Los métodos de SAC y RI-E fueron evaluados de manera individual y en conjunto, haciendo uso de clasificadores como MVS, RB, K-NN, C.4.5 y RNA, aplicados a los dataset originales como los reducidos. El proceso experimental se puede observar con más detalle en la Figura 13.

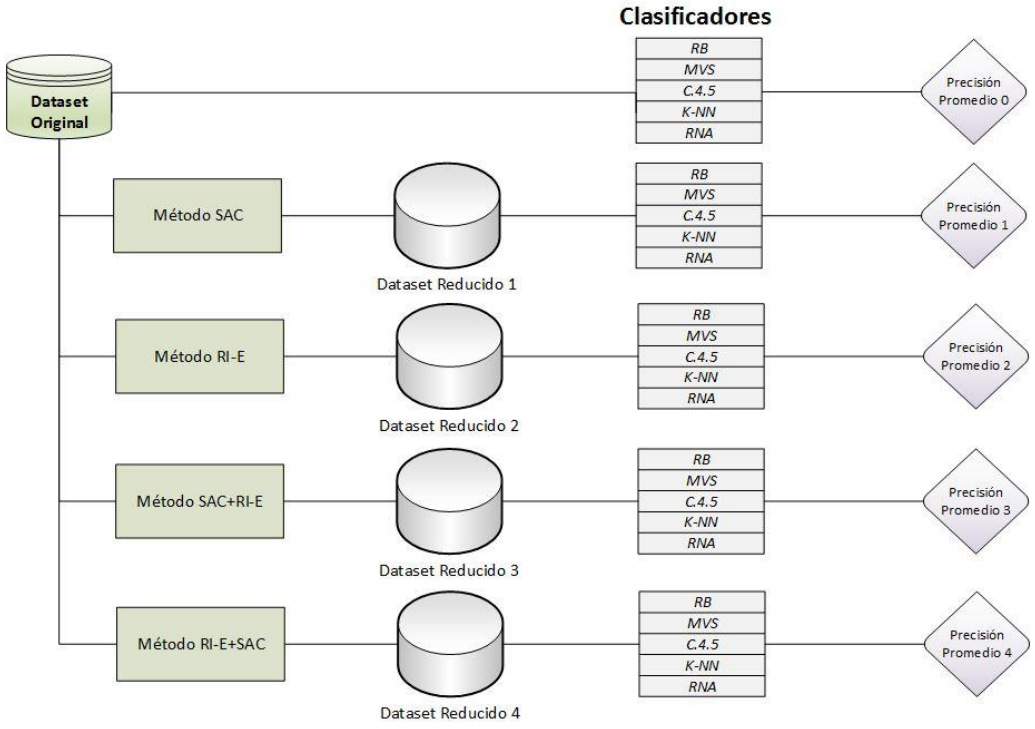


Figura 13. Proceso experimental reducción de la dimensión

Como se indica en la figura anterior, el esquema general de pruebas consta de cinco procedimientos diferentes, con los cuales se verifica cual es el mejor método de reducción

de la dimensión de los conjuntos de datos en cuestión. A continuación, se describe cada uno de estos procedimientos.

Procedimiento 1: consiste en realizar la clasificación de los dataset originales sin ninguna clase de pre-procesamiento y de este modo poder verificar si los dataset pre-procesados pueden mejorar el proceso de clasificación en comparación a una clasificación sin una etapa de pre-procesamiento.

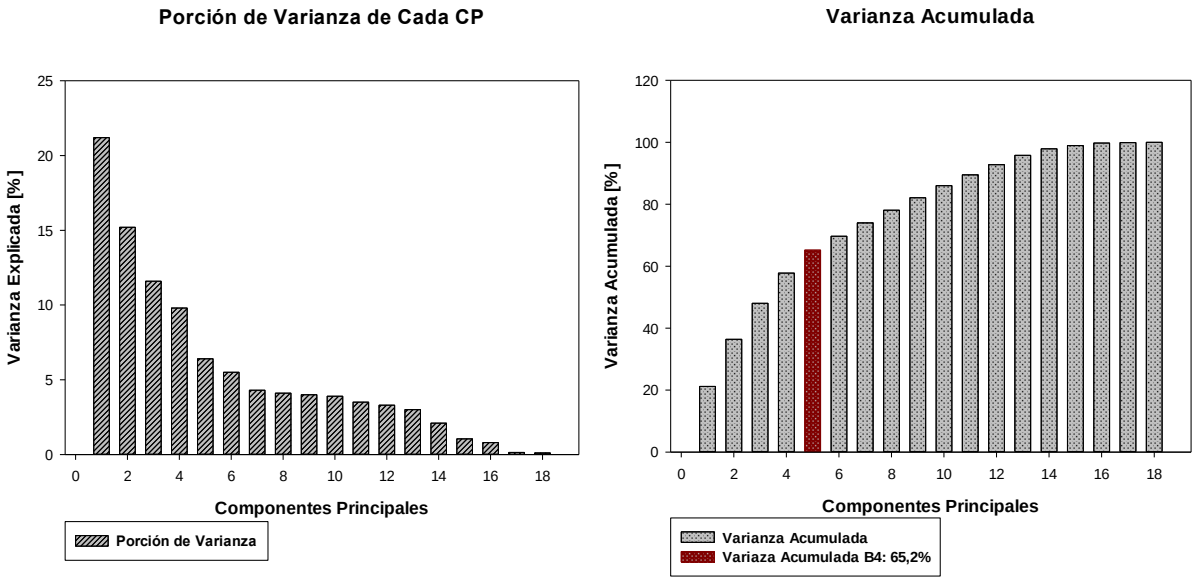
Procedimiento 2: consiste en seleccionar los atributos o características más importantes de los dataset originales. El nuevo conjunto de datos que contiene dichas características se le realiza un proceso de clasificación. Como resultado, podemos entender si realizar el proceso de selección de características puede resultar en un mejor rendimiento de clasificación.

Procedimiento 3: en este paso se selecciona las instancias o ejemplos más relevantes del conjunto de datos original. El nuevo subconjunto de datos se le realiza un proceso de clasificación. Los resultados pueden ser utilizados para comparar si el clasificador seguido por el proceso de selección de instancias puede proporcionar un mejor rendimiento que el clasificador seguido de la etapa de selección de características y sin la etapa de pre-procesamiento.

Procedimiento 4 y 5: aquí se realiza el proceso de reducción de atributos-instancias e instancias-atributos de manera conjunta. El subconjunto de datos generado se le realiza un proceso de clasificación. Los resultados obtenidos pueden ser utilizados para comparar si el clasificador seguido por estas secuencias de mecanismos pueden proporcionar un mejor rendimiento que el clasificador seguido de la etapa de reducción de características, la etapa de reducción de instancias y sin la etapa de pre-procesamiento.

RESULTADOS EXPERIMENTALES DE REDUCCIÓN DE ATRIBUTOS

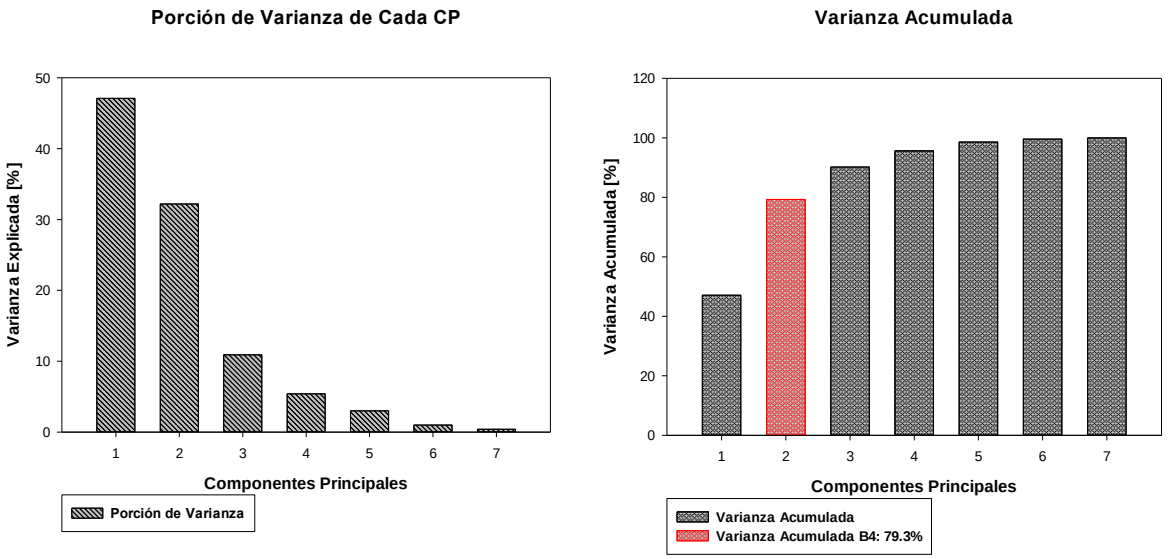
Los resultados del proceso de extracción de características ACP sobre el dataset del Rio Las Piedras se exponen en la Figura 14. Como se observa en la Figura 13(a), el proceso de ACP genera 18 componentes principales (CP), donde la primera CP representa el 21.2% de la información total, la segundo CP explica el 15.2% de la varianza original, la tercera explica el 11.6%, y así sucesivamente hasta la CP 18, la cual representa el 0.1% de la información total. Como se expuso anteriormente, los datos de varianza explicada son importantes para conocer el número de CP que van a ser utilizados en el análisis. Si se toma como criterio de selección de componentes la metodología B4, se considera que el número de CP óptimo (hay que recordar que se trata de reducir la cantidad de CP en la medida de lo posible) es de 5 componentes, cuya varianza acumulada es del 65.2%. Esto significa que las cinco (5) primeras CP representan el 65.2% de la información total del dataset. Tal y como se indica en la Figura 13(b), los otros componentes explican porcentajes significativamente más bajos comparativamente a los cinco primeros componentes.



(a) Porción de varianza explicada por CP (b) Varianza acumulada

Figura 14. Resultados PCA sobre el dataset Rio Las Piedras.

De manera análoga, se ejecuta el proceso de ACP sobre el dataset Estuario California, cuyos resultados se exponen en la Figura 15. Si se toma como referencia los métodos de retención de CP B4, se considera que el número óptimo de componentes principales es de 2. Estos dos componentes representan el 79.3% de la información total, mientras los otros componentes explican porcentajes significativamente bajos comparados a los dos primeros componentes. Así pues, es razonable quedarse con los 2 primeros CP, debido a que añadiendo uno más se gana aproximadamente un 11% de información, mientras que quitando un CP se pierde un 32.2% de información.



(a) Porción de varianza explicada por CP (b) Varianza acumulada

Figura 15. Resultados PCA sobre el dataset Estuario California

Una vez aplicado el proceso de ACP sobre los dataset, se aplica el mecanismo propuesto SAC, que consiste como primera medida, en construir dataset con $\{1\}, \{1+2\}, \{1+2+3\}, \ldots, \{1+2+3+\cdots 18\}$ componentes principales. Posteriormente, se entrena los clasificadores mencionados anteriormente con cada uno de estos dataset. Finalmente, se obtiene como resultado el subconjunto de datos que mejor precisión obtuvo y el número de CP asociado a este dataset. Estos resultados son expuestos en la Tabla 6.

Dataset	No. Atributos	Representación de Información	Precisión de los clasificadores					Precisión Promedio
			RB	SVM	C.4.5	K-NN	RNA	
Rio Las Piedras								
No pre-procesado	18	100%	80.2%	92.8%	77.1%	73.2%	96.1%	83.8%
SAC	**5**	65.2%	91%	91.4%	91.4%	86.9%	89%	89.9%
Estuario California								
No pre-procesado	7	100%	98.7%	99.8%	99.8%	99.8%	99.8%	99.5%
SAC	**3**	90.3%	98.9%	99.3%	99.6%	99.2%	99.2%	99.2%

Tabla 6. Resultados obtenidos por el mecanismo de reducción de atributos SAC

Como se observa para el caso del dataset del Rio las Piedras, el mecanismo SAC seleccionó el dataset conformado por las primeras 5 CP como el mejor conjunto de datos, debido a que este último obtuvo la mejor precisión promedio (89.9%) entre todos los dataset conformados. La reducción de 13 componentes obtenida por el método SAC, llevó a mejorar el rendimiento de clasificación en aproximadamente un 6%.

En términos prácticos, el problema inicialmente representado en un hiperespacio de 18 dimensiones ha sido reducido a un hiperespacio más reducido (5 dimensiones), rescatando el 65.2% de la varianza original. Este resultado implica una mayor y fácil interpretabilidad de los datos, proceso que, como se mencionó anteriormente, se realiza analizando la relación entra las CP y los atributos iniciales.

Siguiendo con la misma línea, se procedió de manera análoga con el dataset Estuario California, cuyos resultados se muestran en la Tabla 6. Como se observa, el método SAC determinó para este caso que el mejor representante de los datos es el dataset compuesto por 3 CP, representantes del 90.3% de la información total, ya que al momento de entrenar a los clasificadores con este dataset se obtuvieron los mejores resultados de rendimiento (aproximadamente del 100%). Esto indica que el método SAC logró reducir en 4 el número de componentes sin degradar de manera considerable el rendimiento de clasificación.

A partir de estos resultados, se puede asumir que el método SAC puede realizar una selección bastante contundente de los datos de entrenamiento sin deteriorar las capacidades de los clasificadores.

Ahora, para contrastar el método SAC con el método de selección B4, en la Tabla 7 se plasman los resultados obtenidos.

Dataset	Método SAC			Método de selección de CP B4		
	No. CP	**Representación de la información**	**Precisión Promedio**	**No. CP**	**Representación de la información**	**Precisión Promedio**
Rio Las Piedras	**5**	**65.2%**	**90.1%**	**5**	**65.2%**	**90.1%**
Estuario California	3	90.3%	99.2%	2	79.3%	93.1%

Tabla 7. Contraste del método de reducción de atributos SAC y el método B4

El mecanismo para la selección automática de componentes principales (SAC) aplicada sobre el dataset del Rio Las Piedras logró reducir el espacio del mismo a cinco componentes. Estos resultados correspondieron con los obtenidos con el método de selección B4, donde sugiere que el nuevo dataset esté constituido con 5 componentes y cuya cantidad de información se encuentre dentro del intervalo de confianza de varianza acumulada 60% - 95%. Además, el mecanismo incrementó tanto el rendimiento del proceso de clasificación como la interpretabilidad del modelo (menor número de atributos).

Por otro lado, al aplicar la técnica SAC sobre el dataset Estuario California, se sugiere que el nuevo dataset (reducido) este conformado por los primeros tres CP, representantes del 90.3% de la información total, a diferencia del método B4 que sugiere utilizar 2 CP que explican el 79.3% de toda la información. Por su parte, utilizar 3 CP implica una ganancia de un 6.1% de rendimiento de clasificación con respecto a utilizar 2 CP.

RESULTADOS EXPERIMENTALES REDUCCIÓN DE INSTANCIAS

En esta sección se exponen los resultados obtenidos por el método para la reducción de instancias propuesto (RI-E) y su antecesor (SIB), los cuales fueron presentados en la sección anterior. Para contrastar los resultados de estos dos mecanismos se hizo uso de 5 dataset del repositorio Universidad de California Irvine (UCI **http://archive.ics.uci.edu/ml/index.html**) utilizados en la experimentación de método SIB y publicados en [60], los cuales fueron procesados por RI-E. En la Tabla 8 se presentan las medidas de precisión y porcentaje de instancias reducidas tanto del método de reducción de instancias original (SIB), como del método de reducción de instancias propuesto.

Dataset	**No. Atributos**	**No. Instancias**	**Precisión SIB**	**Reducción instancias SIB**	**Precisión RI-E**	**Reducción instancias RI-E**
Cáncer	9	699	93.4%	15.3%	62.8%	15.7%
Ecoli	7	336	76%	44.5%	81.5%	35.4%
Iris	4	150	79.2%	37.5%	75%	38.6%
Mammographic	5	961	71.9%	47.5%	53.2%	42.8%
Vehicle	18	846	68.9%	58.3%	22.5%	23.5%

Tabla 8. Resultados del método de reducción de instancias original y el método propuesto

De esta forma, al observar la tabla anterior, el porcentaje de reducción de instancias de los dos algoritmos siguen aproximadamente el mismo comportamiento, con excepción de los resultados obtenidos sobre el dataset Vehicle, donde el porcentaje de reducción de instancias tiene una diferencia bastante marcada (34.8%). Además, se visualiza que el método RI-E no mejora su rendimiento de clasificación, excepto cuando reduce el 35.4% de instancias del dataset Ecoli, donde obtiene su mejor resultado (81.5%).

Adicionalmente, se puede observar que el valor de precisión del método propuesto (RI-E) se encuentra entre 50% y 82% (exceptuando los resultados obtenidos por el dataset Vehicle), a diferencia del método SIB, el cual obtiene una precisión que oscila entre el 68.9% y el 93.4%. Lo anterior hace posible que los dos métodos de reducción de instancias seleccionen información importante del mismo. Estos valores indican que el funcionamiento del algoritmo tiene un buen desempeño de clasificación como de reducción.

Ahora bien, aplicando el método RI-E sobre los dataset de calidad del agua, presentados en la sección 2.1.1, se obtuvieron los resultados presentados en la Tabla 9.

Dataset	No. Atributos	No. Instancias	No. Clases	Precisión de los clasificadores					Precisión Promedio
				RB	SVM	C.4.5	K-NN	RNA	
Rio Las Piedras									
No pre-procesado	18	645	3	80.2%	92.8%	77.1%	73.2%	96.1%	83.8%
RI-E	18	336	3	50.6%	57.1%	62.8%	55.2%	55.2%	56.1%
Estuario California									
No pre-procesado	7	2505	4	96.6%	97%	97%	97%	99.8%	99.5%
RI-E	7	1262	4	80.9%	97.2%	97.6%	97%	97.3%	94%

Tabla 9. Resultados obtenidos por el método de reducción de instancias RI-E sobre los dataset de calidad de agua

Como se nota, la técnica RI-E logro reducir en 309 y 1243 el número de instancias del dataset Rio las Piedras y el dataset Estuario California respectivamente. Sin embargo, para el primer dataset el método en cuestión redujo un 27.7% la precisión de los clasificadores, lo que indica que se eliminó instancias importantes de este dataset, perdiendo de esta forma información importante. Por su parte, ocurre el caso contrario con el dataset Estuario California, donde a pesar de reducir aproximadamente la misma proporción de instancias, sólo redujo un 5.5% la precisión de los clasificadores, lo que indica que el método RI-E fue más efectivo al momento de eliminar instancias irrelevantes del dataset Estuario California que en el dataset del Rio las Piedras.

RESULTADO EXPERIMENTAL REDUCCIÓN DE ATRIBUTOS E INSTANCIAS

Bajo el mismo contexto, en la Tabla 10 se resumen los datos conseguidos producto de entrenar los clasificadores con los nuevos dataset adquiridos después de aplicar de manera conjunta las técnicas de reducción presentadas anteriormente. Como se observa, el método de reducción de atributos/instancias se denotan como SAC+RI-E, mientras que el método de reducción de instancias/atributos se representa como RI-E+SAC.

Dataset	No. Atributos	No. Instancias	Representación de la información	Precisión de los clasificadores					Precisión Promedio
				RB	SVM	C.4.5	K-NN	RNA	
Rio Las Piedras									
SAC+RI-E	5	493	65.2%	72.7%	73.8%	76%	72.7%	74.2%	73.9%
RI-E+SAC	3	225	41.5%	44.7%	41%	60.9%	65.7%	59.1%	54.3%
Estuario California									
SAC+RI-E	3	1630	90.3%	87.2%	81.8%	99.6%	97.2%	96.7%	92.5%
RI-E+SAC	6	1947	99%	84.4%	85.8%	86.5%	86.7%	87.2%	86.1%

Tabla 10. Precisión de clasificación de los cuatro procedimientos.

Es interesante notar que para el caso del dataset Rio Las Piedras, la secuencia de métodos RI-E+SAC logro reducir en mayor medida tanto el número de atributos como el número de instancias que la técnica SAC+RI-E. La primera técnica filtro 15 componentes y 420 instancias del dataset, mientras que la segunda logro reducir 13 componentes y 152 instancias. Sin embargo, al entrenar los clasificadores con el dataset procesado con esta último secuencia (SAC+RI-E) se obtiene una precisión promedio más alta (73.9%) que al entrenarlos con el dataset obtenido al aplicar el método RI-E+SAC (54.3%). Además, puede notarse que el dataset reducido por RI-E+SAC está representado por 3 características que explican tan solo el 41.5% de la varianza total, la cual se encuentra por fuera del rango de confianza sugerido por el criterio B4. Esto implica que este método no resulta apropiado para reducir la dimensión del dataset Rio Las Piedras.

Por otro lado, al procesar el dataset Estuario California con los métodos SAC+RI-E y RI-E+SAC, se obtiene que el primero obtuvo los mejores resultados tanto en reducción como en precisión de clasificación, ya que logró reducir 4 componentes y 875 instancias y consiguió una precisión promedio del 92.5%, mientras que el segundo solo redujo 1 componente y 558 instancias, alcanzando una precisión promedio del 86.1%.

Como resultado, se puede observar que el mecanismo SAC+RI-E es una solución adecuada para la reducción de la dimensión de los dataset de la calidad del agua, el cual permite a los clasificadores proporcionar una precisión de clasificación similar a la obtenida al clasificar los dataset originales (sin pre-procesamiento). De lo anterior, se puede decir que el método RI-E+SAC no resulta apropiado para reducir la dimensión de los dataset.

Ahora, para contrastar los resultados obtenidos por cada una de las técnicas de reducción expuestas anteriormente, en la Figura 16 se condensan los resultados de cada una de ellas.

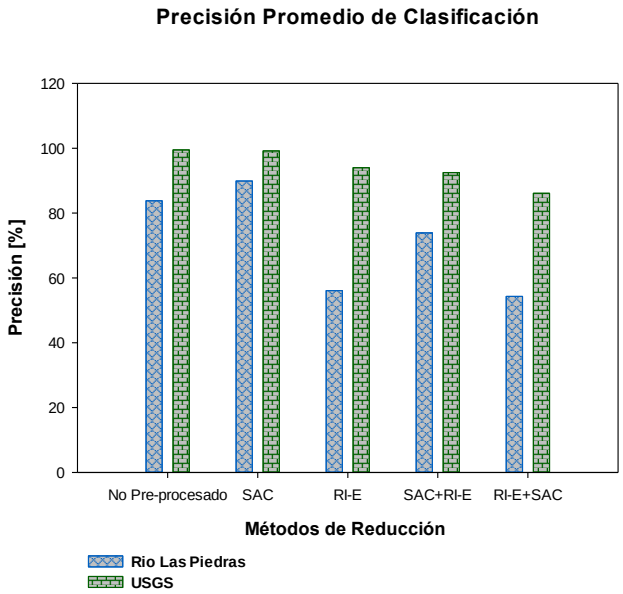


Figura 16. Precisión promedio de clasificación de los métodos de reducción de la dimensión

Al comparar los dataset reducidos con el dataset Rio Las piedras, se muestra que las técnicas de reducción de la dimensión filtran gran cantidad de atributos e instancias sin degradar de manera considerable el rendimiento de los clasificadores, excepto cuando se realiza la reducción con los métodos individuales RI-E y RI-E+SAC. Caso contrario ocurre al realizar la comparación con el dataset Estuario California, donde todas las técnicas de reducción lograron reducir la dimensión del dataset sin degradar de manera considerable la precisión de clasificación.

Es interesante notar que el proceso de SAC de forma individual hace que el rendimiento de los clasificadores sea mucho mejor que el rendimiento obtenido por las demás técnicas de reducción, e incluso el obtenido con el dataset original. Con base a este resultado, es posible asumir que la existencia de características redundantes en el dataset influye en mayor medida en el desempeño de la clasificación.

TIEMPO PROMEDIO DE ENTRENAMIENTO DE LOS CLASIFICADORES

Es importante mencionar que otra forma de evaluar el rendimiento de los algoritmos de reducción es mediante el coste computacional (tiempo de ejecución) que tienen los clasificadores al momento de procesar un dataset. Para este análisis, no se evalúa el tiempo de ejecución de cada clasificador por separado, sino que de manera conjunta se calcula el promedio aritmético del tiempo que tarda cada clasificador en procesar un dataset (esto debido a que esta sección se enfatiza en evaluar el comportamiento de los mecanismos de reducción de la dimensionalidad). Para este caso de estudio, los dataset serían el dataset no pre-procesado u original y los dataset procesados o reducidos del Rio Las Piedras y Estuario California. En la Figura 17 se resume estos resultados.

En primera medida se observa que la técnica SAC, además de reducir el número de características (13) del dataset del Rio Las Piedras, redujo el tiempo de entrenamiento de los clasificadores en comparación al tiempo de entrenamiento de la clasificación del

dataset original en 228ms. Del mismo modo, para el dataset Estuario California, el método SAC redujo el tiempo de clasificación en 134ms, al reducir en 4 el número de características del mismo. Esto indica que la existencia de características redundantes en el dataset influye en el desempeño de los clasificadores.

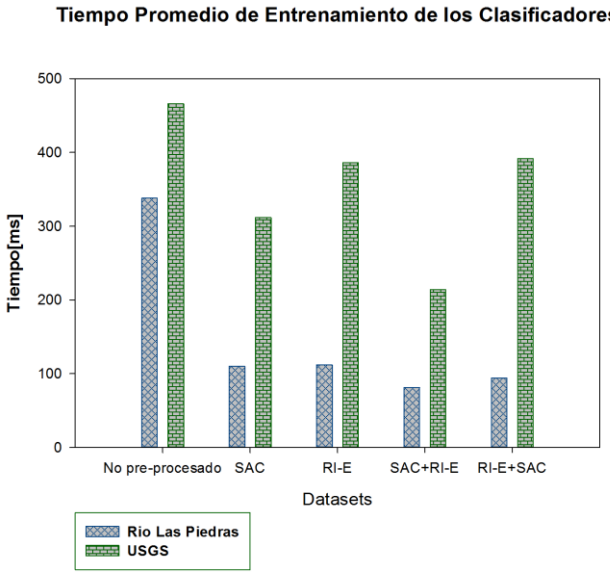


Figura 17. Tiempo promedio de entrenamiento de los clasificadores

Siguiendo con el mismo comportamiento, se observa que la técnica RI-E al reducir en 309 y 1243 el número de instancias del dataset Rio las Piedras y el dataset Estuario California respectivamente, también logró mitigar el tiempo de entrenamiento de los clasificadores en 226 ms y 80 ms de respectivamente. El resultado obtenido da a entender que al minimizar cantidad de instancias redundantes y con ruido, es posible reducir los tiempos de entrenamiento de los clasificadores (coste computacional).

Ahora, para el caso de los métodos de reducción combinados SAC+RI-E y RI-E+SAC, aplicados sobre el dataset Rio Las Piedras, se tiene que en ambos casos se redujo en gran medida el tiempo de entrenamiento de los clasificadores (257ms y 244ms respectivamente). De igual forma, al procesar el dataset Estuario California con estas técnicas se logró reducir el tiempo de clasificación en 252ms y 74ms respectivamente. Adicionalmente, se observa que el enfoque SAC+RI-E permite realizar un proceso de clasificación significativamente más rápido que los métodos individuales e incluso que el método combinado RI-E+SAC. Estos resultados eran de esperarse, debido a que este tipo de mecanismos eliminan tanto características como instancias del dataset, lo que representa una reducción de coste computacional de las tareas de minería de datos (clasificación y/o agrupación).

Como resultado, la estrategia propuesta obtuvo precisiones similares a los dataset originales y además logró disminuir en gran medida el costo computacional, haciendo que esta propuesta sea la más adecuada para reducir la dimensión de los dataset de la calidad del agua.

3.4 Resumen

En este capítulo se describieron de manera detallada las técnicas de reducción de atributos ACP y de reducción de instancias Boosting seleccionadas en la sección 2.2.2 y 2.2.3. De igual manera, se expuso los aportes realizados por este trabajo de grado sobre cada una de dichas técnicas de manera individual (SAC y RI-E). Adicionalmente, se exhibe las pruebas ejecutadas a los mecanismos propuestos, tanto de forma individual como conjunta, haciendo uso de la precisión como métrica de evaluación de los clasificadores y de la varianza para evaluar la calidad de la información del dataset.

Como resultado, se obtuvo que el mecanismo SAC+RI-E es una solución adecuada para la reducción de la dimensión de los dataset de la calidad del agua, la cual permite a los clasificadores proporcionar una precisión de clasificación similar a la obtenida al clasificar los dataset originales (sin pre-procesamiento).

Capítulo 4

4 Desbalanceo de clases

En este capítulo se el aporte llevado a cabo para mejorar el proceso de balanceo de clases. Por último, se exhibe la experimentación realizada y los resultados obtenidos.

4.1 Selección Automática Del Porcentaje Óptimo De Datos Sintéticos (SADS)

El algoritmo SMOTE recibe como parámetros: el número de instancias de la clase minoritaria T, el porcentaje de instancias a sobre-muestrear P y el número de vecinos más cercanos K, para finalmente, retornar el dataset con la clase positiva balanceada al grado del valor de P.

La estrategia propuesta consiste en asegurar que el nivel de desbalance IR del dataset original (DO) cumpla con un umbral de desbalanceo de clases mínimo (línea 4 del Algoritmo 3). Si el dataset cumple con el umbral de desbalanceo, se procede a aplicar sobre la clase minoritaria la definición de SMOTE (línea 7 del Algoritmo 3). Para este caso se define el umbral de desbalanceo en $IR \geq 3.9$ y se utilizarán porcentajes de instancias a sobre-muestrear (valor de P) igual a 50, 100, 150 y 200%, evitando de este modo que la proporción de instancias de C+ supere a las de C-. Además, basados en [81], donde exponen que para alcanzar un sobre-muestreo del 200% tan solo se requieren de dos vecinos más cercanos, suficientes para el caso de estudio, se toma K=2 como el número de vecinos más cercanos para SMOTE.

Una vez aplicado el proceso de sobre-muestreo para un *P* determinado, se evalúa el comportamiento del subconjunto generado haciendo uso de los algoritmos de clasificación: MVS, RB, K-NN, C.4.5 y RNA, y utilizando las métricas de evaluación del rendimiento de datasets desbalanceados: medida-F y ROC (Receiver Operating Characteristic - Característica Operativa del Receptor), las cuales se explican con mayor detalle en la sección de resultados.

Para finalizar, se selecciona el subconjunto de datos que obtenga los mejores valores de las métricas mencionadas anteriormente (línea 16 y 17 del Algoritmo 3), y se selecciona como el mejor porcentaje de datos sintéticos, aquel porcentaje asociado a este dataset. En él Algoritmo 3 se describe el proceso planteado.

Algoritmo 3. Selección automática datos sintéticos(T,K)

***Datos de entrada**:*

- *Conjunto de entrenamiento: T*
- *Número de vecinos más cercanos: K*

***Datos de salida**:*

- *Dataset balanceado: DB_M*

1. *Calcular número de instancias de la clases mayoritaria* (C_M) *y minoritaria* (C_m)
2. *Calcular umbral de desbalanceo de clases*
   $IR = \frac{C_M}{C_m}$
3. /**Si se cumple con el umbral de desbalanceo de clases se aplica la definición de SMOTE**/
4. $\boldsymbol{if}(IR \geq 3.9)$
5. $\boldsymbol{for}\ i \rightarrow 50\ \boldsymbol{to}\ 200$
6. /*Se *genera un dataset balanceado con i instancias sintéticas* DB_i */
7. *Aplicar algoritmo SMOTE*: $\boldsymbol{Smote}(C_m, i, K)$
8. /* *Se entrena los clasificadores:* C_{RN}, C_{MVS}, C_{K-NN}, $C_{C.4.5}$ *y* C_{RNA}, *se calcula las métricas F y ROC*/
9. $F_{RN_i} = C_{RN}(DB_i);\quad ROC_{RN_i} = C_{RN}(DB_i)$
10. $F_{MVS_i} = C_{MVS}(DB_i);\quad ROC_{MVS_i} = C_{MVS}(DB_i)$
11. $F_{K-NN_i} = C_{K-NN}(DB_i);\quad ROC_{K-NN_i} = C_{K-NN}(DB_i)$
12. $F_{C.4.5_i} = C_{C.4.5}(DB_i);\quad ROC_{C.4.5_i} = C_{C.4.5}(DB_i)$
13. $F_{RNA_i} = C_{RNA}(DB_i);\quad ROC_{RNA_i} = C_{RNA}(DB_i)$
14. $i \mathrel{+}= 50$
15. $\boldsymbol{end\ for}$
16. *Seleccionar la mayor métricas* F_M
17. *Seleccionar la mayor métrica* ROC_M
18. *Selección el mejor dataset* $DB_M(F_M, ROC_M)$
19. $\boldsymbol{return}\ (DB_M)$
20. $\boldsymbol{else}$
21. /**Retorna el dataset de entrada**/
22. $\boldsymbol{return}\ (T)$
23. $\boldsymbol{end\ if}$

4.2 Experimentación Y Resultados

Aquí, es presentado el análisis de los resultados obtenidos después de la aplicación del método de desbalanceo de clases sobre los dataset expuestos en la sección 2.1.1. Como primera medida, se expone brevemente la definición de las principales métricas utilizadas en la evaluación del proceso de clasificación del dataset desbalanceado, seguido de la exhibición de los resultados obtenidos con la técnica planteada y se contrastan con los resultados conseguidos con respecto al datase original.

4.3.1 Métricas De Evaluación Del Rendimiento

Las medidas de evaluación más comunes para evaluar el rendimiento de los clasificadores se basan en la matriz de confusión. A partir de esta matriz se calcula la tasa de error, la exactitud (el porcentaje de clasificación correcta), la sensibilidad, la especificidad y la precisión.

Esta matriz permite observar, mediante una tabla de contingencia, la distribución de los errores cometidos por un clasificador. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa la clasificación real [86].

En la Tabla 11, es presentado un ejemplo de la matriz de confusión para la detección de la calidad del agua. De esta manera, 230 instancias indican que la calidad del agua es excelente, 6 instancias con buena calidad y 10 instancias con calidad del agua regular. Para las instancias con buena calidad del agua, 201 instancias pertenecen a esta clase, 15 tienen calidad regular y 3 excelente calidad. Adicionalmente, para las instancias con calidad del agua regular, el algoritmo detectó que 8 instancias tienen buena calidad del agua y 5 excelente calidad.

A partir de la matriz de confusión, se puede decir que el algoritmo tiene un inconveniente al clasificar instancias con buena calidad del agua en comparación con las instancias pertenecientes a la clase regular y excelente calidad.

		Clase Estimada		
		Regular	**Buena**	**Excelente**
Clase Real	**Regular**	201	8	5
	Buena	15	183	3
	Excelente	10	6	214

Tabla 11. Matriz de confusión – calidad del agua

En este sentido, a partir de la matriz de confusión se pueden extraer conceptos, tales como los son los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Estos conceptos se presentan a continuación:

Se tiene la variable objetivo, la cual contiene las clases $C_1, C_2, \dots, C_n$. De este modo, por cada clase se deben aplicar los siguientes conceptos:

- **Falsos Positivos (FP)**: número instancias incorrectamente clasificadas en la clase $C_{x.}$
- **Falsos Negativos (FN)**: instancias de la clase C_x que fueron incorrectamente clasificadas en otra clase.
- **Verdaderos Positivos (VP)**: número instancias correctamente clasificadas en la clase C_x.
- **Verdaderos Negativos (VN)**: todas las instancias restantes correctamente clasificadas diferente a la clase C_x.

Una vez obtenido el número de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos por cada clase, se calculan las métricas empleadas en la medida del rendimiento de los sistemas de búsqueda, reconocimiento de información y reconocimiento de patrones que abordan el problema del desbalanceo de clases [67, 80, 87, 88]. Estas métricas se definen de manera formal a continuación:

PRECISIÓN

Está definida como la proporción de instancias verdaderas del conjunto de instancias predichas como positivas (capacidad del clasificador para evitar el ruido), y se calcula con la Ecuación 9:

EXHAUSTIVIDAD

Esta medida conocida en Ingles como recall, es la encargada de calcular la proporción de verdaderos positivos predichos entre todos los positivos (instancias relevantes clasificadas). La Ecuación 10 se expresa como:

$$e = \frac{VP}{VP + FN} \qquad \text{Ecuación 10}$$

MEDIDA-F

Es un balance de la precisión y la exhaustividad. En otras palabras, es considerada una media ponderada de la precisión y exhaustividad, donde la puntuación alcanza su mejor valor en 1 y el peor en 0. En la Ecuación (11) es presentada:

$$F = \frac{2 * P * e}{P + e} \qquad \text{Ecuación 11}$$

ESPACIO ROC (RECEIVER OPERATING CHARACTERISTIC)

El espacio ROC se interpreta a través de gráficas bidimensionales, en las que la tasa de verdaderos positivos (TVP) o sensibilidad está definida en el eje y, mientras la tasa de falsos positivos (TFP) o (especificidad -1) en el eje x. A continuación, son presentadas las ecuaciones para calcular los valores mencionados anteriormente:

- **Tasa de Verdaderos Positivos (TVP):** es conocida como sensibilidad o exhaustividad (métrica presentada anteriormente), y su cálculo se realiza mediante la Ecuación 10.

- **Tasa de Falsos Positivos (TFP):** se calcula a través de la Ecuación 12:

$$TFP = \frac{FP}{VP + FP} \qquad \text{Ecuación 12}$$

- **Especificidad:** es la proporción de verdaderos negativos, y se obtiene de la siguiente forma:

$$Especificidad = \frac{VN}{VN + FN} \qquad \text{Ecuación 13}$$

Este tipo de análisis es utilizado en problemas de clasificación (la salida representa solo la etiqueta de una clase), donde cada clasificador produce un par (TFP, TVP), correspondiente a un único punto en el espacio ROC [89]. En la Figura 18 es presentado un ejemplo de un espacio ROC con 5 puntos (TFP, TVP) identificados con las letras A hasta E:

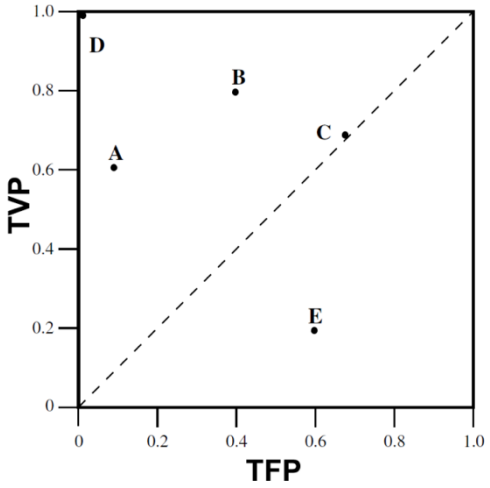


Figura 18. Espacio ROC para 5 puntos. Tomado de [90]

De esta manera, en la Figura 18 vale la pena destacar los puntos más relevantes sobre el espacio ROC [89]:

- El punto $(0,0)$, se caracteriza por no emitir una clasificación positiva, así un clasificador no comete errores de falsos positivos, pero tampoco reconoce verdaderos positivos.
- El punto $(1,1)$, siempre emitirá clasificaciones positivas.
- El punto $(0,1)$, representa la *clasificación perfecta*, como se puede observar en la Figura 18, el cual se encuentra identificado con la letra D.

Adicionalmente, es importante mencionar que los puntos situados en el lado izquierdo de un espacio ROC cerca del eje x son considerados conservadores ya que clasifican positivamente sólo con una fuerte evidencia, por lo que tienen un bajo número de falsos positivos y verdaderos positivos. En el mismo sentido, los puntos que estén situados en la parte superior del lado derecho son considerados liberales, ya que clasifican positivamente con una débil evidencia, por lo que clasifican la mayoría de positivos correctamente, obteniendo altos números de falsos positivos. Como resultado, en la Figura 18, A es considerado más conservador que B. Por otra parte, los puntos que estén ubicados sobre la diagonal $y = x$, representan una clasificación aleatoria. Esto indica que si un clasificador predice positivo la mitad de las veces, se puede esperar que la otra mitad prediga negativos produciendo el punto $(0.5, 0.5)$ en el espacio ROC [90].

CURVAS ROC (RECEIVER OPERATING CHARACTERISTIC)

Según los conceptos expuestos anteriormente, un clasificador produce un único punto en el espacio ROC. Sin embargo, existen clasificadores que producen la probabilidad de puntuación (score) de pertenencia de una clase u otra, según umbrales establecidos para cada instancia de la clase. Siguiendo la notación establecida por [91], un clasificador que produce una puntuación Y se definirá como:

$$Positivo\ si\ Y \geq c; \qquad Negativo\ si\ Y < c$$

En esta expresión, c esta definido como el umbral de decisión que para este caso es un clasificador binario, en donde cada punto de corte origina un punto diferente en el espacio ROC, el cual representa una coordenada (TFP, TVP). Ahora bien, las tasas de verdaderos y falsos positivos generadas por c, pueden presentarse como:

$$TVP(c) = P[Y \geq c \mid D = 1]; \quad TFP(c) = P[Y < c \mid D = 0]$$

De esta manera, la curva ROC es el conjunto de posibles tasas de verdaderos y falsos positivos obtenidos por la utilización de diferentes puntos de corte, y se define de la siguiente manera:

$$ROC(.) = \{(TVP(c), TFP(c)), c \in (-\infty, \infty)\}$$

Ahora bien, la curva ROC estudia el comportamiento de una función de clasificación, donde el área bajo la curva (ABC) cobra gran importancia. Esta área posee un valor comprendido entre 0.5 y 1, donde 1 representa una clasificación perfecta y ABC menores o iguales a 0.5 son consideradas clasificaciones sin capacidad discriminatoria [92].

El objetivo principal de todos los algoritmos de aprendizaje es mejorar la exhaustividad sin sacrificar la precisión. Sin embargo, los objetivos de estas métricas son a menudo contradictorios (son inversamente proporcionales) y atacarlos simultáneamente puede no funcionar bien, especialmente cuando existen clases desbalanceadas. Para afrontar este problema, se necesita que la precisión y la exhaustividad se encuentren compensadas (ya que un sistema con una exhaustividad muy alta pero con baja precisión y viceversa no será adecuado). Para ello se puede utilizar la Medida-F que, como se dijo anteriormente, obtiene un balance entre la precisión y la exhaustividad (mide la mejor precisión y la mejor exhaustividad) y la curva ROC, la cual analiza el comportamiento de la precisión, exhaustividad y la Medida-F [93-95].

De lo anteriormente dicho, en este trabajo de investigación se utiliza como medidas de evaluación del desempeño las métricas ROC y la Medida-F.

4.3.2 Resultados experimentales

Para evaluar el mecanismo propuesto, se utilizó validación cruzada con k=10 para los clasificadores: MVS, RB, K-NN, C.4.5 y RNA. Sobre este enfoque, se evaluará una versión desbalanceada de los dataset Rio las Piedras y Estuario California (Tabla 12) y los dataset balanceados obtenidos del proceso de sobre-muestreo SADS. La clase mayoritaria y minoritaria es denotada como C+ y C- respectivamente.

Dataset originales	No. Clases	No. Instancias	No. Instancias C-	No. Instancias C+	IR
Río Piedras	3	493	307	54	5.6
Estuario California	4	1630	502	127	3.9

Tabla 12. Descripción general de los dataset desbalanceados

El proceso experimental del desbalanceo de clases se puede observar de manera gráfica en la Figura 19.

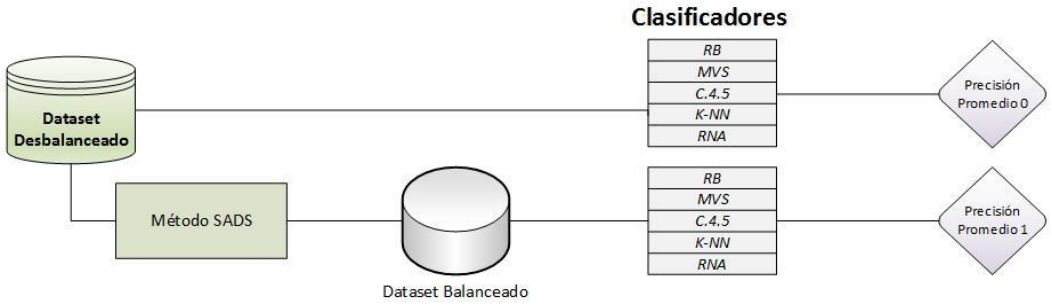


Figura 19. Proceso experimental desbalance de clases

Como se mencionó anteriormente, el rendimiento de cada uno de estos clasificadores se obtiene en términos de las métricas ROC y Medida-F (medidas en %), consideradas para evaluar tanto el dataset original como los dataset sobre-muestreados con diferentes cantidades de muestras sintéticas (50, 100, 150 y 200). Los resultados son expuestos a lo largo de esta sección.

ANÁLISIS DEL RENDIMIENTO DEL DATASET RIO LAS PIEDRAS

En este punto, se realizó el análisis y evaluación del rendimiento de cada uno de clasificadores en cuestión de manera individual, cuyos resultados se dan en términos de las métricas ROC y Medida-F.

A continuación son, presentadas las gráficas asociadas a los resultados de las pruebas definidas en la Figura 19.

a) Comportamiento del rendimiento del clasificador C.4.5

Como se nota en la Figura 20, al entrenar el clasificador C.4.5 con el dataset original Rio las Piedras ($IR = 5.6$) las instancias pertenecientes a la clase negativa tienden a ser bien clasificadas (F=88 y ROC=75.6), mientras que las que pertenecen a la clase positiva tienden a ser mal clasificadas (F=0 y ROC=60.8), esto debido a que los métodos de clasificación tienden a favorecerla clase negativa.

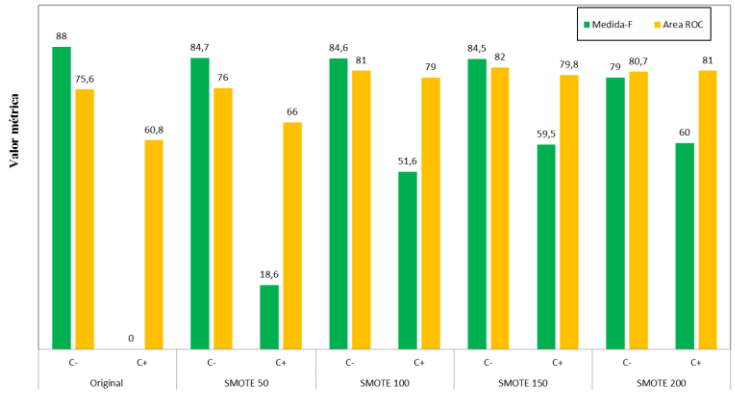
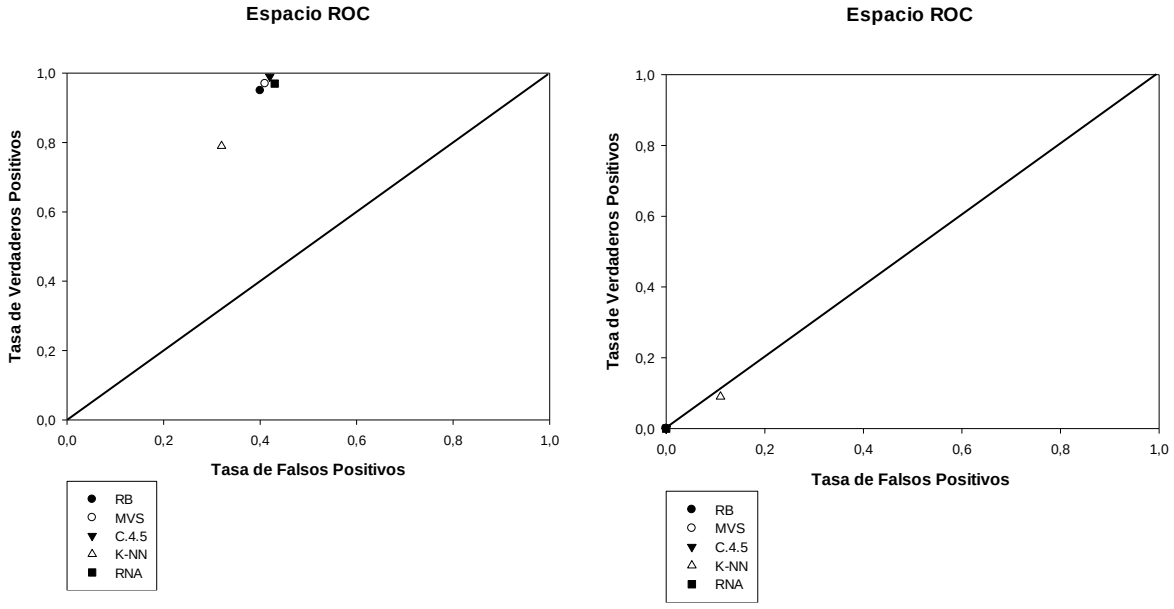


Figura 20. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Rio las Piedras sobre el clasificador C.4.5, aplicando SMOTE

En la Figura 21 se expone el comportamiento del espacio ROC al clasificar las clases C- y C+ del dataset original con los modelos de clasificación mencionados anteriormente.



(a) Clase C- (b) Clase C+

Figura 21. Espacio ROC para los clasificadores: RB, MVS, C.4.5, K-NN y RNA sobre dataset original

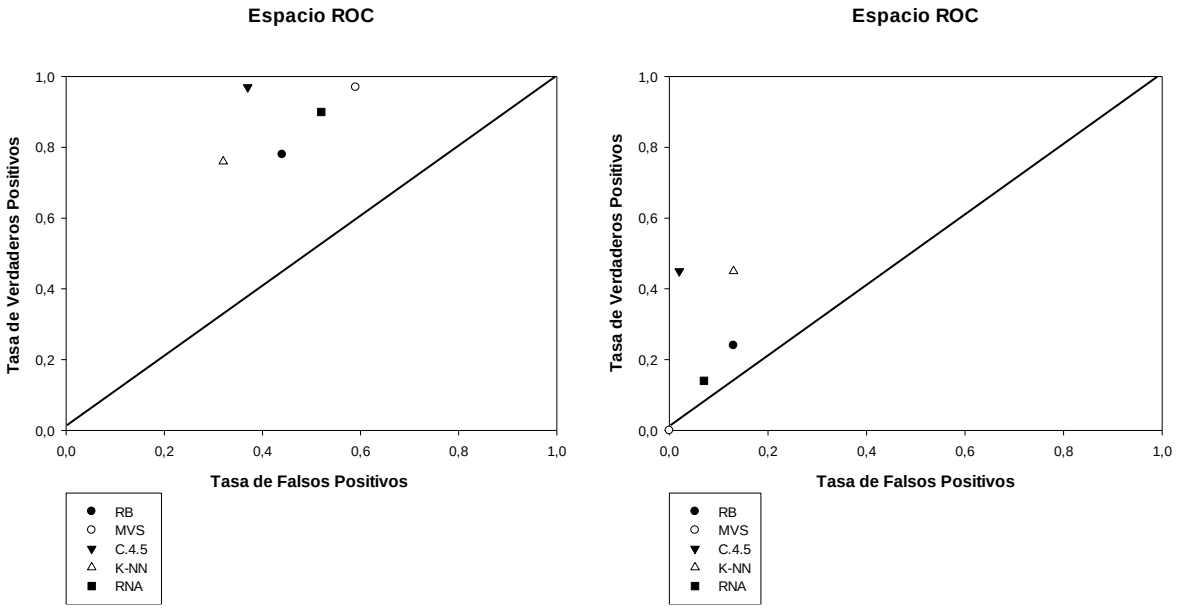
Aquí se observa que para la clase C- las coordenadas (TFP, TVP) obtenidas se encuentran por encima de la línea punteada, en donde el algoritmo supervisado clasifica las instancias de manera correcta (no presenta ninguna aleatoriedad). Caso contrario ocurre cuando se clasifica las instancias de la clase C+, cuyas coordenadas de los métodos RB, MVS, C.4.5 y RNA se encuentran en el origen del plano (0,0) en donde el clasificador no comete errores de falsos positivos, pero tampoco reconoce verdaderos positivos (no hay clasificación de esta clase).

Para el caso del clasificador K-NN, las coordenadas (TFP,TVP) se localizan sobre la diagonal lo que representa una clasificación aleatoria de estas instancias (representa una clasificación mala o inútil).

De la Figura 20 se puede agregar que al incrementar el número de instancias de la clase positiva con datos sintéticos, el rendimiento de la clasificación de esta última mejora, obteniendo los mejores resultados cuando se generan datos sintéticos al 150% y 200%, con F=59.5%, ROC=79.8% y F=60%, ROC=81% respectivamente.

Ahora bien, aunque no hay una diferencia significativa en estos resultados y teniendo en cuenta que lo que se busca con el balaceo de clases, es hacer que el proceso de clasificación pueda predecir correctamente instancias de C+ sin afectar de manera considerable el rendimiento de las detecciones de C-, por esta razón, se puede decir que los mejores resultados de clasificación se obtienen cuando se balancea la C+ con un 150% de instancias sintéticas, debido a que mejora las métricas de evaluación del

rendimiento F y ROC de la clase C+ en un 59.5% y 19% respectivamente, sin afectar la detección de la clase C-, es más, afecta en solo un 3.5% la métrica F e incrementando el valor de la medida ROC en un 6.4%.

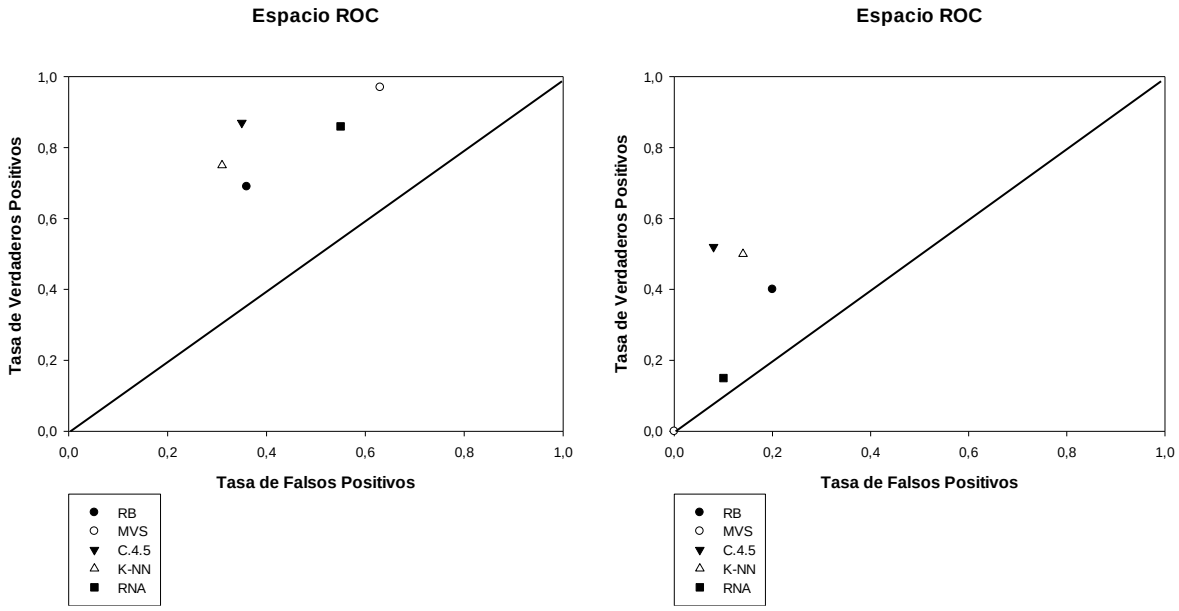


(a) Clase C- (b) Clase C+

Figura 22. Espacio ROC para los clasificadores: RB, MVS, C.4.5, K-NN y RNA sobre dataset sobre-muestreado al 150%

Ahora bien, en la Figura 22, es presentado en detalle las coordenadas (TFP, TVP) para las instancias de la clase C- (Figura 22 (a)) y C+ (Figura 22 (b)) pertenecientes al dataset sobre-muestreado con 150% de instancias sintéticas, generadas por cada algoritmo: RB, MVS, C.4.5, K-NN y RNA. Según estos resultados, se puede afirmar que todos los puntos que se encuentran dibujados en la Figura 22 (a) tienen un alto número de verdaderos positivos, aunque se incrementó levemente el número de falsos positivos (0.2), lo que indica una reducción del rendimiento de los clasificadores. Para el caso de la clase C+ (Figura 22 (b)), se nota que el número de verdaderos positivos para cada algoritmo se incrementó, con excepción del clasificador MVS que no logró clasificar ningún verdadero positivo y ningún falso positivo (0,0), es decir, el método MVS no logro clasificar la clase C+.

De igual manera se puede decir, que para el dataset sobre-muestreado con 200% de instancias sintéticas, los clasificadores en cuestión entregan el espacio ROC que se describe en la Figura 23.



(a) Clase C- (b) Clase C+

Figura 23. Espacio ROC para los clasificadores: RB, MVS, C.4.5, K-NN y RNA sobre dataset sobre-muestreado al 200%

En la figura anterior se muestra el mismo comportamiento anterior, con la diferencia que incrementó en mayor medida el número de falsos positivos. Como se dijo anteriormente, esto perjudica el rendimiento de clasificación.

b) Comportamiento del rendimiento del clasificador K-NN

En el mismo sentido, se entrena el clasificador K-NN con el dataset en cuestión obteniendo los resultados expuestos en la Figura 24.

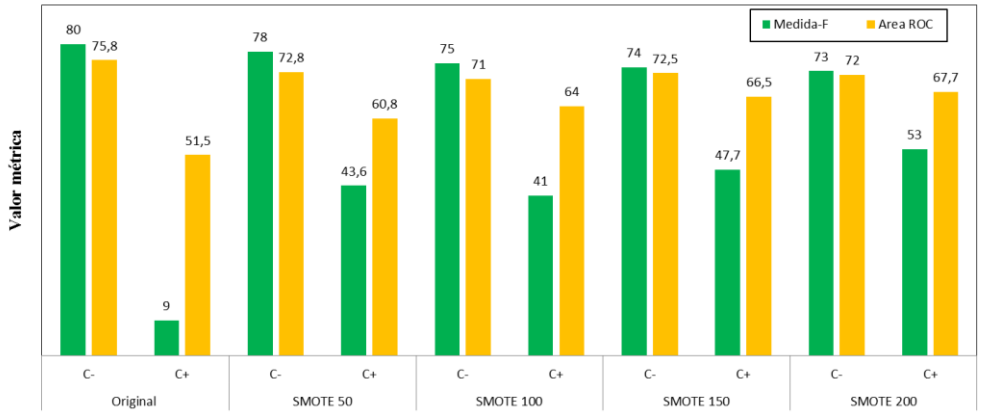


Figura 24. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Rio las Piedras sobre el clasificador K-NN, aplicando SMOTE

En la Figura 24 se observa de manera clara el mismo comportamiento del modelo anterior (clasificador C.4.5), en donde al sobre-muestrear la clase C+ con un 150% de instancias

sintéticas permite al clasificador discriminarla cada vez mejor y que sea detectada de manera correcta (incremento de las métricas de rendimiento F y ROC en 38.7% y 15% respectivamente), a costa de la reducción del 6% de la sensibilidad y/o precisión (Medida-F) y 3.3% del área ROC de la clase C-. Por su parte, al sobre-muestrear la clase C+ con un nivel de muestreo del 200% mejora en 5.3% y 1.2% las métricas F y ROC con respecto a la anterior, reduciendo aún más el rendimiento de clasificación de la clase C-.

c) Comportamiento del rendimiento del clasificador MVS

Al entrenar el clasificador MVS se exhibe en cada caso un valor de la Medida-F nulo (F=0) sobre la clase C+. Caso contrario ocurre en la clase C-, donde se obtiene resultados buenos (Figura 25). De aquí se puede decir que la MVS no escapa al problema del desbalanceo de clases y por el contrario, es muy sensible a este inconveniente.

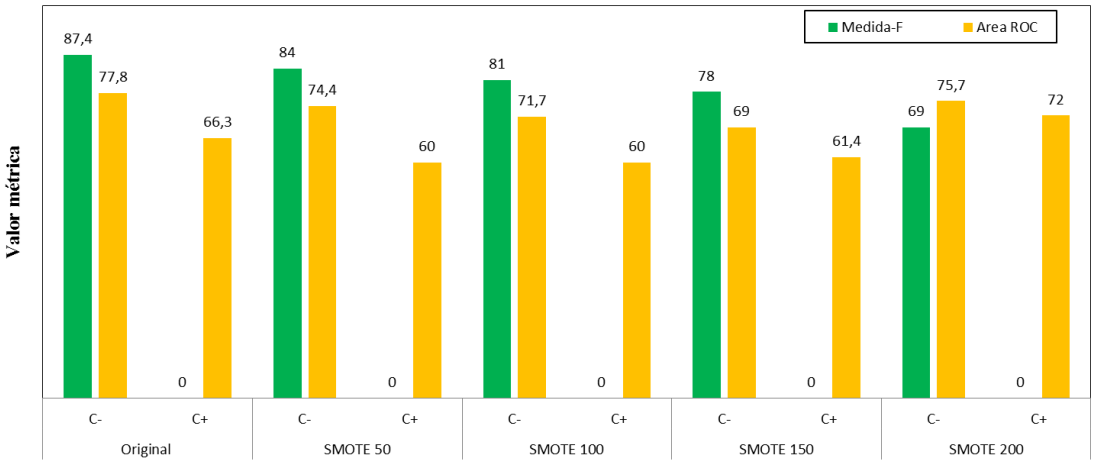


Figura 25. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Rio las Piedras sobre el clasificador MVS, aplicando SMOTE

Para corroborar estos resultados, se pueden observar en la Figura 21(b), Figura 22, Figura 22(b) y Figura 23(b), donde se nota que el algoritmo MVS en ninguno de los casos logró clasificar las instancias perteneciente a la clase C+, ya que sus coordenadas (TFP, TVP) se encuentran en el origen del plano (0,0) donde el clasificador no comete errores de falsos positivos, pero tampoco reconoce verdaderos positivos.

d) Comportamiento del rendimiento del clasificador RB

De la misma forma, al momento de entrenar el clasificador RB se consigue el mismo comportamiento anterior (Figura 26), exceptuando los casos en los que se entrena con los dataset muestreados al 150% y 200% donde las RB empieza a darle importancia a la clase minoritaria. Igualmente, se observa que en estos puntos el rendimiento de la clase mayoritaria se reduce en un 14.6% y en 18.9% para el caso de la métrica F y, de 3% y 4% para el caso de la ROC, de manera respectiva.

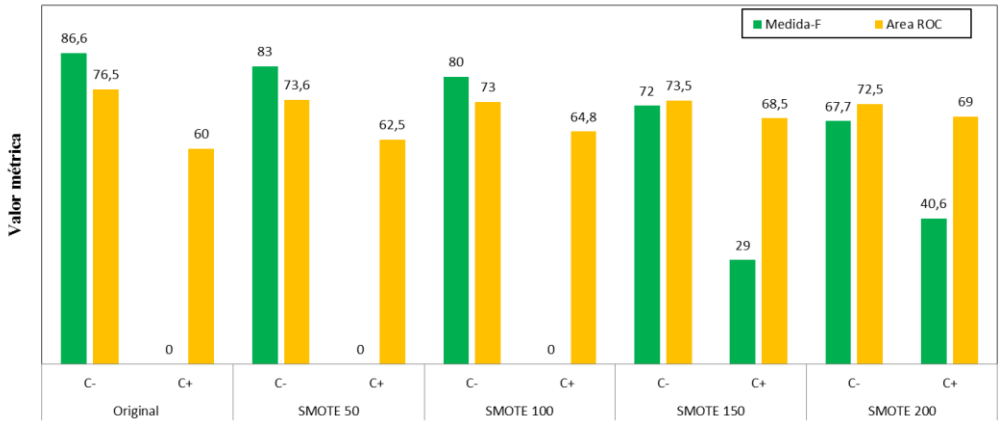


Figura 26. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Rio las Piedras sobre el clasificador RB, aplicando SMOTE

Observando las figuras: Figura 22(b) y Figura 23(b) se visualiza que cuando se sobre-muestrea el dataset con un 150% de instancias sintéticas, se incrementa en proporciones muy similares el número de verdaderos positivos que haciéndolo con un 200% de instancias sintéticas. No obstante, agregando un 150% de instancias sintéticas se obtiene una menor cantidad de falsos positivos que utilizando el un 200%. Esto significa que incluir en el dataset con un 150% de instancias sintéticas se obtiene un mejor rendimiento de clasificación de la clase C+.

e) Comportamiento del rendimiento del clasificador RNA

De la misma forma a los experimentos mostrados anteriormente, se clasifica el dataset mencionado utilizando la técnica RNA. Los resultados se pueden observar en la Figura 27.

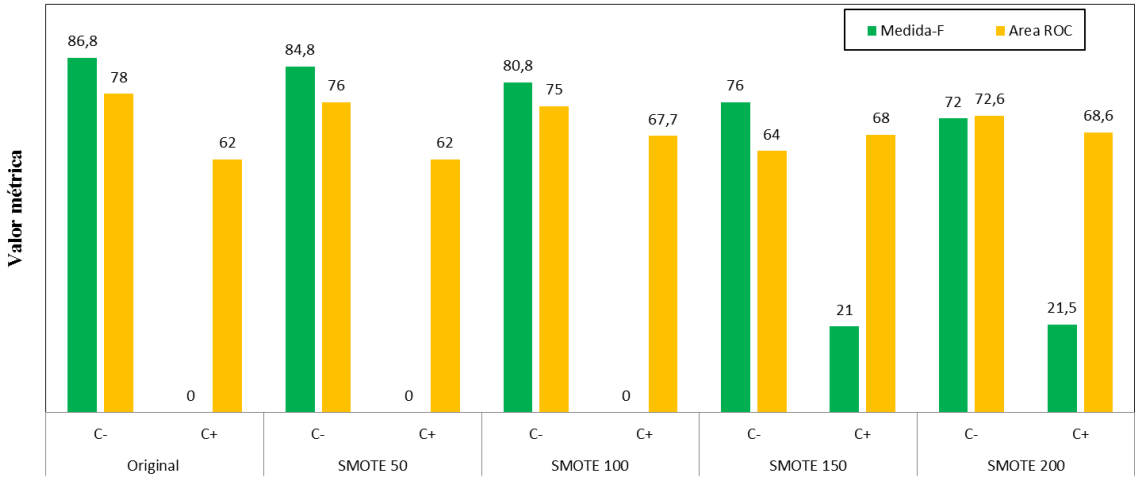


Figura 27. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Rio las Piedras sobre el clasificador RNA, aplicando SMOTE

Al igual que los métodos MVS y RB, las RNA presentan mayor dificultad para obtener ganancias cuando se sobre-muestrea el dataset con un porcentaje de instancias sintéticas menor a 150%. De hecho, al igual que los anteriores casos, sobre-muestrear el dataset con este valor se obtiene los mejores resultados para todos los casos. Es decir, se obtiene el mejor balance entre el número de verdaderos positivos y falsos positivos.

A partir de estos resultados, se observa cómo a pesar de balancear el dataset mediante SMOTE, el porcentaje de instancias clasificadas correctamente por los algoritmos MVS, RN y RNA no aumenta significativamente, e incluso es inferior a los resultados obtenidos entrenando directamente los clasificadores con el dataset desbalanceado (original). Por tanto, se puede decir que el hecho de utilizar muestras sintéticas generadas con SMOTE supone una ganancia en la capacidad discriminante de las clases, aunque no por igual en todos los clasificadores.

ANÁLISIS DE RENDIMIENTO DEL DATASET ESTUARIO CALIFORNIA

En este punto, se realizó el análisis y evaluación del rendimiento de cada uno de clasificadores mencionados anteriormente de manera individual, cuyos resultados se dan en términos de las métricas ROC y Medida-F.

A continuación son presentadas las gráficas asociadas a los resultados de las pruebas de definidas en la Figura 19.

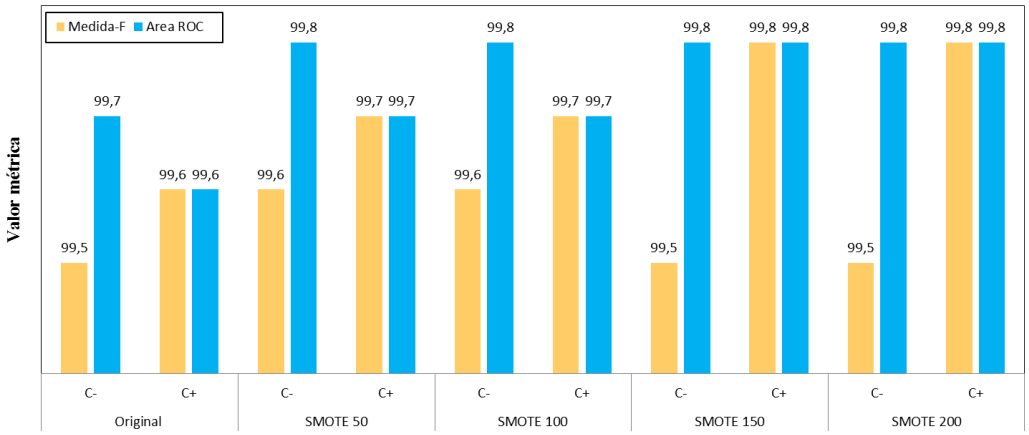
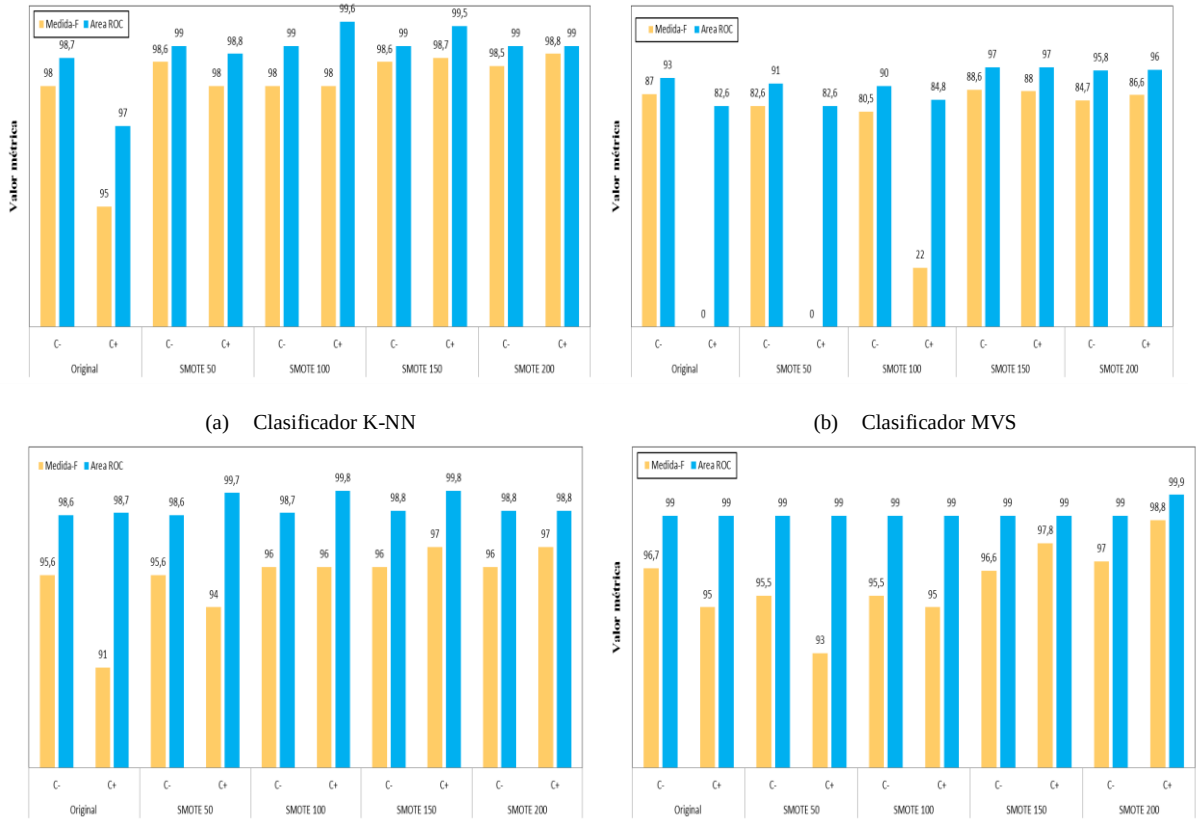


Figura 28. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Estuario California sobre el clasificador C.4.5, aplicando SMOTE

Con el mismo propósito, se entrenan los métodos de clasificación en cuestión con el dataset Estuario California original (IR=3.9) y los constituidos por datos sintéticos, obteniendo en todos los casos resultados de las métricas F y ROC (aproximados a 100% y 1, respectivamente) tal y como se muestra en las Figura 28 y Figura 29.



(a) Clasificador K-NN

(b) Clasificador MVS

(c) Clasificador RN

(d) Clasificador RNA

Figura 29. Resultados de eficacia de las clases mayoritaria C- y minoritaria C+ del dataset Estuario California aplicando SMOTE

En las figuras anteriores se observa que el desbalanceo de clases no perjudica el rendimiento de las detecciones de ninguno de los clasificadores. Incluso, se muestra que no existen diferencias significativas entre los resultados obtenidos por los clasificadores MVS, RB, K-NN, C.4.5 y RNA utilizando el dataset original y los resultados obtenidos al usar los niveles de sobre- muestreo 50%, 100%, 150% y 200%.

Estos resultados se pueden corroborar en la Figura 30, la cual presenta el espacio ROC obtenido por cada uno de los clasificadores mencionado en el transcurso de este trabajo al clasificar los dataset: original, sobre-muestreados al 150% y 200%. Debido a que los resultados son exactamente iguales en los tres casos se presenta una única grafica que describe el comportamiento de los tres dataset.

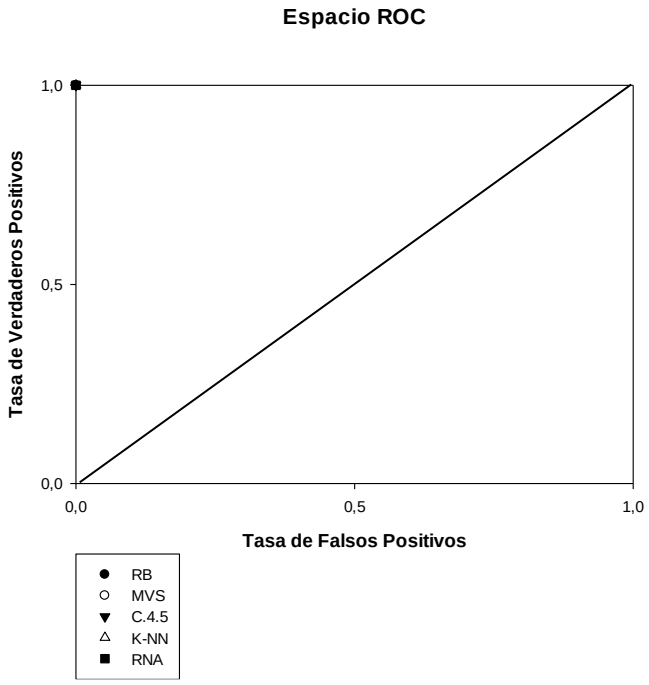


Figura 30. Espacio ROC para los clasificadores: RB, MVS, C.4.5, K-NN y RNA sobre los dataset: original y los sobre-muestreados a 150% y 200%

Aquí se observa que las coordenadas (TFP, TVP) de cada uno de los clasificadores están localizadas en el punto (0,1), en donde se considera una clasificación perfecta tanto de la clase C+ como la clase C-.

En este orden de ideas, los resultados experimentales mostraron que sobre-muestrear el dataset desbalanceado con niveles de 150% y 200% mejora la detección de los clasificadores en la clase C+ sin perjudicar significativamente la detección de la clase C-. Además, de obtener resultados bastante aproximados. No obstante, crear demasiadas instancias sintéticas produce sobre-entrenamiento, lo que afecta el poder de la detección de la clase minoritaria. En este sentido, se considera que el mejor nivel de sobre-muestreo de instancias sintéticas es 150%. Además, se puede decir que el algoritmo SMOTE es considerado un método práctico para la generación instancias sintéticas. Sin embargo, hay que tener especial cuidado en la cantidad de instancias de este tipo que se generen, ya que puede afectar la predicción de la C-.

Partiendo de este resultado, en la Tabla 13 se describen los dataset balanceados con un 150% de instancias sintéticas.

Dataset	No. Atributos	No. Instancias	No. Clases
Río Piedras	5	547	3
Estuario California	3	1757	4

Tabla 13. Descripción general de los dataset balanceados

Como se observa en la tabla anterior, sobre-muestrear los dataset con un 150% de instancias sintéticas incrementó en 54 ejemplos del dataset Rio Las Piedras y en 127 la cantidad de ejemplos del dataset Estuario de California.

4.3 Resumen

Este capítulo expuso de manera detallada el funcionamiento del algoritmo SMOTE seleccionado en la sección 2.2.3 y, posteriormente, fueron expuestas las métricas de evaluación de clases desbalanceadas: medida-F y ROC. De igual manera, se expuso el aporte realizado tomando como punto de partida dicha técnica.

Como resultado, se obtuvo que el mejor nivel de sobre-muestreo de instancias sintéticas es 150%. Además, se concluyó que el algoritmo SMOTE es considerado como un método práctico para la generación instancias sintéticas.

Capítulo 5

5 Algoritmos para la detección de la calidad del agua

En este capítulo son presentados los clasificadores para llevar a cabo la detección de la calidad del agua en sistemas lóticos. Primero, se justifica la selección de un conjunto de algoritmos de aprendizaje supervisado. Adicionalmente, se realiza la evaluación de los clasificadores y, finalmente, son seleccionados los algoritmos adecuados para dar solución al problema de investigación planteado.

5.1 Selección De Los Clasificadores

Para la selección de los clasificadores, fueron tomados 4 trabajos de investigación como punto de partida [78, 96-98], en los cuales evalúan teóricamente un conjunto de algoritmos de aprendizaje supervisado como: AD, RNA, RB, K-NN y MVS teniendo en cuenta las métricas: precisión, tolerancia al ruido, capacidad de explicación, velocidad de aprendizaje y velocidad de clasificación. Estas métricas pueden tomar valores entre 1 y 4, siendo 4 el mejor y 1 el peor rendimiento. En la Figura 31 se resumen los resultados obtenidos de la revisión.

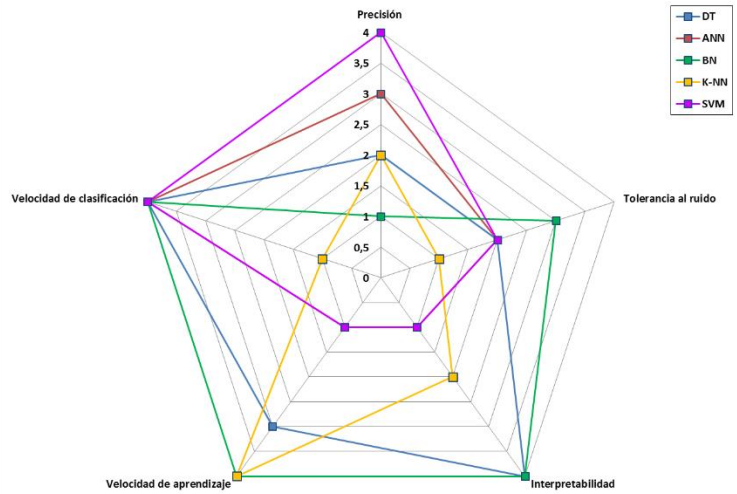


Figura 31. Evaluación de algoritmos de aprendizaje supervisado ([78])

De esta forma en la Figura 31 se puede observar que los algoritmos MVS y RNA obtienen mejor precisión respecto a los otros algoritmos. Sin embargo, estos presentan deficiencias en su interpretación, a diferencia de los algoritmos AD y RB que permiten al usuario observar el clasificador mediante una representación basada grafos, sin embargo carecen de precisión. Ahora bien, se destacan los algoritmos K-NN y RB por su velocidad de aprendizaje (entrenamiento), pero K-NN es lento para clasificar nuevos datos de entrada a diferencia de los otros algoritmos. Finalmente, de los algoritmos expuestos anteriormente, la técnica RB es la única que presenta un comportamiento aceptable para soportar datos erróneos en el proceso de entrenamiento (tolerancia al ruido).

De esta manera, cabe destacar que los algoritmo MVS y RNA son más utilizados en sectores donde su prioridad es obtener un alto grado de precisión en las predicciones, mientras que los AD son utilizados en ámbito donde es más importante generar clasificadores fáciles de interpretar.

Del análisis comparativo de los algoritmos de aprendizaje supervisado evaluados en los trabajos [78, 96-98], se puede decir que ningún algoritmo satisface todas las métricas de evaluación. Además, dependiendo del conjunto de datos utilizado, el algoritmo tiene un comportamiento diferente.

En este sentido, el presente trabajo de grado realiza una evaluación experimental de los algoritmos de aprendizaje supervisado mencionados en la Figura 31, con el fin de seleccionar un clasificador que genere un alto grado de precisión en la predicción y un clasificador de fácil interpretación, con base en el conjunto de datos de entrenamiento que permite detectar la calidad del agua en sistemas lóticos. Las métricas de evaluación utilizadas fueron descritas en la sección 4.3.1.

5.2 Experimentación Y Resultados

Esta sección presenta las evaluaciones y el análisis de los resultados de los clasificadores seleccionados, aplicados sobre los conjuntos de datos descritos en el apartado 2.1.1 y sobre los datos procesados después de aplicar el enfoque de reducción de la dimensionalidad y balanceo de clases (ver Figura 32).

5.2.1 Resultados experimentales

En la Figura 32 se expone el proceso experimental que se llevó a cabo para evaluar el mecanismo propuesto.

Por otro lado, el rendimiento de cada uno de estos clasificadores se obtiene en términos de las métricas: precisión, exhaustividad, Medida-F y ROC, consideradas para evaluar tanto el dataset original como los dataset procesados (mecanismo propuesto). Los resultados son expuestos a lo largo de esta sección.

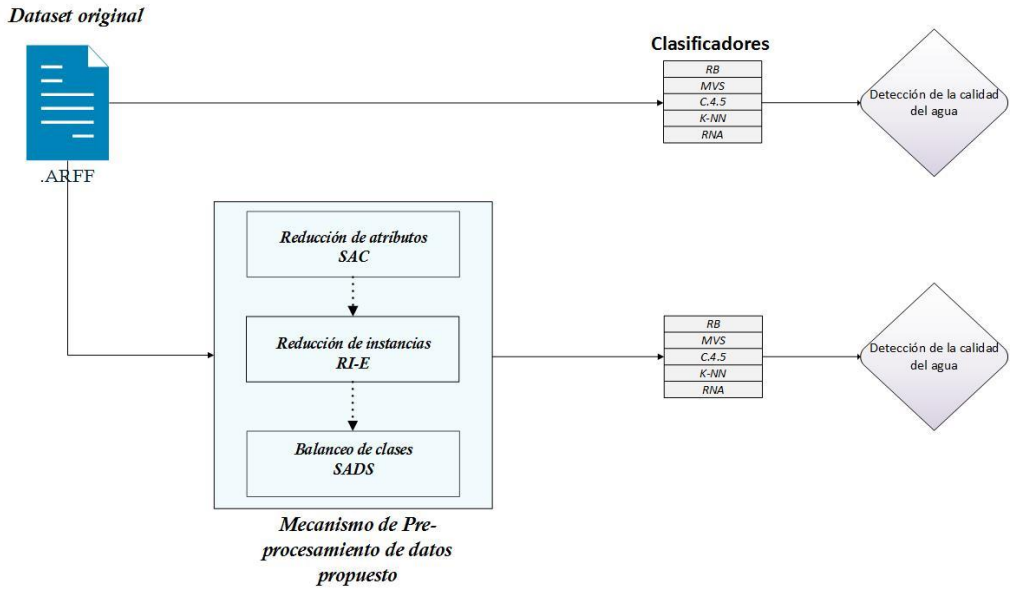


Figura 32. Proceso experimental selección de los clasificadores para la detección de la calidad del agua

ANÁLISIS DEL RENDIMIENTO DEL DATASET RIO LAS PIEDRAS

En la Figura 33 son presentadas las métricas: precisión, exhaustividad, Medida-F y ROC, de los algoritmos: MVS, RB, K-NN, C.4.5 y RNA, aplicado en el dataset rio las piedras, el cual contiene 18 atributos, 645 instancias y cuya variable objetivo es de tipo discreta (sección 2.1.1).

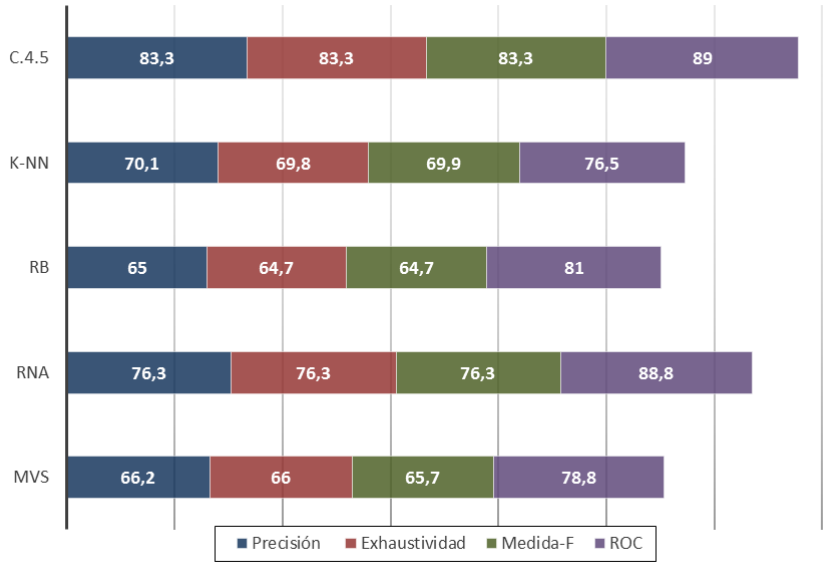


Figura 33. Resultados de la evaluación de los clasificadores: MVS, RNA, K-NN, RB y C.4.5 sobre el dataset de Rio Las Piedras (fuente propia)

Los algoritmos C.4.5 y RNA para el dataset original del Rio las Piedras, obtuvieron los mejores resultados entre todos los modelos evaluados, clasificando de manera incorrecta

un 16% y 24% de las instancias (108 y 153 instancias de 645) a comparación de los otros métodos que clasificaron incorrectamente más del 30% de las de instancias del dataset (MVS = 34% - 220, K-NN = 30.2% - 195 y RB=35.3% - 228) (*Anexo B*), lo cual se puede comprobar con los resultados expuestos en la Figura 33, donde la *precisión* obtenida por los modelos C.4.5 (83%) y RNA (76%) es superior en comparación con los clasificadores MVS (62.6%), K-NN (70.1%), y RB (65%).

Adicionalmente, es importante mencionar que el clasificador C.4.5 obtuvo la mayor proporción de verdaderos positivos que las otras técnicas, debido a que alcanzo una *exhaustividad* mayor al 83%, mientras que los demás algoritmos obtuvieron un valor de *exhaustividad* del 76.3% para el caso de RNA y no superior al 70% para el caso de MVS, RB y K-NN. En cuanto a la *medida-F*, los clasificadores C.4.5 y RNA siguieron el mismo comportamiento, ya que lograron obtener los mejores valores (83.3% y 76.3% respectivamente) de entre todas las técnicas evaluadas. Estos resultados dan a entender que los algoritmos supervisados C.4.5 y RNA son los que mejor comportamiento presentan para trabajar con este dataset.

De manera similar, se entrenan los cinco algoritmos de aprendizaje supervisado con el dataset del Rio Las Piedras procesado (Tabla 13), como se observa en la Figura 34. En esta gráfica se puede observar que para todos los modelos de clasificación se redujo la cantidad de instancias mal clasificadas (RB = 18.6% - 102, RNA = 18.2% - 100, MVS = 17.8% - 98 y K-NN = 27.5% - 151) con excepción del modelo C.4.5 en donde se mantuvo aproximadamente constante (C.4.5 = 16.4% - 90). Este comportamiento se ve reflejado con el incremento en la *precisión* de los clasificadores: RB (72.7%), MVS (73.8%) y K-NN (72.7%) con excepción de RNA que, aunque redujo la cantidad de instancias mal clasificadas, no se incrementó su *precisión*. Esto debido a que aumentó un 15.7% el número de Falsos Positivos (Anexo B).

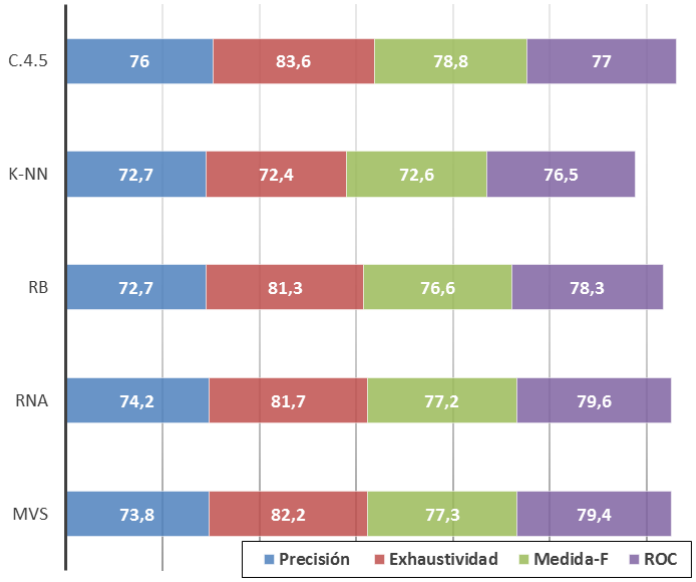
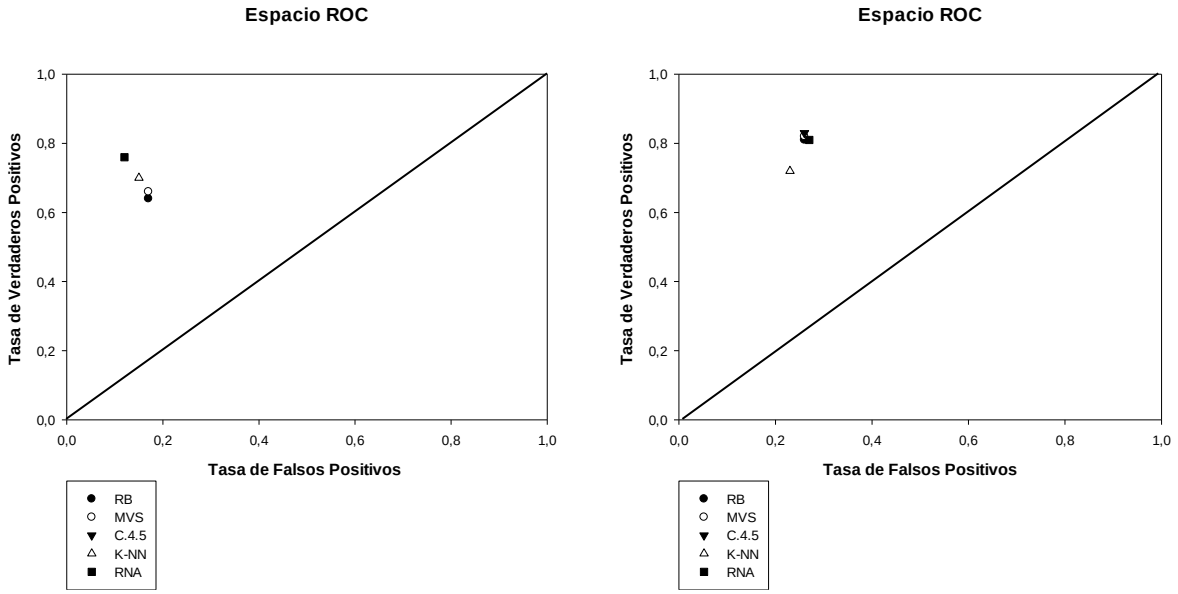


Figura 34. Resultados de la evaluación de los clasificadores: MVS, RNA, K-NN, RB y C.4.5 sobre el dataset de Rio Las Piedras procesado (fuente propia)

Sin embargo, se puede decir que la proporción de Verdaderos Positivos con relación a los Falsos Positivos es alta (Figura 35), debido a que los valores de *exhaustividad* de los modelos superan el 72% (MVS = 82.2%, RNA = 81.7%, RB = 81.3%, K-NN = 72.4% y C.4.5 = 83.6%). De estos resultados se puede decir que los 5 modelos evaluados son buenos ya que el número de falsos positivos es bajo y el número instancias relevantes clasificadas es alta, como se puede contrastar en el cálculo la métrica *Medida-F* (MVS=77.3%, RNA=77.2%, RB=76.6%, K-NN=72.6% y C.4.5=78.8%).

Debido a la poca diferencia de las métricas evaluadas anteriormente, resulta difícil ver cuál de los modelos de clasificación es el mejor. Por ende, se hace necesario evaluar el comportamiento de los clasificadores sobre el espacio ROC. En la Figura 35 se presenta en detalle el espacio ROC para los clasificadores expuestos anteriormente.



(a) Dataset Rio Las Piedras Original (b) Dataset Rio Las Piedras procesado

Figura 35. Espacio ROC para los clasificadores: RB, MVS, C.4.5, K-NN y RNA sobre dataset Rio Las Piedras original y procesado

En la figura anterior se observa que para los dos dataset, las coordenadas (TFP, TVP) obtenidas se encuentran por encima de la línea $TFP = TVP$, lugar en donde el algoritmo supervisado clasifica las instancias de manera correcta (no presenta ninguna aleatoriedad). Según estos resultados, se puede afirmar que todos los modelos de clasificación evaluados obtuvieron un alto número de verdaderos positivos, y con ello un buen comportamiento frente a los dos dataset evaluados. Sin embargo, aplicado sobre el dataset original (Figura 35(a)), el algoritmo supervisado RNA se encuentra más cercano a la clasificación perfecta (0,1) que los otros métodos, a pesar de esto, aplicado sobre el dataset procesado todos los modelos de clasificación, con excepción de K-NN, presentan un comportamiento similar en sus resultados y, por este motivo, sus puntos se traslapan (Figura 35(b)).

A partir de esto, se puede observar que los algoritmos evaluados presentan buenos resultados. Sin embargo, RNA y C.4.5 fueron los más precisos, los que clasificaron menor cantidad instancias incorrectas y, además, mantuvieron el mismo comportamiento en los

dos datasets. Por ende, se eligen estos modelos de clasificación, como posibles clasificadores para detectar la calidad del agua.

ANÁLISIS DEL RENDIMIENTO DEL DATASET ESTUARIO CALIFORNIA

En la Figura 36 son presentadas las métricas: precisión, exhaustividad, Medida-F y ROC, de los algoritmos: MVS, RB, K-NN, C.4.5 y RNA, aplicados en el dataset del Estuario California, el cual contiene 7 atributos, 2505 instancias y cuya variable objetivo es de tipo discreta (sección 2.1.1).

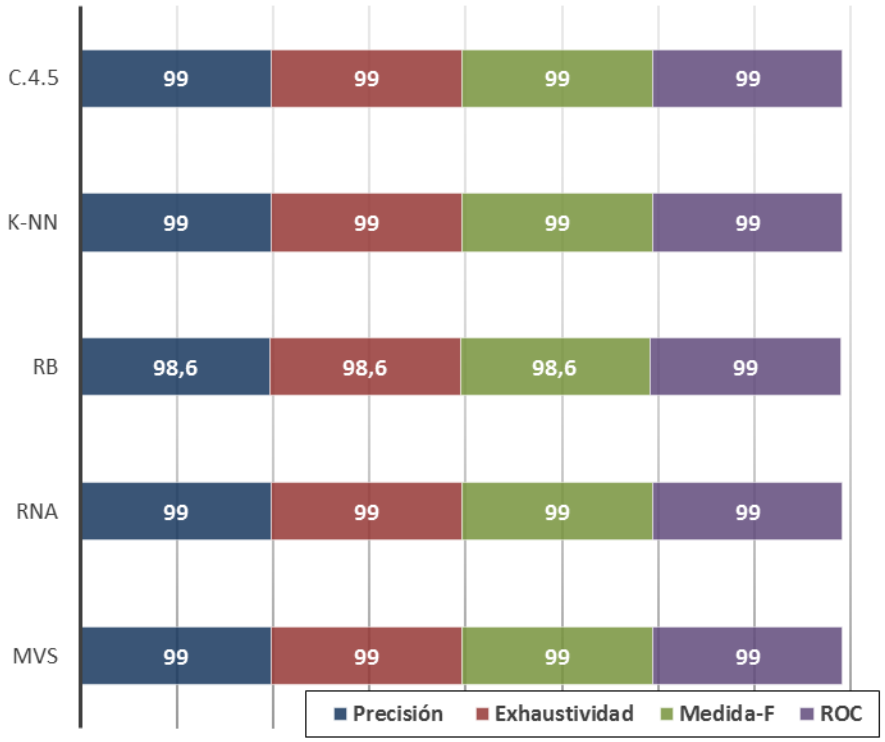


Figura 36. Resultados de la evaluación de los clasificadores: MVS, RNA, K-NN, RB y C.4.5 sobre el dataset de Estuario California (fuente propia)

Como se observa todos los modelos obtienen resultados en las métricas por encima de 98.6%, debido a que solo alcanzaron a clasificar de manera incorrecta 1 instancia para el caso de MVS, RNA, C.4.5 y K-NN, mientras que RB clasificó incorrectamente 9 instancias. Esto se puede comprobar con la *precisión* obtenida por estas técnicas la cual fue superior al 98% (Anexo B). De manera adicional, se observa que todos los modelos alanzaron una TVP alta, debido a que logró obtener una *exhaustividad* mayor al 98%.

Siguiendo con la misma línea, se entrenó los cinco clasificadores con el dataset del Estuario California procesado, obteniendo como resultado los datos expuestos en la Figura 37. En esta gráfica se puede observar que el pre-procesamiento del dataset afectó negativamente el comportamiento de los modelos MVS y RB. La MVS clasificó incorrectamente 207 instancias lo que produjo la reducción de la precisión en aproximadamente un 17%, mientras que RB clasificó de manera incorrecta 249 instancias que conlleva al decremento del 11.8% de precisión. Para el caso de la exhaustividad, la MVS y RB disminuyeron un 12% y 14.3% respectivamente, lo que se traduce en

incremento de la TFP en comparación con los demás modelos evaluados. A diferencia de RNA, C.4.5 y K-NN que mantuvieron el mismo comportamiento anterior (Figura 36).

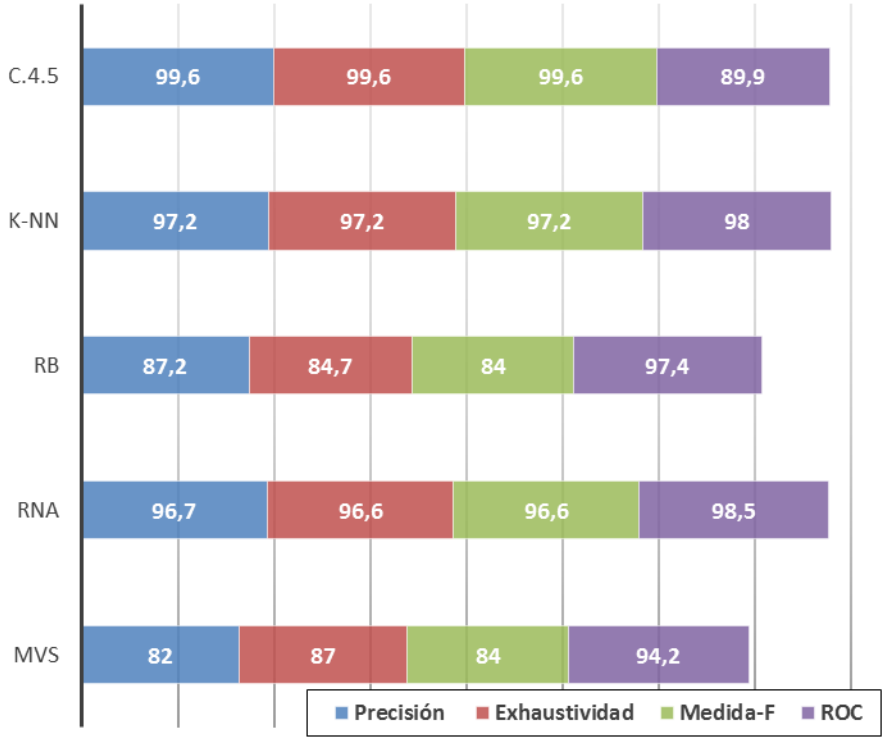


Figura 37. Resultados de la evaluación de los clasificadores: MVS, RNA, K-NN, RB y C.4.5 sobre el dataset de Estuario California procesado (fuente propia)

Según estos resultados, se puede afirmar que todos los modelos de clasificación evaluados obtuvieron un alto número de verdaderos positivos, y con ello un buen comportamiento frente a los dos dataset evaluados. Estos resultados se pueden corroborar con el alto valor que tomo la métrica ROC, la cual fue superior al 89%.

A partir de esto, se puede observar que los algoritmos evaluados presentan buenos resultados. Sin embargo, RNA y C.4.5 fueron los más precisos, los que clasificaron menor cantidad instancias incorrectas y, además, mantuvieron el mismo comportamiento en todos los experimentos. Por ende, se eligen estos modelos de aprendizaje supervisado, como los clasificadores que generen un balance entre precisión e interpretación, con base en el conjunto de datos de entrenamiento que permite detectar la calidad del agua en sistemas lóticos.

Por otro lado, se puede observar que el mecanismo propuesto es una solución adecuada para el pre-procesamiento de los dataset de la calidad del agua, el cual permite a los clasificadores proporcionar una precisión de clasificación similar a la obtenida al clasificar los dataset originales. Como resultado, la estrategia propuesta obtuvo precisiones similares a los dataset originales y, además, logró disminuir en gran medida el costo computacional.

5.3 Resumen

Este capítulo presentó la evaluación de los clasificadores más utilizados en diferentes contextos, tomando como punto de partida las investigaciones [78, 96-98]. Finalmente fueron seleccionados los algoritmos: RNA y C.4.5 con el fin de asegurar un alto grado de precisión en la predicción de la calidad del agua, y para realizar un análisis de la información contenida en los datos de entrenamiento, con el objetivo de detectar cualquier tipo de problema.

Capítulo 6

6 Prototipo para la detección de la calidad del agua

Este capítulo provee una visión global del enfoque de desarrollo a usar para realizar el prototipo para la detección de la calidad del agua y la manera como será gestionada dicha realización. Además, se presentan los resultados y artefactos generados en el proceso de ingeniería llevado a cabo. El proyecto fue desarrollado con base en la metodología de desarrollo software UP Ágil (Agile Unified Process) [99] y la metodología para el desarrollo de proyectos de minería de datos CRISP-DM (Cross Industry Standard Process for Data Mining) [100].

6.1 Proceso Unificado Ágil (PUA)

El Proceso Unificado Ágil es una versión simplificada del Proceso Unificado Racional (RUP- Rational Unified Process) y de la metodología de Desarrollo Extrema (XP- eXtreme Programming), en la cual se basa en adoptar técnicas ágiles como: desarrollo guiado por pruebas, refactorización de Base de Datos, modelado y gestión del cambio ágil, manteniendo la formalidad del proceso RUP. Es decir, PUA describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones software usando técnicas ágiles y utilizando la estructura RUP [101].

PUA además de estar dirigido por los Casos de Uso, está Centrado en la Arquitectura, y por ser iterativo e incremental, se compone de cuatro fases: iniciación, elaboración, construcción y transición. En cada una de estas fases se realizan un número determinado de iteraciones según el alcance del proyecto. Estas iteraciones están constituidas por siete flujos de trabajo o disciplinas, cuatro denominadas ingenieríles (Modelado, Implementación, Pruebas y Desarrollo) y tres de apoyo (Gestión de configuración, Gestión de proyectos y entorno). El Modelado agrupa las tres primeras disciplinas de RUP (Modelado de negocio, Requisitos, y Análisis y Diseño), tal y como se indica en la Figura 38. Adicionalmente, es importante mencionar que este proceso ágil sigue un modelo en cascada y su esfuerzo varía dependiendo de la fase en la cual se encuentra el proyecto. A continuación, se describe cada una de las fases mencionadas anteriormente.

Fase de Iniciación

Esta fase tiene como propósito definir el alcance del proyecto y los requisitos funcionales para el desarrollo del prototipo. Se elabora la propuesta inicial del plan de trabajo, y se establecen las fases e iteraciones a seguir. Además, se identifican los riesgos y todas las entidades externas con las que se trata (actores), definiendo sus interacciones a un alto nivel de abstracción (identificación de casos de uso).

Fase de Elaboración

En la fase de elaboración son seleccionados los casos de uso que permiten definir la arquitectura base del sistema, además de la especificación de los casos de uso.

Fase de Desarrollo

Esta fase está enfocada en la construcción de un producto completamente operativo y eficiente.

Fase de Transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

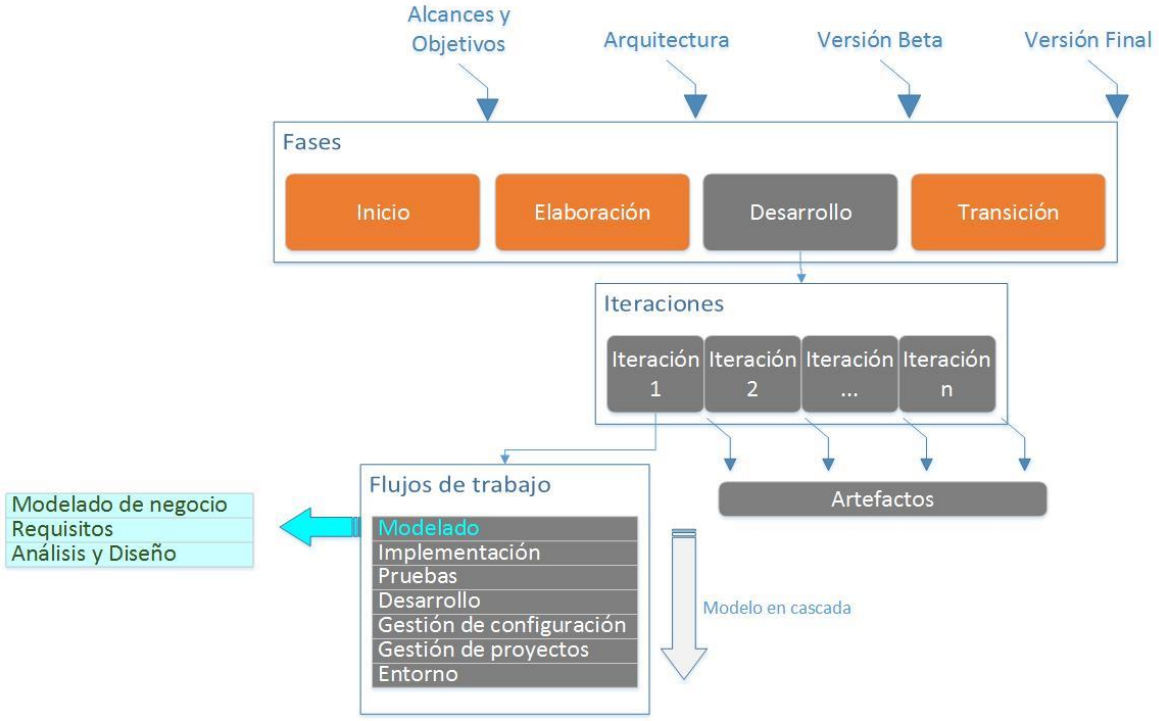


Figura 38. Ciclo de vida de la metodología PUA (fuente propia)

6.2 CRISP-DM

La metodología CRISP-DM está constituida por cuatro niveles de abstracción, organizados de manera jerárquica en tareas que van desde el nivel más general hasta los casos más específicos: fase, tareas generales, tareas específicas e instancias de proceso, tal y como se observa en la Figura 39.

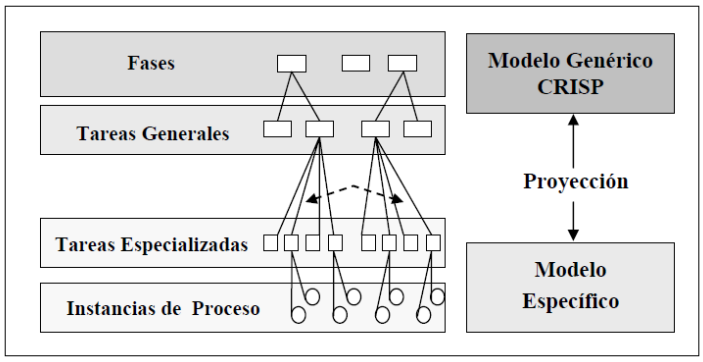


Figura 39. Esquema de los 4 niveles de CRISP-DM ([100])

Además ofrece una descripción del ciclo de vida del proyecto, como se puede observar en la Figura 40.

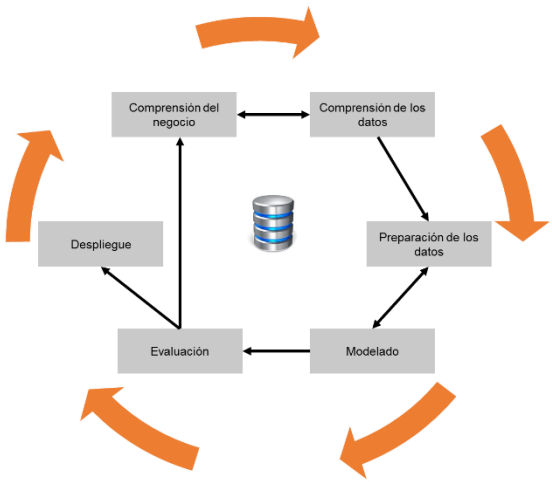


Figura 40. Fases del modelo de referencia CRISP-DM ([100])

Como se observa en la figura anterior, el ciclo de vida del proyecto contiene seis fases, con flechas que indican las dependencias más importantes y frecuentes entre fases. La secuencia de las fases no es estricta, de hecho, la mayoría de los proyectos avanzan y retroceden entre fases si es necesario [102]. Para el presente trabajo de grado fueron construidas las primeras cinco fases (Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, y Evaluación), la fase del despliegue no se ejecuta debido al alcance del objetivo general del proyecto, el cual llega hasta el desarrollo de un mecanismo para la detección de la calidad del agua en sistemas lóticos a través de técnicas de aprendizaje automático. A continuación, se explican las fases utilizadas:

Fase de Comprensión del Negocio

La fase de comprensión del negocio reúne las tareas de comprensión de los objetivos y requisitos del proyecto desde una perspectiva empresarial o institucional, con el fin de convertirlos en objetivos técnicos. En esta fase, es muy importante la capacidad de poder

convertir el conocimiento adquirido del negocio, en un problema de aprendizaje automático cuya meta sea el alcanzar los objetivos del negocio.

Fase de comprensión de los datos

Esta fase comprende la recolección inicial de datos con el objetivo de establecer un primer contacto con el problema, familiarizarse con los datos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis.

Fase de preparación de los datos

Una vez efectuada la recolección de datos, se procede a su preparación para adaptarlos a los algoritmos de Aprendizaje Automático que serán utilizados posteriormente, tales como técnicas de visualización de datos, de búsqueda de relaciones entre variables u otras medidas para exploración de los datos. La preparación de datos incluye las tareas generales de selección de datos a los que se va a aplicar una determinada técnica de modelado, tales como limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.

Fase de modelado

En esta fase de la metodología CRISP-DM, se describe el conjunto de algoritmos más apropiados que van a conforman el proyecto.

Fase de evaluación

Esta fase es la encargada de evaluar el proyecto. Sin embargo, es importante revisar el proceso, teniendo en cuenta los resultados obtenidos, para poder repetir algún paso anterior en el que se haya posiblemente cometido algún error.

Una vez analizadas las principales características de cada metodología a emplear, se hace necesario integrarlas de manera que se cuente con una sola línea de desarrollo. Para ello, se ha establecido la combinación de fases CRISP-DM con los flujos de trabajo del PUA expuesta en la Tabla 14.

Fase CRISP-DM	Fase PUA
Comprensión del negocio Comprensión de los datos	Modelo de negocio
Preparación de los datos	Requisitos
Modelado	Elaboración
Evaluación	Prueba

Tabla 14. Integración de las fases utilizadas de CRISP-DM y PUA

A continuación, son detalladas cada una de las fases aplicadas al presente trabajo de grado.

6.3 Fase De Inicio

En esta fase se determina la viabilidad y alcance del presente proyecto, mediante las bases conceptuales, investigación de presuntas alternativas y demás actividades que permiten estructurar el presente trabajo. Esta fase se compone de las siguientes actividades: modelado, captura y análisis de requerimientos funcionales e identificación de los casos de uso.

6.3.1 Modelado

Tomando como base la investigación documental realizada, donde se presentó la información acerca de las bases teóricas y trabajos relacionados (Capitulo 2), surgieron diferentes consideraciones para tener en cuenta en el desarrollo del prototipo a construir. Dichas consideraciones son presentadas a continuación:

Plan del Proyecto

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La Tabla 15 muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación preliminar).

Fase	No. Iteraciones	Duración
Fase de Inicio	1	3 semanas
Fase de Elaboración	1	2 semanas
Fase de Construcción	2	8 semanas
Fase de Transición	-	-

Tabla 15. Distribución de tiempos y el número de iteraciones de cada fase

Cada fase se concluye con un hito o conjunto de artefactos bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase. Ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los artefactos que marcan el final de cada fase se describen en la Tabla 16.

Fase	Artefactos
Fase de Inicio	• Lista de requerimientos • Casos de uso detectados • Diagrama de casos de uso • Prototipos de interfaz de usuario • Cronograma de actividades • Lista de riegos
Fase de Elaboración	• Casos de uso extendidos • Diseño de plan de pruebas • Arquitectura del sistema • Requisitos no funcionales
Fase de Construcción	• Diagramas de secuencia • Diagrama de clases • Diagrama de paquetes • Reporte de ejecución de pruebas • Código fuente del prototipo desarrollado
Fase de Transición	• Manual de usuario

Tabla 16. Artefactos que marcan el final de cada fase

En el presente capítulo, se muestran los principales resultados obtenidos en la aplicación de la metodología de desarrollo PUA. Sin embargo, la totalidad de los artefactos se encuentran en el Anexo C.

Modelo de Negocio

Como se dijo anteriormente, se realizó un análisis previo de los datos de la calidad del agua con el objetivo de determinar y organizar la información relevante para el desarrollo del presente trabajo. En el Anexo A se encuentra un artículo generado en la etapa de investigación denominado: "***A cluster analysis for water quality warnings in Piedras river basin***", donde se realiza el análisis de las agrupaciones generadas con el algoritmo de agrupamiento K-Means desde el punto de vista biológico, con el fin de generar alertas de la calidad del agua en la cuenca del Rio Las Piedras, mediante el uso de las técnicas de ACP y C.4.5. A partir de aquí, se identificó tanto el significado de las clases (Excelente, buena y regular calidad del agua) del dataset como los mejores atributos del mismo.

Adicionalmente, se identificaron los siguientes requisitos para la construcción del prototipo:

- **Objetivo del negocio**: detectar la calidad del agua en sistemas lóticos mediante técnicas de aprendizaje automático.
- **Fuente de datos**: la fuente de datos con la que se cuenta se describe en la sección 2.1.1.
- **Evaluación de la situación**:
  - ***Fuente de datos***: Los datos utilizados en el presente estudio provienen de dos sistemas lóticos, que proveen información acerca de la calidad del agua. El primer conjunto de datos de la Cuenca Rio Piedras en el Departamento del Cauca (Colombia), mientras que el segundo del estuario del Estado de California (Estados Unidos), suministrado por el repositorio USGS (United States Geological Survey). Estos datos se describen con mayor detalla en la sección 2.1.1.

  - ***Comprensión del problema***: se cuenta con la asesoría de un Biólogo, quien tiene el conocimiento necesario para detectar la calidad del agua de manera manual y con quien se construyó la primera versión del conjunto de datos de entrenamiento [31].

- **Objetivos desde la perspectiva del aprendizaje automático**:
  - Definir los clasificadores para la evaluación de la calidad del agua, enfoques para la reducción de la dimensionalidad, y mecanismos para el tratamiento del desbalanceo de clases (sección 2.2).

  - Definir una combinación estratégica de técnicas de aprendizaje automático para la preparación de los datos (capítulos 3 y 4).

  - Seleccionar los algoritmos de aprendizaje supervisado que permiten detectar la calidad del agua tomando como entrada el conjunto de datos de entrenamiento procesado (Capitulo 5).

6.3.2 Análisis y diseño

Alcance del Sistema

El prototipo permite la detección de la calidad del agua, a partir de técnicas de aprendizaje automático, dado un dataset que contenga información de la calidad del agua. La entrada

del sistema corresponde a un documento con extensión .arff [2]que corresponda con las características descritas en la sección 2.1.1 (conjunto de entrenamiento). Seguidamente, se realiza un pre-procesamiento de los datos (Reducción de atributos e instancias y balanceo de clases) y se entrena al modelo de clasificación con el dataset procesado. Ahora, para detectar la calidad del agua se ingresa al sistema otro documento con extensión .arff que contiene la información que se desea detectar (conjunto de prueba). Como resultado se visualiza una ventana (Figura 41) donde se expone el valor de la detección de la calidad del agua obtenida por el prototipo.

Figura 41. Resultados detección de la calidad del agua

6.3.3 Requisitos

Una vez efectuada la recolección de datos (sección 2.1.1), se procede a su preparación para adaptarlos a los algoritmos de Aprendizaje Automático que serán utilizados posteriormente. Para ello, se propuso un mecanismo que aborde de manera secuencial tareas de reducción de atributos e instancias y balanceo de clases, con el fin de preparar el conjunto de datos de entrenamiento (capítulos 3 y 4).

Por otro lado, teniendo en cuenta las características de la solución a desarrollar, se procedió a realizar el análisis y diseño de las funcionalidades necesarias para dar cumplimiento a las necesidades identificadas al inicio de la investigación. A continuación, se realiza la captura, análisis de los requerimientos funcionales e identificación de los casos de uso:

Identificación de los Casos de Uso

En la Figura 42 se presenta el diagrama de casos de uso para el prototipo desarrollado. Los casos de uso se identificaron a partir de las funcionalidades que ofrece el sistema. A continuación se realiza una breve descripción de cada uno de ellos:

- **CU-001 Reducir atributos**: este caso de uso permite reducir la cantidad de atributos dentro del dataset.
- **CU-002 Reducir instancias**: este caso de uso permite reducir la cantidad de instancias del dataset.

[2] ARFF, por sus siglas en inglés Attribute Relation File Format, es un archivo de texto ASCII que describe una lista de instancias que comparten un conjunto de atributos.

- **CU-003 Balancear clases**: este caso de uso permite al usuario emparejar o balancear la cantidad de instancias de las clases del dataset.
- **CU-004 Todos los procesos**: este caso de uso permite al usuario reducir el número de atributos e instancias del dataset. Además, permite realizar el balanceo de las clases (combinación de los casos de uso **CU-001-CU-003**.
- **CU-005 Visualizar detección de la calidad del agua**: este caso de uso determina y visualiza la detección de la calidad del agua de los casos de uso individual **CU-001-CU-003** y en conjunto **CU-004.**

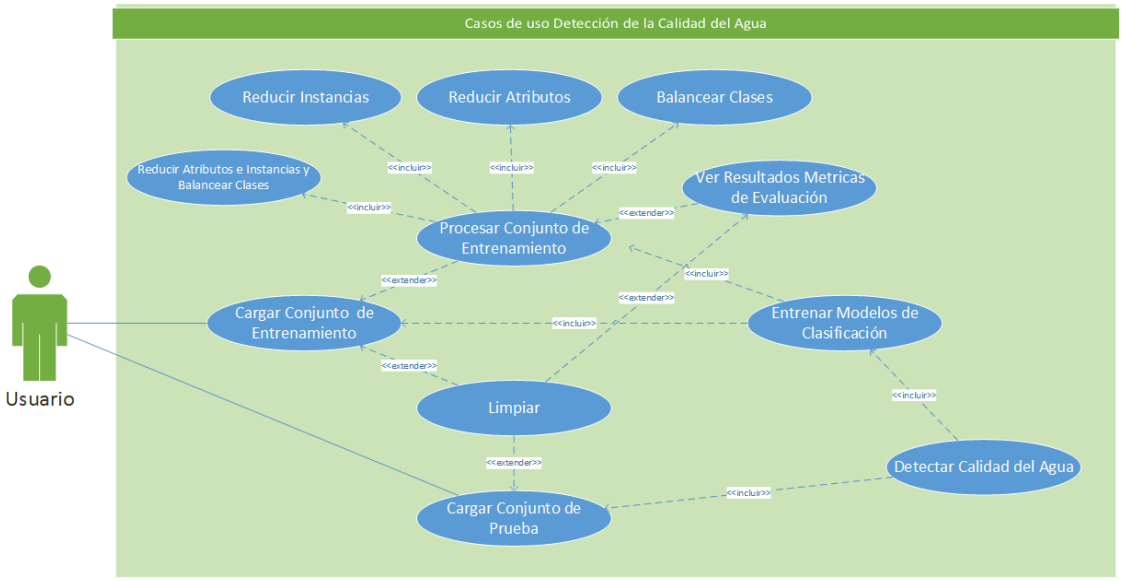


Figura 42. Diagrama de casos de uso del sistema

Captura y Análisis de Requisitos Funcionales

Los requisitos funcionales identificados para el prototipo son los siguientes:

- **RF1**. El usuario puede cargar un archivo con extensión .arff desde el computador donde se encuentra instalado el programa.
- **RF2**. El usuario tiene la opción de deshacer el archivo cargado.
- **RF3**. El prototipo es capaz de verificar si se ha cargado un archivo .arff, y notificárselo al usuario.
- **RF4**. Una vez cargado el archivo con extensión .arff, el sistema permite al usuario seleccionar el procesamiento de datos que desee (Reducción de atributos e instancias y balanceo de clases).
- **RF5**. El prototipo es capaz de procesar un archivo .arff y detectar la calidad del agua, a partir del archivo .arff cargado.
- **RF6**. El usuario tiene la opción de elegir visualizar los resultados de la detección de la calidad del agua obtenidos por el prototipo.
- **RF7**. El aplicativo es capaz de construir los dataset procesados, es decir, a partir del archivo cargado, el prototipo podrá crear un nuevo archivo .arff con los datos procesados.

6.4 Fase De Elaboración

En esta fase de elaboración, se describe el conjunto de algoritmos que conforman el prototipo para la detección de la calidad del agua. Para la construcción del prototipo, se analizaron trabajos de investigación recientes (Capítulo 2) y se analizaron diferentes métodos de aprendizaje automático (Capítulos 3, 4 y 5), en donde se propusieron mecanismos para la reducción de la dimensionalidad, balanceo de clases y además se seleccionaron los clasificadores para la detección de la calidad del agua.

A partir de las características de la solución a desarrollar se realizó el análisis y diseño de las funcionalidades necesarias para dar cumplimiento a las necesidades identificadas anteriormente. Para continuar, se expone la manera en que se relacionan entre si los componentes del prototipo.

Definición de la Arquitectura

A continuación, se presenta una breve descripción de cada uno de los subsistemas que componen el prototipo.

- **Usuario**: Corresponde a los usuarios que interactúan con el prototipo. Los usuarios cargan un archivo con extensión .arff, con el fin de detectar la calidad del agua que representa el mismo. El prototipo entrega al usuario los resultados de la detección de la calidad del agua de manera visual, tanto del dataset procesado, como del dataset sin procesar (original).
- **Dataset:** este componente contiene los datos de calidad del agua a detectar y tiene como extensión el formato .arff.
- **Dataset procesado:** este componente contiene la información del dataset generado por el módulo de procesamiento de datos (Reducción de atributos e instancias y balanceo de clases).
- **Procesamiento de datos**: este módulo permite procesar los datos del dataset ya sea: reducción de atributos, reducción de instancias y balanceo de clases de manera individual y conjunta.
- **Validación cruzada**: este módulo evalúa los resultados del análisis estadístico y garantiza que la partición de datos de entrenamiento y prueba sean independientes.
- **Librería Weka**: este componente está compuesto por la librería que se utilizó para modelar los mecanismos de procesamiento de datos y los algoritmos de clasificación.
- **Modelo de clasificación**: recibe como parámetros el conjunto de datos de entrenamiento y el conjunto de datos de prueba, resultado del módulo de validación cruzada. Además, este componente es el encargado de realizar la detección de la calidad del agua, a partir de dataset que recibió como parámetro.
- **Detección de la calidad del agua**: este bloque recibe los datos resultantes del modelo de clasificación y los muestra al usuario. Muestra tanto los resultados de la detección de la calidad del agua del dataset original (sin procesar) como del procesado.

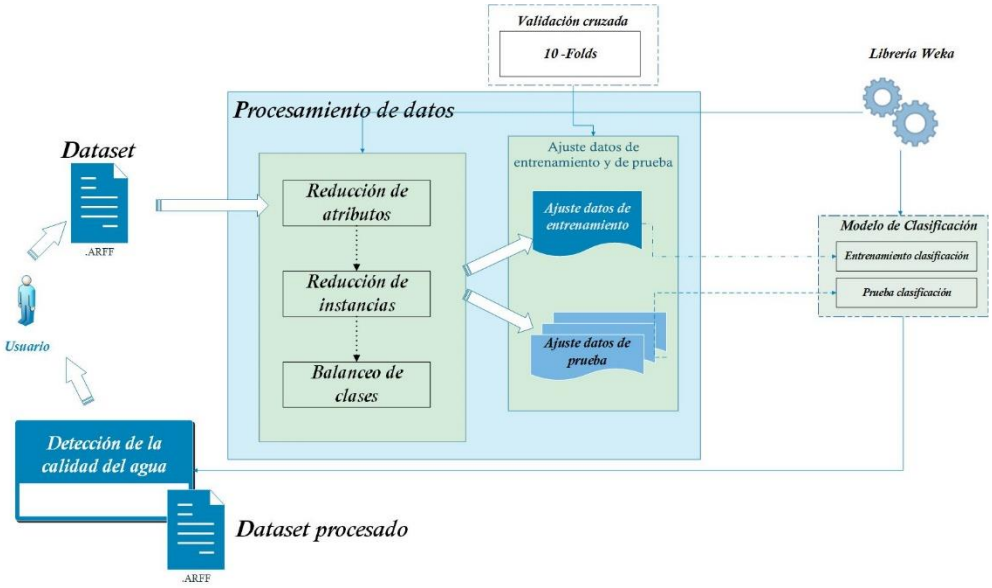


Figura 43. Arquitectura del prototipo

6.5 Fase De Desarrollo

En esta sección se llevan a cabo los aspectos relacionados con el desarrollo y la construcción de la plataforma final. Para ello se hace uso de UML (Lenguaje unificado de modelado) [103].

Las cuatro iteraciones definidas anteriormente (Tabla 15) para la fase de construcción, han sido divididas de acuerdo a los casos de uso detectados en la sección 6.3.3, de la siguiente manera:

Iteración	Casos de uso a realizar
Iteración 1	CU-001 y CU-005
Iteración 2	CU-002 y CU-005
Iteración 3	CU-003 y CU-005
Iteración 4	CU-004 y CU-005

Tabla 17. Definición de las iteraciones

Al final de cada una de las iteraciones, se debe realizar la ejecución de las pruebas diseñadas en la fase de elaboración. Las pruebas, que son diseñadas en la fase de elaboración y ejecutadas en la fase de construcción, están enfocadas en las funcionalidades y validación del sistema.

6.5.1 Diagrama de clases del sistema

Esta sección se enfoca en el desarrollo del diagrama de clases del prototipo implementado. La Figura 44 presenta el diagrama de clases de la aplicación, que son descritas a continuación.

- **GUI**: implementa la lógica necesaria para la presentación de la interfaz gráfica del prototipo.

- **PrincipalComponents**: esta clase realiza el proceso de Análisis de Componentes Principales, calculando la varianza y transformando el dataset en componentes ortogonales.
- **ACP**: esta clase contiene la lógica suficiente y necesaria para realizar la reducción de atributos del dataset.
- **ArffSaver**: esta clase es la encargada de gestionar la creación de un archivo con extensión .arff.
- **FilesArff**: la función principal de esta clase consiste crear un archivo con extensión .arff y escribir sobre este el listado de componentes más adecuado.
- **EvaluationClassifier**: esta clase calcula el promedio de las precisiones de los clasificadores.
- **ConverterUtils**: esta clase permite la manipulación de las características y atributos de los archivos de tipo .arff.
- **Instances**: la función principal de esta clase es aplicar los diferentes tipos de filtros a las instancias del dataset.
- **LoadBoostingIS**: esta clase contiene toda la configuración inicial de las variables del prototipo.
- **ManagerFileRips, Drop3Algo, Ib3Algo, MsAlgo, RnnAlgo y RandomAlgo**: estas clases implementan su respectivo algoritmo de reducción de instancias.
- **BoostingIS**: esta clase contiene la lógica suficiente y necesaria para realizar la reducción de instancias del dataset.
- **SMOTE**: esta clase implementa la definición del algoritmo con el mismo nombre.
- **SelectClases**: calcula el grado de desbalanceo que tienen las clases del dataset y las organiza de manera descendente.
- **LogicSmote**: la función principal de esta clase es encontrar la cantidad más adecuada de instancias sintéticas, para sobre-muestrear con ellas la clase minoritaria del dataset.

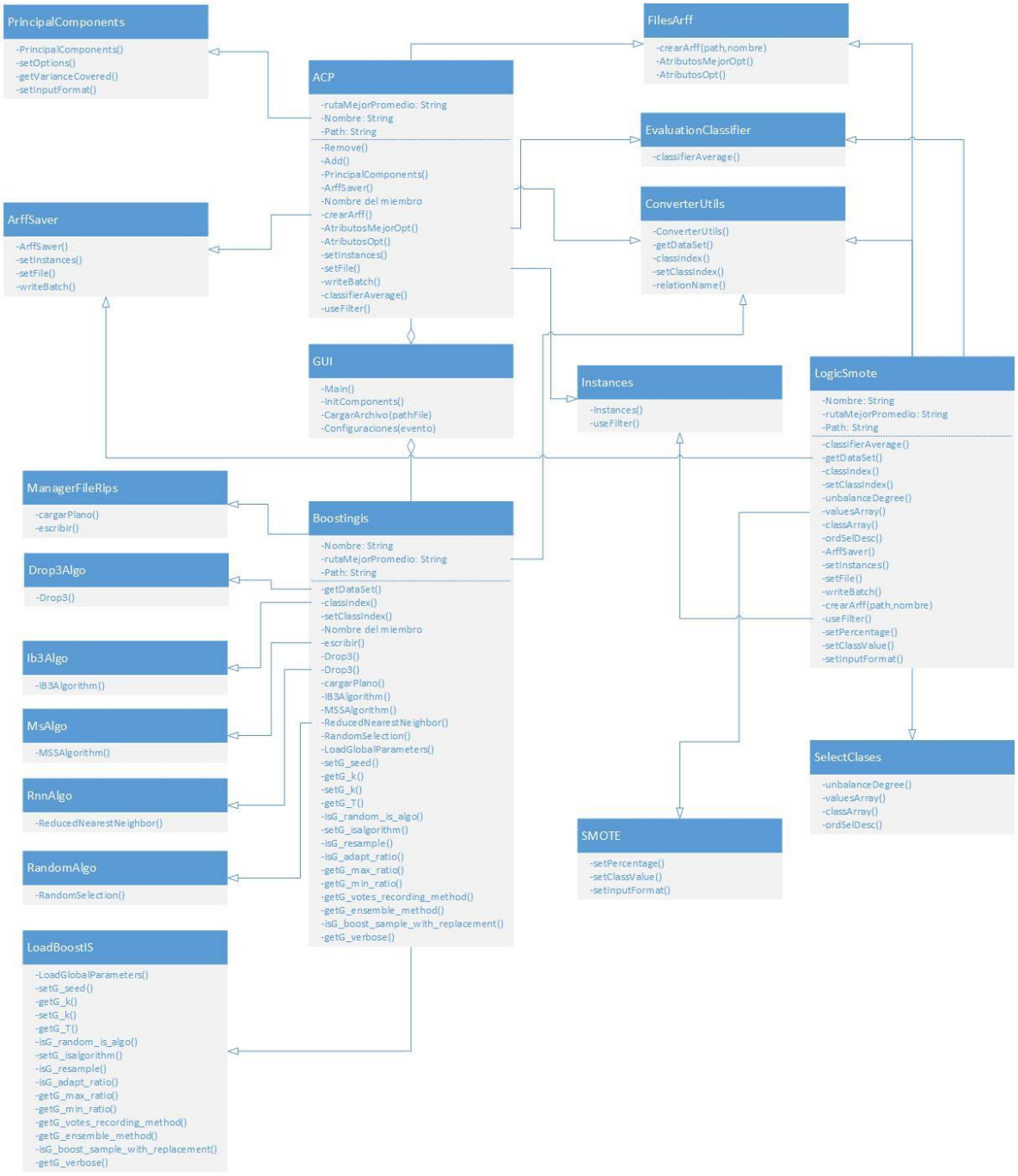


Figura 44. Diagrama de clases del prototipo

6.5.2 Diagrama de paquetes

Según [104], los diagramas de paquetes exponen la vista lógica de las aplicaciones software que componen el sistema. Dichos diagramas están organizados en paquetes, subsistemas y capas (Presentación y lógica del negocio, acceso a datos), mostrando la interacción existente entre capas, así como también los paquetes más relevantes que las componen. La Figura 45 presenta el diagrama de paquetes del prototipo.

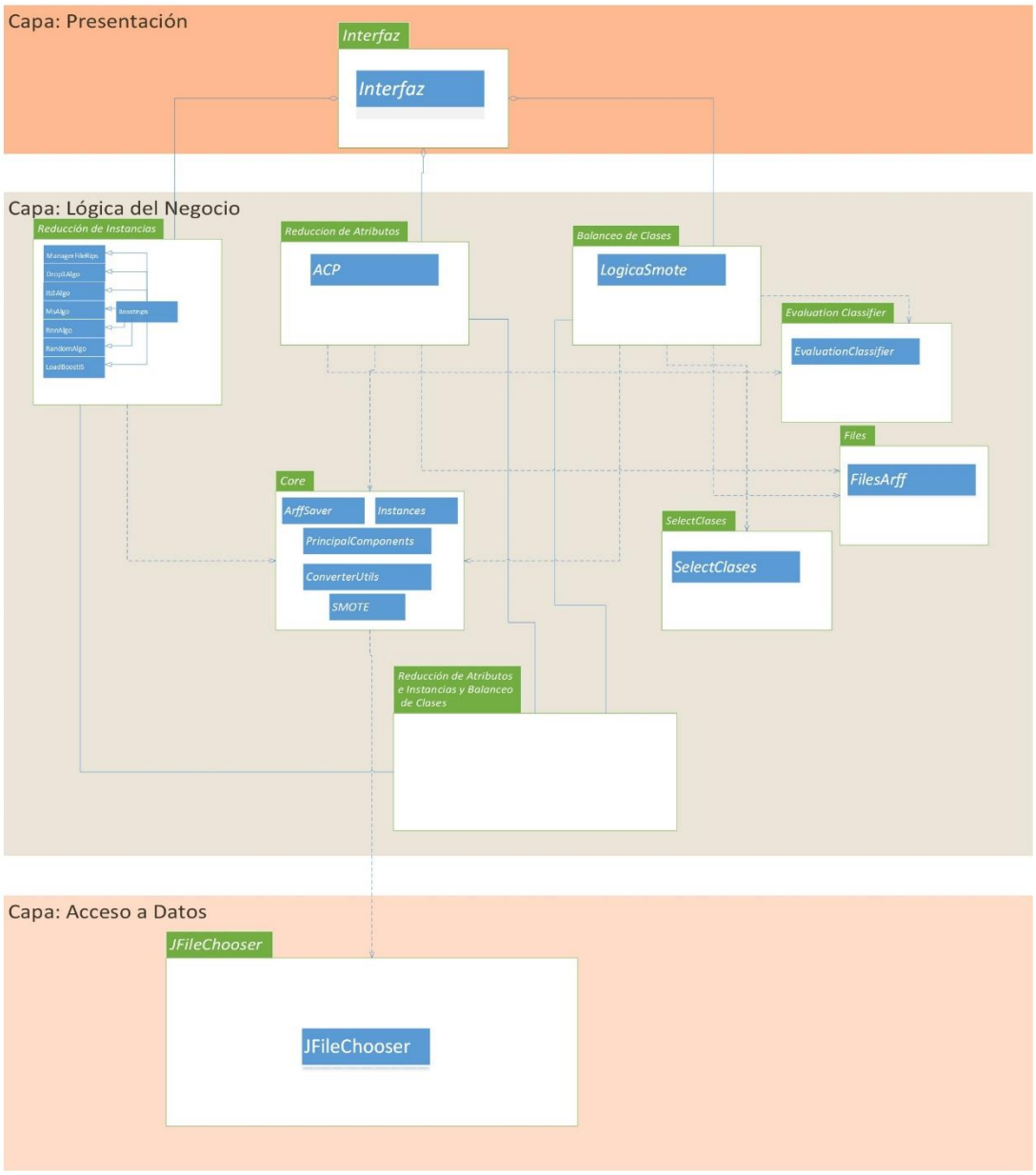


Figura 45. Diagrama de paquetes de la aplicación

A continuación se describen cada una de las capas y su respectiva interacción.

Capa de Presentación

Esta capa es implementada por medio del paquete Interfaz, el cual se describe a continuación.

- **Interfaz**: Ofrece una interfaz gráfica al usuario, que permite cargar un archivo .arff (dataset) con el fin de detectar la calidad del agua.

Capa Lógica del Negocio

Los paquetes que implementan esta capa representan las funcionalidades más importantes del sistema propuesto. En esta sección se realiza una breve descripción de cada uno de estos paquetes.

- **Core**: Paquete con las clases e interfaces que conforman la infraestructura de WEKA [105]. Define las estructuras de datos que contienen los datos a manejar por los algoritmos de aprendizaje. Este paquete encapsula un dataset, los atributos e instancias, junto con los métodos para manejarlos (creación y copia, división en sub-datasets, aleatorización, gestión de pesos, etc.).
- **SelectClases**: contiene la clase encargadas de seleccionar la clase minoritaria y mayoritaria del dataset, así como también calcular el grado de desbalanceo de estas clases.
- **Files**: este paquete representa la clase necesaria para seleccionar un conjunto de atributos que se desean suprimir del dataset. Además, permite crear el datasets procesado con extensión .arff.
- **Evaluation Classifier**: Representa a un paquete contenedor de la clase que permite evaluar el comportamiento de un conjunto de modelos de clasificación de manera simultáneas. También calcula y promedia la precisión de los algoritmos de aprendizaje supervisado utilizados.
- **Reducción de instancias**: este paquete contiene toda la funcionalidad para reducir las instancias irrelevantes y redundantes del dataset.
- **Reducción de atributos**: representa un paquete contenedor de la clase que permite reducir el número de atributos del dataset, eliminando las características más irrelevantes, redundantes y que representan poca información.
- **Balanceo de clases**: contiene la clase que permite determinar la cantidad de instancias sintéticas adecuada para balancear las clases del dataset.

Capa de acceso a Datos

Esta capa tiene por objetivo servir como intermediaria entre la capa lógica de negocio y el dataset cargado en memoria. Los paquetes que hacen parte de esta capa son descritos a continuación:

- **JFileChooser**: Representa a un paquete contenedor de la clases que proporciona un mecanismo sencillo para que el usuario seleccione un archivo y lo cargue en memoria.

6.6 Fase de pruebas

Esta fase es la encargada de evaluar el prototipo construido. Sin embargo, es importante revisar el proceso, teniendo en cuenta los resultados obtenidos, para poder repetir algún paso anterior en el que se haya cometido algún error. Las pruebas de calidad y rendimiento se hacen sobre los módulos que realizan las funcionalidades más importantes del prototipo: Modulo de Reducción de la dimensionalidad, Modulo de Balanceo de Clases y Modulo del modelo de clasificación. En los Capítulos 3, 4 y 5 son explicadas las métricas utilizadas para medir la calidad y el rendimiento del sistema. Además, son presentados los resultados obtenidos.

6.7 Fase De Transición

En esta fase se presenta al usuario una breve descripción del manual de usuario del prototipo ejecutable desarrollado, que le permite detectar la calidad del agua mediante técnicas de aprendizaje automático, a partir de un dataset con información relacionada al agua.

6.7.1 Interfaz de usuario

En esta sección son presentadas las interfaces graficas de usuario (GUI) del prototipo desarrollado, basado en la metodología PUA. En la Figura 46 se puede observar la interfaz principal de usuario del prototipo.

Interfaz principal

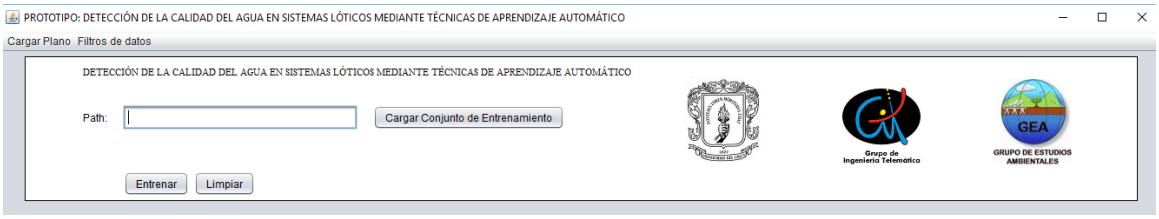


Figura 46. Interfaz principal de usuario

De manera seguida, se describen los principales controles de la interfaz gráfica principal del usuario:

- **Caja de texto Path**: esta caja de texto permite al usuario ingresar la dirección donde se encuentra el dataset.
- **Botón Cargar Conjunto de Entrenamiento**: este botón es utilizado para cargar el dataset de entrenamiento en memoria.
- **Botón Limpiar**: permite inicializar el estado del prototipo.
- **Botón Entrenar**: este botón da inicio al procesamiento de los datos y el respectivo proceso de entrenamiento de los modelos de clasificación.

Interfaz de configuración de procesos

En la Figura 47 se expone la interfaz de configuración del prototipo, en donde el usuario podrá seleccionar el o los procesamientos de datos de manera individual y combinada.

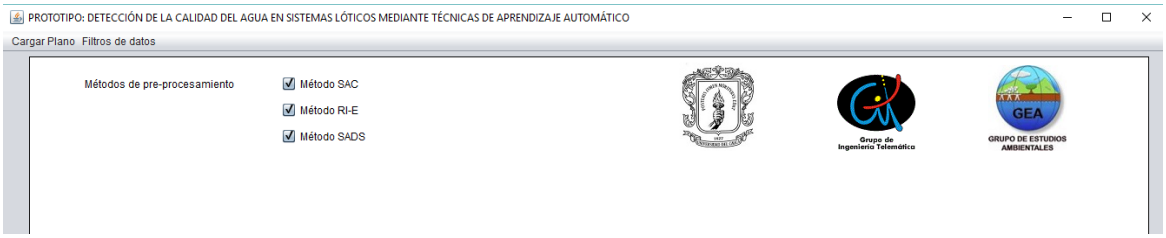


Figura 47. Interfaz de configuración de procesos

Una vez seleccionados los métodos de procesamiento de datos deseados, el usuario debe dirigirse a la interfaz principal (Figura 46) y presionar el botón *Entrenar* para dar inicio al entrenamiento de los clasificadores.

De igual manera, una vez finalizado el proceso de entrenamiento en la interfaz principal se mostrara los botones que se indican en la Figura 48.

- **Botón Cargar Conjunto de Prueba**: este botón es utilizado para cargar el dataset de prueba (dataset al que se le detectara la calidad del agua) en memoria.
- **Botón Ver Resultados de Entrenamiento**: permite visualizar los resultados del proceso de la detección de la calidad del agua.
- **Botón Volver a Entrenar**: este botón re-direcciona a la interfaz principal (Figura 46) y permite volver a entrenar los modelos de clasificación.
- **Botón Detectar la Calidad del Agua**: da inicio al proceso de detección de la calidad del agua.

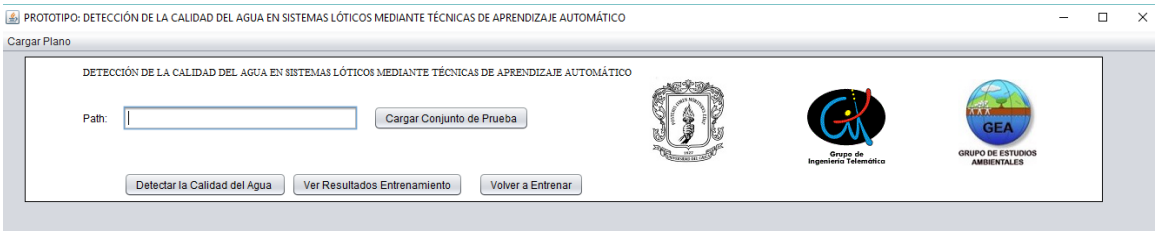


Figura 48. Interfaz principal – Resultados

Al presionar sobre el botón *Ver Resultados de Entrenamiento* se despliega el panel donde se exponen los resultados del proceso de entrenamiento obtenidos por el prototipo (Figura 49).

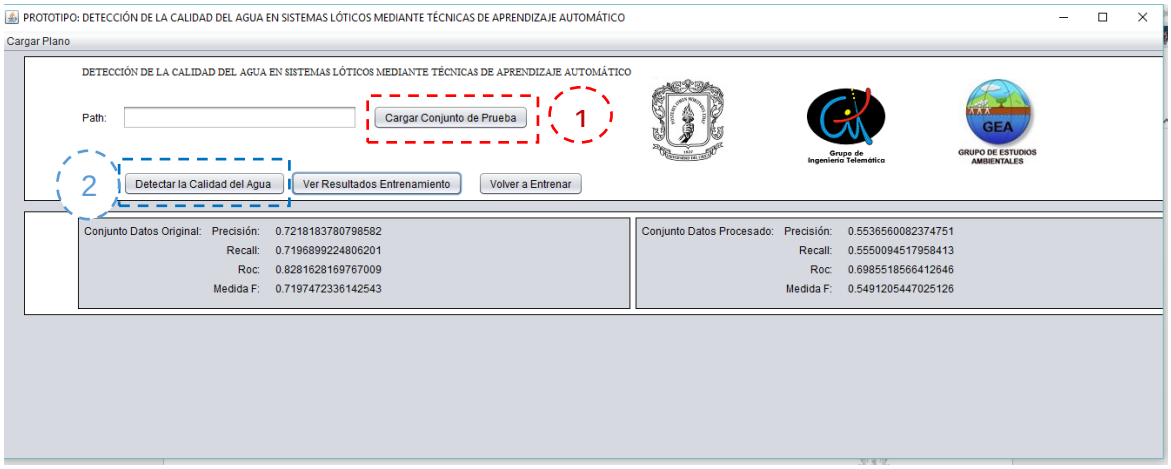


Figura 49. Interfaz principal – resultados de la detección de la calidad del agua

Como se indica en la figura anterior, el prototipo contrasta de manera visual los resultados obtenidos de entrenar los clasificadores con el dataset sin procesar (original) y el dataset procesado.

Por otra parte, el usuario puede realizar una limpieza de la interfaz principal presionando sobre el *Botón Limpiar*, y así volver al estado inicial del prototipo para poder empezar otro proceso.

Ahora, para detectar la calidad del agua el usuario debe cargar en el dataset de prueba, presionando sobre el botón *Cargar Conjunto de Prueba* y seleccionado el dataset

respectivo. Una vez cargado dicho dataset, se procede a detectar la calidad del agua presionando sobre el botón *Detectar la Calidad del Agua (*Figura 49*)*.

Figura 50. Resultado proceso de la detección de la calidad del agua

Como resultado final se despliega una ventana emergente que contiene la detección de la calidad del agua, tal y como se indica en la Figura 50.

6.8 Resumen

En este capítulo se expone el proceso de ingeniería realizado para definir la arquitectura que soporta el prototipo desarrollado de detección de la calidad del agua (integración de las metodologías CRISP-DM y PUA). Entre los aspectos teóricos expuestos en este capítulo se describieron las consideraciones previas a la definición de las arquitecturas y desarrollo de las mismas. Como resultado de aplicar la metodología de desarrollo PUA, en el presente proyecto surgieron una serie de artefactos para cada una de las fases concernientes a la metodología.

Por otro lado, en este capítulo sólo fueron mostrados los artefactos más relevantes junto con su descripción. Sin embargo, la totalidad de los artefactos están plasmados en el ANEXO C, estos artefactos están distribuidos en diferentes fases de la siguiente forma:

- **Fase iniciación**: Lista de requisitos, lista de riesgos, Diagrama de casos de uso, prototipos de interfaz de usuario, cronograma de actividades.

- **Fase elaboración:** Diagrama de casos de uso extendidos, plan de pruebas (ANEXO C), arquitectura, requisitos no funcionales.

- **Fase Construcción:**
  - **Análisis:** Diagramas de secuencia, Diagramas de colaboración
  - **Diseño:** Diagrama de clases, Diagrama de paquetes.
  - **Implementación:** Código fuente (aplicación)
  - **Pruebas:** Ejecución de las pruebas

- **Despliegue:** Manual de usuario.

Finalmente, en este capítulo se muestran las diferentes interfaces de usuario que hacen posible el entendimiento de la aplicación.

Capítulo 7

7 Conclusiones y trabajos futuros

Con la realización de este proyecto de fin de carrera, se pretendió aplicar técnicas de aprendizaje automático para la detección de la calidad del agua. En este capítulo se presentan las conclusiones y los trabajos futuros.

7.1 Conclusiones

Una vez construido el mecanismo para la detección de la calidad del agua en sistemas lóticos, se llegaron a las siguientes conclusiones:

- Debido a la ausencia de clases para detectar la calidad del agua en los datasets con los que contamos, en [31] propusimos un mecanismo para generar un dataset para la generación de alertas de la calidad del agua basado en el análisis del algoritmo de agrupamiento K-Means y el modelo de clasificación C.4.5. Como resultado, obtuvimos tres clases de alertas (calidad biológica del agua alta, buena y regular) para el dataset del Rio Las Piedras y cuatro alertas de calidad del agua para el dataset del Estuario California (calidad biológica del agua alta, buena, regular y mala).

- Para dar cumplimiento a los objetivos planteados para este trabajo, se realizó una revisión literaria de los trabajos de investigación más relevantes desde el 2006 hasta el presente tomando como fuentes de búsqueda: IEEE, ScienceDirect y Google Schoolar, y enfocados en tres áreas de estudio: clasificadores para la detección de la calidad del agua, enfoques para la reducción de la dimensionalidad, y mecanismos para el tratamiento del desbalanceo de clases. Para la primer área de investigación, se analizaron 19 artículos donde los algoritmos MVS (5 artículos) y RNA (7 artículos) son utilizados con mayor frecuencia para predecir y monitorear la calidad del agua, y los algoritmos C.4.5 (2 artículos), K-NN (2 artículos) y RB (3 artículos), fueron utilizados con el fin de generar alertas de calidad de agua. Para el caso de la reducción de la dimensionalidad, se analizaron 44 artículos enfocados en la reducción de atributos, donde el algoritmo de Análisis de Componentes Principales (ACP) es el más utilizado (33 artículos). Y para reducir la cantidad de instancias, los métodos “Envoltura” y “Ensamble” son los fueron los más utilizados con 12 y 16 artículos respectivamente. Por último, para dar solución al desbalanceo de clases se analizaron 30 trabajos donde 17 de los artículos utilizan SMOTE, con el fin de mejorar la eficacia de clasificación y las 13 investigaciones restantes utilizan los métodos internos para el mismo propósito.

- Se propuso un mecanismo para eliminar información redundante del conjunto de datos de entrenamiento, el cual logro realizar una selección bastante contundente de los datos (mayor interpretabilidad) sin deteriorar las capacidades de los clasificadores.
- Una Dataset con un número alto de instancias incrementa las capacidades de los clasificadores.
- Eliminar instancias de un conjunto de datos pequeño incrementa las posibilidades de eliminar información importante del mismo, y con ello, reducir las capacidades del clasificador.
- Los resultados muestran que el mecanismo propuesto es una solución adecuada para la reducción de la dimensión de los dataset de la calidad del agua, la cual permite a los clasificadores reducir el tiempo de entrenamiento, y a la vez, permite una precisión de clasificación similar a la obtenida al clasificar los dataset originales (sin pre-procesamiento).
- Se diseñó un mecanismo hibrido que integra una etapa de procesamiento de datos seguido de un modelo de clasificación que detecta la calidad del agua.
- Antes de aplicar los algoritmos de clasificación, es importante una etapa de preparación de los datos en donde se unifiquen, se limpien y se desechen los atributos redundantes o irrelevantes.

7.2 Trabajos futuros

El presente trabajo de grado ha aportado soluciones al problema para la detección de la calidad del agua en sistemas lóticos, a través del uso de algoritmos de aprendizaje automático. Así, con relación al campo de estudio de esta investigación se propone los siguientes trabajos futuros:

- Desarrollar esquemas similares a los propuestos en este trabajo de grado, considerando otros algoritmos de aprendizaje automático y otras métricas de evaluación.

- En la preparación de los datos, abordar otros problemas que pueden presentarse como: la ausencia en los datos y los valores atípicos.

- Probar el prototipo en un ambiente de producción.

- Crear un mecanismo tanto para la detección como para la predicción de la calidad del agua basado en series de tiempo.

- Las publicaciones llevadas a cabo en la última década apuntan a que la manera de resolver el problema de la selección de instancias en datasets de alta dimensionalidad es mediante algoritmos ensambladores, de envoltura y tipo filtro. A partir de aquí, se propone como trabajo futuro combinar las ventajas de cada una de estas técnicas.

- La adquisición de nuevos dataset que incluyan muchas más características referentes a la calidad del agua, ya que incrementando el número datos mejorará el proceso de selección de variables y a su vez la precisión y el comportamiento de los modelos de clasificación.

8 Referencias bibliográficas

[1] V. Y. D. T. Ministerio De Ambiente, "Política Nacional para la Gestión Integral del Recurso Hídrico," *Ministerio de Ambiente, Vivienda y Desarrollo Territorial, Bogotá, D.C.: Colombia.,* p. 124, 2010 2010.

[2] Á. L. F. Arango María Cecilia, Arango Gloria Alexandra, Torres Orlando Elí, Monsalve Asmed de Jesús, "Calidad Del Agua De Las Quebradas La Cristalina Y La Risaralda, San Luis, Antioquia," *EIA,* pp. 121-141, Julio 2008 2008.

[3] A.-T. Javier, "Macroinvertebrados Acuaticos Y Calidad De Las Aguas De Los Rios," *IV Simposio del Agua en Andalucia,* vol. 2, pp. 203-213, 1996.

[4] M. G. D. Pino Chalá Wilber , Mosquera Martha Lucia, Caicedo Kelly Patricia, Palacios Jhon Arley , Castro Anilio Alberto, GuerreroJair Enrique "Diversidad De Macroinvertebrados Y Evaluación De La Calidaddel Agua De La Quebrada La Bendición, Municipio De Quibdó (Chocó, Colombia)," *Acta Biológica Colombiana,* vol. 8 No.2, p. 8, Octubre 2003 2003.

[5] M. P. Claudia Rico, Nelson Fernandez, "Modelación De La Estructura Jerárquica De Macroinvertebrados Bentónicos A Través De Redes Neuronales Artificiales Acta Biológica Colombiana," *Open Journal Systems,* vol. 3, pp. 71-96, 2009.

[6] R. C. Young-Seuk Park, Arthur Compin, Sovan Lek, "Applications Of Artificial Neural Networks For Patterning And Predicting Aquatic Insect Species Richness In Running Waters," *Ecological Modelling,* pp. 265-280, 2003.

[7] T.-S. C. Young-Seuk Parka, Inn-Sil Kwakb, Sovan Leka, "Hierarchical Community Classification And Assessment Of Aquatic Ecosystems Using Artificial Neural Networks," *Science of the Total Environment,* pp. 105-122, 2004.

[8] G. R. Pérez, "Bioindicación De La Calidad Del Agua En Colombia: Uso Del Metodo Bmwp," vol. Primera edicion, p. 165, 2003.

[9] S. G. Kunwar P. Singh, "Artificial Intelligence Based Modeling for Predicting the Disinfection by-Products in Water," *Chemometrics and Intelligent Laboratory Systems,* vol. 114, pp. 122–131, 15 May 2012.

[10] Y.-S. P. Mi-Jung Bae, "Biological Early Warning System Based on the Responses of Aquatic Organisms to Disturbances: A Review," *Science of The Total Environment,* vol. 466-467, pp. 635–649, 1 January 2014 2014.

[11] I. O. K. Bucak, Bekir, "Detection of Drinking Water Quality Using CMAC Based Artificial Neural Networks," *Ekoloji Dergisi,* vol. 20, pp. 75-81, 2011.

[12] W. S. Hao Liao, "Forecasting and Evaluating Water Quality of Chao Lake Based on an Improved Decision Tree Method," *Procedia Environmental Sciences,* vol. 2, pp. 970–979, 2010.

[13] H. T. Shuangyin Liua, Qisheng Ding, Daoliang Li, Longqin Xu, Yaoguang Wei, "A Hybrid Approach of Support Vector Regression With Genetic Algorithm Optimization for Aquaculture Water Quality Prediction," *Mathematical and Computer Modelling,* vol. 58, pp. 458-465, August 2013 2012.

[14] P. S. Sirilak Areerachakul, Chidchanok Lursinsap, "Integration of Unsupervised and Supervised Neural Networks to Predict Dissolved Oxygen Concentration in Canals," *Ecological Modelling,* vol. 261-262, pp. 1-7, 24 July 2013 2013.

[15] J. X. Yue Liao, Wenjing Wang, "A Method of Water Quality Assessment Based on Biomonitoring and Multiclass Support Vector Machine," *Procedia Environmental Sciences,* vol. 10, pp. 451–457, 2011.

[16] J. Y. Guohua Tan, Chen Gao, Suhua Yang, "Prediction of Water Quality Time Series Data Based on Least Squares Support Vector Machine," *Procedia Engineering,* vol. 31, pp. 1194–1199, 2012.

[17] N. B. Kunwar P. Singh, Shikha Gupta, "Support Vector Machines in Water Quality Management," *Analytica Chimica Acta,* vol. 703, pp. 152–162, 10 October 2011 2011.

[18] M. G. Javier, "Modelos Híbridos De Inteligencia Computacional Aplicados En La Segmentación De Imágenes De Resonancia Magnética," Doctor Tesis Del Doctorado En Ingeniería, Orientación Electrónica, Ingeniería, Universidad Nacional de Mar del Plata, Facultad de Ingeniería, 2008.

[19] R. Rohwer, M. Wynne-Jones, and F. Wysotzki, "Neural Networks," in *Machine Learning, Neural and Statistical Classification*, S. Michie, Taylor, Ed., ed: Prentice Hall, 1994, pp. 84-105.

[20] G. S. H. J. Sun, Y.S. Wong, M. Rahman, Z.G. Wang, "Effective Training Data Selection In Tool Condition Monitoring System," *International Journal of Machine Tools and Manufacture,* vol. 46, pp. 218-224, February 2006 2006.

[21] S. C. Hyunjung Shin, "Fast Pattern Selection for Support Vector Classifiers," *San,* vol. 56 No.1, pp. 376--387, 2003.

[22] I. T. Abe Shigeo, "Fast Training of Support Vector Machines by Extracting Boundary Data," *Lecture Notes in Computer Science,* vol. 2130 - Artificial Neural Networks - ICANN 2001, pp. 308-313, Desember 2013 2008.

[23] D. E. Alejandro Hadad, Bartolomé Drozdowicz, "Modelo Para El Tratamiento De Datos Desbalanceados Basado En Redes Neuronales Auto-Organizadas," *Universidad Nacional de Entre Ríos,* January 2009 2009.

[24] S. Oh, "A new dataset evaluation method based on category overlap," *Computers in Biology and Medicine,* vol. 41, pp. 115-122, February 2011 2011.

[25] K. W. B. Nitesh V. Chawla, Lawrence O. Hall, W. Philip Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Artificial Intelligence Research,* vol. 16, pp. 321–357, January 2002 2002.

[26] D. C. Daniel Boley "Training Support Vector Machine using Adaptive Clustering," *SIAM International Conference on Data Mining,* p. 12, April 2004 2004.

[27] P. R. Seoung Bum Kim, "Unsupervised Feature Selection Using Weighted Principal Components," *Expert Systems with Applications,* vol. 38, pp. 5704–5710, May 2011 2011.

[28] S.-Y. C. Yok-Yen Nguwi, "An Unsupervised Self-Organizing Learning With Support Vector Ranking For Imbalanced Datasets," *Expert Systems with Applications,* vol. 37, pp. 8303–8312, December 2010 2010.

[29] R. W. Sebastián Maldonado, "A Wrapper Method For Feature Selection Using Support Vector Machines," *Information Sciences: an International Journal,* vol. 179, pp. 2208–2217, June 2009 2009.

[30] G. R. Pérez, *Bioindicación de la Calidad del Agua en Colombia: Propuesta Para el Uso del Método BMWP Col*, Primera ed. vol. 1: Universidad de Antioquia, 2003.

[31] W. F. G. Edwin Ferney Castillo, David Camilo Corrales, Iván Darío López, Miller Guzmán Hoyos, Apolinar Figueroa and Juan Carlos Corrales, "Water quality warnings based on cluster analysis in Colombian rivers basins," *Sistemas y Telemática (S&T),* 2015.

[32] R. P. Gabriel, "Bioindicación De La Calidad Del Agua En Colombia: Uso Del Metodo Bmwp," vol. Primera edicion, p. 165, 2003.

[33] L. d. Vargas, "Tratamiento de agua para consumo humano "Plantas de filtración rápida"," *Centro Panamericano de Ingeniería Sanitaria y Ciencias del Ambiente,* vol. 1, 2004.

[34] M. A. F. a. C. E. Brodley, "Decision Tree Classification Of Land Cover From Remotely Sensed Data," *Elsevier,* vol. 61, pp. 399-409, FeBruary 1997 1997.

[35] R. Pavón, "Redes Bayesianas Para El Ajuste De Parámetros De Algoritmos Genéticos Usados En Problemas De Satisfacción De Restricciones Geométricas," *Iberoamericana de Inteligencia Artificial,* vol. 14 No. 45, pp. 5-8, 2010.

[36] P. R. Guallart, "Mineria de datos aplicada al analisis del tratamiento informatico de la drogadiccion," Doctor Trabajo de Investigacion, Ciencias Fisicas, Matematicas y de la Computacion, Universidad Cardenal Herrera 2010.

[37] J. R. Fabian Guiza, Maurice Bruynooghe, Geert Meyfroidt, "Machine Learning Techniques To Examine Large Patient Databases," *Best Practice & Research Clinical Anaesthesiology,* vol. 23, pp. 127-143, 2009 2009.

[38] M. M. Munir Andrés Jalil, "Evaluación De Pronósticos Del Tipo De Cambio Utilizando Redes Neuronales Y Funciones De Pérdida Asimétricas," *Revista Colombiana de Estadística,* vol. 30 No. 1, pp. 143-161, Junio 2007 2007.

[39] R. Salas, "Redes Neuronales Artificiales," *Universidad de Valparaıso. Departamento de Computación,* p. 7, 2004.

[40] W. R. a. R. O. Juan David Gutierrez, "Bioindicación de la Calidad del Agua con Macroinvertebrados Acuáticos en la Sabana de Bogotá, Utilizando Redes Neuronales Artificiales," *Caldasia,* vol. 26, pp. 151-160, 2004.

[41] J. G. R. Ignacio García, Felipe López y Yenisse M. Tenorio, "Transporte de Contaminantes en Aguas Subterráneas Mediante Redes Neuronales Artificiales," *Información tecnológica,* vol. 21, pp. 79-86, 2010.

[42] G. M.-E. y. R. B.-C. Juan Mora-Florez, "Evaluación Del Clasificador Basado En Los K Vecinos Más Cercanos Para La Localización De La Zona En Falla En Los Sistemas De Potencia," *Ingeniería e Investigación,* vol. 28 No. 3, Septiembre-Diciembre 2008 2008.

[43] M. G. Campos, "Aplicacion De Tecnicas De Clustering Para La Mejora Del Aprendizaje," Ingeniero de Telecomunicaciones, Universidad Carlos III De Madrid, 2009.

[44] T. O. Ayodele, *Computer and Information Science. Artificial Intelligence "New Advances in Machine Learning"*: February 1, 2010 under CC BY-NC-SA 3.0 license, 2010.

[45] S. S. G. Jorge E. Hernandez L., " Implementación De Una Maquina De Vectores Soporte Empleando Fpga," *Scientia Et Technica,* vol. 7 No. 31, pp. 47-52, Agosto 2006 2006.

[46] M. A. C. M. PhD. German Hernandez, "Utilizacion De Las Maquinas Con Vectores De Soporte Para Regresion. M2 De Contruccion En Bogota," *Revista Avances en Sistemas e Informatica,* vol. 6 No. 6, p. 8, Septiembre 2009, Medellin 2009.

[47] K. Fukunaga, *introduction to statistical pattern recognition* School of Electrical Engineering-Purdue University-West Lafa yet te, Indiana.

[48] I. I. a. P. L. Yvan Saeys, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, pp. 2507-2517, 2007/10/1 2007.

[49] T. K. a. S. N. Samina Khalid, "A survey of feature selection and feature extraction techniques in machine learning," *Science and Information Conference (SAI),* pp. 372 - 378, August 27-29, 2014 2014.

[50] X. W. a. K. K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition,* vol. 36, pp. 2429 – 2439, 2002.

[51] R. S. a. T. D. L.Ladha, Lecturer, "Feacture selection methods and algorithms " *International Journal on Computer Science and Engineering (IJCSE),* vol. 3, May 2011 2011.

[52] M. C. P. a. L. M. Sasu, "Feature Extraction, Feature Selection and Machine Learning for Image Classification: A Case Study," *IEEE,* 2014.

[53] K. K. Paliwal, "Dimensionality Reduction of the Enhanced Feature Set for the HMM-Based Speech Recognizer," *DIGITAL SIONAI, PROCESSING,* pp. 157-173, 1992.

[54] S. S. S. A. a. W. Pedrycz, "Feature andInstance Selection Via Cooperative PSO," *IEEE,* pp. 2127 - 2132, 9-12 Oct. 2011 2011.

[55] C.-F. T. a. C.-W. Chang, "SVOIS: Support Vector Oriented Instance Selection for text classification," *Information Systems,* vol. 38, pp. 1070–1083, November 2013 2013.

[56] Z.-Y. C. a. S.-W. K. Chih-Fong Tsaia, "Evolutionary instance selection for text classification," vol. 90, pp. 104–113, April 2014 2014.

[57] N. J. a. M. Grochowski, "Comparison of Instances Seletion Algorithms I. Algorithms Survey," *Springer Berlin Heidelberg,* vol. 3070, pp. 598-603, 2004.

[58] J. A. C.-O. J. Arturo Olvera-López, J. Francisco Martínez-Trinidad and Josef Kittler, "A review of instance selection methods," *Artificial Intelligence Review* vol. 34, pp. 133-143 27 May 2010 2010.

[59] M. Blachnik, "Ensembles of Instance Selection Methods based on Feature Subset," *Procedia Computer Science,* vol. 35, pp. 388–396, 2014.

[60] N. G. P. a. A. d. H. García, "Boosting instance selection algorithms," *Knowledge-Based Systems,* vol. 67, pp. 342–360, 14 April 2014 2014.

[61] M. B. a. M. Kordos, "Bagging of Instance Selection Algorithms," *Artificial Intelligence and Soft Computing,* vol. 8468, pp. 40-51, 2014.

[62] X. Ming Gao, ShengChen and ChrisJ.Harris, "A combinedSMOTEandPSObasedRBFclassifierfortwo-class imbalancedproblems," *Neurocomputing,* vol. 74, pp. 3456–3466, 2011.

[63] A. F. Salvador Garcia, Francisco Herrera, "Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems," *Applied Soft Computing,* vol. 9, pp. 304–1314, 18 April 2009 2009.

[64] S. M. A. E. a. A. Abraham, "A Review of Class Imbalance Problem," *Journal of Network and Innovative Computing,* vol. 1, pp. 332-340, 2013.

[65] K. P. N. V. S. a. Dr.J.V.R.MURTHY, "An Exhaustive Literature Review on Class Imbalance Problem," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* vol. 2, June 2013 2013.

[66] E. R. Nele Verbiest, Chris Cornelis, and Francisco Herrera, "Improving SMOTE with Fuzzy Rough Prototype Selection to detect Noise in Imbalanced Classification data," 2009.

[67] S.-J. Y. a. Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications,* vol. 36, pp. 5718–5727, 2009.

[68] M. S. K. Yanmin Sun, Andrew K.C.Wong and Yang Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition,* vol. 40, pp. 3358 – 3378, 17 April 2007 2007.

[69] M. I. J. a. R. M. K. Eric P. Xing, "Feature Selection for High-Dimensional Genomic Microarray Data," *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning,* pp. 601-608, 2001.

[70] E. W. C. Hien M. Nguyen, Katsuari Kamei, "Borderline Over-sampling for Imbalanced Data Classification," *Fifth International Workshop on Computational Intelligence & Applications,* 2009.

[71] M. Khosrowpour and I. R. M. Association, *Machine Learning: Concepts, Methodologies, Tools and Applications* vol. 2: Information Science Reference, 2012.

[72] R. C. P. a. M. C. M. Gustavo E. A. P. A. Batista, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter,* vol. 6, pp. 20-29, June 2004 2004.

[73] J. S. S. a. R. A. M. V. García, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowledge-Based Systems,* vol. 25, pp. 13-21, 2012.

[74] C.-m. W. a. Y. Z. Yuan LUO, "Facial expression feature extraction using hybrid PCA and LBP," *The Journal of China Universities of Posts and Telecommunications. ScienceDirect,* vol. 20, pp. 120-124, 3 June 2013 2013.

[75] D. X. a. Y. Wang, "An automated feature extraction and emboli detection system based on the PCA and fuzzy sets," *Computers in Biology and Medicine,* vol. 37, pp. 861–871, June 2007 2007.

[76] B. Xiao, "Principal component analysis for feature extraction of image sequence," *International Conference on Computer and Communication Technologies in Agriculture Engineering,* vol. 1, pp. 250 - 253, 12-13 June 2010 2010.

[77] J. R. K. A. D. A. JACKSON, "Variable Selection In Large Environmental Data Sets Using Principal Components Analysis," *Environmetrics,,* vol. 10, p. 67±77, 1999.

[78] J. C. C. David Camilo Corrales, and Apolinar Figueroa-Casas, "Towards Detecting Crop Diseases and Pest by Supervised Learning," *Ing. Univ,* vol. 19, pp. 207-228, 2015.

[79] A. J. W. a. A. B. David Mease "Boosted Classification Trees and Class Probability/Quantile Estimation," *Journal of Machine Learning Research,* vol. 8, pp. 409-439 2007.

[80] A. F. a. F. H. Victoria López, "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," *Information Sciences,* vol. 257, pp. 1-13, 1 February 2014 2014.

[81] K. W. B. Nitesh V. Chawla, Lawrence O. Hall and W. Philip Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321–357, 2002.

[82] M. Á. S. a. J. C. Riquelme, "SMOTE-I: mejora del algoritmo SMOTE para balanceo de clases minoritarias," *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos,* vol. 3, 2009.

[83] J. F. M. T. y. M. G. B. Octavio Loyola González, "Clasificadores Supervisados basados en Patrones Emergentes para Bases de Datos con Clases Desbalanceadas," Coordinacio´n de Ciencias Computacionales INAOE, Sta. Ma. Tonantzintla2014.

[84] B. M. Kung-Jeng Wang, Kun-Huang Chen and Kung-Min Wang, "A hybrid classifier combining SMOTE with PSO to estimate 5-yearsurvivability of breast cancer patients," *Applied Soft Computing,* vol. 20, pp. 15–24, 2014.

[85] R. R. Sánchez, "Heuristicas de seleccion de atributos para datos de gran dimensionalidad," Doctor en Informática, Departamento de Lenguajes y Sistemas Informaticos, Universidad de Sevilla, Universidad de Sevilla, 2006.

[86] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment,* vol. 62, pp. 77–89, 1997.

[87] P. J. A. a. D. S. S. Satyam Maheshwari, "A New approach for Classification of Highly Imbalanced Datasets using Evolutionary Algorithms," *International Journal of Scientific & Engineering Research,* vol. 2, 2011.

[88] X. Y. Yang Liu, Jimmy Xiang, Huang and Aijun An "Combining integrated sampling with SVM ensembles for learning from imbalanced datasets," *Information Processing and Management,* vol. 47, pp. 617–631, 2011.

[89] T. Noguera, "Metodología ROC en la Evaluación de Medidas Antropométricas como Marcadores de la Hipertensión Arterial.," Master, Facultad de Matemáticas, Universidad Santiago de Compostela, España, 2010.

[90] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, pp. 861-874, 2006.

[91] M. Sullivan, *The Statistical Evaluation of Medical Tests for Classification and Prediction*: Oxford University Press, Incorporated, 2003.

[92] J. Cervantes, "Aprendizaje automático y modelos de clasificación. Aplicación ed la calificación crediticia de los gobiernos autónomosn descentralizados municipales como clientes del banco del estado," Ingeniero Matemático, Facultad de Ciencias, Escuela Politécnica Nacional, Ecuador, 2012.

[93] A. L. Nitesh V. Chawla, Lawrence O. Hall and Kevin Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," *Knowledge Discovery in Databases,* vol. 2838, pp. 107-119, 2003.

[94] N. V. Chawla, "Data Mining For Imbalanced Datasets: An Overview," *Data Mining and Knowledge Discovery Handbook,* pp. 853-867, 2005.

[95] S. Cooper, "The Paradoxical Role of Unesamined Documents in the Evaluation of Retrieval Effectiveness," *Information Processing and Management,* vol. 12, pp. 367-375, 1976.

[96] H. B. a. A. Ganatra, "A Comparative Study of Training Algorithms for Supervised Machine Learning," *International Journal of Soft Computing and Engineering (IJSCE),* vol. 2, pp. 74-81, 2012.

[97] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica,* vol. 31, pp. 249-268, 2007.

[98] I. D. Z. a. P. E. P. S. B. Kotsiantis, "Machine learning: a review of classification and combining techniques," *Springer Science,* vol. 2, pp. 159-190, 2006.

[99] S. B. Michele Sliger, *The Software Project Manager's Bridge to Agility*, Primera ed., 2008.

[100] J. C. S. Pete Chapman (NCR), Randy Kerber (NCR), Thomas Khabaza (SPSS), Thomas Reinartz (DaimlerChrysler), Colin Shearer (SPSS) and Rüdiger Wirth (DaimlerChrysler, "CRISP-DM 1.0 Step-by-step data mining guide," 2000.

[101] C. Edeki, "Agile Unified Process," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND MOBILE APPLICATIONS - IJCSMA,* vol. 1, pp. 13-17, September-2013 2013.

[102] R. Wirth, "CRISP-DM: Towards a standard process model for data mining," in *Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, Manchester, UK, 2000, pp. 29–39.
[103] **D. Maldonado**, "Metodología de Desarrollo UML," 2008.
[104] S. W. Ambler, "**UML Package Diagrams** " 2005.
[105] E. F. Mark Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations,* vol. 11, 2009.