



**INSTITUTO POLITÉCNICO NACIONAL**

---

**CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**

**Predicción de las condiciones de una planta  
depuradora de aguas residuales**

**T E S I S**

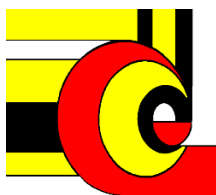
**QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS EN INGENIERÍA DE CÓMPUTO**

**PRESENTA:**

**ALEMBERT ALEJANDRO ROA DINORÍN**

**DIRECTORES DE TESIS:**

**DR. AMADEO JOSÉ ARGÜELLES CRUZ  
DR. OLEKSIY POGREBNYAK**



**MÉXICO, D.F.**

**NOVIEMBRE DE 2013**



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

SIP-14 bis

**ACTA DE REVISIÓN DE TESIS**

En la Ciudad de México, D.F. siendo las 11:30 horas del día siete del mes de noviembre de 2013 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

**Centro de Investigación en Computación**

para examinar la tesis titulada:

**“Predicción de las condiciones de una planta depuradora de aguas residuales”**

Presentada por el alumno:

<b>Roa</b>	<b>Dinorín</b>	<b>Alembert Alejandro</b>							
Apellido paterno	Apellido materno	Nombre(s)							
		Con registro: <table border="1"> <tr> <td>A</td> <td>1</td> <td>2</td> <td>0</td> <td>6</td> <td>4</td> <td>5</td> </tr> </table>	A	1	2	0	6	4	5
A	1	2	0	6	4	5			

aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

**LA COMISIÓN REVISORA**  
Directores de Tesis

Dr. Amadeo José Argüelles Cruz

Dr. Ofeksiy Pogrebnyak

Dr. Sergio Suárez Guerra

Dr. Herón Molina Lozano

Dr. Oscar Camacho Nieto

Dr. Cornelio Várez Márquez

PRESIDENTE DEL COLEGIO DE PROFESORES

Luis Antonio Vargas  
INSTITUTO POLITÉCNICO NACIONAL  
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO  
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN  
DIRECCIÓN




**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de México, D.F. el día 22 del mes de noviembre del año 2013, el (la) que suscribe Alembert Alejandro Roa Dinorín alumno(a) del Programa de Maestría en Ciencias en Ingeniería de Cómputo con opción en Sistemas Digitales, con número de registro A120645, adscrito(a) al Centro de Investigación en Computación, manifiesta que es autor(a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Amadeo José Argüelles Cruz y Dr. Oleksiy Pogrebnyak y cede los derechos del trabajo titulado Predicción de las condiciones de una planta depuradora de aguas residuales, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección(es) jackthelegendary@gmail.com, jamadeo@cic.ipn.mx y olek@cic.ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
Alembert A. Roa Dinorín

Nombre y firma del alumno(a)

## Resumen

En este trabajo de tesis se desarrolla, implementa y evalúa un método para realizar la predicción de las condiciones que presenta una planta depuradora de aguas residuales empleando un algoritmo inteligente de reconocimiento y clasificación de patrones perteneciente a la rama del cómputo no convencional. El algoritmo seleccionado es el clasificador Gamma el cual forma parte del ascendente enfoque asociativo Alfa-Beta, mismo que emplea los operadores Alfa-Beta como base de sus operaciones fundamentales y que sirvieron de inspiración para el desarrollo de las memorias asociativas Alfa-Beta.

La predicción se realiza tomando como base un banco de datos que describe y abstrae el comportamiento de una planta depuradora de aguas residuales por un periodo de tiempo de aproximadamente dos años, en los cuales se tiene un registro tanto de condiciones de trabajo normales como anormales. Las pruebas realizadas para hacer la predicción son presentadas en forma de una serie de tiempo.

La finalidad de este trabajo es presentar una propuesta que contribuya de forma importante para el posterior desarrollo de herramientas de apoyo que sirvan de respaldo en la toma de decisiones respecto al funcionamiento de una planta depuradora de aguas residuales para lograr que siempre trabaje en las mejores condiciones.

## **Abstract**

In this thesis, a method for prediction of the operation conditions of a wastewater treatment plant is developed, implemented and evaluated. The proposed method uses an intelligent algorithm for pattern recognition and classification that belongs to the branch of non-conventional computing. The selected algorithm is the Gamma classifier that forms a part of the ascending Alpha-Beta associative approach. This approach uses the Alfa-Beta operators as a basis of their core operations, and these operators inspired the development of Alpha-Beta associative memories.

The prediction is performed over a database that describes and abstracts the behavior of a wastewater treatment plant for a period of time two years approximately, in which both normal and abnormal working conditions were recorded. The accomplished prediction tests are presented in form of a time series.

This work presents a proposal to contribute significantly for the further development of support tools that help to the decision making regarding the wastewater treatment plant functioning. With the proposed approach, the best possible conditions for the plant operation are attained.

A la memoria de mi papá

René Roa Lima

(1953 – 2013)

## **Agradecimientos**

A mi mamá, mi papá y mis abuelos:

Por ser los pilares de todo lo que soy y puedo llegar a ser.

A mi gran familia:

Por todo el apoyo, el cariño y su admiración.

A mis amigos:

Por toda la ayuda, compañía y sobre todo su amistad.

A mis maestros:

Por permitirme ser su alumno y enseñarme el camino de la ciencia.

A mis directores de tesis:

Por su puntual y grandiosa guía en el desarrollo de mi trabajo.

Al CIC:

Por darme la oportunidad de demostrar mí valía y dejarme desarrollar como profesionalista.

Al IPN:

Por permitirme formar parte de una gran institución.

A todos simplemente:

**¡MUCHAS GRACIAS!**

**G M G**

# Índice general

<b>Resumen</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Agradecimientos</b>	<b>vi</b>
<b>Índice general</b>	<b>vii</b>
<b>Índice de figuras</b>	<b>x</b>
<b>Índice de tablas</b>	<b>xi</b>
<b>Glosario de términos</b>	<b>xii</b>
<b>1 Capítulo 1: Introducción</b>	<b>1</b>
1.1 Antecedentes . . . . .	1
1.2 Justificación . . . . .	2
1.3 Objetivo general . . . . .	3
1.4 Objetivos específicos . . . . .	3
1.5 Contribuciones . . . . .	3
1.6 Organización del documento . . . . .	4
<b>2 Capítulo 2: Estado del arte</b>	<b>5</b>
2.1 Planta depuradora de aguas residuales. . . . .	5
2.2 Modelos de inteligencia artificial. . . . .	6
2.2.1 Redes neuronales artificiales . . . . .	6
2.2.2 Redes bayesianas . . . . .	8
2.2.3 Regresión . . . . .	8
2.2.4 Análisis por componentes principales . . . . .	9
2.2.5 Mínimos cuadrados parciales. . . . .	9
2.2.6 Regresión polinomial multivariada . . . . .	10
2.2.7 Sistemas adaptativos de inferencia neuro-difusos . . . . .	10



2.2.8	Análisis por agrupamiento ( <i>clustering</i> ) . . . . .	11
2.2.9	Minería de datos. . . . .	12
2.2.10	Control automático. . . . .	13
2.2.11	Enfoque asociativo. . . . .	14
<b>3</b>	<b>Capítulo 3: Materiales y métodos</b>	<b>16</b>
3.1	Requerimientos para el clasificador. . . . .	16
3.1.1	Operadores Alfa ( $\alpha$ ) y Beta ( $\beta$ ). . . . .	16
3.1.2	Operador $u_\beta$ . . . . .	17
3.1.3	Módulo . . . . .	17
3.1.4	Código binario Johnson-Möebius modificado. . . . .	18
3.2	Clasificador Gamma. . . . .	19
3.2.1	Operador Gamma ( $\gamma$ ) de similitud . . . . .	20
3.2.2	Algoritmo del Clasificador Gamma. . . . .	21
3.3	Banco de datos . . . . .	29
3.3.1	Descripción del banco de datos . . . . .	30
3.4	Waikato Environment for Knowledge Analysis . . . . .	32
3.4.1	MultiLayerPerceptron . . . . .	32
3.4.2	RBFNetwork. . . . .	33
3.4.3	BayesNet. . . . .	33
3.4.4	NaiveBayes. . . . .	33
3.4.5	RandomTree. . . . .	33
3.4.6	REPTree . . . . .	33
3.4.7	SVM . . . . .	34
3.4.8	IBk . . . . .	34
<b>4</b>	<b>Capítulo 4: Modelo propuesto</b>	<b>35</b>
4.1	Pre-procesamiento del banco de datos. . . . .	35
4.1.1	Valores faltantes. . . . .	35
4.1.2	Número de clases . . . . .	37
4.2	Valores iniciales del clasificador Gamma. . . . .	37
4.2.1	Asignación de pesos. . . . .	38
4.2.2	Condiciones de paro. . . . .	39
4.2.3	Umbral de pausa. . . . .	39
4.2.4	Grado de similitud . . . . .	40
4.3	Modelo para la predicción . . . . .	40
<b>5</b>	<b>Capítulo 5: Resultados y discusión</b>	<b>44</b>
5.1	Aplicación del modelo propuesto. . . . .	44
5.2	Prueba de predicción 1: 100-427 . . . . .	44

5.3	Prueba de predicción 2: 150-377 .....	47
5.4	Prueba de predicción 3: 200-327 .....	49
5.5	Prueba de predicción 4: 235-292 .....	51
5.6	Prueba de predicción 5: 250-277 .....	53
5.7	Prueba de predicción 6: 422-105 .....	55
5.8	Discusión .....	58
5.8.1	Discusión: histórico de datos. ....	58
5.8.2	Discusión: valores iniciales. ....	61
5.8.3	Discusión: banco de datos .....	62
<b>6</b>	<b>Capítulo 6: Conclusiones y trabajo a futuro</b>	<b>64</b>
6.1	Conclusiones. ....	64
6.2	Trabajo a futuro .....	65
	<b>Referencias bibliográficas</b>	<b>66</b>

# Índice de figuras

2.1	Procesos de una <i>PDAR</i> . . . . .	5
3.2.2.1	Parte 1 del clasificador Gamma: Pre-procesamiento . . . . .	24
3.2.2.2	Parte 2 del clasificador Gamma: Funcionamiento del algoritmo. . . . .	25
4.1.1	Serie de tiempo del rasgo 24 ó <i>BOD</i> . . . . .	34
4.3.1	Comportamiento de la planta depuradora en los años 1990-1991 . . . . .	41
4.3.2	Definición del conjunto fundamental y del conjunto de prueba. . . . .	42
5.2.1	Conjuntos fundamental y de prueba para la prueba de predicción 1. . . . .	45
5.2.2	Predicción realizada con el clasificador Gamma para prueba 1. . . . .	45
5.3.1	Conjuntos fundamental y de prueba para la prueba de predicción 2. . . . .	47
5.3.2	Predicción realizada con el clasificador Gamma para prueba 2. . . . .	48
5.4.1	Conjuntos fundamental y de prueba para la prueba de predicción 3. . . . .	49
5.4.2	Predicción realizada con el clasificador Gamma para prueba 3. . . . .	50
5.5.1	Conjuntos fundamental y de prueba para la prueba de predicción 4. . . . .	52
5.5.2	Predicción realizada con el clasificador Gamma para prueba 4. . . . .	52
5.6.1	Conjuntos fundamental y de prueba para la prueba de predicción 5. . . . .	54
5.6.2	Predicción realizada con el clasificador Gamma para prueba 5. . . . .	54
5.7.1	Conjuntos fundamental y de prueba para la prueba de predicción 6. . . . .	56
5.7.2	Predicción realizada con el clasificador Gamma para prueba 6. . . . .	57
5.8.1.1	Gráfica de rendimiento del clasificador Gamma. . . . .	59
5.8.1.2	Gráfica de rendimiento del Multilayer Perceptron. . . . .	59
5.8.1.3	Gráfica de rendimiento del Bayes Net. . . . .	59
5.8.1.4	Gráfica de rendimiento de la Maquina de Vector Soporte . . . . .	60
5.8.1.5	Gráfica de rendimiento del Naive Bayes. . . . .	60
5.8.1.6	Gráfica de rendimiento del RBF Network . . . . .	60
5.8.2	Comportamiento del clasificador Gamma respecto del grado de similitud. . . . .	62

# Índice de tablas

3.1.1.1	Operador Alfa .....	17
3.1.1.2	Operador Beta .....	17
3.1.4	Codificación obtenida para el ejemplo 3.1.4. ....	19
3.3.1.1	Descripción de atributos. ....	30
3.3.1.2	Descripción de clases. ....	31
4.1.2	Clases que van emplearse en el proceso de predicción. ....	37
4.2.1	Rasgos más importantes del banco de datos .....	38
4.2.2	Pesos asignados a cada rasgo. ....	38
5.2.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 1 .....	46
5.3.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 2 .....	48
5.4.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 3 .....	50
5.5.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 4 .....	53
5.6.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 5 .....	55
5.7.1	Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis <i>Weka</i> para la prueba de predicción 6 .....	57

# Glosario de términos

## Términos generales:

- PDAR** - Planta Depuradora de Aguas Residuales.
- Weka** - Waikato Environment for Knowledge Analysis.

## Términos en los que trabaja una PDAR:

- BOD** - Demanda Biológica de Oxígeno.
- COD** - Demanda Química de Oxígeno.
- SS** - Sólidos Suspendidos.
- TSS** - Total de Sólidos Suspendidos.
- TN** - Total de Nitrógeno.
- pH** - Potencial de Hidrógeno.
- Zn** - Zinc.
- SSV** - Sólidos Suspendidos Volátiles.
- SED** - Sedimentos.
- COND** - Conductividad.
- MLSS** - Sólidos Suspendidos en Licor Mezclado.
- VFA** - Ácido Volátil Graso.
- P** - Fosforo.
- NH<sub>4</sub>** - Amonio.
- IVLA** - Índice de Volumen de Lodos Activos.

## Términos de los modelos matemáticos empleados para la predicción:

- RNA** - Redes Neuronales Artificiales.
- RB** - Redes Bayesianas.
- ACP** - Análisis por Componentes Principales.
- MCP** - Mínimos Cuadrados Parciales.
- DVS** - Descomposición de Valor Singular.
- SAIND** - Sistemas Adaptativos de Inferencia Neuro-Difusos.

- SND** - Sistemas Neuro-Difusos.
- SID** - Sistemas de Inferencia Difusos.

**Términos utilizados por Weka:**

- Multilayer Perceptron** - Perceptrón Multicapa.
- RBFBNetwork** - Red Neuronal de Funciones de Base Radial.
- BayesNet** - Red Bayesiana.
- NaiveBayes** - Bayes Ingenuo.
- RandomTree** - Árboles Aleatorios.
- REPTree** - Árboles Basados en Reducción de Poda de Error.
- SVM** - Máquina de Soporte Vectorial.
- IBk** - Instancia Basada: Implementación del algoritmo  $k$  vecinos más cercanos.

# Capítulo 1

## Introducción

En este trabajo de tesis se realiza la predicción de las condiciones que presenta una planta depuradora de aguas residuales mediante el uso de algoritmos inteligentes de reconocimiento y predicción de patrones provenientes del reciente enfoque asociativo Alfa-Beta.

### 1.1 Antecedentes

Las aguas residuales o aguas contaminadas son aquellas que tienen incorporados agentes y productos de desecho ajenos a la composición básica del agua. En México las principales fuentes de aguas residuales provienen de uso doméstico, urbano y de desechos industriales; éstas constituyen el 90% de todas las aguas residuales del país [1]. El tratamiento inadecuado de las mismas puede volverse un problema tanto ambiental como de salud para la población en general [1], [11], [15], [18], [20], [24], [26].

En la actualidad, las plantas depuradoras de aguas residuales (*PDAR*) son necesarias para la vida diaria de las personas, ya que su importancia radica en que proporcionan un servicio básico y un equilibrio entre la acumulación de aguas residuales y el medio ambiente en el que vivimos [2]. En México, un tratamiento adecuado de estas aguas, más otros servicios públicos, genera un bienestar social, el cual conlleva a ser una pieza fundamental de sustentabilidad en el uso correcto del agua dentro del territorio nacional [3].

Dentro de una *PDAR* el tratamiento de aguas residuales, es un proceso complejo en el que intervienen varias combinaciones de factores físicos, químicos y biológicos, los cuales pueden determinar las condiciones de trabajo de la misma y que pueden variar de una a otra, dependiendo del tipo de comunidad que se trate y el estilo de vida que se lleve [4], [8], [18]. Para lograr lo anterior, es necesario realizar una caracterización de la composición de las aguas residuales en función de algunos términos de concentraciones en los que labora la *PDAR*, entre ellos se pueden encontrar: la demanda biológica de oxígeno (*BOD*), la

demanda química de oxígeno (*COD*) y la acidez del potencial de hidrógeno (*pH*), entre otros más [1].

Durante los últimos años se ha venido trabajando el concepto de optimizar el funcionamiento de una *PDAR* con la finalidad de que opere siempre en las mejores condiciones. Entre las formas que logran la tarea de optimizar, están el monitoreo, la clasificación, el control automatizado y la predicción [14]. El presente trabajo propone un modelo para realizar la predicción de las condiciones de trabajo de una *PDAR*.

Haciendo referencia a la literatura especializada, existen varias propuestas que implementan un modelo de predicción empleando diferentes enfoques de orden matemático, entre los que se encuentran:

- Las redes neuronales artificiales [7], [8], [11], [12], [14], [15], [18], [19], [20], [21], [22], [23], [25], [26].
- Las redes Bayesianas [9], [41].
- La regresión [10],
- El análisis por componentes principales [12].
- La técnica de mínimos cuadrados parciales [13], [15].
- El modelo de regresión polinomial multivariada [15].
- Los sistemas adaptativos de inferencia neuro-difusos [12], [16], [19], [27].
- El análisis por agrupamiento (*clustering*) [10], [17].

De la lista anterior existe un modelo cuyo enfoque aún no ha sido tomado en cuenta para el contexto del trabajo actual y de acuerdo con la teoría es capaz de competir incluso con los mejores en su tipo. El enfoque a tratar es el asociativo y su máximo exponente radica en las memorias asociativas Alfa-Beta creadas en el 2002 en el Centro de Investigación en Computación del IPN [28]. Estas memorias en el 2007 sirvieron de inspiración para introducir un nuevo clasificador de alto desempeño desarrollado por investigadores del mismo centro y nombrado como el clasificador Gamma [29].

El propósito de esta tesis es emplear el enfoque asociativo Alfa-Beta con el clasificador Gamma para hacer un análisis de la predicción de las condiciones que pueden presentarse dentro de una *PDAR*.

## 1.2 Justificación

El agua es un elemento esencial para la vida, ya que con el paso de los años al ir en aumento el crecimiento de la población, conlleva en grado proporcional un mayor uso del vital líquido. Esto hace que la responsabilidad sobre la forma de emplear el agua afecta no sólo a las personas sino al medio ambiente en el que viven.



México es un país preocupado por el uso y depuración del agua, lo cual, ha propiciado la creación de un número considerable de plantas depuradoras en todo el territorio nacional. Para el año 2010 se tiene un registro de 2186 plantas municipales en operación cuya capacidad suma el 44.8% en el tratamiento de las aguas residuales provenientes de los sistemas municipales de alcantarillado del país [5].

Con base en el dato anterior se hace énfasis para presentar una propuesta que pueda servir como herramienta de apoyo para realizar la predicción de las condiciones que presenta una *PDAR* en un momento específico y a partir de la información procedente de las diferentes combinaciones que yacen contenidas en sus procesos operativos. Así mismo poder incidir en la toma de decisión que evalúe el comportamiento y mejore el desempeño de la *PDAR*.

### 1.3 Objetivo general

Desarrollar, implementar y evaluar un modelo basado en algoritmos de cómputo no convencional, tomados del enfoque asociativo Alfa-Beta, para la predicción de las condiciones de trabajo que pueden presentarse dentro de una *PDAR*.

### 1.4 Objetivos específicos

1. Hacer una revisión detallada al estado del arte y describir de manera general los diferentes modelos utilizados para la predicción de las condiciones que presenta una *PDAR*.
2. Revisar la estructura del algoritmo seleccionado y llevarlo a la práctica haciendo uso de un banco de información correspondiente a una *PDAR*.
3. Realizar la experimentación del algoritmo a fin de obtener resultados y establecer un alcance con el objetivo general.
4. Comparar los resultados obtenidos con otros mencionados en el estado del arte.

### 1.5 Contribuciones

Las aportaciones que este trabajo realiza sobre el tema de estudio son las siguientes:

Elaborar un análisis de rendimiento para la predicción de las condiciones que presenta una *PDAR* empleando el algoritmo del clasificador Gamma y cuyo enfoque asociativo Alfa-Beta aún no ha sido utilizado en el campo de estudio del presente documento.

Maximizar el rendimiento con el clasificador Gamma realizando las modificaciones pertinentes al modelo Alfa-Beta seleccionado para predecir las condiciones de una *PDAR*.

## **1.6 Organización del documento**

El trabajo de tesis está distribuido de la siguiente manera:

En este primer capítulo se encuentra la descripción de los antecedentes, la justificación, los objetivos, la contribución que aporta el trabajo, así como la organización del documento.

En el capítulo 2 se presenta el estado del arte que da a conocer de manera general la investigación que se realizó en la predicción de las condiciones de una *PDAR*.

El capítulo 3 presenta todas las herramientas (materiales y métodos) que van a ser utilizadas para el desarrollo del presente documento.

El cuarto capítulo, base del trabajo a presentarse, describe, sustenta y ejemplifica de forma teórica el modelo a proponerse con base en el algoritmo a utilizarse.

El quinto capítulo desarrolla el modelo del capítulo anterior y presenta resultados de experimentos realizados en comparación con el desempeño de otros clasificadores.

El capítulo final presenta las conclusiones, las aportaciones y el trabajo a futuro que se propone desarrollar.

Un apartado extra incluye todas las referencias utilizadas para este trabajo.

## Capítulo 2

# Estado del arte

La predicción de las condiciones de una *PDAR* ha sido un tema de interés actual y su importancia ha tenido trascendencia internacional, esto ha generado una gran variedad de trabajos que buscan mejorar las condiciones que presenta una planta depuradora con base en la certeza de tener una buena predicción. Para este capítulo se tienen dos secciones, la primera describe de manera general la forma de trabajo de una *PDAR* y la segunda enumera los varios modelos de inteligencia artificial que describen diferentes enfoques que permiten implementar la predicción.

### 2.1 Planta depuradora de aguas residuales

El funcionamiento de una *PDAR*, en general, sigue el esquema presentado en la figura 2.1. Esta figura describe el proceso biológico conocido como de *lodos activados*.

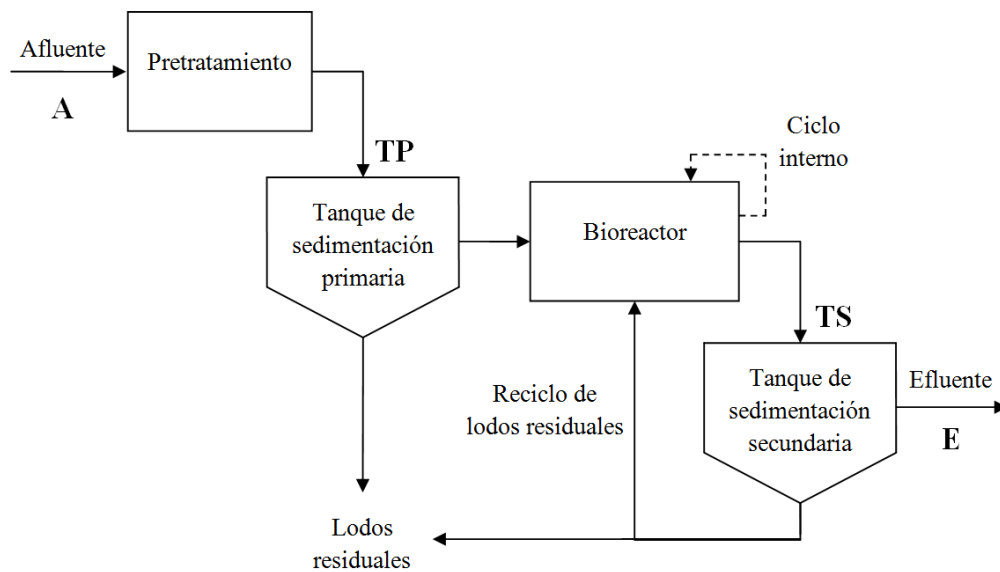


Figura 2.1: Procesos de una *PDAR*.

El proceso de *lodos activados* es el más utilizado en el tratamiento de aguas residuales y consiste en transformar el contaminante biodegradable en una biomasa; dicha biomasa se crea usando una mezcla de varios microorganismos dentro de un *bioreactor* que incorpora oxígeno disuelto por algún sistema de aeración y el cual contiene aguas residuales tratadas en una primera fase [53].

El pre-tratamiento consiste en recibir el afluente (A) o entrada de las aguas hacia la planta para remover sólidos suspendidos mayores y pesados por medio de rejillas, desmenuzadores o cernedores. En el primer tanque de sedimentación (TP) se eliminan aquellos sólidos que no pudieron quitarse en el pre-tratamiento al asentarse estos en el fondo del tanque. Después, el flujo de agua es tratada en un *bioreactor*, donde por la acción de microorganismos los niveles de sustratos son reducidos; es en esta etapa donde se desarrolla el proceso de *lodos activados*. Ahora el agua fluye al segundo tanque de sedimentación (TS), donde la biomasa creada previamente se asienta haciendo que el agua limpia permanezca por encima del sedimento; el agua tratada es llevada hacia afuera de la planta como efluente (E). Una fracción de los lodos es regresada a la entrada del *bioreactor*, cuya finalidad es mantener niveles apropiados de biomasa y permitir la oxidación de la materia orgánica. El resto de los lodos es purgado [53].

## 2.2 Modelos de inteligencia artificial

### 2.2.1 Redes neuronales artificiales

Básicamente, las redes neuronales artificiales (*RNA*) son modelos matemáticos inspirados en el proceso biológico de aprendizaje y característico del cerebro humano cuyos elementos básicos de procesamiento, las neuronas [40], están interconectadas conforme una topología específica (arquitectura) organizada en capas, y con la capacidad de auto-modificar el valor de los enlaces (pesos) que pertenecen a los parámetros de los elementos procesados (aprendizaje) [8], [11], [12], [14], [15], [20], [22], [23].

Las *RNA* tienen una ventaja muy importante respecto de otros modelos ya que pueden ser utilizadas tanto para modelar sistemas lineales como no lineales. En el presente contexto, las *PDAR* son sistemas no lineales y altamente variables con el tiempo [11], [18], [20], esto hace que las *RNA* sean convenientes para tratar la complejidad de las condiciones en que se encuentra una planta depuradora.

De acuerdo con los diferentes artículos revisados, la predicción se realiza con base en determinados términos de concentraciones que miden el grado de contaminación en que se encuentra el agua. En [8], [23] y [25], la predicción en el efluente, o a la salida de la planta, se da en términos de la demanda biológica de oxígeno (*BOD*), la demanda química de oxígeno (*COD*) y el total de sólidos suspendidos (*TSS*). Estas mediciones son las de mayor

uso en la literatura disponible [8]. En [20] la tarea se extiende al agregar una concentración más que es el total de nitrógeno (*TN*). En [14] y [22] se sustituye el *BOD* por otro término de medición, sólidos suspendidos en licor mezclado (*MLSS*) y producción de biogás respectivamente. Los trabajos mencionados cumplen el objetivo principal con resultados muy aceptables y con la característica de que a la entrada de los respectivos modelos ocupan tres o más términos de concentraciones del afluente para realizar la predicción.

Otras referencias como [7], [11], [12], [15] y [18] enfocan su esfuerzo en realizar la predicción en uno o dos términos máximo: *COD* y ácido volátil graso (*VFA*) en [7], *TSS* en [11], *COD* en [12], *BOD* y *COD* en [15] y *BOD* en [18]. En todos estos trabajos la predicción realizada es exitosa. De acuerdo con [8] y [23], la unión de varias concentraciones a la entrada de la respectiva topología de *RNA* genera mejores resultados de predicción que usando sólo algunas. De las referencias presentadas, y como ejemplo de lo comentado, [12] emplea, a la entrada de su modelo de *RNA*, 10 términos de concentraciones para realizar la tarea de predicción; otros trabajos suman más o menos términos en el modelo de su respectiva topología.

En [19] y [21] se contempla una consecuencia directa de los procesos de depuración: la obtención de energía en forma de gas metano. En [19] la predicción se realiza en la producción de dicho gas y en [21] se realiza con la finalidad de obtener la máxima producción del mismo. Tanto en [19] como en [12] la utilización de las *RNA* no es tan buena como en otros trabajos ya que el rendimiento de lo que se quiere predecir es superado por el enfoque neuro-difuso.

Un caso extra se tiene con [26]. En este trabajo se realiza la predicción de *BOD* y *COD* en el efluente que descarga una mina de carbón. A diferencia de otros trabajos la recolección de datos no es mediante sensores dentro de una *PDAR* sino de mediciones en el lugar donde el efluente de las aguas es descargado. El uso de las *RNA* al igual que en otros trabajos anteriores es muy satisfactorio.

Las *RNA* pueden ser apoyadas con métodos de pre-procesamiento de datos que ayudan a mejorar los resultados de la predicción. En [7] se utiliza la regresión lineal que muestra una correlación cercana entre los términos de entrada de la *RNA*. En [11] se hace uso de las *wavelets* que son las responsables de la extracción de características de los datos que van a ser usados en la *RNA*. En [12] se utiliza el análisis de componentes principales (*ACP*) para reducir al mínimo los términos de entrada de la *RNA* y finalmente en [18] se utiliza la interpolación entre los datos para estimar valores faltantes.

De toda la literatura consultada el mayor empleo de las *RNA* es utilizando el modelo topológico perceptrón multicapa, de alimentación hacia delante (*Feed Forward*) con el algoritmo de entrenamiento basado en la propagación del error hacia atrás (*Back*

*Propagation*). No obstante en [22] se hacen uso de las *RNA* de funciones de base radiales y de las *RNA* de regresión generalizada. En este trabajo la tarea de predicción se realiza a la par con las *RNA* perceptrón multicapa, siendo las últimas las que mejores resultados generan.

### 2.2.2 Redes bayesianas

Teóricamente, las redes bayesianas (*RB*) son modelos gráficos que permiten representar y establecer las relaciones de probabilidad entre un conjunto de variables. El modelo matemático se basa estrictamente en datos numéricos cuya representación gráfica es por medio de grafos acíclicos dirigidos. Forma parte del enfoque probabilístico-estadístico y algunas veces es conocido como modelo de Bayes, red probabilística o mapa del conocimiento [9].

Las *RB* ofrecen fuertes argumentos para su uso. Se destaca la estricta fundamentación en la teoría de la probabilidad y el fácil entendimiento de los resultados del diagnostico [9]. Otros puntos a favor de las *RB* es que pueden manejar conjuntos de datos incompletos, así como facilitar la combinación de conocimiento de datos y dominio en conjunto con técnicas estadísticas bayesianas. Finalmente ofrecen un enfoque eficaz para evitar el sobreajuste de datos en relación con otro tipo de modelos [42].

En [9] el modelo propuesto de *RB* expone dos objetivos. El primero es para predecir fosforo ( $P$ ), amonio ( $NH_4$ ) y estados de nitrato en el efluente, y el segundo es para diagnosticar el estado de los procesos reales o predichos de la planta con los datos censados. El trabajo muestra resultados prometedores tanto en los términos a predecir como en el diagnóstico de la planta, reduciendo las alarmas de advertencia/precaución en cerca de un 33%.

Las *RB* también pueden ser usadas en otros ámbitos como es el caso de [41] que realiza la predicción de ozono ( $O_3$ ) en el ambiente. Este trabajo compara el uso de árboles de decisión con las *RB* reportando que el enfoque probabilístico es el que mejores resultados genera.

### 2.2.3 Regresión

La regresión es una herramienta matemática estadística que permite investigar la relación entre variables de un conjunto de datos; el objetivo es identificar la mejor combinación de variables que representen de manera discreta el conjunto de datos en función de un indicador comprensible que ayude a establecer la calidad del modelo. El uso de las técnicas de regresión permite realizar tareas de pronóstico y predicción [10], [31].

En [10] se utilizan dos modelos de regresión: lineal y no lineal, para predecir las concentraciones de *COD* y *VFA*. El modelo lineal ocupa el cuadrado de la función de correlación de Pearson ( $R^2$ ) como indicador para relacionar las variables o los términos de concentraciones utilizadas. Los resultados arrojan una buena predicción en *COD* pero en *VFA* la predicción no resultó ser tan buena. En cuanto al modelo no lineal, son utilizadas las *RNA* y en comparación con el modelo lineal, los resultados tanto en *COD* como en *VFA* tienden a ser similares, esto es, el rendimiento de la predicción es mejor en términos de *COD*.

### 2.2.4 Análisis por componentes principales

El análisis por componentes principales (*ACP*) es un método para reducir la dimensión de un conjunto de datos identificando las correlaciones lineales entre variables aleatorias. Puede ser obtenida mediante métodos matemáticos como la descomposición de valor singular (*DVS*) y la no linealidad iterativa de mínimos cuadrados parciales (*MCP*). El *ACP* funciona como una proyección para reducir el conjunto original de variables a una variable latente o un componente principal. Dentro de la práctica, sólo unos pocos componentes principales son a menudo suficientes para explicar las variaciones de los datos [12].

En [12] la utilización del *ACP* para la predicción de *COD* sirve para reducir la cantidad de parámetros utilizados inicialmente. De los 10 parámetros de entrada, al implementar el *ACP*, el conjunto fue reducido a 3 componentes principales y a una variable latente. Lo anterior posibilita la mejor comprensión entre las variaciones de todo el conjunto de datos. Finalmente en combinación con un modelo de *RNA* los resultados de la predicción son aceptables pero están por debajo de los resultados de rendimiento que se obtienen al combinar el *ACP* con el enfoque neuro-difuso.

### 2.2.5 Mínimos cuadrados parciales

El modelo de mínimos cuadrados parciales es una técnica lineal de calibración multivariante mediante la extracción de combinaciones lineales de las características del conjunto de datos, permitiendo encontrar las relaciones entre el conjunto de datos original independiente  $X$  y el conjunto de datos de trabajo dependiente  $Y$  [13], [15]. El uso de esta técnica ha sido empleada como un enfoque eficiente en el monitoreo de procesos complejos, ya que el conjunto de datos original puede ser reducido a un conjunto mucho más pequeño de variables latentes para su mejor interpretación [13].

El trabajo mostrado en [13] expone la tarea de predicción en términos del índice de volumen de lodos activos (*IVLA*) generando resultados de correspondencia razonables al utilizar un coeficiente de correlación que se obtiene entre los valores observados y los

valores predichos. En este trabajo se hace uso del análisis de imagen para la extracción de características y generar el conjunto de datos con los que se va a trabajar.

En [15] el objetivo de predecir se realiza en términos de *BOD* y *COD*. Los resultados del trabajo muestran estimaciones buenas del efluente tanto para *BOD* como para *COD* en base al coeficiente de correlación obtenido. El modelo lineal descrito compite en este trabajo con modelos no lineales. Los resultados de los modelos no lineales, como lo son las *RNA*, dan mejores resultados que los obtenidos con la técnica de *MCP*.

### 2.2.6 Regresión polinomial multivariada

El modelo de regresión polinomial multivariada es un modelo no lineal de orden inferior que maximiza la covarianza entre el conjunto de datos originales *X* (independiente) y el conjunto de datos de trabajo *Y* (dependiente) y cuya finalidad es explicar las mejores estimaciones del conjunto *Y*. De la misma manera que en *APC* y *MCP* este modelo utiliza coeficientes de correlación que explican la relación entre las variables [15].

En [15] la predicción en el efluente se da en términos de *BOD* y *COD*. Se utilizan coeficientes de correlación estimados por el método de mínimos cuadrados. Este enfoque entrega buenos resultados en los términos de concentraciones que se quieren predecir, siendo el *BOD* el término que mejor rendimiento proporciona. Este trabajo compite con el modelo lineal *MCP* y el modelo no lineal de *RNA*.

En general, los resultados de rendimiento finales de [15] pueden ser comparados casi de igual manera con los obtenidos por las *RNA* estando por encima del modelo realizado con *MCP*.

### 2.2.7 Sistemas adaptativos de inferencia neuro-difusos

Los sistemas adaptativos de inferencia neuro-difusos (*SAIND*) son sistemas que están compuestos por dos partes: los sistemas neuro-difusos (*SND*) y los sistemas de inferencia difusa (*SID*). Los *SND* combinan ventajas tanto de redes neuronales como de sistemas difusos. La arquitectura del modelo consiste en emplear por un lado la lógica difusa para representar el conocimiento de manera interpretable y por otro la capacidad de aprendizaje de la red neuronal. Los *SID* proveen una solución para la toma de decisión basada en información ambigua, imprecisa y faltante. La lógica difusa aporta las reglas de tipo condicional (si-entonces) para definir parámetros tanto de entrada como de salida. La inferencia difusa depende de los parámetros estimados anteriormente y al utilizar reglas difusas, funciones de pertenencia y fusificación, y operaciones de defusificación permite producir una salida que es fácil de ser entendida e interpretada [27].



Con lo anterior, los *SAIND*, son modelos de red neuronal perceptrón multicapa donde los nodos de la red son adaptativos, esto significa que cada salida depende de los parámetros relativos de entrada en los nodos, así como una regla de aprendizaje que especifica la actualización de los parámetros para minimizar una medida de error prescrita. La combinación entre *SND* y *SID* constituyen un modelo que permite interpretar datos además de ser capaz de aprender con base en el conocimiento previo de un problema dado [27].

En el trabajo presentado en [12], la predicción en el efluente se realiza en términos de *COD*. El resultado de rendimiento del modelo de predicción es superior al propuesto por el enfoque neuronal, siendo el *SAIND* superior por 5 y 1.1 veces en entrenamiento y validación respectivamente. El modelo es complementado con un pre-procesamiento por *ACP* que acentúa el rendimiento del modelo.

En [16], la tarea de predecir se elabora en términos de *COD*, *SS* y *TN*. En este trabajo se utilizan solamente los *SID* empleando reglas condicionales para los parámetros de entrada del modelo. Esto permite predecir y clasificar con una alta precisión el estado del efluente así como la identificación visual de las razones de calidad del mismo.

La predicción realizada en [19] se basa en términos de producción de gas metano. En este artículo el modelo de *SAIND* compite contra otros 4 enfoques como las *RNA* y las máquinas de soporte vectorial, por nombrar algunos, siendo el *SAIND* el que mejor rendimiento de predicción tiene frente a las otras métricas utilizadas. Tanto en este trabajo como en [16] se emplean árboles de decisión para extraer información y generar las reglas condicionales que sirven de entrada al modelo utilizado.

Finalmente en [27] el objetivo de predecir es implementado en términos de la tasa de dosificación de coagulante. En este trabajo se utilizan dos modelos de *SAIND*, uno enfocado en partición de red (*GRID*) y el otro en clúster substractivo (*SUB*). Los resultados indican que la predicción más precisa y confiable está en el modelo *SAIND-SUB* ya que los coeficientes de correlación son muy cercanos a la unidad en la mayoría de los casos. El resultado de este y los otros trabajos muestran que el enfoque neuro-difuso es una buena herramienta para el modelado de sistemas complejos y no lineales.

## **2.2.8 Análisis por agrupamiento (*clustering*)**

El análisis por agrupamiento (*clustering*) es fundamental en el área de minería de datos que a diferencia de la tarea de clasificación utiliza un aprendizaje no supervisado. El objetivo del *clustering* es describir el conjunto original de datos en subconjuntos de los mismos, agrupados de manera tal que casos de instancias similares se junten, mientras que

las distintas instancias pertenecen a grupos diferentes para poder interpretar mejor el análisis que se está realizando [43].

El trabajo presentado en [10] realiza el *clustering* no sólo para identificar grupos coherentes de datos en el espacio de proceso multidimensional, sino para crear una simple variable representativa de los mismos. Así mismo se utiliza una regla de inducción que asocia los rangos de las múltiples variables para predecir diferentes valores de una variable simple seleccionada.

Los resultados obtenidos en [10] son variados ya que los valores de entrada varían tanto en la cantidad de datos como en términos utilizados para la formación de clústeres. No obstante en general todos los resultados están por encima del 65% de predicción correcta llegando incluso en algunos casos al 100% de rendimiento en la predicción.

En [17] el trabajo presenta un enfoque de *clustering* de dos etapas. En la primera etapa consiste en un mapa auto-organizado el cual se usa para generar un pequeño pero representativo subconjunto de datos del conjunto original. En la segunda etapa el conjunto de datos reducido es agrupado jerárquicamente por el algoritmo de aglomeración de Ward cuyo objetivo es obtener dos clústeres altamente relacionados en función con su varianza.

La tarea de predicción en [17] se enfoca en detectar y distinguir las diferentes composiciones de aguas residuales en el desagüe con la finalidad de conocer el origen de las mismas. Los resultados del trabajo son muy buenos ya que se tiene un aproximado del 98% en los términos de predicción.

## 2.2.9 Minería de datos

Una de las herramientas en común y en la que están basados los enfoques descritos anteriormente es la que proporciona la minería de datos. La minería de datos sirve de apoyo para la extracción de conocimiento presente en bancos de información que es, de cierta manera, la materia prima con la que se realizan los trabajos descritos [47]. Estos bancos son una representación abstracta y descriptiva de las condiciones en las que se encuentra una planta depuradora de aguas residuales. Estos bancos después de ser procesados van a entregar un resultado que va a servir de apoyo en la toma de decisión que en el actual contexto es respecto del estado de una *PDAR*.

El uso de la minería de datos tiene un alcance bastante amplio debido a que permite, dependiendo de las características del banco de información a utilizar, realizar varias tareas como pueden ser el monitoreo del estado de la planta, el monitoreo de los sensores que utiliza, la clasificación del estado de las aguas, el control de los procesos de depuración y como anteriormente se describió, la predicción [14].

En la mayoría de los trabajos descritos [7], [8], [10], [11], [13], [14], [19], [17], [18], [16] y [20], el objetivo principal consiste en proponer un modelo de predicción. Dicho modelo, de cada una de las referencias expuestas, utiliza un historial de información compuesto de datos relevantes que describen el comportamiento de las condiciones en las que se encuentra una *PDAR* a fin de entregar un análisis que muestre el rendimiento del modelo propuesto. También hay algunos trabajos [6], [9] y [25] que además de proponer un modelo, con su respectivo análisis, proponen una aplicación en la cual los resultados son mostrados de manera más amigable al usuario que le compete dicha información.

Para el actual trabajo se tiene un referente importante que es el proyecto *TELEMAC* (*TELEMonitoring and Advanced teleControl*) [44]. *TELEMAC* fue un proyecto que reunió a cerca de 15 organizaciones a través de Europa y América Latina, México, para dar seguimiento y mejorar el control de las plantas de tratamiento de aguas residuales perteneciente al sector industrial de bebidas alcohólicas [6].

La finalidad fue desarrollar un sistema confiable de monitoreo remoto para el control de plantas depuradoras sin presencia de experiencia local. La idea general es que una red de expertos en el tema del tratamiento de aguas residuales brinde asistencia remota a técnicos de alguna planta local para que juntos tomen decisiones que lleven al buen funcionamiento de la planta depuradora con auxilio de información, previamente recopilada, que pueda ser consultada, comparada y verificada [6], [7], [10].

Un aspecto importante del proyecto *TELEMAC* es el acceso a la información del estado de las plantas involucradas a través de una red de comunicación vía acceso a internet. Es aquí donde interviene la minería de datos al permitir que el conocimiento sea derivado y a la vez compartido del monitoreo de información del estado, presente o pasado, de una o múltiples plantas involucradas [6], [10], [44].

Los trabajos [7] y [10] utilizan algunos bancos de información que son consecuencia directa del conocimiento derivado en forma de bancos de datos del proyecto *TELEMAC*.

## **2.2.10 Control automático**

Un enfoque final y que fue tocado de manera indirecta en el apartado anterior es el control automático. El objetivo de este enfoque es mantener las variables de operación de un proceso dado en un determinado valor, tales como la temperatura, la presión, los flujos, los compuestos, entre otros. Estas variables son de naturaleza dinámica y por tanto no lineal que si no se realizan las acciones pertinentes, dichas variables no cumplirán con las condiciones establecidas de un modelo propuesto [48].

Los trabajos [24] y [45] proponen un control automático para modificar y preservar las condiciones en las que se encuentra una planta depuradora.

La propuesta de [24] es utilizando el enfoque de inteligencia artificial mediante un control de agente inteligente, que a su vez está basado en el enfoque emergente de aprendizaje por refuerzo, el cual depende de la interacción con el medio en el que se utiliza. El trabajo fue aplicado en el proceso de eliminación de  $NH_4$ .

En [45] el trabajo propuesto consiste en una técnica predictiva adaptativa para controlar la salida de la concentración de sustrato contaminante. El modelo matemático para el control de un modelo simple de *PDAR* es obtenido en el contexto por estimación recursiva. Este trabajo posibilita gestionar cambios en un amplio rango de parámetros que se efectúan en los procesos de depuración de la planta.

En ambos trabajos se compite con el modelo de control tradicional, el cual requiere de presencia de un experto para que pueda llevarse a cabo. Las propuestas descritas realizan el control automático sin presencia de expertos en los procesos de depuración cumpliendo los objetivos planteados.

## 2.2.11 Enfoque asociativo

Este enfoque aún no ha recibido la oportunidad de ser utilizado para el contexto actual, no obstante, ha sido empleado en otros rubros con resultados bastante competitivos e incluso superiores a otros enfoques como las *RNA*, los *SAIND* o las máquinas de soporte vectorial.

El clasificador gamma, como digno representante del enfoque asociativo Alfa-Beta, ha tenido un recibimiento bastante bueno en tareas de predicción. Los trabajos [31] y [46] enriquecen la propuesta del algoritmo al realizar el análisis de predicción en contaminantes atmosféricos.

La siguiente línea de tiempo resalta los momentos más representativos del enfoque asociativo y que sustenta las bases para el desarrollo del clasificador Gamma:

- 1961: Primer modelo de memoria asociativa, por Karl Steinbuch: la Lernmatrix. Modelo que puede funcionar como clasificador de patrones binarios [32].
- 1969: Se da a conocer el Correlograph, por Willshaw, Buneman y Longuet-Higgins. Dispositivo que funciona como memoria asociativa y que trabaja para recuperar patrones binarios [33].
- 1972: Se presentan varios trabajos:
  - La Memoria Interactiva de James A. Anderson [34].

- 
- Las Memorias de Correlación Matricial de Teuvo Kohonen [35].
  - El Associatron de Kaoru Nakano [36].
- 1982: Se presenta la Memoria Hopfield, por John J. Hopfield. Es una memoria autoasociativa cuyo desempeño es pobre comparado a otros clasificadores actuales ya que en etapa de recuperación no rebasa la capacidad del 15% [37]. Con este trabajo se retoma el interés por el enfoque asociativo.
- 1988: Bart Kosko presenta la memoria asociativa bidireccional (*BAM*). Este modelo de memoria es de tipo heteroasociativa y fue basada en el modelo de 1982. Esta memoria gana notoriedad, no obstante la capacidad de recuperación es comparada a la de Hopfield [38].
- 1998: Ritter, Sussner y Díaz de León introducen la memoria asociativa morfológica. Esta memoria logra capacidades de aprendizaje mayores a los obtenidos hasta ese momento ya que su estructura está conformada por conceptos de morfología matemática [39].
- 2002: Se crean las memorias asociativas Alfa-Beta por investigadores del Centro de Investigación en Computación. Estas memorias utilizan los operadores Alfa-Beta para clasificar patrones binarios [28].
- 2007: El clasificador Gamma [29].

El presente documento propone extender el área del conocimiento para la implementación del clasificador Gamma en la predicción de las condiciones de una *PDAR* usando el enfoque asociativo Alfa-Beta.

## Capítulo 3

# Materiales y métodos

Para la elaboración del modelo a presentarse en el capítulo siguiente es necesario describir las herramientas que van a permitir la realización del trabajo. Para la primera sección de este capítulo se describen todos los elementos necesarios y con los cuales el clasificador Gamma puede funcionar. En la segunda sección se describe la estructura y funcionamiento del algoritmo. La tercera sección describe el banco de datos que abstrae el funcionamiento de una *PDAR*. Finalmente la cuarta sección describe el entorno de análisis *Weka* junto con algunos algoritmos de clasificación de patrones que serán utilizados para comparar la predicción frente a la obtenida con el clasificador Gamma.

### 3.1 Requerimientos para el clasificador

#### 3.1.1 Operadores Alfa ( $\alpha$ ) y Beta ( $\beta$ )

El empleo del ingenio dio paso a la formalidad expuesta en los operadores  $\alpha$ - $\beta$  y cuya definición y descripción se encuentra a detalle en [28]. Estos operadores generan y fundamentan el enfoque asociativo Alfa-Beta en todos sus modelos; el clasificador Gamma no es la excepción.

La manera de emplear los dos operadores es la siguiente: para la fase de aprendizaje es utilizado el operador Alfa, mientras que en la fase de recuperación se utiliza el operador Beta.

La definición de ambos operadores se muestra a continuación.

Dado el conjunto binario  $A = \{0, 1\}$  y el conjunto ternario  $B = \{0, 1, 2\}$ , se generan las siguientes tablas:

Tabla 3.1.1.1: Operador Alfa.

$\alpha: A \times A \rightarrow B$		
$x$	$y$	$\alpha(x,y)$
0	0	1
0	1	0
1	0	2
1	1	1

Tabla 3.1.1.2: Operador Beta.

$\beta: B \times A \rightarrow A$		
$x$	$y$	$\beta(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

### 3.1.2 Operador unario $u_\beta$

La definición detallada de este operador se puede encontrar en [29]. Su importancia radica en que es empleado dentro de la estructura del clasificador Gamma.

**Definición 3.1.2:** Sean: el conjunto  $A = \{0, 1\}$ , un número  $n \in \mathbb{Z}^+$  y  $x \in A^n$  un vector binario de dimensión  $n$ , con la  $i$ -ésima componente representada por  $x_i$ . Se define el operador  $u_\beta(x)$  de la siguiente manera:  $u_\beta(x)$  tiene como argumento de entrada un vector binario  $n$ -dimensional  $x$  y la salida es un número entero no negativo que se calcula así:

$$u_\beta = \sum_{i=1}^n \beta(x_i, x_i)$$

A continuación se muestra el uso del operador con unos ejemplos.

**Ejemplo 3.1.2:** Sea  $x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ ; obtener  $u_\beta(x)$ :

Haciendo uso de la definición 3.1.2.1 se tiene  $u_\beta = \sum_{i=1}^5 \beta(x_i, x_i)$ , por lo que desarrollando obtenemos  $u_\beta(x) = \beta(1,1) + \beta(0,0) + \beta(1,1) + \beta(1,1) + \beta(0,0) = 1 + 0 + 1 + 1 + 0 = 3$ . Entonces  $u_\beta(x) = 3$ .

### 3.1.3 Módulo

Dentro del concepto de la división existen circunstancias donde resulta de mayor y mejor interés el uso del residuo (entero) en vez del cociente o resultado de la operación. El concepto de módulo formalmente puede encontrarse en [49] donde el operador *módulo* está denotado por *mod*.

**Definición 3.1.3:** Sea  $a$  un entero y  $m$  un entero positivo. Se denota por  $a \bmod m$  al residuo de dividir  $a$  por  $m$ .

En otras palabras,  $a \bmod m$  es el número entero  $r$  tal que  $a = qm + r$  y  $0 \leq r < m$ .

**Ejemplo 3.1.3:** El módulo de  $46 \bmod 6 = 4$ .

El uso de este concepto junto con el de la sección anterior son de gran importancia para el desarrollo del clasificador Gamma.

### 3.1.4 Código binario Johnson-Möebius modificado

Uno de los requisitos para el funcionamiento del clasificador Gamma es que los datos en forma de vectores de entrada estén codificados de cierta manera. La codificación utilizada es la que proporciona el código Johnson-Möebius modificado. Dicha codificación está basada en los trabajos [50] y [51].

**Algoritmo 3.1.4:** Código Johnson-Möebius modificado.

1. Sea un conjunto de números reales:

$$\{r_1, r_2, \dots, r_i, \dots, r_n\},$$

donde  $n$  es un número entero positivo fijo.

2. Si algún elemento del conjunto es negativo (por ejemplo  $r_i$ ), crear un nuevo conjunto transformado a través de la siguiente operación: restar  $r_i$  a cada uno de los  $n$  elementos del conjunto.

El conjunto transformado queda de la siguiente forma:

$$\{t_1, t_2, \dots, t_i, \dots, t_n\},$$

donde  $t_j = r_j - r_i \forall j \in \{1, 2, \dots, n\}$  y particularmente  $t_i = 0$ .

Nota: En caso de existir más de un elemento negativo, entonces, trabajar con el mayor de todos.

3. Seleccionar un número fijo  $d$  de decimales para truncar cada uno de los elementos del conjunto transformado a  $d$  decimales. El conjunto transformado no contiene elementos negativos.
4. Realizar un escalamiento de  $10d$  en el conjunto del paso anterior, para obtener un conjunto de  $n$  enteros no negativos. El nuevo conjunto de enteros positivos es:

$$\{e_1, e_2, \dots, e_i, \dots, e_m, \dots, e_n\},$$

donde  $e_m$  es el número mayor.

5. El código Johnson-Möebius modificado para cada  $j = 1, 2, \dots, n$  se obtiene al generar  $(e_m - e_j)$  ceros concatenados por la derecha con  $e_j$  unos.

El empleo del código se muestra en el siguiente ejemplo:



**Ejemplo 3.1.4:** Sea el conjunto  $r = \{2.4, 1.76, 0.99\}$ ;  $r \in \mathbb{R}$ . Se sigue el algoritmo descrito anteriormente:

**Paso 1:**  $r = \{2.4, 1.76, 0.99\}$ .

**Paso 2:** El conjunto  $r$  no tiene elementos negativos, entonces no se crea un nuevo conjunto transformado y se ocupa el mismo conjunto  $r$ .

**Paso 3:** Se selecciona  $d = 1$  para obtener:  $r = \{2.4, 1.7, 0.9\}$ .

**Paso 4:** Se escala cada elemento del conjunto  $r$  con  $10d$  y se obtiene:

$e = \{24, 17, 9\}$ , donde  $e_m = 24$ .

**Paso 5:** Para cada elemento de  $e$ , se generan  $e_m - e_i$  ceros concatenados con  $e_i$  unos.

1. Siendo que 24 es el número mayor, entonces cada número del conjunto será codificado con 24 bits.
2. Para el número  $e_1 = 24$ , se tienen  $24 - 24 = 0$  ceros concatenados de 24 unos.
3. Para el número  $e_2 = 17$ , se tienen  $24 - 17 = 7$  ceros concatenados de 17 unos.
4. Para el número  $e_3 = 9$ , se tienen  $24 - 9 = 15$  ceros concatenados de 9 unos.

La tabla 3.1.4 muestra los códigos correspondientes del conjunto  $e$ .

Tabla 3.1.4: Codificación obtenida para el ejemplo 3.1.4.

Número	Códigos Johnson-Möebius modificado
24	111111111111111111111111
17	000000011111111111111111
9	000000000000000111111111

El empleo de esta codificación se debe a la carencia de ruido combinado en cada código binario obtenido, esto es, o se utiliza ruido sustractivo o aditivo pero nunca ambos [50]. El clasificador Gamma emplea esta codificación para encontrar similitudes entre patrones de forma más exacta. En la sección siguiente se ahondará más a detalle su utilización.

## 3.2 Clasificador Gamma

Este modelo asociativo es un clasificador de alto desempeño presentado inicialmente en [29]. Hace uso de todos los elementos descritos en la sección 3.1 y añade un elemento más a su composición: el operador gamma ( $\gamma$ ) de similitud el cual es presentado igualmente en [29].

En esta sección se describe tanto el operador gamma ( $\gamma$ ) de similitud como el correspondiente algoritmo para el desarrollo del clasificador Gamma.

### 3.2.1 Operador Gamma ( $\gamma$ ) de similitud

De este operador resaltan las siguientes características:

1. Está basado en las operaciones fundamentales que dan lugar a las memorias asociativas Alfa-Beta (descritas en la sección 3.1.1).
2. El operador tiene la tarea de observar si dos vectores son parecidos o no, mediante un grado de disimilitud  $\theta$ , indicando la tolerancia en que al comparar los vectores sean considerados similares, no obstante que son diferentes.
3. El empleo de este operador permite trabajar con vectores binarios de dimensiones diferentes.
4. El nombre se le dio en base a los operadores Alfa-Beta, siendo gamma ( $\gamma$ ) la letra griega que está a continuación de las otras dos.

Formalmente el operador se define de la siguiente manera: [29]

**Definición 3.2.1:** Sean: el conjunto  $A = \{0, 1\}$ , un número  $n \in \mathbb{Z}^+$  y  $x, y \in A^n$  dos vectores binarios de  $n$ -dimensionales, con la  $i$ -ésima componente representada por  $x_i$  y  $y_i$ , respectivamente, y además,  $\theta$  un número entero no negativo. Se define el operador Gamma de similitud  $\gamma(x, y, \theta)$  de la siguiente manera:  $\gamma(x, y, \theta)$  tiene como argumentos de entrada dos vectores binarios  $n$ -dimensionales  $x$  y  $y$ , y un número entero no negativo  $\theta$ , y la salida es un número binario que se calcula así:

$$\gamma(x, y, \theta) = \begin{cases} 1 & \text{si } n - u_\beta[\alpha(x, y) \bmod 2] \leq \theta \\ 0 & \text{en otro caso} \end{cases}$$

A continuación se muestra un ejemplo que emplea el operador Gamma de similitud.

**Ejemplo 3.2.1:** Sean  $x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$ ,  $y = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$  y  $\theta = 3$ ; calcular  $\gamma(x, y, \theta)$ .

**Paso 1:** En este ejemplo se tiene que  $n = 4$ .

**Paso 2:** Calculando  $\alpha(x, y)$ , obtenemos:  $\begin{pmatrix} 2 \\ 0 \\ 2 \\ 1 \end{pmatrix}$

**Paso 3:** Al aplicar el módulo 2 a cada elemento del vector resultante en el paso

anterior, se obtiene el vector siguiente:  $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

**Paso 4:** Aplicando el operador  $u_\beta$  en el vector del paso 3, se obtiene:  $u_\beta = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 1$ .

Del resultado obtenido, se realiza la resta con  $n = 4$ :  $4 - 1 = 3$ ;  $3 \leq \theta = 3$  y por lo tanto  $\gamma(x, y, \theta) = 1$ .

### 3.2.2 Algoritmo del clasificador Gamma

El empleo del algoritmo se justifica con base en el trabajo presentado en [30] al caracterizarse la operación del clasificador y al definirse el conjunto fundamental ideal con el cual se realiza una recuperación correcta.

**Definición 3.2.2:** Sea el conjunto fundamental del clasificador Gamma el conjunto de patrones asociados a una clase, de la forma  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$ ; donde  $x^\mu$  es un patrón y  $y^\mu$  es su clase correspondiente. Además, para este conjunto fundamental se cumplen las siguientes tres afirmaciones:

$$x^i \neq x^j \quad \forall i, j \in \{1, 2, \dots, p\} \text{ tal que } i \neq j$$

*Esto implica que no hay patrones repetidos.*

$$x^i = x^j \Rightarrow y^i = y^j \quad \forall i, j \in \{1, 2, \dots, p\}$$

*Un patrón dado no puede tener asociada más de una clase.*

$$y^i = y^j \Rightarrow x^i = x^j \quad \forall i, j \in \{1, 2, \dots, p\}$$

*Clases diferentes tienen asociados patrones diferentes.*

*Dicho de otra manera, el conjunto fundamental debe incluir una relación entre el conjunto de patrones y el conjunto de clases, de tal manera que dicha relación cumpla con las características de una función.*

Desde su concepción, el clasificador Gamma ha sufrido algunas modificaciones que han repercutido en su desempeño de forma positiva. La última modificación registrada se presenta en [31], no obstante y después de un análisis detallado del problema a tratar, el

algoritmo elegido para la propuesta del modelo a presentarse en el capítulo siguiente está basado en el correspondiente trabajo [30].

**Algoritmo 3.2.2:** *Sea el conjunto fundamental del clasificador Gamma de acuerdo con la definición 3.2.2. Al presentarse un patrón a clasificar  $\tilde{x}$ , donde  $\tilde{x}$  es un vector real  $n$ -dimensional  $\tilde{x} \in \mathbb{R}^n$ , con  $n \in \mathbb{Z}^+$ , se realiza lo siguiente:*

1. Codificar las componentes de cada patrón del conjunto fundamental con el código Johnson-Möebius modificado. Se resta el menor valor a todos los componentes de cada patrón y se obtiene un elemento trasladado  $e_m = V_{i=1}^p x_j^i$  por cada componente esto con la finalidad de trabajar en un rango de 0 a  $e_m$ . Así, la componente  $x_j^i$  se transforma en un vector binario de dimensión  $e_m(j)$ .
2. Codificar las componentes de cada patrón a clasificar con el código Johnson-Möebius modificado, utilizando las mismas condiciones del paso 1. En caso de que alguna componente del patrón a clasificar sea mayor al elemento  $e_m$  correspondiente, esto es ( $\tilde{x}_\xi > e_m(\xi)$ ), igualar esa componente a  $e_m(\xi)$  y guardar su valor anterior en la variable  $mgamma_\xi$ . Por otro lado, si alguna componente da un valor negativo una vez desplazada, igualar esa componente a 0 y asignar el valor  $e_m(\xi) + |\tilde{x}_\xi|$  a  $mgamma_\xi$ .
3. Calcular el parámetro de paro  $\rho$  y el parámetro de pausa  $\rho_0$ . Dependiendo del problema a tratar, algunas posibilidades sugeridas para estos parámetros son las siguientes:
  - $\rho = \bigwedge_{j=1}^n (V_{i=1}^p x_j^i)$
  - $\rho = \frac{1}{n} \sum_{j=1}^n (V_{i=1}^p x_j^i)$
  - $\rho = V_{j=1}^n (V_{i=1}^p x_j^i)$
  - $\rho_0 = \bigwedge_{j=1}^n (V_{i=1}^p x_j^i)$ , sobre todo si  $\rho = V_{j=1}^n (V_{i=1}^p x_j^i)$
  - $\rho_0 = \rho$ , cuando se desea asignar forzosamente una clase conocida a los patrones desconocidos.
4. Determinar el umbral de pausa  $u$ . Considerando que el valor de este umbral depende fuertemente de las características del problema y las propiedades del conjunto fundamental, se ofrecen las siguientes sugerencias como valores iniciales:
  - $u = 0$ .
  - $u = n$ .
5. Determinar los pesos de cada dimensión  $w_i \in \mathbb{R}^+ | i = 1, 2, \dots, n$ . Dentro de este contexto, se sugieren los siguientes rangos como valores iniciales empíricos:
  - Dentro del rango  $[1.5, 2]$  a las dimensiones que sean puntualmente separables para todas las clases.

- Dentro del rango  $[1, 1.5]$  a las dimensiones que sean puntualmente separables para algunas clases o bien, que sean puntualmente segmentables para todas las clases.
  - Dentro del rango  $[0.8, 1.2]$  a las dimensiones que sean puntualmente segmentables para todas o algunas clases.
  - Dentro del rango  $(0, 0.5]$  a las dimensiones que sean puntualmente no separables.
6. Realizar una conversión de índices en los patrones del conjunto fundamental, de manera que el índice que tenía un patrón originalmente, por ejemplo  $x^\mu$ , se convierta en dos índices: uno para la clase a la que pertenece (ejemplo clase  $i$ ) y otro para el orden que le corresponde dentro del conjunto (ejemplo orden  $\omega$ ). Bajo estas condiciones, la notación para el patrón  $x^\mu$  será ahora  $x^{i\omega}$ , esto es el patrón  $x$  pertenece a la clase  $i$  en la posición  $\omega$ . Lo anterior se realiza para todos los patrones del conjunto fundamental.
  7. Inicializar  $\theta$  a 0.
  8. Realizar la operación  $\gamma_g(x_j^\mu, \tilde{x}_j, \theta)$  para las componentes de cada uno de los patrones del conjunto fundamental y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$  en caso necesario.
  9. Calcular la suma ponderada inicial  $c_i^0$  de los resultados obtenidos en el paso anterior, para cada patrón fundamental  $\mu = 1, 2, \dots, p$ :
 
$$c_\mu^0 = \sum_{j=1}^n w_j \cdot \gamma_g(x_j^\mu, \tilde{x}_j, \theta).$$
  10. Evaluar si existe un máximo único, cuyo valor es además igual a  $n$ , asignando al patrón a clasificar la clase correspondiente a ese máximo:  $\tilde{y} = y^i$  tal que  $\bigvee_{\mu=1}^p c_\mu^0 = c_i^0 = n$ . En otro caso, continuar.
  11. Realizar la operación  $\gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)$  para cada clase y para cada componente de cada uno de los patrones del conjunto fundamental que corresponden a esa clase, y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$  si es necesario.
  12. Calcular la suma ponderada  $c_i$  de los resultados obtenidos en el paso 11, para cada clase  $i = 1, 2, \dots, m$ :
 
$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n w_j \cdot \gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)}{k_i}$$
  13. Evaluar, si existe más de un máximo entre las sumas ponderadas por clase, incrementar  $\theta$  en 1 y repetir los pasos 11 y 12, hasta que:
    - a. exista un máximo único;
    - b. o se cumpla con la condición de pausa:  $\theta = \rho_0$ ;
    - c. o se cumpla con la condición de pausa:  $\theta \geq \rho$ .

14. Evaluar, si se cumple con la condición de pausa  $\theta = \rho_0$ , se compara el valor máximo de las sumas ponderadas con el umbral de pausa.
  - a. Si  $\bigvee_{i=1}^m c_i \leq u$  entonces se asigna la clase desconocida al patrón a clasificar:  $C_{\tilde{x}} = C_0$ .
  - b. Si  $\bigvee_{i=1}^m c_i > u$  entonces se continúa en el paso 11.
15. Evaluar, si existe un máximo único, asignar al patrón a clasificar la clase correspondiente a ese máximo:  $\tilde{y} = y^j$  tal que  $\bigvee_{i=1}^m c_i = c_j$ .
16. En caso contrario: si  $\lambda$  es el índice más pequeño de clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase  $\tilde{y} = y^\lambda$ .

A continuación se presentan dos diagramas de bloques. El primero describe el pre-procesamiento (pasos 1 a 7) que debe realizarse en los datos para posteriormente proseguir con el segundo diagrama que describe el algoritmo del clasificador Gamma.

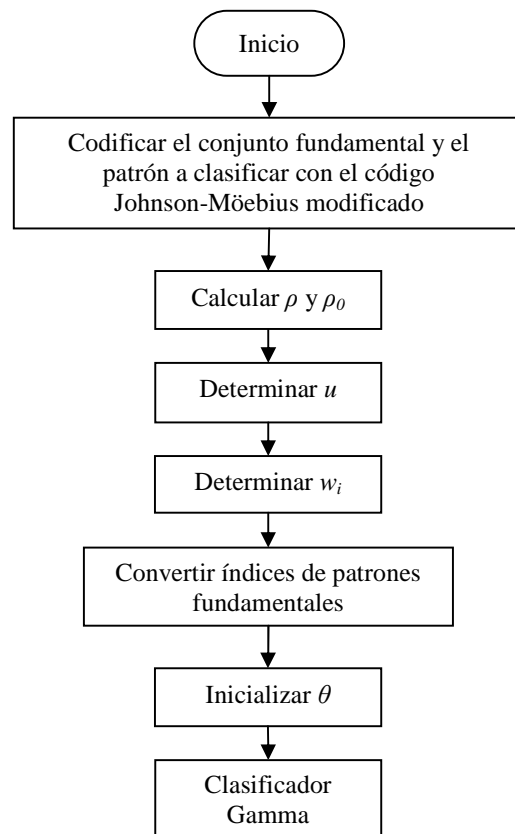


Figura 3.2.2.1. Parte 1 del clasificador Gamma: Pre-procesamiento.

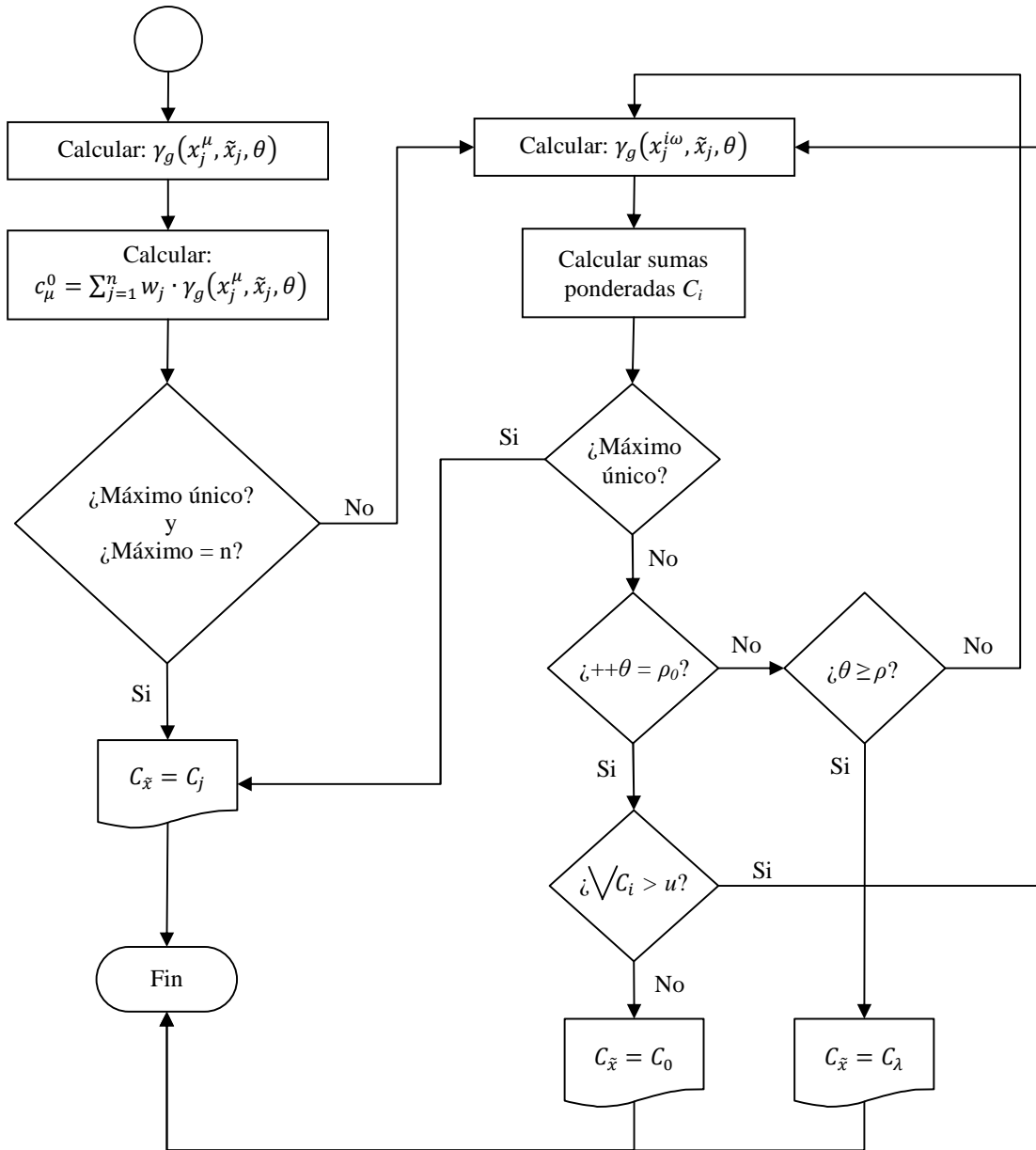


Figura 3.2.2.2. Parte 2 del clasificador Gamma: Funcionamiento del algoritmo.

A continuación se muestra un ejemplo del uso del clasificador Gamma.

**Ejemplo 3.2.2:** Sean  $x^1 = \begin{pmatrix} 6 \\ 7 \\ 4 \end{pmatrix}, x^2 = \begin{pmatrix} 6 \\ 4 \\ 3 \end{pmatrix}, x^3 = \begin{pmatrix} 4 \\ 3 \\ 3 \end{pmatrix}, x^4 = \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix}, x^5 = \begin{pmatrix} 5 \\ 4 \\ 2 \end{pmatrix}$ , los patrones fundamentales agrupados en 3 clases  $c_1 = \{x^1, x^2\}, c_2 = \{x^3, x^4\}, c_3 = x^5$ , clasificar los patrones  $y^1 = \begin{pmatrix} 4 \\ 2 \\ 5 \end{pmatrix}, y^2 = \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}$ .

Para este ejemplo, la dimensión de los patrones es  $n = 2$ , y se tiene  $m = 3$  clases, donde el número de patrones están agrupados en  $k_1 = 2$ ,  $k_2 = 2$  y  $k_3 = 1$  clases, respectivamente. Con la finalidad de clasificar los patrones de muestra, se siguen los pasos respectivos del algoritmo.

1. Dentro de las componentes de cada uno de los patrones del conjunto fundamental, se observa que el menor valor es 1, entonces, este valor se resta de cada una de las componentes de cada patrón para obtener patrones desplazados dentro del rango de 0 a  $e_m$ . Los nuevos patrones son:

$$x^1 = \begin{pmatrix} 5 \\ 6 \\ 3 \end{pmatrix}, x^2 = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}, x^3 = \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}, x^4 = \begin{pmatrix} 2 \\ 0 \\ 3 \end{pmatrix}, x^5 = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}.$$

Para obtener el tamaño de bits a utilizar se emplea el valor  $e_m$ :

$$e_m = V_{i=1}^5 x_j^i = \begin{pmatrix} 5 \\ 6 \\ 3 \end{pmatrix}, \text{ esto es, } e_m(1) = 5, e_m(2) = 6, e_m(3) = 3.$$

Los patrones codificados en código Johnson-Möebius modificado son:

$$x^1 = \begin{pmatrix} 11111 \\ 111111 \\ 111 \end{pmatrix}, x^2 = \begin{pmatrix} 11111 \\ 000111 \\ 011 \end{pmatrix}, x^3 = \begin{pmatrix} 00111 \\ 000011 \\ 011 \end{pmatrix}, x^4 = \begin{pmatrix} 00111 \\ 000000 \\ 111 \end{pmatrix},$$

$$x^5 = \begin{pmatrix} 01111 \\ 000111 \\ 001 \end{pmatrix}.$$

2. Se observa que para  $\tilde{x}_3^1 > e_m(3)$  ó  $5 > 3$ , entonces  $\tilde{x}_3^1 = 3$  y  $m\gamma_{\xi} = 5$ .

Los patrones de prueba codificados son:

$$y^1 = \begin{pmatrix} 01111 \\ 000011 \\ 111 \end{pmatrix}, y^2 = \begin{pmatrix} 00011 \\ 000111 \\ 011 \end{pmatrix}$$

3. Los parámetros de paro ( $\rho$ ) y pausa ( $\rho_0$ ) son calculados de la siguiente manera:

$$\rho = V_{j=1}^n (V_{i=1}^p x_j^i) = V_{j=1}^3 (V_{i=1}^5 x_j^i) = V_{j=1}^3 \begin{pmatrix} 5 \\ 6 \\ 3 \end{pmatrix} = 6.$$

$$\rho_0 = V_{j=1}^n (\wedge_{i=1}^p x_j^i) = V_{j=1}^3 (\wedge_{i=1}^5 x_j^i) = V_{j=1}^3 \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = 2.$$

4. El umbral de pausa es  $u = 0$ .
5. En este ejemplo la asignación de pesos a cada rasgo es  $w_i = 1, i = 1, 2, \dots, n$ .
6. La conversión de índices para cada patrón en su respectiva clase queda de la siguiente manera:

$$c_1 = \{x^1, x^2\}: x^1 = x^{11}, x^2 = x^{12}$$

$$c_2 = \{x^3, x^4\}: x^3 = x^{21}, x^4 = x^{22}$$

$$c_3 = x^5: x^5 = x^{31}$$

7. La tolerancia entre la similitud inicial es  $\theta = 0$ .



Hasta el paso anterior termina el pre-procesamiento de los datos, los siguientes pasos se encargan del proceso respectivo de clasificación.

Para clasificar el patrón  $y^1$ , se siguen los pasos restantes del algoritmo.

8. Se obtiene la similitud entre el patrón  $y^1$  y los  $x_j^\mu$  patrones del conjunto fundamental.

- $\gamma_g(x_1^{11}, y_1^1, 0)$ :  $x_1^{11} = (11111)$  y  $y_1^1 = (01111)$

Para el primer rasgo se tiene que  $n = e_m(1) = 5$ , entonces:

$$\alpha(x_1^{11}, y_1^1) = (21111)$$

$$\alpha(x_1^{11}, y_1^1) \bmod 2 = (01111)$$

$$u_\beta[\alpha(x_1^{11}, y_1^1) \bmod 2] = \beta(0,0) + \beta(1,1) + \beta(1,1) + \beta(1,1) + \beta(1,1) \\ = 0 + 1 + 1 + 1 + 1 = 4$$

$$n - u_\beta[\alpha(x_1^{11}, y_1^1) \bmod 2] = 5 - 4 = 1 \leq 0: \text{No}$$

$$\therefore \gamma_g(x_1^{11}, y_1^1, 0) = 0$$

- $\gamma_g(x_2^{11}, y_2^1, 0)$ :  $x_2^{11} = (111111)$  y  $y_2^1 = (000011)$

Para el segundo rasgo se tiene que  $n = e_m(2) = 6$ , entonces:

$$\alpha(x_2^{11}, y_2^1) = (222211)$$

$$\alpha(x_2^{11}, y_2^1) \bmod 2 = (000011)$$

$$u_\beta[\alpha(x_2^{11}, y_2^1) \bmod 2] = \\ = \beta(0,0) + \beta(0,0) + \beta(0,0) + \beta(0,0) + \beta(1,1) + \beta(1,1) \\ = 0 + 0 + 0 + 0 + 1 + 1 = 2$$

$$n - u_\beta[\alpha(x_2^{11}, y_2^1) \bmod 2] = 6 - 2 = 4 \leq 0: \text{No}$$

$$\therefore \gamma_g(x_2^{11}, y_2^1, 0) = 0$$

- $\gamma_g(x_3^{11}, y_3^1, 0)$ :  $x_3^{11} = (111)$  y  $y_3^1 = (111)$

Para el tercer rasgo se tiene que  $n = m\gamma_{\xi} = 5$ , entonces:

$$\alpha(x_3^{11}, y_3^1) = (111)$$

$$\alpha(x_3^{11}, y_3^1) \bmod 2 = (111)$$

$$u_\beta[\alpha(x_3^{11}, y_3^1) \bmod 2] = \beta(1,1) + \beta(1,1) + \beta(1,1) = 1 + 1 + 1 = 3$$

$$n - u_\beta[\alpha(x_3^{11}, y_3^1) \bmod 2] = 5 - 3 = 2 \leq 0: \text{No}$$

$$\therefore \gamma_g(x_3^{11}, y_3^1, 0) = 0$$

- $\gamma_g(x_1^{12}, y_1^1, 0) = 0, \gamma_g(x_2^{12}, y_2^1, 0) = 0, \gamma_g(x_3^{12}, y_3^1, 0) = 0$
- $\gamma_g(x_1^{21}, y_1^1, 0) = 0, \gamma_g(x_2^{21}, y_2^1, 0) = 1, \gamma_g(x_3^{21}, y_3^1, 0) = 0$
- $\gamma_g(x_1^{22}, y_1^1, 0) = 0, \gamma_g(x_2^{22}, y_2^1, 0) = 0, \gamma_g(x_3^{22}, y_3^1, 0) = 0$
- $\gamma_g(x_1^{31}, y_1^1, 0) = 1, \gamma_g(x_2^{31}, y_2^1, 0) = 0, \gamma_g(x_3^{31}, y_3^1, 0) = 0$

9. Calcular las sumas ponderadas inicial  $C_i^0$ :

$$C_1^0 = 0 + 0 + 0 = 0$$

$$C_2^0 = 0 + 0 + 0 = 0$$

$$C_3^0 = 0 + 1 + 0 = 1$$

$$C_4^0 = 0 + 0 + 0 = 0$$

$$C_5^0 = 1 + 0 + 0 = 1$$

10. ¿Existe un máximo único en  $C_i^0$ ? : **No.**

11. Repetir el paso 8.

12. Calcular las sumas ponderadas por clase  $C_i$ :

$$C_1 = \frac{0 + 0}{2} = 0$$

$$C_2 = \frac{1 + 0}{2} = \frac{1}{2}$$

$$C_3 = \frac{1}{1} = 1$$

13. ¿Existe más de un máximo único en  $C_i$ ? : **No.**

14. Este paso no se hace como consecuencia del anterior.

15. ¿Existe un máximo único en  $C_i$ ? : **Si**, por lo tanto:

El patrón  $y^1$  se asigna a la clase  $C_3$ .

16. Este paso no se hace como consecuencia del anterior.

Para clasificar el patrón  $y^2$ , se siguen los pasos restantes del algoritmo.

8. Se obtiene la similitud entre el patrón  $y^2$  y los  $x_j^\mu$  patrones del conjunto fundamental.

$$\gamma_g(x_1^{11}, y_1^2, 0) = 0, \gamma_g(x_2^{11}, y_2^2, 0) = 0, \gamma_g(x_3^{11}, y_3^2, 0) = 0$$

$$\gamma_g(x_1^{12}, y_1^2, 0) = 0, \gamma_g(x_2^{12}, y_2^2, 0) = 1, \gamma_g(x_3^{12}, y_3^2, 0) = 1$$

$$\gamma_g(x_1^{21}, y_1^2, 0) = 0, \gamma_g(x_2^{21}, y_2^2, 0) = 0, \gamma_g(x_3^{21}, y_3^2, 0) = 1$$

$$\gamma_g(x_1^{22}, y_1^2, 0) = 1, \gamma_g(x_2^{22}, y_2^2, 0) = 0, \gamma_g(x_3^{22}, y_3^2, 0) = 0$$

$$\gamma_g(x_1^{31}, y_1^2, 0) = 0, \gamma_g(x_2^{31}, y_2^2, 0) = 1, \gamma_g(x_3^{31}, y_3^2, 0) = 0$$

9. Calcular las sumas ponderadas inicial  $C_i^0$ :

$$C_1^0 = 0 + 0 + 0 = 0$$

$$C_2^0 = 0 + 1 + 1 = 2$$

$$C_3^0 = 0 + 0 + 1 = 1$$

$$C_4^0 = 1 + 0 + 0 = 1$$

$$C_5^0 = 0 + 1 + 0 = 1$$

10. ¿Existe un máximo único en  $C_i^0$ ? : **Si.**

¿El máximo es igual a  $n$ ? : **No.**

11. Repetir el paso 8.

12. Calcular las sumas ponderadas por clase  $C_i$ :

$$C_1 = \frac{0 + 2}{2} = 1$$

$$C_2 = \frac{1 + 1}{2} = 1$$

$$C_1 = \frac{1}{1} = 1$$

13. ¿Existe más de un máximo único en  $C_i$ ? : **Si**, entonces se incrementa  $\theta$  en 1.

Con  $\theta = 1$ :

$\theta = \rho_0$  ( $1 = 2$ ): **No**.

$\theta \geq \rho$  ( $1 = 6$ ): **No**.

Repetir el paso 11.

11. Establecer la similitud con  $\theta = 1$ :

- $\gamma_g(x_1^{11}, y_1^2, 1) = 0, \gamma_g(x_2^{11}, y_2^2, 1) = 0, \gamma_g(x_3^{11}, y_3^2, 1) = 1$
- $\gamma_g(x_1^{12}, y_1^2, 1) = 0, \gamma_g(x_2^{12}, y_2^2, 1) = 1, \gamma_g(x_3^{12}, y_3^2, 1) = 1$
- $\gamma_g(x_1^{21}, y_1^2, 1) = 1, \gamma_g(x_2^{21}, y_2^2, 1) = 1, \gamma_g(x_3^{21}, y_3^2, 1) = 1$
- $\gamma_g(x_1^{22}, y_1^2, 1) = 1, \gamma_g(x_2^{22}, y_2^2, 1) = 0, \gamma_g(x_3^{22}, y_3^2, 1) = 1$
- $\gamma_g(x_1^{31}, y_1^2, 1) = 0, \gamma_g(x_2^{31}, y_2^2, 1) = 1, \gamma_g(x_3^{31}, y_3^2, 1) = 1$

12. Calcular las sumas ponderadas por clase  $C_i$ :

$$C_1 = \frac{1 + 2}{2} = 1.5$$

$$C_2 = \frac{3 + 2}{2} = 2.5$$

$$C_1 = \frac{2}{1} = 2$$

13. ¿Existe más de un máximo único en  $C_i$ ? : **No**.

14. Este paso no se hace como consecuencia del anterior.

15. ¿Existe un máximo único en  $C_i$ ? : **Si**, por lo tanto:

El patrón  $y^2$  se asigna a la clase  $C_2$ .

16. Este paso no se hace como consecuencia del anterior.

### 3.3 Banco de datos

Para el actual trabajo de tesis, uno de los principales objetivos se presenta en emplear un modelo de predicción. Para ello y de acuerdo con los artículos del estado del arte, es indispensable recurrir a un banco de datos que describa el comportamiento de una *PDAR*. El banco de datos seleccionado tiene por título “*Water Treatment Plant Data Set*” y es de uso público [52].

El banco de datos mencionado describe el comportamiento de una *PDAR* urbana localizada en la ciudad de Manresa, dentro de la provincia de Barcelona, España. La

recopilación de la información fue realizada periódicamente, mediante sensores ubicados en puntos específicos de entrada-salida de cada uno de los procesos de la planta. Dicha recopilación fue realizada entre los años 1990-1991. La sección 2.1 describe a detalle el comportamiento de los procesos de la *PDAR*.

### 3.3.1 Descripción del banco de datos

Originalmente el banco de datos está constituido por 527 instancias. Debido a que la información fue recopilada diariamente, las instancias representan los días cuando se realizaron las mediciones y en total se cubre un aproximado de 1 año y medio de información. Para este trabajo y de acuerdo con el algoritmo a emplearse, dichas instancias, van a ser nombradas como patrones. Cada patrón está conformado por 38 atributos que describen el comportamiento de la *PDAR* en un día determinado. La tabla siguiente detalla en qué consiste cada atributo del banco de datos.

Tabla 3.3.1.1: Descripción de atributos.

No.	Atributo	Significado
1	Q-A	Afluente de flujo a la planta
2	ZN-A	Afluente de zinc a la planta
3	PH-A	Afluente de pH a la planta
4	BOD-A	Afluente de demanda biológica de oxígeno a la planta
5	COD-A	Afluente de demanda química de oxígeno a la planta
6	SS-A	Afluente de sólidos suspendidos a la planta
7	SSV-A	Afluente de sólidos suspendidos volátiles a la planta
8	SED-A	Afluente de sedimentos a la planta
9	COND-A	Afluente de conductividad a la planta
10	PH-TP	Entrada de pH al tanque de sedimentación primaria
11	BOD-TP	Entrada de demanda biológica de oxígeno al tanque de sedimentación primaria
12	SS-TP	Entrada de sólidos suspendidos al tanque de sedimentación primaria
13	SSV-TP	Entrada de sólidos suspendidos volátiles al tanque de sedimentación primaria
14	SED-TP	Entrada de sedimentos al tanque de sedimentación primaria
15	COND-TP	Entrada de conductividad al tanque de sedimentación primaria
16	PH-TS	Entrada de pH al tanque de sedimentación secundaria
17	BOD-TS	Entrada de demanda biológica de oxígeno al tanque de sedimentación secundaria
18	COD-TS	Entrada de demanda química de oxígeno al tanque de sedimentación secundaria
19	SS-TS	Entrada de sólidos suspendidos al tanque de sedimentación secundaria
20	SSV-TS	Entrada de sólidos suspendidos volátiles al tanque de sedimentación secundaria
21	SED-TS	Entrada de sedimentos al tanque de sedimentación secundaria
22	COND-TS	Entrada de conductividad al tanque de sedimentación secundaria
23	PH-E	Salida de pH

24	BOD-E	Salida de demanda biológica de oxígeno
25	COD-E	Salida de demanda química de oxígeno
26	SS-E	Salida de sólidos suspendidos
27	SSV-E	Salida de sólidos suspendidos volátiles
28	SED-E	Salida de sedimentos
29	COND-E	Salida de conductividad
30	RD-BOD-TP	Rendimiento de la entrada de demanda biológica de oxígeno en tanque de sedimentación primaria
31	RD-SS-TP	Rendimiento de la entrada de sólidos suspendidos en tanque de sedimentación primaria
32	RD-SED-TP	Rendimiento de la entrada de sedimentos en tanque de sedimentación primaria
33	RD-BOD-TS	Rendimiento en entrada de demanda biológica de oxígeno en tanque de sedimentación secundaria
34	RD-COD-TS	Rendimiento en entrada de demanda química de oxígeno en tanque de sedimentación secundaria
35	RD-BOD-G	Rendimiento global de entrada de demanda biológica de oxígeno
36	RD-COD-G	Rendimiento global de entrada de demanda química de oxígeno
37	RD-SS-G	Rendimiento global de sólidos suspendidos
38	RD-SED-G	Rendimiento global de sedimentos

Para establecer si la *PDAR* se encuentra en condiciones normales o anormales, los 527 patrones van a distribuirse taxonómicamente en 13 clases. La tabla siguiente detalla las 13 clases junto con el número de patrones que las constituyen, esto es, la descripción del comportamiento de la *PDAR* con base en una determinada cantidad de días.

Tabla 3.3.1.2: Descripción de clases.

Clases	Descripción	No. de días
1	Situación normal	275
2	Problema 1 en sedimentación secundaria	1
3	Problema 2 en sedimentación secundaria	1
4	Problema 3 en sedimentación secundaria	4
5	Situación normal con rendimiento sobre la media	116
6	Sobrecarga de sólidos 1	3
7	Problema 4 en sedimentación secundaria	1
8	Tormenta 1	1
9	Situación normal con bajo afluente	69
10	Tormenta 2	1
11	Situación normal	53
12	Tormenta 3	1
13	Sobrecarga de sólidos 2	1

Una vez descrito el banco de datos se tienen los siguientes comentarios:

- Es indispensable realizar un análisis adecuado para que toda la información del banco de datos sea lo más confiable posible y por tanto, el proceso de predicción también; en este sentido, es necesario realizar un pre-procesamiento para que todos los datos sean uniformes y con la menor cantidad de ruido que no altere la información disponible, hablando principalmente por la existencia de valores faltantes en algunos patrones.
- Respecto a las clases mencionadas, a simple vista se puede observar que el número de clases que presentan condiciones anormales son más que si se presentan condiciones normales, no obstante, la cantidad de patrones de las clases con condiciones anormales es demasiado pequeño; es por ello que, en el capítulo siguiente es necesario analizar si se conservan éstas clases o se hace una reducción de las mismas.
- Este banco de datos al ser recopilado de forma periódica va a ser tratado como una serie de tiempo en función de las condiciones que presenta cada uno de los días de que se tienen registro.

### 3.4 Waikato Environment for Knowledge Analysis

*Weka* es un entorno de análisis para aprendizaje automático y minería de datos escrito en la plataforma de *Java* y realizado por la Universidad de Waikato, ubicada en Nueva Zelanda. La plataforma de trabajo contiene una vasta colección de algoritmos de análisis de datos y modelos predictivos en los que se incluyen métodos que abordan problemas de regresión, clasificación, clustering y selección de atributos. El software permite a un banco de datos ser pre-procesado, si se requiere, e introducirlo en un esquema de aprendizaje con la finalidad de analizar tanto los resultados como el desempeño de un clasificador seleccionado [55].

En las siguientes sub-secciones se describen de manera general algunos algoritmos que emplea *Weka* para clasificar patrones y que serán utilizados para el presente documento. La información siguiente puede ser consultada a detalle en [55].

#### 3.4.1 MultiLayerPerceptron

Es un clasificador basado en *RNA* perceptrón multicapa que utiliza el algoritmo de retropropagación del error para realizar el entrenamiento. En *Weka* la red neuronal puede ser construida a mano, monitoreada y modificada durante la etapa de entrenamiento. Los nodos en esta red son sigmoidales con excepción de los nodos de salida para clases numéricas que son transformados automáticamente en unidades lineales sin umbral.

### 3.4.2 RBFNetwork

Este clasificador implementa una red neuronal de funciones de base radial con normalización gaussiana. Emplea el algoritmo por agrupamiento *k*-medias (*k-means clustering algorithm*) para proveer las funciones base que permitan el aprendizaje ya sea para regresión logística (problemas de clases discretas) o para regresión lineal (problemas de clases numéricas). Es posible especificar *k* como el número de clústeres a utilizar. Si la clase es nominal (clase discreta) el algoritmo *k-means* es aplicado por separado a cada clase para obtener *k* clústeres en cada clase.

### 3.4.3 BayesNet

Basado en el teorema de Bayes, una red bayesiana es un modelo probabilístico que abstrae un conjunto de variables aleatorias y sus dependencias condicionales mediante una representación gráfica llamada grafo acíclico dirigido. El aprendizaje de la red de Bayes se da haciendo uso de algoritmos de búsqueda y de mediciones previamente realizadas, esto es, conocimiento previo.

### 3.4.4 NaiveBayes

Al igual que la red bayesiana, el clasificador de Bayes ingenuo es un modelo probabilístico que está fundamentado en el teorema de Bayes y algunas hipótesis de simplificación o independencia entre variables predictoras. De la independencia entre variables es que recibe el apelativo de ingenuo.

### 3.4.5 RandomTree

Este algoritmo es un modelo de predicción que dado un conjunto de datos se construyen diagramas lógicos, parecidos a los sistemas de predicción basados en reglas, que sirven para representar y categorizar condiciones que ocurren de forma secuencial para la resolución de un determinado problema. *Weka* permite construir un árbol que considera escoger aleatoriamente *k* atributos de cada nodo u hoja del árbol.

### 3.4.6 REPTree

Es un algoritmo de aprendizaje rápido que emplea árboles de decisión, los cuales se construyen usando una reducción de ganancia/varianza de la información y se poda empleando un criterio de reducción de error. Clasifica valores solamente una vez para atributos numéricos. Los valores que faltan se obtienen partiendo las correspondientes instancias.

### 3.4.7 SVM

Son un conjunto de algoritmos de aprendizaje supervisado que están propiamente relacionados con problemas de clasificación y regresión. Una máquina de soporte vectorial construye un hiperplano o conjunto de hiperplanos, en un espacio de  $n$ -dimensión, para buscar separar y diferenciar de forma óptima los elementos entre una y otra clase o varias clases. La forma de generar los hiperplanos es mediante funciones kernel que buscan separar lo más eficientemente las clases de un problema dado. Entre las funciones kernel que emplea *Weka* están: kernel lineal, kernel polinomial, kernel de base radial y kernel sigmoidal.

### 3.4.8 IBk

IBk emplea el algoritmo de los  $k$  vecinos más cercanos ( $k$ -nn ó  $k$ -nearest neighbor). Es un método de clasificación supervisada basado en un entrenamiento mediante ejemplos cercanos en el espacio de los  $k$ -nn elementos de diversas clases y que permiten clasificar un patrón  $x$  en una clase  $c$  determinada. El valor de  $k$  establece el número de vecinos que se quieren detectar entorno a un patrón determinado. El valor por default es  $k = 1$ , aunque puede ser variado por cualquier número en el orden de los naturales.

El uso del entorno de análisis *Weka* en el presente documento es con la intención de comparar los resultados generados entre el entorno de análisis y el resultado obtenido por el clasificador Gamma. Dichos resultados serán expuestos a detalle en el posterior capítulo 5.



## Capítulo 4

# Modelo propuesto

Con base en las herramientas presentadas en el capítulo anterior, este capítulo describe el modelo propuesto para realizar la predicción de las condiciones que presenta una *PDAR*. La primera sección realiza un análisis del banco de datos a utilizarse. La segunda sección define algunos valores iniciales necesarios para el correcto funcionamiento del clasificador Gamma. La tercera sección presenta el modelo de predicción sobre el banco de datos y cómo actúa al emplear el algoritmo del clasificador Gamma.

### 4.1 Pre-procesamiento del banco de datos

En la sub-sección 3.3.1 se hizo mención que existen algunas consideraciones previas para hacer uso del banco de datos y que deben ser tomadas en cuenta antes de realizar el proceso de predicción. Las consideraciones previas en general son dos: valores perdidos o faltantes y número de clases a utilizar.

#### 4.1.1 Valores faltantes

Para realizar el proceso de predicción que se busca, es indispensable que la información con la que se está trabajando esté lo más completa posible. El banco de datos que se ocupa en esta tesis tiene la desventaja de que cuenta con una cantidad considerable de valores faltantes y esto provoca que el proceso de predicción no pueda realizarse. Para atacar este problema se consideraron dos propuestas para cubrir dichos valores.

La primera propuesta consiste en omitir aquellos patrones donde en alguno de sus rasgos exista un valor del cual no se tiene información. Llevando a cabo la propuesta, el resultado obtenido es que de los 527 patrones del banco de datos, se deben descartar 147 patrones, debido a que en uno o más rasgos existe un valor que es desconocido. Lo anterior permite que el banco de datos sea reducido a 380 patrones con los cuales se puede trabajar.

La propuesta, si bien en algunos artículos del estado del arte es realizada [12], [18], no es una opción viable para el presente trabajo, ya que el banco de datos reducido representa un 72.11 % del banco de datos total, mientras que los patrones descartados representan un 27.89 %, esto es, se discrimina casi 1/3 parte del total de la información con la cual se dispone. La inviabilidad de utilizar 2/3 partes es debido a que dentro de la 1/3 parte discriminada, se encuentra información importante que es vital para el éxito del proceso de predicción. Esta aseveración se verá más a detalle en secciones posteriores.

La segunda propuesta es utilizar un modelo matemático que permita cubrir la información desconocida. El modelo matemático empleado fue la interpolación polinómica de Hermite [54] para la extrapolación de valores desconocidos o faltantes. Este método consiste en buscar un polinomio por partes  $H_n(x)$  que sea cúbico en cada subintervalo  $[x_{i-1}, x_i]$ ,  $1 \leq i \leq n$  y que cumpla  $f'(x)$  en los puntos  $\{x_0, \dots, x_n\}$ , donde  $f(x)$  es la función que se quiere interpolar.

Llevando a la práctica la propuesta, los valores obtenidos tratan de cubrir de la mejor forma posible los valores desconocidos. Para ejemplificar lo anterior se realiza la prueba con un rasgo que tenga valores perdidos. Dicho rasgo va a ser tratado como una serie de tiempo.

La Figura 4.1.1 muestra un ejemplo:

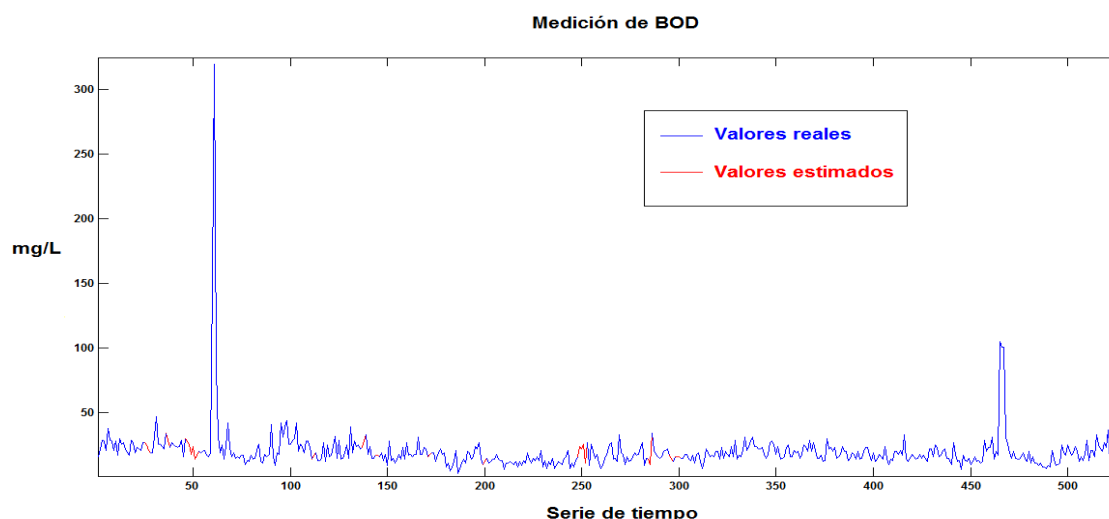


Figura 4.1.1: Serie de tiempo del rasgo 24 ó BOD.

La figura anterior muestra como los valores perdidos son reemplazados, por otros ya estimados, vía la interpolación polinómica de Hermite. La razón final para ocupar este método se debe a que fue el que mejores resultados arrojó en la estimación de dichos valores. Este método fue aplicado a todos los rasgos del banco de datos.

### 4.1.2 Número de clases

Haciendo referencia a la tabla 3.3.2, se mostró cuantas y cuáles son las clases que maneja el banco de datos que se están utilizando. Para el proceso de predicción que se busca, es importante tener a la mano suficiente información como sea posible, esto es, *mientras más información pasada se tenga mejor será la predicción que se pueda realizar*.

Después de llevar a cabo varias pruebas, la mejor respuesta obtenida fue realizando una reducción de 13 a 2 clases. La decisión se tomó debido a que la mayoría de las clases que establecen condiciones anormales, aún juntándolas entre sus similares, no reunían información suficiente para realizar un proceso de predicción adecuado, es por ello que se optó por reunir las en una sola clase. En cuanto a las clases en condiciones normales se decidió unir las con base en el mismo criterio.

Tabla 4.1.2: Clases que van emplearse en el proceso de predicción.

Clases	Descripción	No. de días
1	Situación normal	513
2	Situación anormal	14

Al revisar la tabla 4.1.2, se observa que existe una desproporción entre ambas clases, no obstante, la clase 2 aún con 14 patrones cumple perfectamente con suficiente información para utilizarse en el proceso de predicción.

## 4.2 Valores iniciales del clasificador Gamma

Como se mencionó en la sección 3.2.2 que hace referencia al clasificador Gamma, en la parte correspondiente a su pre-procesamiento (figura 3.2.2.1), es necesario establecer algunos valores iniciales para su correcto funcionamiento.

Después de un exhaustivo análisis del problema, se llegó a la conclusión de que el seguimiento establecido para el algoritmo en el proceso de predicción originaba conclusiones no satisfactorias de acuerdo con los objetivos de la tesis.

Las características propias del problema hicieron énfasis en que la información más importante se encuentra contenida dentro del banco de datos y principalmente en la información de cada uno de sus rasgos. Lo anterior implica que, si es necesario analizar los datos para trabajar, entonces, el clasificador Gamma debe adaptarse al problema que se plantea. Es por ello que el paso 5 del algoritmo del clasificador Gamma, asignación de pesos, es uno de los más importantes en este trabajo. Posteriormente se revisarán otros pasos del algoritmo cuyo comportamiento es con base en los pesos establecidos.

### 4.2.1 Asignación de pesos

De acuerdo con el estado del arte [53] los rasgos de mayor importancia y que definen las condiciones en que se encuentra la *PDAR* son principalmente tres:

Tabla 4.2.1: Rasgos más importantes del banco de datos.

No.	Atributo	Descripción
24	BOD-E	Salida de demanda biológica de oxígeno
25	COD-E	Salida de demanda química de oxígeno
26	SS-E	Salida de sólidos suspendidos

Estos rasgos describen el comportamiento del efluente (*E*) una vez tratado por las diferentes etapas del proceso de depuración.

Haciendo referencia nuevamente a [53], se dan a conocer algunos parámetros establecidos por expertos para taxonómicamente diferenciar una clase de la otra. Dichos parámetros son expuestos en forma de regla lógica para determinar cuando la planta se encuentra en condiciones normales. La regla lógica es la siguiente:

$$Si (x_{24}^{\omega} < 25) \wedge (x_{25}^{\omega} < 75) \wedge (x_{26}^{\omega} < 35) \rightarrow \text{condición normal}$$

Los valores 25, 75 y 35 son cantidades medidas en *mg/L*.

Tomando como apoyo la información anterior, los pesos designados para buscar que el clasificador Gamma tenga un rendimiento adecuado son los siguientes:

Tabla 4.2.2: Pesos asignados a cada rasgo.

Rasgo	Peso asignado	Rasgo	Peso asignado	Rasgo	Peso asignado
1	10	14	1	27	1
2	1	15	1	28	1
3	1	16	1	29	1
4	1.25	17	1.25	30	1.25
5	1.75	18	1.75	31	1.35
6	1.35	19	1.35	32	1
7	1	20	1	33	1.25
8	1	21	1	34	1.75
9	1	22	1	35	1.25
10	1	23	1	36	1.75
11	1.25	24	1.25	37	7.35
12	1.35	25	1.75	38	1
13	1	26	1.35		

Como se puede observar en la tabla anterior, los tres rasgos de la tabla 4.2.1 no son los únicos que dictan cuando una planta se encuentra en condiciones normales. Fue necesario hacer una revisión de los datos de otros rasgos para observar su comportamiento y optar por decidir cuántos y cuáles rasgos ocupar para finalmente asignar un valor de peso tomando como base la regla lógica establecida.

## 4.2.2 Condiciones de paro

Una vez definido el valor de los pesos en función de cada uno de los rasgos, se realiza la implementación del algoritmo y su comportamiento dicta que la clasificación de patrones siempre se da en el paso número 13 del clasificador Gamma, esto es, siempre se encuentra un máximo entre las sumas ponderadas por clase, lo que equivale a tener una similitud entre patrones idóneo para asignar un patrón desconocido a una clase determinada.

La observación anterior hace que las condiciones de paro ( $\rho$ ) y pausa ( $\rho_0$ ) del algoritmo sean determinados con base en valores seleccionados del banco de datos de la siguiente manera:

- $\rho = [\wedge_{j=1}^n (V_{i=1}^p x_j^i)]$
- $\rho_0 = \wedge_{j=1}^n (V_{i=1}^p x_j^i)$

Los valores seleccionados hacen converger al clasificador Gamma de manera muy rápida, esto es, el margen de trabajo es bastante reducido, no obstante, suficiente para generar un resultado satisfactorio respecto al proceso que se quiere realizar.

## 4.2.3 Umbral de pausa

Como se mencionó al inicio de la sub-sección anterior, el uso de los pesos que se han establecido, ayudan al algoritmo a concretar el proceso de clasificación en el paso número 13 (descrito en la sección 3.2.2). Lo anterior implica que el paso del algoritmo que hace uso del umbral de pausa nunca es alcanzado. Sin embargo, se asigna un valor al umbral con la idea de seguir lo mejor posible la mayoría de los pasos que han sido diseñados para el funcionamiento del algoritmo.

El valor asignado para el umbral de pausa es el siguiente:

- $u = 0$

Se hace la aclaración de que este valor, dentro del presente problema, puede o no ser utilizado una vez que se lleva el algoritmo a la implementación.

## 4.2.4 Grado de similitud

El último valor que debe definirse es el grado de similitud. Este valor juega un papel importante en el proceso de clasificación del algoritmo ya que mediante su grado de similitud un patrón puede parecerse a otro en un número determinado de rasgos.

La teoría diseñada para el algoritmo establece que el grado de similitud  $\theta$ , debe comenzar en cero. Después de realizar las pruebas pertinentes con el banco de datos de la planta residual, se obtuvieron resultados no muy aceptables para los fines que persigue la tesis.

La razón de lo anterior se debe a que si el valor de  $\theta$  comienza en cero, al menos para este problema, la similitud que se encuentra entre un patrón y otro está entre 5 y 13 rasgos como máximo. Si se habla de un banco de datos con 38 rasgos lo mínimo que se esperaría para obtener un resultado aceptable es que el proceso de similitud esté por encima del 50 % entre un patrón y otro. Hablar de un porcentaje menor al 50 % significa que la similitud entre patrones es pobre y por lo tanto el rendimiento de clasificación también.

Lo anterior genera un planteamiento que tiene que ver con el funcionamiento del clasificador Gamma: si se quiere obtener una similitud entre patrones que sea confiable, para concluir que un patrón es parecido a otro, entonces, el grado de tolerancia en la similitud, respecto de un patrón y otro, debe satisfacer la mayor cantidad de semejanzas entre los rasgos que involucran a los patrones que se están comparando. Este planteamiento arrojó, después de varias pruebas, que existen varios valores ideales iniciales para  $\theta$ , no obstante y exclusivamente en este trabajo, el valor inicial para  $\theta$  y que va ser utilizado en las pruebas posteriores es el siguiente:

- $\theta = 125$

El valor de  $\theta$  concluye el pre-procesamiento que requiere el clasificador Gamma para su buen funcionamiento.

## 4.3 Modelo para la predicción

En la sección 3.3 del capítulo anterior, se describe de forma general el banco de datos “*Water Treatment Plant Data Set*”, el cual es una recopilación de información realizada entre los años 1990-1991 y su contenido describe el funcionamiento de una *PDAR*.

De acuerdo con el título de la tesis, lo que se busca con el proceso de predicción es determinar las condiciones en las que se encuentra una planta. El banco de datos a utilizar ya proporciona esa información estableciendo que días la planta funcionó en condiciones

óptimas y que días no (sección 4.1.2 habla de ello). Por lo tanto, el modelo de predicción a proponerse debe ser similar al que se ocupa en la clasificación de patrones haciendo uso del método de validación *hold-out*, el cual toma del banco de datos original un porcentaje determinado para conformar dos conjuntos de datos: *conjunto de fundamental* (aprendizaje) y *conjunto de prueba* (datos a predecir).

Para definir qué patrones van a pertenecer al conjunto fundamental y que patrones al conjunto de prueba (valores a predecir) se utilizará la siguiente gráfica (figura 3.2.1) que muestra cual fue el comportamiento de las condiciones de la planta depuradora entre los años 1990-1991.

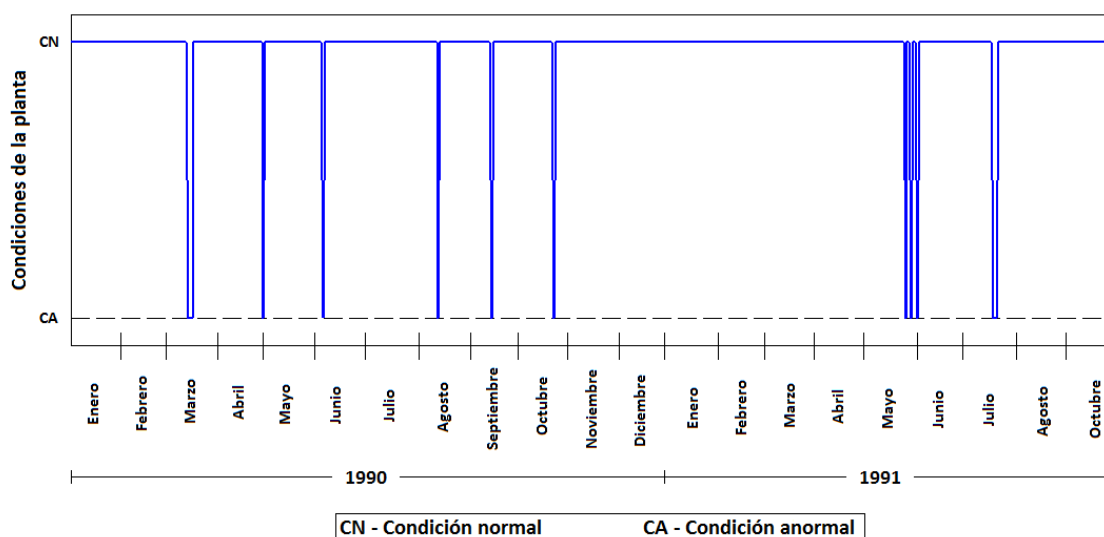


Figura 4.3.1: Comportamiento de la planta depuradora en los años 1990-1991.

Al revisar la figura anterior se pudo observar con detenimiento lo comentado al final de la sección 4.1.2 respecto al hecho de que el banco de datos registra un alto porcentaje de días en que la planta depuradora se comportó en condiciones normales, para ser más precisos, se tiene un registro del 97.34 % de *condiciones normales*, mientras que sólo el 2.66 % de los datos se reportan en *condiciones anormales*.

Con la figura 4.3.1 ya es posible definir cuál va a ser el conjunto fundamental y cuál el de prueba. La elección es en función de los días que arrojan un registro de condiciones anormales. Para obtener el conjunto fundamental se hace seleccionando un número considerable de meses o días en los que se tenga información suficiente de ambas clases dándole más importancia a la clase que tiene menor registro.

Lo anterior se puede resumir en el siguiente ejemplo:

**Ejemplo 4.3:** Realizar la predicción del año 1991 con la información del año 1990.

La figura 4.3.2 gráficamente muestra lo que se tiene que hacer:

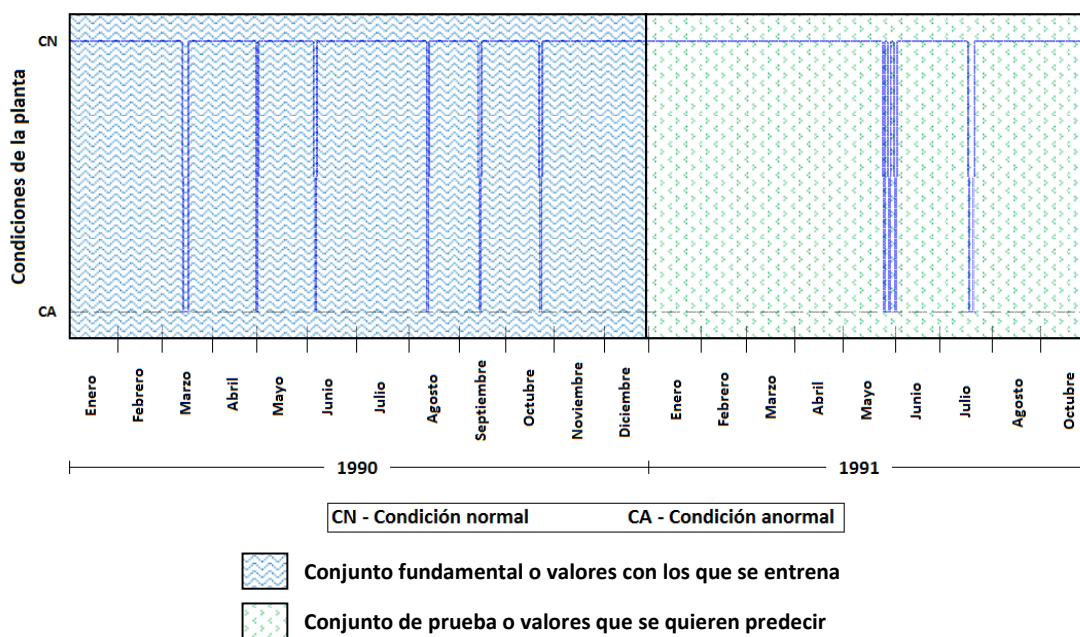


Figura 4.3.2: Definición del conjunto fundamental y del conjunto de prueba.

Para este ejemplo el conjunto fundamental es de 300 patrones y el de prueba de 227 patrones.

Del conjunto fundamental se tienen la siguiente información:

- 292 patrones (97.33%) de todo el conjunto para la clase 1 (condición normal).
- 8 patrones (2.67%) de todo el conjunto para la clase 2 (condición anormal).

Del conjunto de prueba se tiene el siguiente registro:

- 221 patrones, esto es, un 97.35 % que se quiere predecir para la clase 1.
- 6 patrones, esto es, un 2.65 % que se quiere predecir para la clase 2.

Con este ejemplo se observa que para realizar una buena predicción, el clasificador Gamma tiene el reto de predecir correctamente los patrones que definen a la condición anormal. Por el lado de la clase que define a la condición normal no existe mucho problema debido a que la información disponible es suficiente para que se obtengan buenos resultados.



Así como en el ejemplo anterior se propondrán otros que permitan realizar un proceso de predicción ya sea aumentando o disminuyendo el número de patrones del conjunto fundamental o de entrenamiento.

En el capítulo siguiente se muestran los resultados obtenidos de las pruebas propuestas para observar el comportamiento del algoritmo y posteriormente dar una interpretación y conclusión a esos resultados.

## Capítulo 5

# Resultados y discusión

En este capítulo se presentan los resultados correspondientes a las pruebas realizadas con el clasificador Gamma, así mismo, se muestran los resultados que corresponden al empleo de otros algoritmos de reconocimiento de patrones y con los cuales se estableció una comparación de rendimiento en la predicción.

### 5.1 Aplicación del modelo propuesto

Para el presente capítulo se exponen las pruebas realizadas seguidas al modelo propuesto del capítulo anterior. Se efectuaron 6 experimentos que muestran el desempeño del clasificador Gamma, los cuales fueron elegidos con base en lo planteado en la sección 4.3. Los resultados de las pruebas realizadas son presentados en las posteriores secciones.

### 5.2 Prueba de predicción 1: 100-427

La primera prueba contempla 100 días consecutivos de aprendizaje para conformar el conjunto fundamental, mientras que los restantes 427 días conforman el conjunto de prueba o los patrones a predecir.

La información contenida en el conjunto fundamental es la siguiente:

- 96 patrones de aprendizaje para la clase 1 (condición normal).
- 4 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 417 patrones que se esperan predecir para la clase 1 (condición normal).
- 10 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.2.1 muestra de forma gráfica el planteamiento descrito.

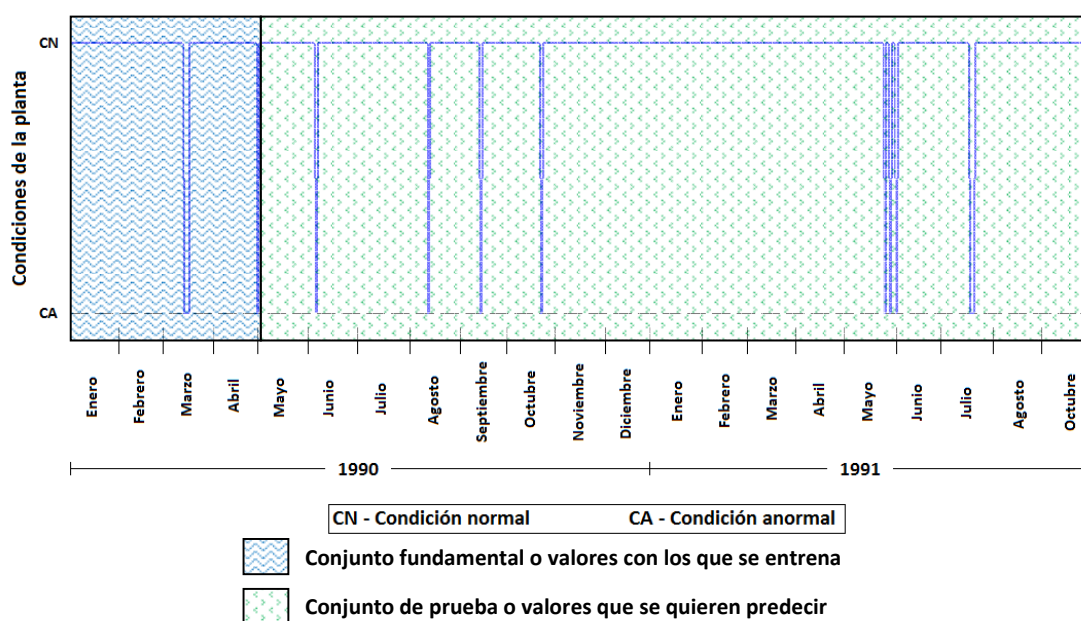


Figura 5.2.1: Conjuntos fundamental y de prueba para la prueba de predicción 1.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

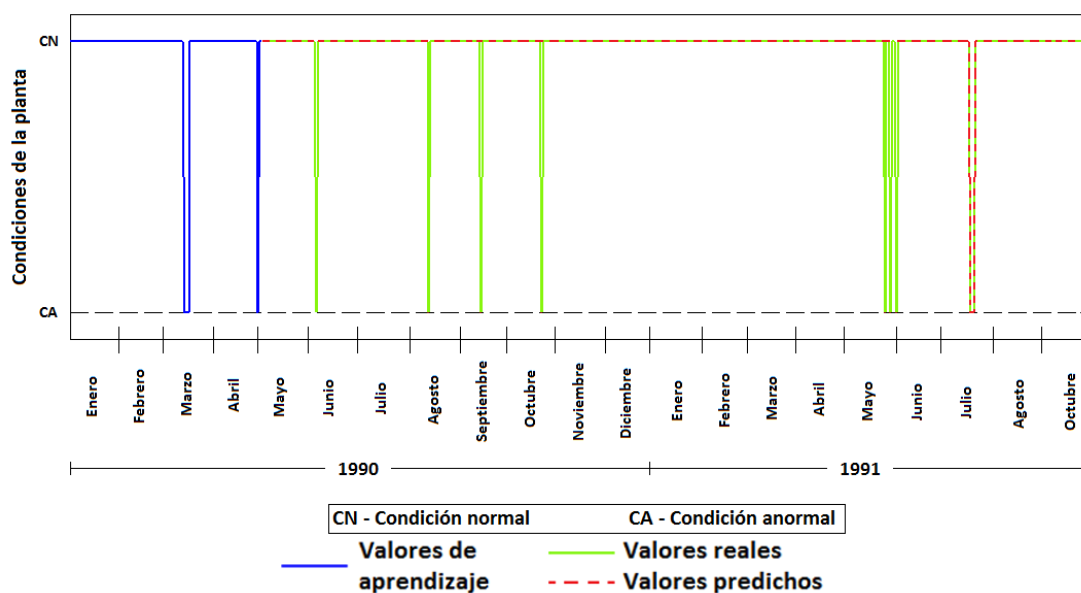


Figura 5.2.2: Predicción realizada con el clasificador Gamma para la prueba 1.

La forma de conocer el rendimiento de predicción con cualquiera de los algoritmos de clasificación empleados, es sacando el cociente del número de días predichos entre el total de días que se querían predecir; el cociente obtenido se

multiplica por cien para dar como resultado el rendimiento de trabajo del algoritmo de clasificación.

El párrafo anterior se resume en la siguiente *ecuación de rendimiento*:

$$\frac{\text{días predichos}}{\text{total de días a predecir}} \times 100 = \text{rendimiento } \%$$

Para este ejemplo de un total de 427 días (tamaño del conjunto prueba) se logró una predicción de 420. Siguiendo la *ecuación de rendimiento*, se obtiene:

$$\frac{420}{427} \times 100 = 98.3607 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, tanto del clasificador Gamma como de los algoritmos de clasificación del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.2.1: Rendimientos entre el clasificador Gamma y los algoritmos del entorno de análisis *Weka* para la prueba de predicción 1.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
Multilayer perceptron	417	3	0	7	420	7	100.00	30.00	98.3607
Bayes Net	417	3	0	7	420	7	100.00	30.00	98.3607
Naïve Bayes	417	3	0	7	420	7	100.00	30.00	98.3607
<b>Gamma</b>	<b>417</b>	<b>3</b>	<b>0</b>	<b>7</b>	<b>420</b>	<b>7</b>	<b>100.00</b>	<b>30.00</b>	<b>98.3607</b>
RBFNetwork	417	2	0	8	419	8	100.00	20.00	98.1265
Random Tree	416	2	1	8	418	9	99.76	20.00	97.8923
1 - Nearest Neighbor	417	1	0	9	418	9	100.00	10.00	97.8923
LibSVM con kernel radial bassis	417	0	0	10	417	10	100.00	0.00	97.6581
LibSVM con kernel sigmoidal	417	0	0	10	417	10	100.00	0.00	97.6581
REP Tree	417	0	0	10	417	10	100.00	0.00	97.6581
3 - Nearest Neighbor	417	0	0	10	417	10	100.00	0.00	97.6581
LibSVM con kernel lineal	415	1	2	9	416	11	99.52	10.00	97.4239
LibSVM con kernel polinomial	413	1	4	9	414	13	99.04	10.00	96.9555

Las pruebas posteriores siguen el mismo esquema al mostrado en esta primera prueba.

### 5.3 Prueba de predicción 2: 150-377

La segunda prueba consiste en entrenar con 150 días consecutivos para conformar el conjunto fundamental. Los 377 días restantes conforman el conjunto de patrones a predecir.

La información contenida en el conjunto fundamental es la siguiente:

- 145 patrones de aprendizaje para la clase 1 (condición normal).
- 5 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 368 patrones que se esperan predecir para la clase 1 (condición normal).
- 9 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.3.1 muestra de forma gráfica el planteamiento descrito.

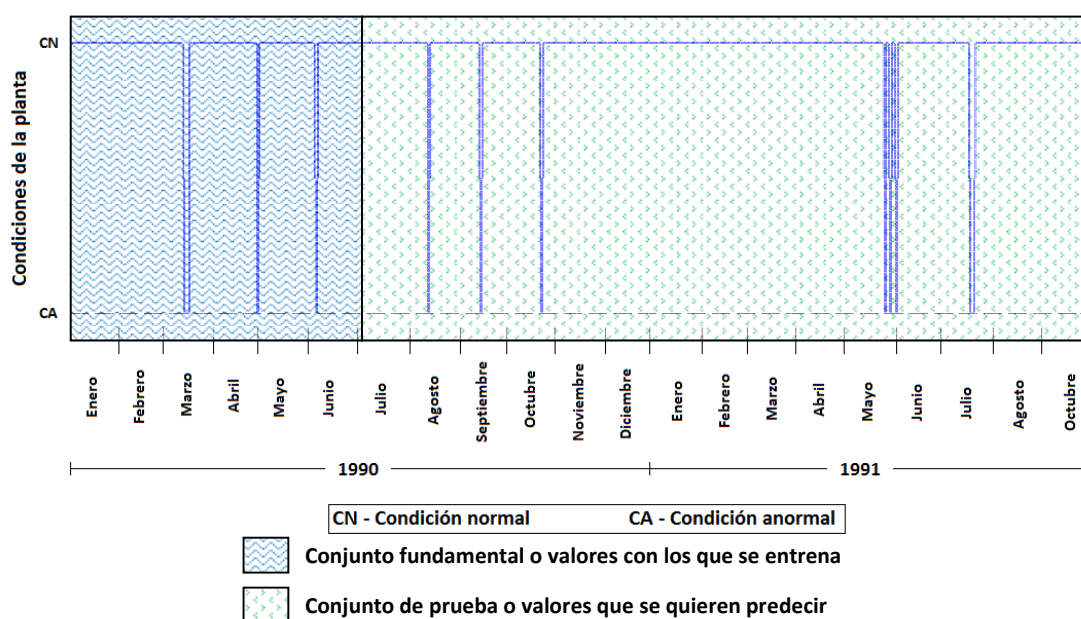


Figura 5.3.1: Conjuntos fundamental y de prueba para la prueba de predicción 2.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

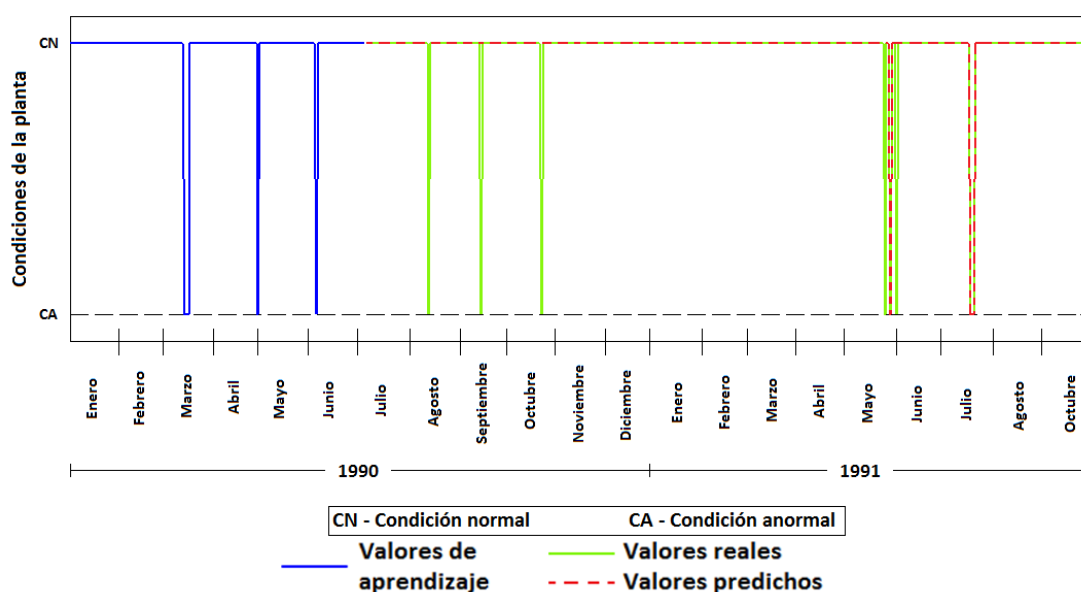


Figura 5.3.2: Predicción realizada con el clasificador Gamma para prueba 2.

Para este ejemplo de un total de 377 días (tamaño del conjunto prueba) se logró una predicción de 372. Siguiendo la *ecuación de rendimiento*, se tiene:

$$\frac{372}{377} \times 100 = 98.6737 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, del clasificador Gamma junto con los algoritmos de clasificación seleccionados del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.3.1: Rendimientos entre el clasificador Gamma y los algoritmos del entorno de análisis *Weka* para la prueba de predicción 2.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
<b>Gamma</b>	<b>368</b>	<b>4</b>	<b>0</b>	<b>5</b>	<b>372</b>	<b>5</b>	<b>100.00</b>	<b>44.44</b>	<b>98.6737</b>
RBFNetwork	366	6	2	3	372	5	99.46	66.67	98.6737
Multilayer perceptron	368	3	0	6	371	6	100.00	33.33	98.4085
Bayes Net	368	3	0	6	371	6	100.00	33.33	98.4085
1 - Nearest Neighbor	368	2	0	7	370	7	100.00	22.22	98.1432
Naive Bayes	364	6	4	3	370	7	98.91	66.67	98.1432
LibSVM con kernel lineal	364	5	4	4	369	8	98.91	55.56	97.8780
LibSVM con kernel radial bassis	368	0	0	9	368	9	100.00	0.00	97.6127

LibSVM con kernel sigmoidal	368	0	0	9	368	9	100.00	0.00	97.6127
REP Tree	368	0	0	9	368	9	100.00	0.00	97.6127
3 - Nearest Neighbor	368	0	0	9	368	9	100.00	0.00	97.6127
Random Tree	362	6	6	3	368	9	98.37	66.67	97.6127
LibSVM con kernel polinomial	361	3	7	6	364	13	98.10	33.33	96.5517

## 5.4 Prueba de predicción 3: 200-327

La tercera prueba tiene como base 200 días consecutivos que conformar el conjunto fundamental. Para conformar el conjunto de prueba se toman los 327 días restantes.

La información contenida en el conjunto fundamental es la siguiente:

- 194 patrones de aprendizaje para la clase 1 (condición normal).
- 6 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 319 patrones que se esperan predecir para la clase 1 (condición normal).
- 8 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.4.1 muestra de forma gráfica el planteamiento descrito.

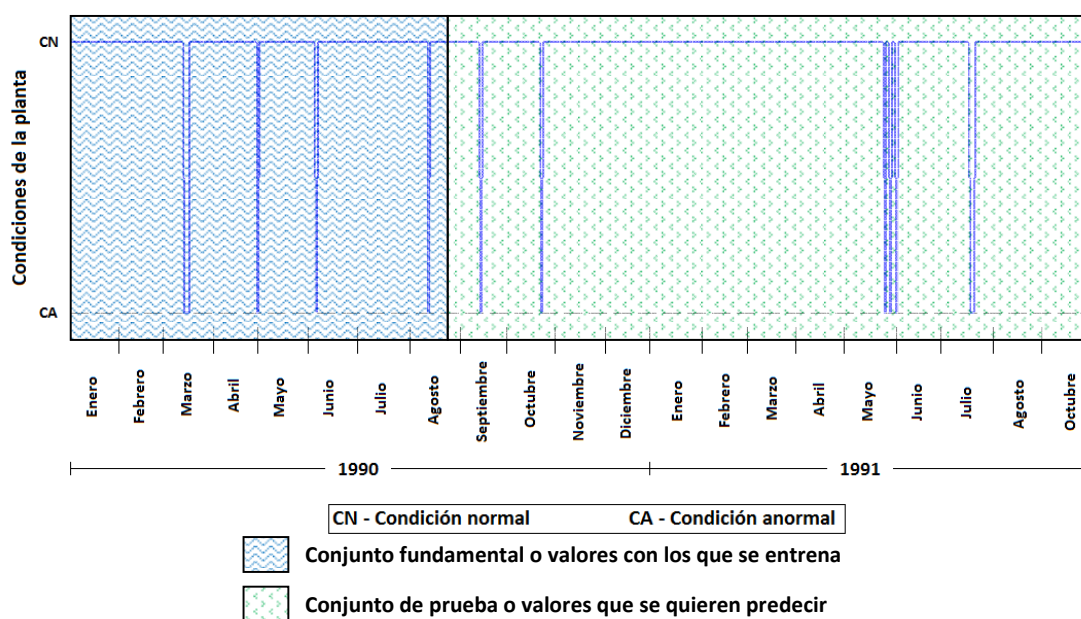


Figura 5.4.1: Conjuntos fundamental y de prueba para la prueba de predicción 3.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

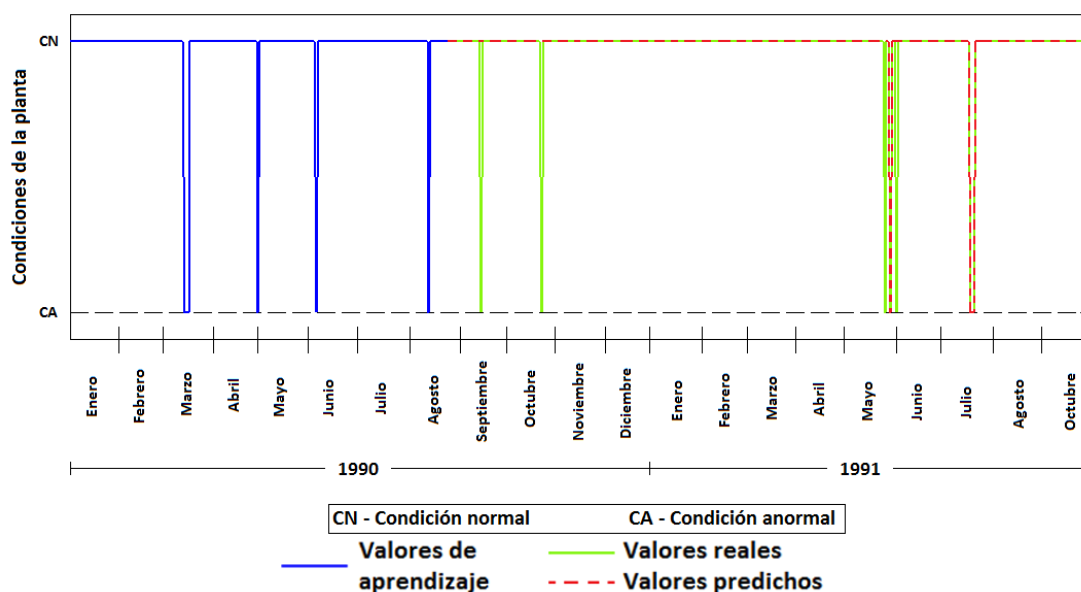


Figura 5.4.2: Predicción realizada con el clasificador Gamma para prueba 3.

Para este ejemplo de un total de 327 días (tamaño del conjunto prueba) se logró una predicción de 323. Siguiendo la *ecuación de rendimiento*, se tiene:

$$\frac{323}{327} \times 100 = 98.7768 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, del clasificador Gamma junto con los algoritmos de clasificación seleccionados del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.4.1: Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis *Weka* para la prueba de predicción 3.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
Multilayer perceptron	319	7	0	1	326	1	100.00	87.50	99.6942
<b>Gamma</b>	<b>319</b>	<b>4</b>	<b>0</b>	<b>4</b>	<b>323</b>	<b>6</b>	<b>100.00</b>	<b>50.00</b>	<b>98.7768</b>
RBFNetwork	319	3	0	5	322	5	100.00	37.50	98.4709
Bayes Net	319	3	0	5	322	5	100.00	37.50	98.4709
LibSVM con kernel lineal	319	3	0	5	322	5	100.00	37.50	98.4709



LibSVM con kernel polinomial	319	3	0	5	322	5	100.00	37.50	98.4709
Random Tree	317	5	2	3	322	5	99.37	62.50	98.4709
1 - Nearest Neighbor	319	2	0	6	321	6	100.00	25.00	98.1651
Naive Bayes	314	7	5	1	321	6	98.43	87.50	98.1651
LibSVM con kernel radial bassis	319	0	0	8	319	8	100.00	0.00	97.5535
LibSVM con kernel sigmoidal	319	0	0	8	319	8	100.00	0.00	97.5535
REP Tree	319	0	0	8	319	8	100.00	0.00	97.5535
3 - Nearest Neighbor	319	0	0	8	319	8	100.00	0.00	97.5535

## 5.5 Prueba de predicción 4: 235-292

Para la cuarta prueba se cuenta con 235 días consecutivos que dan forma al conjunto fundamental. El conjunto de prueba se conforma con los 292 días restantes.

La información contenida en el conjunto fundamental es la siguiente:

- 228 patrones de aprendizaje para la clase 1 (condición normal).
- 7 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 285 patrones que se esperan predecir para la clase 1 (condición normal).
- 7 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.5.1 muestra de forma gráfica el planteamiento descrito.

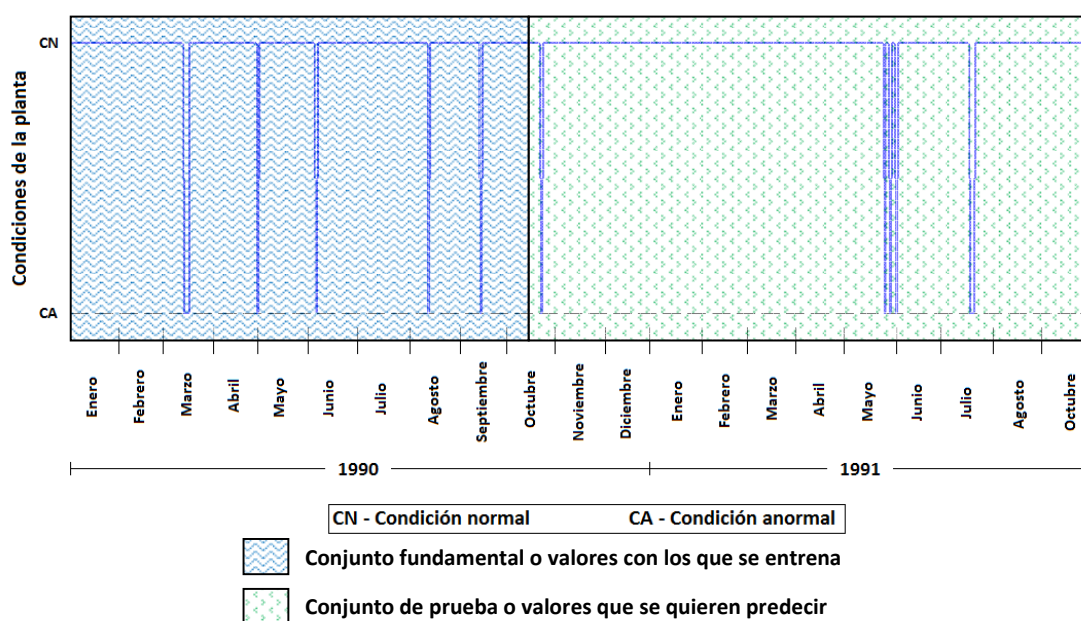


Figura 5.5.1: Conjuntos fundamental y de prueba para la prueba de predicción 4.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

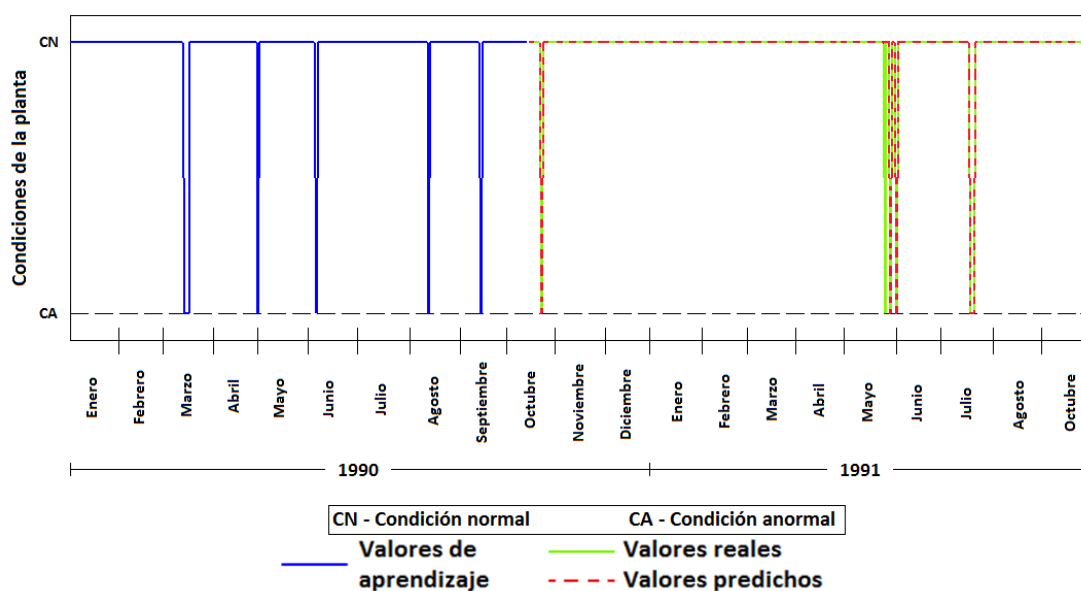


Figura 5.5.2: Predicción realizada con el clasificador Gamma para prueba 4.

Para este ejemplo de un total de 292 días (tamaño del conjunto prueba) se logró una predicción de 291. Siguiendo la *ecuación de rendimiento*, se tiene:

$$\frac{291}{292} \times 100 = 99.6575 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, del clasificador Gamma junto con los algoritmos de clasificación seleccionados del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.5.1: Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis *Weka* para la prueba de predicción 4.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
<b>Gamma</b>	<b>285</b>	<b>6</b>	<b>0</b>	<b>1</b>	<b>291</b>	<b>1</b>	<b>100.00</b>	<b>85.71</b>	<b>99.6575</b>
LibSVM con kernel lineal	285	5	0	2	290	2	100.00	71.43	99.3151
Naïve Bayes	283	7	2	0	290	2	99.30	100.00	99.3151
LibSVM con kernel polinomial	285	4	0	3	289	3	100.00	57.14	98.9726
RBFNetwork	284	5	1	2	289	3	99.65	71.43	98.9726
Multilayer perceptron	285	3	0	4	288	4	100.00	42.86	98.6301
Bayes Net	285	3	0	4	288	4	100.00	42.86	98.6301
1 - Nearest Neighbor	285	2	0	5	287	5	100.00	28.57	98.2877
LibSVM con kernel radial bassis	285	0	0	7	285	7	100.00	0.00	97.6027
LibSVM con kernel sigmoidal	285	0	0	7	285	7	100.00	0.00	97.6027
REP Tree	285	0	0	7	285	7	100.00	0.00	97.6027
3 - Nearest Neighbor	285	0	0	7	285	7	100.00	0.00	97.6027
Random Tree	285	0	0	7	285	7	100.00	0.00	97.6027

## 5.6 Prueba de predicción 5: 250-277

Para la quinta prueba se tomaron 250 días consecutivos para el conjunto fundamental. El conjunto de prueba está conformado con los 277 días restantes.

La información contenida en el conjunto fundamental es la siguiente:

- 242 patrones de aprendizaje para la clase 1 (condición normal).
- 8 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 271 patrones que se esperan predecir para la clase 1 (condición normal).
- 6 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.6.1 muestra de forma gráfica el planteamiento descrito.

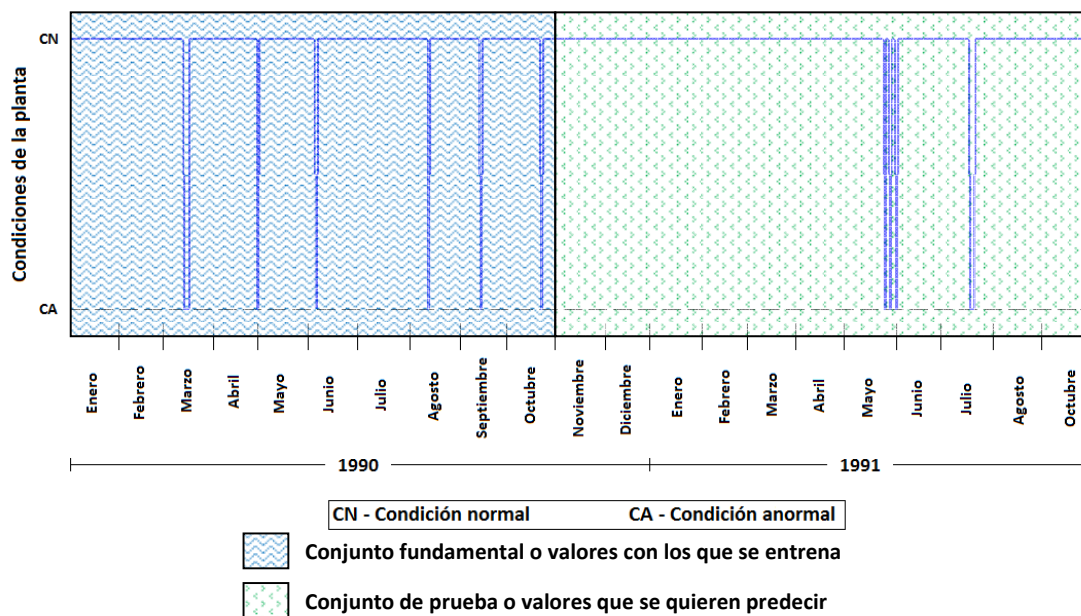


Figura 5.6.1: Conjuntos fundamental y de prueba para la prueba de predicción 5.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

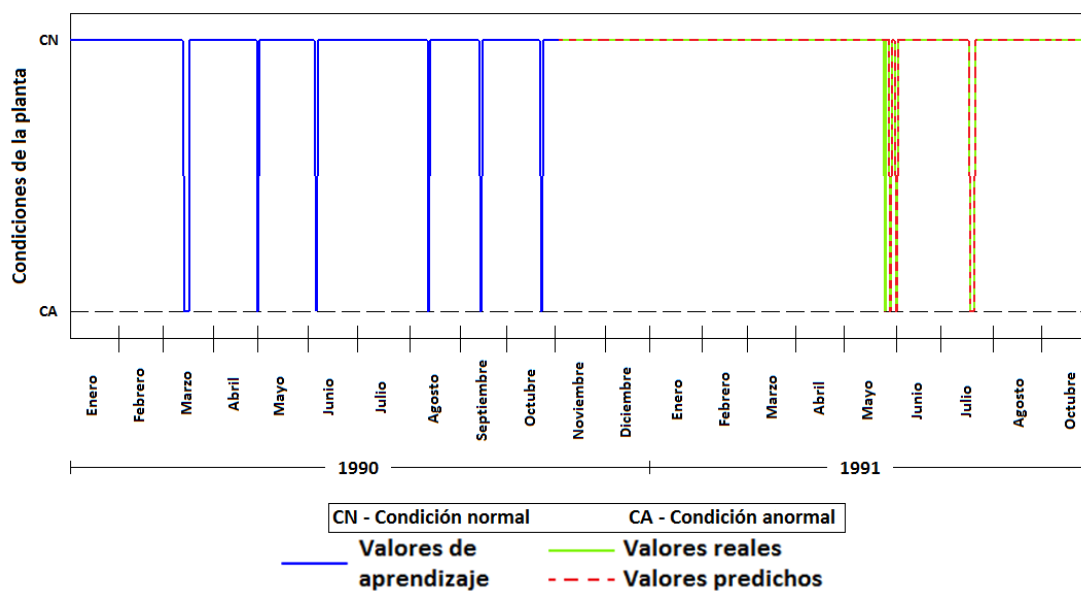


Figura 5.6.2: Predicción realizada con el clasificador Gamma para prueba 5.

Para este ejemplo de un total de 277 días (tamaño del conjunto prueba) se logró una predicción de 276. Siguiendo la *ecuación de rendimiento*, se tiene:

$$\frac{276}{277} \times 100 = 99.6390 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, del clasificador Gamma junto con los algoritmos de clasificación seleccionados del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.6.1: Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis *Weka* para la prueba de predicción 5.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
Multilayer perceptron	271	6	0	0	277	0	100.00	100.00	100.000
Bayes Net	271	6	0	0	277	0	100.00	100.00	100.000
<b>Gamma</b>	<b>271</b>	<b>5</b>	<b>0</b>	<b>1</b>	<b>276</b>	<b>1</b>	<b>100.00</b>	<b>83.33</b>	<b>99.6390</b>
LibSVM con kernel lineal	271	4	0	2	275	2	100.00	66.67	99.2780
Naïve Bayes	269	6	2	0	275	2	99.26	100.00	99.2780
REP Tree	271	3	0	3	274	3	100.00	50.00	98.9170
LibSVM con kernel polinomial	271	3	0	3	274	3	100.00	50.00	98.9170
RBFNetwork	268	6	2	0	274	2	98.89	100.00	98.9170
1 - Nearest Neighbor	271	2	0	4	273	4	100.00	33.33	98.5560
Random Tree	270	2	1	4	272	5	99.63	33.33	98.1949
LibSVM con kernel radial basis	271	0	0	6	271	6	100.00	0.00	97.8339
LibSVM con kernel sigmoidal	271	0	0	6	271	6	100.00	0.00	97.8339
3 - Nearest Neighbor	271	0	0	6	271	6	100.00	0.00	97.8339

## 5.7 Prueba de predicción 6: 422-105

En la última prueba se eligen 422 días consecutivos que conforman el conjunto fundamental, mientras que el conjunto de prueba está conformado con los restantes 105 días.

La información contenida en el conjunto fundamental es la siguiente:

- 413 patrones de aprendizaje para la clase 1 (condición normal).
- 9 patrones de aprendizaje para la clase 2 (condición anormal).

La información contenida en el conjunto de prueba es la siguiente:

- 100 patrones que se esperan predecir para la clase 1 (condición normal).
- 5 patrones que se esperan predecir para la clase 2 (condición anormal).

La figura 5.7.1 muestra de forma gráfica el planteamiento descrito.

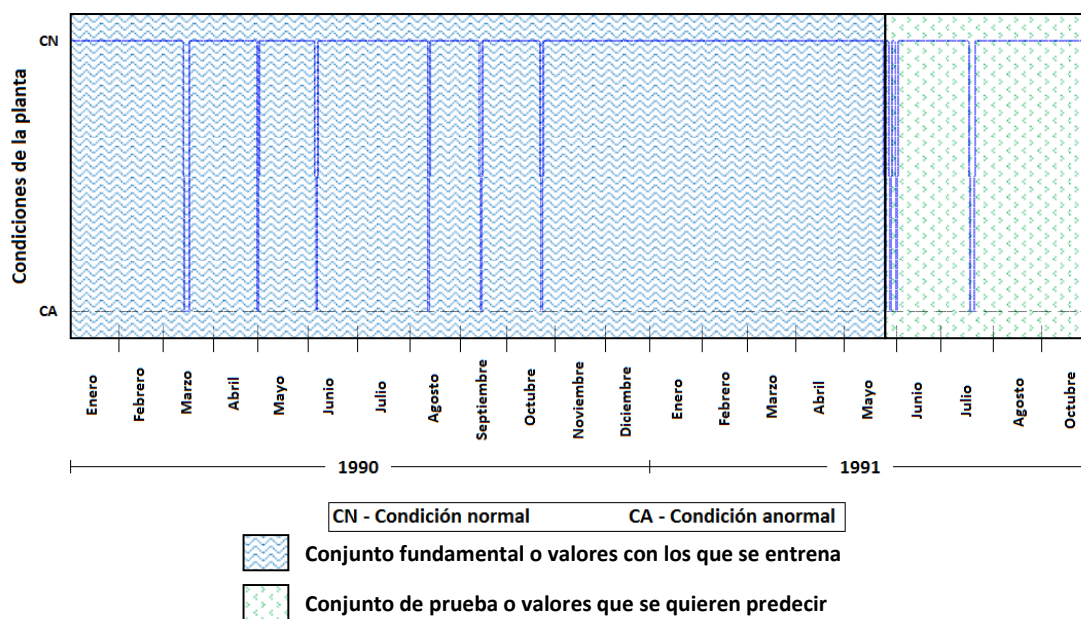


Figura 5.7.1: Conjuntos fundamental y de prueba para la prueba de predicción 6.

La predicción realizada con el clasificador Gamma, de manera gráfica, se muestra en la siguiente figura:

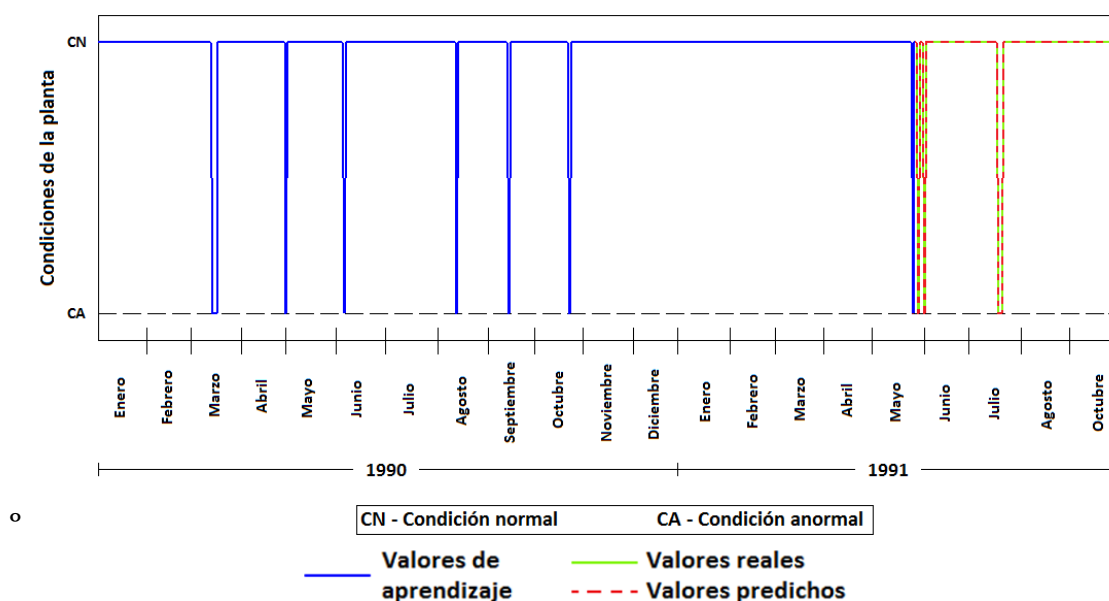


Figura 5.7.2: Predicción realizada con el clasificador Gamma para prueba 6.

Para este ejemplo de un total de 105 días (tamaño del conjunto prueba) se logró una predicción de 105. Siguiendo la *ecuación de rendimiento*, se tiene:

$$\frac{105}{105} \times 100 = 100 \%$$

La tabla siguiente muestra a detalle los resultados obtenidos, para esta prueba, del clasificador Gamma junto con los algoritmos de clasificación seleccionados del entorno de análisis *Weka* que también pueden ser utilizados para la tarea de predicción.

Tabla 5.7.1: Rendimientos entre el clasificador Gamma y algunos algoritmos del entorno de análisis *Weka* para la prueba de predicción 6.

Clasificador	Días bien predichos por clase		Días mal predichos por clase		Predicción general correcta	Predicción general incorrecta	Rendimiento de la predicción (%)		
	C1	C2	C1	C2			C1	C2	General
Bayes Net	100	5	0	0	105	0	100.00	100.00	100.000
<b>Gamma</b>	<b>100</b>	<b>5</b>	<b>0</b>	<b>0</b>	<b>105</b>	<b>0</b>	<b>100.00</b>	<b>100.00</b>	<b>100.000</b>
Naive Bayes	100	5	0	0	105	0	100.00	100.00	100.000
RBFNetwork	100	5	0	0	105	0	100.00	100.00	100.000
Multilayer perceptron	100	3	0	2	103	2	100.00	60.00	98.0952
LibSVM con kernel lineal	100	3	0	2	103	2	100.00	60.00	98.0952
Random Tree	98	5	2	0	103	2	98.00	100.00	98.0952
REP Tree	100	2	0	3	102	3	100.00	40.00	97.1429

LibSVM con kernel polinomial	100	2	0	3	102	3	100.00	40.00	97.1429
1 - Nearest Neighbor	100	2	0	3	102	3	100.00	40.00	97.1429
LibSVM con kernel radial basis	100	0	0	5	100	5	100.00	0.00	95.2381
LibSVM con kernel sigmoidal	100	0	0	5	100	5	100.00	0.00	95.2381
3 - Nearest Neighbor	100	0	0	5	100	5	100.00	0.00	95.2381

## 5.8 Discusión

Haciendo una revisión a los resultados mostrados en este capítulo, se puede observar cual es el comportamiento del rendimiento de predicción realizado tanto por el clasificador Gamma como por los otros algoritmos del entorno de análisis *Weka*.

Enfocándonos primordialmente al clasificador Gamma, conforme se van realizando las pruebas, resalta el hecho de que el rendimiento obtenido tiende a ascender, es decir, el rendimiento de predicción poco a poco se va acercando al 100% o a una predicción correcta. Lo anterior se debe a dos razones principales: la primera es la cantidad de datos que el algoritmo utiliza en su conjunto fundamental, esto es, *mientras más información pasada se tenga (datos de aprendizaje) mejor será la predicción a realizarse*; y la segunda son los valores iniciales que el clasificador Gamma necesita para procesar la información.

### 5.8.1 Discusión: histórico de datos

Con base en la cantidad de datos a utilizar y comparando los resultados mostrados desde la prueba de predicción 1 hasta la prueba de predicción 6, la mejor predicción se da en la última prueba ya que el número de datos históricos utilizados es de casi el 80% del banco de datos total. Por otro lado, la peor prueba de predicción se da en la primera prueba ya que sólo cuenta con un aproximado del 20% de valores del banco de datos para aprender.

Respecto a los algoritmos del entorno de análisis *Weka*, el comportamiento de rendimiento en algunos casos es parecido al del clasificador Gamma, no obstante, existe una variación en el comportamiento.

Las gráficas siguientes muestran el comportamiento de rendimiento del clasificador Gamma y de algunos algoritmos utilizados con *Weka* (los que mejores resultado generan) para cada una de las pruebas de predicción realizadas.



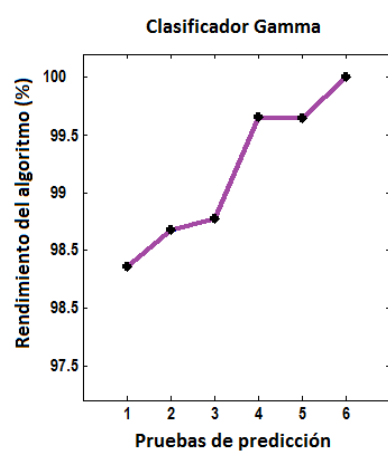


Figura 5.8.1.1: Gráfica de rendimiento del clasificador Gamma.

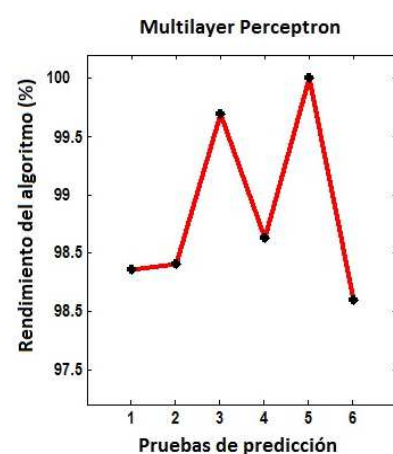


Figura 5.8.1.2: Gráfica de rendimiento del Multilayer Perceptron.

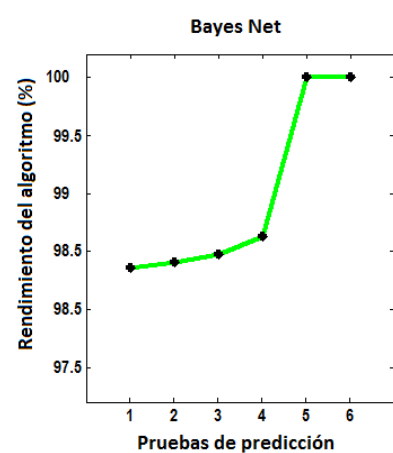


Figura 5.8.1.3: Gráfica de rendimiento del Bayes Net.

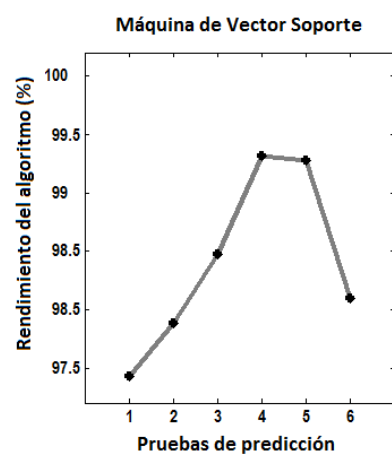


Figura 5.8.1.4: Gráfica de rendimiento de la Máquina de Vector Soporte.

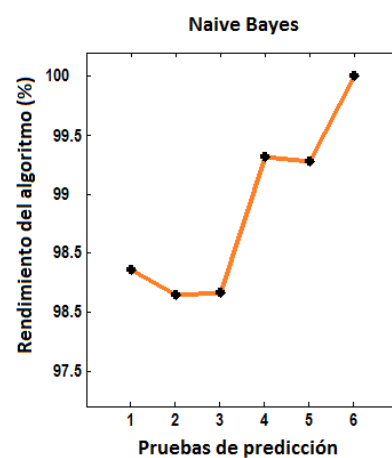


Figura 5.8.1.5: Gráfica de rendimiento del Naive Bayes.

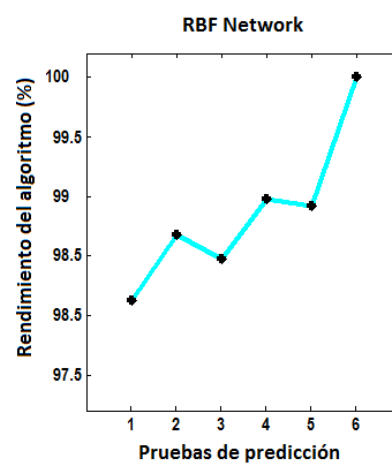


Figura 5.8.1.6: Gráfica de rendimiento del RBF Network.

Tal y como se muestra en las figuras anteriores, el algoritmo del enfoque asociativo Alfa-Beta presenta una mejora más pronunciada conforme las pruebas van en aumento (Figura 5.8.1.1). Los demás algoritmos presentan comportamientos varios. Los algoritmos Bayes Net, Naive Bayes y RBF Network, presentan un comportamiento de rendimiento en ascenso, sin embargo no es tan pronunciado como el del clasificador Gamma; en algunos casos Naive Bayes y RBF Network (Figuras 5.8.1.5 y 5.8.1.6) lo hacen con trabajo, reportando momentos de ascenso y descenso en el rendimiento de las pruebas. En el caso del algoritmo Bayes Net (Figura 5.8.1.3) el rendimiento es de forma pasiva y no es sino hasta la prueba de predicción 5 que su comportamiento se hace notar. El caso del algoritmo Multilayer Perceptron (Figura 5.8.1.2) tiene un comportamiento muy inestable presentando rendimientos buenos y malos conforme el avanzar de las pruebas. Finalmente el algoritmo de la Máquina de Vector Soporte (Figura 5.8.1.4) con kernel lineal presenta un rendimiento de predicción límite; posteriormente va en descenso, cuando debería ser lo opuesto.

## 5.8.2 Discusión: valores iniciales

Con base en la información inicial, extraída del banco de datos, y requerida por el clasificador Gamma para su funcionamiento, se observó que de todos los valores iniciales utilizados, los más importantes y que influyen directamente en el desempeño del algoritmo son: la asignación de pesos a cada rasgo del banco de datos y el grado de similitud.

En cuanto a la asignación de los pesos, estos tuvieron que ser asignados con base en el estado del arte y a la información contenida y observada del banco de datos (ver sub-sección 4.2.1). Los pesos definidos se utilizaron como constantes para todas las pruebas realizadas.

Respecto al grado de similitud, en la sub-sección 4.2.4 se comentó que existen varios valores iniciales para  $\theta$ , no obstante el valor elegido fue de 125. La razón de dicha elección fue la ejecución del clasificador Gamma al menos  $\rho_0$  veces aumentando el valor inicial de  $\theta$  en uno, esto es,  $\theta = \theta + 1$ , siendo el valor inicial por default de  $\theta = 0$ . Lo anterior significa que de las  $\rho_0$  veces que se ejecuta el algoritmo aumentando  $\theta$  al inicio, se observa que el comportamiento varía en el rendimiento de predicción final. La razón de la variación en el rendimiento se explica en la misma sub-sección 4.2.4.

La figura siguiente muestra de forma gráfica el rendimiento del clasificador Gamma empleando el concepto anterior con la prueba de predicción 5.

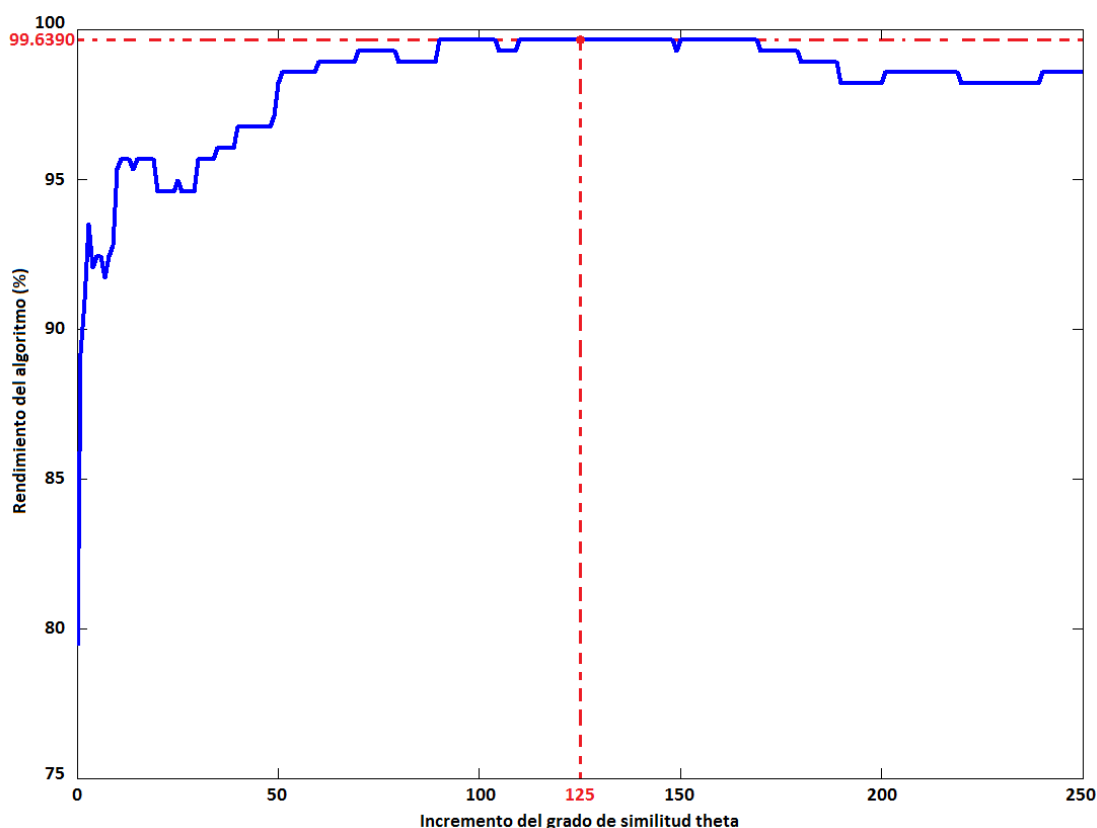


Figura 5.8.2: Comportamiento del clasificador Gamma respecto del grado de similitud.

Como se observa en la figura anterior, el rendimiento de la predicción se incrementa en relación al aumento del grado de similitud, no obstante y al llegar a un tope del rendimiento, el mismo comienza a descender. En esta misma gráfica se puede ver que el valor  $\theta = 125$  no sólo es el único valor ideal que puede ser usado por el clasificador Gamma para realizar la predicción.

### 5.8.3 Discusión: banco de datos

Uno de los inconvenientes del trabajo ya presentado fue la necesidad de recurrir a un banco de datos que tiene más de 20 años de haber sido recopilado y además de origen extranjero. La razón de lo anterior se debe a la indisponibilidad de un banco de datos actual y de origen nacional.

En una primera etapa correspondiente al estado del arte, el autor de este trabajo se vio en la tarea de solicitar a las respectivas dependencias encargadas del tratamiento de aguas residuales, tanto de gobierno como privadas, un banco de datos que abstraiera el comportamiento de una *PDAR*. El resultado de dichas

solicitudes fue en ocasiones de negativas por parte de las dependencias y en otras no existió ninguna respuesta (la mayor parte del tiempo). De acuerdo con el *IMTA* (Instituto Mexicano de Tecnología del Agua) [56] en su división de tratamiento de aguas residuales, no existe un banco de datos nacional y actual que abstraiga la información que se empleó para este documento. Es por ello que un posible trabajo a futuro es generar un banco de datos que pueda ser utilizado con fines como los ya mostrados en este trabajo no solo en beneficio del IPN sino de cualquier comunidad de tipo académico.

## Capítulo 6

# Conclusiones y trabajo futuro

En este capítulo final son presentadas las conclusiones captadas y derivadas del trabajo realizado. Posteriormente se exponen propuestas como trabajos a futuro que podrían realizarse con el fin de expandir y mejorar el presente documento.

### 6.1 Conclusiones

De acuerdo con los objetivos planteados al inicio de este documento, se obtuvieron las siguientes conclusiones:

- De la revisión al estado del arte, los artículos empleados para la realización de este trabajo sirvieron como base tanto para la teoría del contexto en que se desarrolló la tesis (las *PDAR*) como para el enfoque de trabajo al cual se orientó (algoritmo de predicción).
- Se revisó a conciencia la estructura del clasificador Gamma con la finalidad de sacar el mayor provecho a su funcionamiento. Se detectó, al menos para el contexto de trabajo de esta tesis, que es necesario adaptar el algoritmo con base en la información del banco de datos, esto es, algunos valores iniciales deben ser considerados para obtener resultados satisfactorios.
- Al evaluar el desempeño del clasificador Gamma, se observaron desempeños bastante altos. La figura 5.8.1 del capítulo anterior muestra el ascenso del rendimiento del clasificador Gamma conforme las pruebas van avanzando. El promedio del rendimiento de todas las pruebas realizadas está por encima del 99 % de predicción.
- Debido a que los artículos del estado del arte realizan una predicción basada explícitamente en uno o más términos de concentraciones de manera individual, en este trabajo se realiza la predicción empleando la unión de todos los términos de concentraciones que utiliza la planta depuradora; es por ello que se buscó la manera de comparar los resultados del clasificador

---

Gamma definiendo las mismas condiciones utilizadas por el algoritmo, unión de todos los términos de concentraciones, en los algoritmos de clasificación del entorno de análisis *Weka*.

Se cumple el objetivo general ya que se desarrolló, implementó y evaluó un modelo basado en el enfoque asociativo Alfa-Beta con el clasificador Gamma para la predicción de las condiciones de una *PDAR*.

En general el clasificador Gamma, y de acuerdo con los resultados del capítulo anterior, cumple con el planteamiento y los objetivos del trabajo de tesis al realizar la predicción de las condiciones de una *PDAR*. Los resultados arrojados terminan por competir con los resultados de otros modelos utilizados también para el mismo propósito. El clasificador Gamma, al estar dentro del enfoque asociativo Alfa-Beta, termina por ser una opción viable y competitiva frente a modelos matemáticos más añejos y que se encuentran en el gusto de la comunidad científica.

## 6.2 Trabajo a futuro

Los trabajos a futuro que se contemplan son los siguientes:

1. Buscar otro o más bancos de datos que describan el comportamiento de otras *PDAR*. La finalidad es de poner a prueba el clasificador Gamma con esos bancos de datos y observar su comportamiento al momento de realizar la predicción.
2. En caso de no obtener un banco de datos que describa el comportamiento de una *PDAR*, realizar la recopilación de uno con base en normas actuales y de origen mexicano.
3. Buscar maneras para establecer los valores iniciales del algoritmo de forma automática en función de que el banco de datos se adapte al algoritmo y no el algoritmo se adapte al banco de datos.
4. Implementar el análisis de predicción en dispositivos de lógica reconfigurable *FPGA* empleando lenguajes de descripción de hardware *HDL* para realizar una arquitectura completamente basada en el clasificador Gamma.

## Referencias bibliográficas

- [1] Centro Virtual del Agua. Tecnologías del Agua. Tratamiento de Aguas (2013). Disponible en [http://www.agua.org.mx/h2o/index.php?option=com\\_content&view=category&id=51&Itemid=300044](http://www.agua.org.mx/h2o/index.php?option=com_content&view=category&id=51&Itemid=300044).
- [2] Comas, J., Dzeroski, S., Gilbert, K., Rodas, I., & Sànchez-Marrè, M. (2001). Knowledge discovery by means inductive methods in wastewater treatment plant data. *AI communications. The European journal on artificial intelligence*, vol. 14, no. 1, pp: 45-62.
- [3] Comisión Nacional del Agua (CONAGUA). Historia. (2013). Disponible en <http://www.conagua.gob.mx/Contenido.aspx?n1=1&n2=1>.
- [4] Sànchez-Marrè, M. (1995). An Integrated Supervisory Multi-level Architecture for WasteWater Treatment Plants. PhD Thesis, Universidad Politècnica de Catalunya. Barcelona, España.
- [5] Inventario nacional de plantas municipales de potabilización y de tratamiento de aguas residuales en operación. (2010). Disponible en: [http://bva.colech.edu.mx/xmlui/bitstream/handle/1/945/SGAPDS\\_18\\_11InventarioPlantasTratamiento.pdf](http://bva.colech.edu.mx/xmlui/bitstream/handle/1/945/SGAPDS_18_11InventarioPlantasTratamiento.pdf).
- [6] Dixon, M., Gallop, J. R., Lambert, S. C., & Healy, J. V. (2007). Experience with data mining for the anaerobic wastewater treatment process. *Environmental Modelling & Software*, vol. 22, no. 3, pp: 315-322.
- [7] Dixon, M., Gallop, J. R., Lambert, S. C., Lardon L., Healy, J. V., & Steyer, J.-P. (2007). Data mining to support anaerobic WWTP monitoring. *Control Engineering Practice*, vol. 15, no. 8, pp: 987-999.
- [8] Mjalli, F. S., Al-Asheh, S. & Alfadala, H. E. (2007). Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *Journal of Environmental Management*, vol. 83, no. 3, pp: 329-338.



- 
- [9] Cheon, S.-P., Kim, S., Kim, J. & Kim, C. (2008). Learning Bayesian networks based diagnosis systems for wastewater treatment process with sensor data. *Water Science & Technology*, vol. 58, no. 12, pp: 2381-2393.
- [10] Lambert, S. C., Gallop, J. R. & Dixon, M. (2008). Grids in control of anaerobic wastewater treatment plants: Leveraging the knowledge. *Simulation Modelling Practice and Theory*, vol. 16, no. 10, pp: 1546-1560.
- [11] Hanbay, D., Turkoglu, I. & Demir, Y. (2008). Prediction of wastewater treatment plant performance based on wavelet packet decomposition and neural networks. *Expert Systems with Applications*, vol. 34, no. 2, pp: 1038-1043.
- [12] Civelekoglu, G., Yigit, N. O., Diamadopoulos, E. & Kitis, M. (2009). Modelling of COD removal in a biological wastewater treatment plant using adaptive neuro-fuzzy inference system and artificial neural network. *Water Science & Technology*, vol. 60, no. 6, pp: 1475-1487.
- [13] Mesquita, D. P., Dias, O., Amaral, A. L. & Ferreira, E. C. (2009). Monitoring of activated sludge settling ability through image analysis: validation on full-scale wastewater treatment plants. *Bioprocess Biosystems Engineering*, vol. 32, no. 3, pp: 361-367.
- [14] Güçlü, D. & Dursun, Ş. (2010). Artificial neural network modelling of a large-scale wastewater treatment plant operation. *Bioprocess Biosystems Engineering*, vol. 33, no. 9, pp: 1051-1058.
- [15] Singh, K. P., Basant, N., Malik, A. & Jain, G. (2010). Modeling the performance of “up-flow anaerobic sludge blanket” reactor based wastewater treatment plant using linear and nonlinear approaches – A case study. *Analytica Chimica Acta*, vol. 658, no. 1, pp: 1-11.
- [16] Moon, T., Kim, Y., Kim, H., Choi, M. & Kim, C. (2011). Fuzzy rule-based inference of reasons for high effluent quality in municipal wastewater treatment plant. *Korean J. Chem. Eng.*, vol. 28, no. 3, pp: 817-824.
- [17] Dürrenmat, D. J. & Gujer, W. (2011). Identification of industrial wastewater by clustering wastewater treatment plant influent ultraviolet visible spectra. *Water Science & Technology*, vol. 63, no. 6, pp: 1153-1159.
- [18] Vyas, M., Modhera, B., Vyas, V. & Sharma, A. K. (2011). Performance forecasting of common effluent treatment plant parameters by artificial neural network. *ARPJ Journal of Engineering and Applied Sciences*, vol. 6, no.1, pp:38-42.

- 
- [19] Kusiak, A. & Wei, X. (2011). Prediction of methane production in wastewater treatment facility: a data-mining approach. *Annals of Operation Research*, doi: 10.1007/s10479-011-1037-6.
- [20] Lee, J.-W., Suh, C., Hong, Y.-S. T. & Shin, H.-S. (2011). Sequential modeling of a full-scale wastewater treatment plant using an artificial neural network. *Bioprocess Biosystems Engineering*, vol. 34, no. 8, pp: 963-973.
- [21] Kusiak, A. & Wie, X. (2012). A data-driven model for maximization of methane production in a wastewater treatment plant. *Water Science & Technology*, vol. 65, no. 6, pp: 1116-1122.
- [22] Elnekave, M., Celik, S. O., Tatlier, M. & Tufekci, N. (2012). Artificial Neural Network Predictions of Up-Flow Anaerobic Sludge Blanket (UASB) Reactor Performance in the Treatment of Citrus Juice Wastewater. *Polish Journal of Environmental Studies*, vol. 21, no. 1, pp: 49-56.
- [23] Jami, M. S., Husain, I. A. F., Kabashi, N. A. & Abdullah, N. (2012). Multiple Inputs Artificial Neural Network Model For The Prediction Of Wastewater Treatment Plant Performance. *Australian Journal of Basic and Applied Sciences*, vol. 6, no. 1, pp: 62-69.
- [24] Hernández-del-Olmo, F., Llanes, F. H. & Gaudioso, E. (2012). An emergent approach for the control of wastewater treatment plants by means of reinforcement learning techniques. *Expert Systems with Applications*, vol. 39, no. 3, pp: 2355-2360.
- [25] Nasr, M. S., Moustafa, M. A. E., Seif, H. A. E. & Kobrosy, G. E. (2012). Application of Artificial Neural Network (ANN) for the prediction of EL-AGAMY wastewater treatment plant performance-EGYPT. *Alexandria Engineering Journal*, vol. 51, no. 1, pp: 37-43.
- [26] Verma, A. K. & Singh, T. N. (2012). Prediction of water quality from simple field parameters. *Environmental Earth Science*, vol. 69, pp: 821-829.
- [27] Heddiam, S., Bermad, A. & Dechemi, N. (2012). ANFIS-based modeling for coagulant dosage in drinking water treatment plant: a case study. *Environmental Monitoring and Assessment*, vol. 184, no. 4, pp: 1953-1971.
- [28] Yáñez-Márquez, C. (2002). *Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios*. Tesis de Doctorado en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F, México.

- 
- [29] López-Yáñez, I. (2007). Clasificador automático de alto desempeño. Tesis de Maestría en Ciencia de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.
- [30] López-Yáñez, I. (2011). Teoría y aplicaciones del clasificador Gamma. Tesis de Doctorado en Ciencia de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.
- [31] Uriarte-Arcia, Abril. (2012). Procesamiento de datos de monitoreo atmosférico usando clasificación no convencional. Tesis de Maestría en Ciencia de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.
- [32] Steinbuch, K. (1961). Die Lernmatrix. *Kybernetik*, vol. 1, no. 1, pp. 36-45.
- [33] Willshaw, D., Buneman, O. & Longuet-Higgins, H. (1969). Nonholographic associative memory. *Nature*, no. 222, pp. 960-962.
- [34] Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, vol. 14, no. 3-4, pp. 197-220.
- [35] Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers*, vol. C-21, no. 4, pp. 353-359.
- [36] Nakano, K. (1972). Associatron - A model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 2, no. 3, pp. 380-388.
- [37] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554-2558.
- [38] Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 49-60.
- [39] Ritter, G. X., Sussner, P. & Diaz-de-Leon, J. L. (1998). Morphological associative memories. *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 281-293.
- [40] Brapenning, P. J., Thuijsman, F. & Weijters, A. J. M. M. (1995). *Artificial Neural Network: An introduction to ANN theory and practice*. USA: Springer.
- [41] Cheon, S. P., Kim, S., Lee, S. Y. & Lee, C. B. (2009). Bayesian networks based rare event prediction with sensor data. *Knowledge-Based Systems*, vol. 22, no. 5, pp. 336-343.

- 
- [42] Heckerman, D. (1996). A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06. USA: Microsoft Research. Advanced Technology Division. Microsoft Corporation.
- [43] Rokach, L. & Maimon, O. Z. (2005). Clustering methods. In Maimon, O. Z. & Rokach, L. (Eds), *Data Mining and Knowledge Discovery Handbook* pp: 321-352. USA: Springer.
- [44] The European Research Consortium for Informatics and Mathematics (ERCIM). Projects. TELEMAT. (2013). Disponible en <http://www.ercim.eu/activity/projects/telemat.html>
- [45] Vlad, G., Crişan, R., Mureşan, B., Naşcu, I. & Dărab, C. (2010). Development and Application of a Predictive Adaptive Controller to a Wastewater Treatment Process. *Proceedings of the 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, vol. 1, pp: 1-6.
- [46] Yáñez-Márquez, C., López-Yáñez, I. & De-La-Luz-Sáenz-Morales, G. (2008). Analysis and prediction of air quality data with the gamma classifier. *Progress in pattern recognition. Image analysis and applications. Lecture notes in computer science*, vol. 5197, pp: 654-658.
- [47] Han, J. & Kamber, M. (2006). *Data mining. Concepts and techniques*. (2nd. Ed.). USA. Morgan Kaufmann.
- [48] Smith, C. A. & Corripio, A. B. (1991). *Control automático de procesos. Teoría y práctica*. (1er. Ed.). México. Limusa, S.A. de C.V.
- [49] Rosen, K. H. (2004). *Matemática discreta y sus aplicaciones*. (5th Ed.). España. McGraw-Hill.
- [50] Flores-Carapia, R. (2006). *Memorias asociativas Alfa-Beta basadas en el código Johnson-Möbius modificado*. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.
- [51] Yáñez-Márquez, C., Felipe-Riverón, E.M., López-Yáñez, I. & Flores-Carapia, R. (2006). A Novel Approach to Automatic Color Matching. *Progress in Pattern Recognition, Image Analysis and Applications, Lecture Notes in Computer Science* vol. 4225, pp: 529-538.

- 
- [52] Water Treatment Plant Data Set. (2013). Disponible en <http://archive.ics.uci.edu/ml/datasets/Water+Treatment+Plant>.
- [53] Béjar, J., Cortés, U. & Poch, M. (1993). LINNEO+: A Classification Methodology for Ill-structured Domains. Research report RT-93-10-R. Barcelona, España: Dept. Llenguatges i Sistemes Informatics.
- [54] Matlab. Documentation Center (2013). Disponible en <http://www.mathworks.com/help/matlab/>
- [55] Witten, I. H., Frank, E., Hall, M. A. (2011) Data Mining Practical Machine Learning Tools and Techniques. Elsevier.
- [56] IMTA. Servicios tecnológicos: Tratamiento de aguas residuales (2013). Disponible en [http://www.imta.mx/index.php?option=com\\_content&view=article&id=81Itemid](http://www.imta.mx/index.php?option=com_content&view=article&id=81Itemid)