INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

PROCESAMIENTO DE DATOS DE MONITOREO ATMOSFÉRICO USANDO CLASIFICACIÓN NO CONVENCIONAL

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

ABRIL URIARTE ARCIA

DIRECTORES DE TESIS

DR. AMADEO JOSÉ ARGÜELLES CRUZ
DR. CORNELIO YÁÑEZ MÁRQUEZ

MÉXICO, D. F. JUNIO DE 2012

Agradecimientos

Quiero agradecer a mis padres, por su incondicional apoyo y dedicación; sin ellos hubiese sido imposible alcanzar las metas que hasta el día de hoy he alcanzado; han sido un faro que me guía para no perder mi horizonte y un puerto seguro al que siempre quiero regresar; gracias por todo. A mis hermanas, Maya y María Teresa, por ser mis cómplices, por entender mis razones egoistas y a pesar de todos estar siempre a mi lado. A mi abuelita Tere por ser parte integral en mi formación como persona, siempre le estaré agradecida. A mi abuelita Chepita por su eterna preocupación y por sus oraciones.

A todas mis amigos, por sus palabras de aliento. A aquellos que aún en la distancia han estado conmigo y que a pesar de lo difícil que resulta el separarnos, siempre me han impulsado a continuar para alcanzar mis metas. A todas las personas que he conocido en México, por hacer hacerme sentir en casa y evitar que la soledad me invadiera. Todos ustedes han sido un pilar importante para la culminación de este gran esfuerzo.

A mis directores de Tesis por el gran apoyo que me brindaron para la realización de este trabajo. Al Dr. Amadeo José Arguelles Cruz y al Dr. Cornelio Yáñez Márquez por sus consejos, siempre oportunos y por la guía que me han brindado, que hizo posible el que alcanzara mis objetivos. A todos mis compañeros del Laboratorio de Redes Neuronales y Cómputo no Convencional, porque en ellos siempre encontré la mejor disposición para ayudarme cuando lo necesité.

A mis sinodales por sus valiosas aportaciones y críticas, que han enriquecido este trabajo.

Finalmente, se agradece a la Secretaría de Relaciones Exteriores de México, el Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP y CIC), el CONACyT, a la Universidad Nacional de Ingeniería de Nicaragua, el Sistema de Monitoreo Atmosférico de la Ciudad de México (SIMAT) y al ICyTDF (proyectos PIUTE 10-77 y PICSO 10-85), el apoyo recibido en el desarrollo de esta tesis.

Índice de Figuras

Índice de Tablas

Contenido

Resumen

En el presente trabajo de tesis, se diseña e implementa un método para el procesamiento y predicción de datos atmosféricos emanados a través del escape de vehículos automotores, basado en un algoritmo de clasificación no convencional. El clasificador a emplear es el Gamma, el cual pertenece al enfoque asociativo y utiliza los operadores Alfa y Beta, que son a su vez la base de la memorias asociativas Alfa-Beta.

Se realiza el procesamiento y predicción de la concentración de monóxido de carbono (CO)*, bióxido de carbono* (CO_2) *y óxidos nitrosos* (NO_x)*. El banco de datos utilizado para realizar las pruebas del método se representa en forma de una serie de tiempo.*

Con este trabajo, se espera desarrollar una herramienta que contribuya de manera significativa en la disminución de los niveles de contaminación atmosférica generada por los automóviles, una de las principales fuentes de contaminación en la actualidad; este tema es de suma importancia no sólo para México, sino para todo el mundo.

Abstract

In the current document of thesis, a method to process and predict atmospheric data issued through the exhaust pipe of an automobile is designed and implemented, based on an algorithm of non conventional classification. The classifier used is the Gamma classifier, which belongs to the associative approach and uses the Alfa and Beta operators, which are also the base of associative memories Alpha-Beta.

Processing and prediction of the concentration of carbon monoxide (CO)*, carbon dioxide* (CO_2) *and nitrous oxides* (NO_x) *is performed. The data set used for experiments is presented as a time series.*

With this method, it is expected to develop a tool that contributes in a significant way to the decrease of the levels of air pollution generated by automobiles, one of the principal pollution sources in present times; this topic is of great importance not just for Mexico, but for the rest of the world.

Capítulo 1

Introducción

En este trabajo de tesis se presenta un método para la predicción de contaminantes atmosféricos emanados a través del escape de vehículos automotores, utilizando un clasificador no convencional.

1.1 Antecedentes

El aire limpio está compuesto principalmente por nitrógeno y oxígeno y en pequeñas cantidades se puede encontrar vapor de agua y dióxido de carbono. La contaminación atmosférica se define por la presencia de compuestos tóxicos que provocan un daño a la salud humana. Algunos de ellos no son detectados por los sentidos, lo que hace que su exposición pase desapercibida, teniendo efectos inmediatos o a largo plazo [1].

Diversos contaminantes son asociados con efectos adversos a la salud, en especial con enfermedades cardio-pulmonares, siendo estos efectos en ocasiones mortales [2]. Los principales contaminantes relacionados con la calidad del aire son: bióxido de azufre (SO_2), monóxido de carbono (CO), óxidos de nitrógeno (NO_X), partículas suspendidas, compuestos orgánicos volátiles (COV) y ozono (O_3) [3].

Desde hace algunos años el problema de la contaminación ambiental ha tomado una importancia crítica, en especial para las grandes metrópolis, donde el sector industrial y el transporte generan grandes concentraciones de contaminantes atmosféricos. En la Ciudad de México y su área metropolitana, el transporte y la industria tienen la mayor demanda de energía proveniente de combustibles fósiles; por lo que su contribución de emisiones directas a la atmósfera es notable [1].

Se estima que existen 4.5 millones de vehículos automotores en el Distrito Federal y su zona conurbada. El principal aporte de emisiones en la zona metropolitana y Valle de México es generado por el sector transporte y éstas representan el 99% del CO y el 82% de los NO_X que se emiten en la zona [4].

Debido a la importancia de este tema, se han realizado esfuerzos considerables para encontrar métodos que ayuden a prevenir la emisión de contaminantes; desde

este punto de vista, la predicción eficaz y eficiente de los niveles de contaminación resulta un herramienta de gran utilidad para la toma oportuna de decisiones que permitan mitigar el problema. Entre las principales técnicas utilizadas en la actualidad para la predicción de contaminantes se encuentran:

- Redes neuronales: [5] - [16], [19].
- Regresión: [15], [20] - [22].
- Lógica difusa: [23].
- Enfoque neuro - difuso: [10], [18], [22].
- Máquinas de soporte vectorial: [15], [24], [25].
- Enfoque probabilístico - estadístico: [7], [26], [27].
- Árboles de decisión: [7], [15], [26].
- Enfoque asociativo: [28].
- Programación genética: [29].
- Modelos físico - químicos: [30].

Entre las técnicas mencionadas anteriormente para la predicción de contaminantes, se encuentra el enfoque asociativo; el uso del enfoque asociativo no ha sido extenso, pero desde hace algunos años este enfoque ha demostrado su gran potencial para la solución de una amplia variedad de tareas [28], [31] - [34], [47].

El primer modelo de memoria asociativa fue propuesto por Karl Steinbuch en 1961, la *Lernmatrix* [35]; memoria heteroasociativa que puede funcionar como clasificador de patrones binarios. En 1969 Willshaw, Buneman y Longuet-Higgins, presentan el *Correlograph* [36], dispositivo óptico capaz de funcionar como una memoria asociativa.

1972 fue un año de mucha actividad en la investigación sobre memorias asociativa; sobresalen los trabajos de James A. Anderson con su Interactive Memory [37], Teuvo Kohonen presentó sus Correlation Matrix Memories [38], Kaoru Nakano dio a conocer su Associatron [39] y Amari realizó algunos aportes teóricos [40]. Dada la similitud entre los trabajos de Anderson y Kohonen, el modelo recibe el nombre genérico de Linear Associator.

El trabajo de Amari resulta de gran importancia ya que establece un antecedente para uno de los modelos más conocido de memorias asociativas, la memoria de Hopfield [41]. En 1982 el físico John J. Hopfield publicó su artículo, que representa un hito en el desarrollo de las memorias asociativas y redes neuronales, ya que revivió

el interés en este campo de investigación. La memoria Hopfield es autoasociativa y su desempeño es bastante pobre ya que su capacidad de recuperación es sólo de un 15%. En 1988, Bart Kosko presentó un modelo de memoria heteroasociativa con base en el modelo de Hopfield, la memoria asociativa bidireccional BAM [42]. A pesar que este intento por obtener una memoria heteroasociativa fue exitoso, la capacidad de recuperación de este modelo sigue siendo muy baja.

Durante los siguientes años se presentaron una serie de mejoras a modelos ya existentes, pero sin un avance significativo hasta 1998 cuando Ritter et al. [43], presentaron su modelo de memorias asociativas morfológicas, que incorpora conceptos de la morfología matemática y que logra superar las capacidades de aprendizaje de los modelos conocidos hasta ese momento.

Las memorias morfológicas sirvieron de inspiración para el desarrollo de las memorias asociativas Alfa-Beta, creadas en el 2002 por investigadores del Centro de Investigación en Computación del IPN [44]. En el año 2007 investigadores de este mismo centro, presentan un nuevo clasificador de alto desempeño [45], el clasificador Gamma, que utiliza los operadores Alfa y Beta, perteneciente a las memorias asociativas Alfa-Beta, como base para el desarrollo de un nuevo operador de similitud al que se le da el nombre de Gamma y del cual el modelo toma su nombre.

La propuesta de este trabajo de tesis es utilizar el clasificador Gamma para la predicción de la emisión de contaminantes en el escape de vehículos automotores.

1.2 Justificación

La contaminación se ha transformado en un problema de gran importancia a nivel mundial, así como en un factor de gran impacto para la salud humana. El aumento en el número de vehículos automotores en las áreas urbanas es hoy en día uno de los principales factores que contribuyen a los niveles de contaminación atmosférica, por lo que se vuelve de vital importancia el desarrollo de sistemas de predicción que brinden datos confiables sobre la concentración de contaminantes en las emisiones del escape de vehículos automotores. El modelo propuesto en este trabajo de tesis es una herramienta que podrá ayudar, tanto al público en general como a las autoridades, en la toma de decisiones oportunas que ayuden a minimizar los niveles de emisiones y en el desarrollo de planes adecuados de contingencia ambiental.

Según Adamatzky et al. [61], el cómputo no convencional, es la búsqueda de nuevos algoritmos que rompan con los paradigmas tradicionales, así como la implementación física de nuevos y mucho más potentes paradigmas de programación, basados o ins-pirados en los principios de procesamiento de información de sistemas biológicos, químicos, físicos y lógica no tradicional. Algunos de los tópicos que se incluyen en el cómputo no convencional, de acuerdo con [62], son: cómputo cuántico, cómputo químico, cómputo de colisiones y autómatas celulares para cómputo masivo; por mencionar algunos. Los modelos pertenecientes al enfoque asociativo son

emergentes y de uso no generalizado, por lo que se encuentran incluidos dentro del ámbito del cómputo no convencional. En el Laboratorio de Redes Neuronales y Cómputo no Convencional (LRNCC) del Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN) se han desarrollado varios de estos modelos, entre ellos el clasificador Gamma, base fundamental para el desarrollo del presente trabajo.

Diversos métodos han sido utilizados para la predicción de contaminantes; sin embargo el uso del enfoque asociativo para abordar este problema ha sido casi nulo. Dado que las memorias asociativas Alfa-Beta y otros modelos basados en éstas, han demostrado un excelente desempeño en su aplicación a diferentes problemas, se pretende mostrar un modelo basado en estas memorias, el clasificador Gamma, que permita desarrollar un sistema de predicción de contaminantes competitivo con respectos a los existentes en la actualidad.

1.3 Objetivo General

Diseñar, implementar y probar un método basado en algoritmos de clasificación no convencional, que permita realizar el procesamiento y predicción de datos de monitoreo atmosférico, específicamente los relacionados con la emisión de contaminantes a través de los escapes de vehículos automotores.

1.4 Objetivos Específicos

1. Realizar una investigación documental sobre el estado del arte que incluya los aspectos mínimos necesarios relacionados con el procesamiento de datos de monitoreo atmosférico y los sistemas de predicción.

2. Describir las características, ventajas y desventajas de los algoritmos del estado del arte que son utilizados en los sistemas de procesamiento y predicción de datos de monitoreo atmosférico.

3. Investigar sobre las características, aplicaciones y desempeño de los algoritmos de clasificación basados en redes neuronales y cómputo no convencional, haciendo énfasis en el paradigma de los modelos asociativos Alfa-Beta, donde se incluye el Clasificador Gamma.

4. Elegir un conjunto de datos de contaminantes emitidos por motores de combustión interna, a fin de diseñar e implementar un método basado en algoritmos de clasificación no convencional, que permita realizar el procesamiento y predicción de esos datos.

5. Realizar experimentos para probar el nuevo método de predicción, y comparar los resultados experimentales con los reportados en el estado del arte.

1.5 Contribuciones

Un sistema para el procesamiento y predicción de datos de monitoreo atmosférico, emanados a través del escape de vehículos automotores, usando un método de clasificación no convencional.

1.6 Organización del Documento

En este capítulo se describieron los antecedentes, la justificación, los objetivos, contribuciones que aporta el trabajo de tesis y la organización del documento.

El capítulo 2 presenta el estado del arte, donde se muestra un panorama general de la investigación realizada en cuanto a sistemas de monitoreo de contaminantes ambientales se refiere.

En el capítulo 3 se muestran los materiales y métodos, que incluyen los conceptos básicos y las herramientas matemáticas que se requieren para el desarrollo de este trabajo.

El capítulo 4 es la parte principal de la tesis, dado que aquí se presenta, se describe, se ejemplifica y se sustenta teóricamente el método propuesto.

En el capítulo 5 se muestran los experimentos y resultados del nuevo modelo, y un estudio comparativo respecto del rendimiento de otros clasificadores.

El capítulo 6 incluye las conclusiones, las aportaciones y el trabajo a futuro que se propone desarrollar. Finalmente se incluyen las referencias.

Capítulo 2

Estado del Arte

En la actualidad el tema de la contaminación ambiental se ha vuelto un tema de suma importancia a nivel mundial; debido a esto, se han realizado diversos estudios para el desarrollo de técnicas que permitan realizar el monitoreo y la predicción de la concentración de contaminantes presentes en la atmósfera. Entre los principales métodos utilizados en la actualizadad para la predicción de contaminantes, se encuentran:

2.1 Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) son modelos de cómputo inspirados en el funcionamiento del cerebro humano. Existen varios modelos de RNA y a pesar de que puedan existir varias diferencias entre ellos, todos comparten algunas características; cualquier modelo de RNA posee unidades básicas de procesamiento, las neuronas [49].

La estructura de las RNA les permite modelar problemas no lineales muy complejos, lo que las hace apropiadas para su aplicación a problemas de monitoreo y predicción de contaminantes. El uso de RNA, para la predicción de la concentración de contaminantes en el medio ambiente es muy común. Brunelli et al. [6], Kurt et al. [12], Tzima et al. [15], Ozdemir et al. [13] y Zito et al. [19], presentan en sus respectivos trabajos, modelos de RNA para predecir los niveles de contaminación en el ambiente. También se han realizado estudios de la concentración de contaminantes en espacios cerrados usando una RNA [7], con muy buenos resultados. Shakil et al. [14] presentan el uso de una RNA para la predicción de las emisiones de NO_x generadas por la combustión en calderas. Las RNA también ha sido usadas para la predicción de contaminantes generados por las emisiones de motores diesel [5], [9].

Otros trabajos, han presentado técnicas para predecir las emisiones generadas al usar fuentes alternativas de combutibles; Ganapathy et al. [8], presentan en su trabajo una RNA para predecir las emisiones generadas por un motor que utiliza aceite de jatropha como combustible. Kiani et al. [11], realizan el estudio sobre un motor que utiliza una mezcla de etanol y gasolina. Otro ejemplo de la aplicación

de una RNA a la predicción de emisiones generadas por combustibles alternos se presenta en [10], donde el combustible usado es hidrógeno.

Se ha estudiado el uso de redes neuronales en combinación con otros métodos, con el objetivo de mejorar los resultados de las predicciones como en el caso de [5], donde se combinan con algoritmos genéticos. En el trabajo presentado en [14], se utiliza el Análisis de Componentes Principales para reducir el número de variables de entrada a la red, permitiendo que el modelo de RNA sea más sencillo; también se usan algoritmos géneticos para seleccionar los valores más adecuados para los retardos que este modelo utiliza.

En la literatura actual, uno de los modelos de RNA más utilizado para predecir la concentración de contaminantes, son las redes neuronales multicapas feed-forward con backpropagation, pero algunos trabajos [6], [14] y [19], presentan modelos alternativos de RNA como son: redes neuronales recurrentes, redes neuronales de base radial, redes neuronales modulares y redes neuronales dinámicas. Brunelli et al. [6], presentan el uso de una red neuronal Elman, modelo de red neuronal recurrente, para la predicción de las concentraciones máximas de SO_2, O_3, PM_{10}, NO_2 y CO. Shakil et al. [14], utilizan una red neuronal dinámica a la que se le introducen retardos, permitiendole a la red neuronal modelar de forma correcta las emisiones de NO_x generadas por una caldera. Zito et al. [19], comparan el desempeño de 2 modelos de RNA, redes neuronales de base radial y redes neuronales modulares, para predecir la concentración de contaminantes generados por el tráfico vehicular.

2.2 Regresión

Los métodos de regresión son herramientas estadísticas para investigar la relación entre variables; usualmente se busca el efecto que una o varias variables independientes tienen sobre una variable dependiente. La tarea del análisis regresivo es estimar algunos parámetros que caractericen esta relación, a partir de los datos observados del fenómeno que se analiza. Los métodos de regresión son muy comunes para realizar tareas de predicción y pronósticos. En los trabajos presentados en [20], [21] y [22], vemos ejemplos de su aplicación a la predicción de contaminantes.

Chen et al. [20], comparan 2 métodos de regresión para predecir los niveles de NO_x en las emisiones generadas por motores diesel. El objetivo del estudio es comparar el desempeño de la regresión lineal y no lineal en términos de su habilidad para predecir las emisiones de NO_x. De manera general, el modelo que utiliza regresión no lineal presenta mejores resultados pero las diferencias no son sustanciales.

En el trabajo presentado por Pisoni et al. [21], se estudia el uso de un modelo no lineal de autoregresión con variables exógenas (NARX de su nombre en inglés). Este método ha sido usado para pronosticar la calidad del aire haciendo uso de redes neuronales para modelar la no linealidad, mientras que en este trabajo se exploran

las ventajas de usar una expansión polinomial en lugar de una red neuronal. El estudio muestra que el modelo NARX que usa expansión polinomial exhibe resultados similares al NARX que usa redes neuronales.

Polat y Durduran [22], presentan un modelo para la predicción de la concentración diaria de PM_{10}, que combina la regresión lineal con un método de pre-procesamiento de las entradas llamado Output-Dependent Data Scaling. Este método proporciona una mejora significativa a los resultados de la predicciones.

2.3 Lógica Difusa

En este enfoque se crean funciones de pertenencia, las cuales convierten un parámetro medible objetivamente en una pertenencia subjetiva a una categoría; estas categorías en lógica difusa son rangos de valores de un rasgo, que parcialmente coinciden. Posteriormente se usa una regla de conjunción para transformar los valores de pertenencia en una función de discriminación [51]. Debido a la estructura no lineal de los modelos basados en lógica difusa, estos pueden modelar fácilmente complejos sistemas ambientales [16].

Lughofer et al. [23], proponen un sistema que aplica lógica difusa para realizar la predicción de la emisión de NO_x de un motor diesel. En este trabajo se utiliza un enfoque llamado FLEXFIS (FLEXible Fuzzy Inference Systems), el cual automáticamente extrae el número apropiado de reglas y conjuntos difusos.

2.4 Enfoque Neuro - Difuso

Las redes neuronales reconocen patrones y se adaptan por sí mismas, mientras que los sistemas de inferencia difusos incorporan el conocimiento humano, realizan inferencia y toma de decisiones. La integración de estos 2 enfoques, a la par de algunas técnicas de optimización, resultan en una nueva disciplina llamada Neuro - Difusa [54]. Jang presenta su enfoque de esta diciplina en [17], con el nombre de ANFIS (Adaptative Neuro-Fuzzy Inference Systems). ANFIS utiliza una red feedforward para optimizar los parámetros de un sistema de inferencia difuso dado.

El modelo ANFIS ha sido utilizado en [10], [18] y [22], para la predicción de los niveles diarios de contaminación. Noori et al. [18], comparan el desempeño de ANFIS y de una red neuronal para realizar la predicción de la concentración diaria de monóxido de carbono; en este estudio también se muestra el uso de 2 métodos de selección de variables de entrada, Forward Selection (FS) y Gamma Test (GT); el estudio muestra que la combinación de ANFIS con FS y de la red neuronal con FS, son los métodos con mejores resultados.

Polat y Durduran [22], combinan el enfoque Neuro - Difuso con un método de pre-procesamiento de las entradas llamado Output-Dependent Data Scaling, que proporciona una mejora a los resultados de la predicciones. El método es aplicado

para predecir la concentración de los niveles diarios de PM_{10}. Los datos presentados por este estudio muestran que el método de pre-procesamiento mejora de manera significativa los resultados de ANFIS.

Karri et al. [10], aplican ANFIS para la predicción de las emisiones de un motor que usa hidrógeno como combustible; utilizando los datos de la emisiones del motor en la forma de una serie de tiempo. Este método es comparado con los resultados obtenidos por 2 modelos de redes neuronales, siendo el modelo de red neuronal con backpropagation el que muestra los mejores resultados.

2.5 Máquinas de Vectores de Soporte

En la actualidad, las Máquinas de Vectores de Soporte (MVS) han surgido como una herramienta para clasificación, regresión y predicción de series de tiempo. Este método realiza un pre-procesamiento de los datos de entrada, mediante el cual los patrones se representan en un espacio dimensional mucho más grande que el espacio de los rasgos originales. Con un mapeo no lineal $\varphi()$ apropiado a un espacio dimensional lo suficientemente grande, datos de 2 clases siempre podrán ser separados por un hiperplano [51].

Ejemplos de la aplicación de esta metodología para la predicción de contaminantes pueden encontrarse en [15], [24] y [25]. Wang et al. [25], presentan un sistema de MVS en línea para predecir los niveles de contaminantes en el medio ambiente, cuya principal ventaja es la forma de encontrar el modelo óptimo de predicción cuando un nuevo dato es agregado al conjunto de entrenamiento.

Osowski y Garanty [24], combinan las MVS con un método de pre-procesamiento de datos llamado descomposición de wavelets, para realizar la predicción de la concentración diaria de NO_2, CO y SO_2; este pre-procesamiento les permite alcanzar una mayor precisión en las predicciones realizadas por el modelo.

Tzima et al. [15], muestran la aplicación de las MVS para predecir la calidad del aire, especificamente de las concentraciones de PM_{10} y O_3.

2.6 Árboles de Decisión

Este enfoque utiliza clasificación multinivel. Un árbol de decisión cuenta con un nodo raíz, que por convención se encuentra en la parte superior, conectado por medio de ramas a otros nodos; los nodos que ya no tienen ramas son las hojas del árbol. La clasificación de un patrón se inicia en el nodo raíz, donde se pregunta por el valor de un determinado rasgo, las diferentes ramas del nodo corresponden a los posibles valores de la respuesta para este rasgo y se selecciona el siguiente nodo de acuerdo con la respuesta. Este proceso se realiza consecutivamente hasta alcanzar una hoja del árbol, cada hoja tiene la etiqueta de una clase y el patrón es entonces asignado a esa clase [52].

Deleawe et al. [7], presentan el uso de un árbol de decisión para predecir la concentración de CO_2 en espacio cerrados. Tzima et al. [15], utilizan el algoritmo C4.5 para la construcción de un árbol de decisión, que es usado para predecir la concentracion de PM_{10} y O_3.

En el trabajo presentado por Cheon et al. [26], se compara un modelo de redes bayesianas con árboles de decisión para predecir altos niveles de O_3. Los resultados obtenidos por este estudio muestran un mejor desempeño de las redes bayesianas.

2.7 Enfoque Probabilístico - Estadístico

Este enfoque abarca una gran variedad de métodos, de los cuales algunos son utilizados en la actualidad para realizar la predicción de los niveles de contaminación. Algunos de estos métodos son: redes bayesianas, naïve Bayes y modelos de Markov.

Las redes bayesianas son modelos gráficos probabilísticos que se representan como grafos acíclicos dirigidos en los cuales cada nodo representa una variable aleatoria y la falta de un arco entre dos nodos representa la independencia probabilística condicional de esas variables [26]. Cheon et al. [26], comparan una red bayesiana con un árbol de decisión, para predecir valores máximos de O_3; los resultados muestran que las redes bayesianas obtienen mejores resultados. Las redes bayesianas brindan una ventaja adicional; permiten realizar un análisis de causas para determinar los factores que pueden causar el incremento en la concentración de O_3 y poder desarrollar planes de contigencia.

Otro método que se ha aplicado a la predicción de la calidad del aire es el clasificador naïve Bayes; este clasificador probabilístico aplica el teorema de Bayes con la asunción de la independencia entre las probabilidades asignadas a cada rasgo. Deleawe et al. [7], muestran la aplicación de este clasificador a la predicción de los niveles de CO_2 en espacios cerrados.

El modelo oculto de Markov, es un método estadístico que representa secuencias estocásticas donde los estados no son directamente observados sino que son asociados a una función de probabilidad [27]. Dong et al. [27], proponen el uso de un modelo oculto de Markov para la predicción de la concentración de $PM_{2.5}$. El modelo presentado en este estudio incorpora un modificación con respecto a los modelos tradicionales de Markov, que permite incorporar estructuras de tiempo adecuadas para problemas de predicción.

2.8 Enfoque Asociativo

Este enfoque se encuentra basado en el uso de memorias asociativas; el objetivo principal de estas memorias es recuperar patrones completos a partir de patrones de entrada que pueden estar alterados. De esta forma las memorias asociativas pueden

ser vistas como un sistema de entrada salida, donde cada patrón de entrada forma una *asociación* con el correspondiente patrón de salida.

En el trabajo que se presenta en [28], se propone el uso del clasificador Gamma, para la predicción de la concentración de contaminantes en el medio ambiente. Este clasificador utiliza los operadores Alfa y Beta, que son a su vez la base de las memorias asociativas Alfa-Beta. Cabe mencionar que una contribución relevante de este trabajo es la forma en que se codifican los datos, tanto del conjunto de entrenamiento como del conjunto de prueba, para transformar la serie de tiempo original en un conjunto de vectores que puedan ser utilizados por el clasificador.

2.9 Programación Genética

Es una metodología de inteligencia artificial, cuya estrategia de búsqueda se encuentra basada en los algoritmos genéticos. Los algoritmos genéticos utilizan cadenas de bits como cromosomas y se aplica generalmente a problemas de optimización. La programación genética hace uso de las operaciones de: selección, crossover y mutación [29].

En el trabajo presentado por Pires et al. [29], encontramos un modelo basado en programación genética para la predicción de las concentraciones de O_3. En este estudio se aplica el Análisis de Componentes Principales (PCA de su nombre en inglés) a las variables de entrada y luego se aplica programación genética, tanto a las variables originales como a los componentes principales generados por el PCA. Al comparar el modelo de programación genética aplicado a las variables originales contra el mismo modelo aplicado a los componentes generados por el PCA, el primero presenta mejor desempeño durante la etapa de entrenamiento, siendo el caso contrario durante la etapa de prueba.

2.10 Modelos Físico - Químicos

Son modelos que basados en algunas características del motor pueden realizar estimaciones de las emisiones de éste. La formación de los diferentes contaminantes en motores diesel como resultado de la combustión es un proceso complejo que depende de muchas variables; varios modelos físicos y químicos han sido propuestos con este propósito.

En el trabajo presentado por Tinaut et al. [30], se propone un modelo termoquímico para realizar la predicción de las emisiones de un motor que utiliza una mezcla de gas e hidrógeno como combustible. El problema con este tipo de modelos es su complejidad y el hecho de que en ocasiones necesitan parámetros de entrada que son difćiles de medir; para evitar este último punto, algunos parámetros son estimados, lo que puede disminuir la precisión de las predicciones.

2.11 Algoritmos de WEKA

En el capítulo 5, se presentan los resultados obtenidos con el modelo propuesto en este trabajo de tesis y se comparan dichos resultados con algunos algoritmos implementados en WEKA (Waikato Environment for Knowledge Analysis). Por lo que en la presente sección se realiza una descripción de dicho software y de los algoritmos utilizados para la comparación. El contenido de esta sección se encuentra basado en [59]

Weka es una colección de algoritmos de aprendizaje de máquina y herramientas para el pre-procesamiento de datos. Weka incluye los principales métodos para problemas de mineria de datos, tales como: regresión, clasificación, clustering y selección de atributos. Weka permite pre-procesar un banco de datos, introducirlo en un esquema de aprendizaje y analizar los resultados y el desempeño del clasificador resultante.

A continuación, se hace una breve descripción de los algoritmos que se utilizarán en el capítulo 5, para comparar con los resultados obtenidos por el modelo propuesto en este trabajo de tesis.

2.11.1 RepTree

RepTree es un algoritmo de aprendizaje que construye un árbol de decisión usando reducción de ganancia/varianza de la información y poda basada en la reducción del error. Debido a que este algoritmo se encuentra optimizado para ser rápido, ordena los valores numéricos sólo una vez. Los valores perdidos, se manejan dividiendo las instancias existentes en segmentos; al igual que lo hace el C4.5. Los parámetros que pueden ser establecidos son: número mínimo de instancias en cada hoja, la profundidad máxima del árbol y la cantidad de datos usada para la poda.

2.11.2 DecisionStump

Este algoritmo fue diseñado para trabajar en conjunto con algoritmos incrementales; construye árboles de decisión binarios para bancos de datos con clases numéricas o categóricas. Los valores perdidos son tratados como valores separados y se extiende una tercera rama para éstos.

2.11.3 LeastMedSq

LeastMedSq es un método robusto de regresión lineal, que minimiza la mediana (en lugar de la media) de los cuadrados de las divergencias desde la línea de regresión. Aplica regresión lineal estándar de forma repetitiva y presenta la solución que tenga el error cuadrático más pequeño en la mediana.

2.11.4 SimpleLinearRegression

Este algoritmo aprende un modelo de regresión lineal basado en un solo atributo;

seleccionando el atributo que exhiba el menor error cuadrático. Valores perdidos y atributos no numéricos no son permitidos.

2.11.5 MultiLayerPerceptron

MultiLayerPerceptron (MLP) es una red neuronal que se entrena usando backpropagation. Los nodos de la red son sigmoidales, excepto los nodos de salida para clases numéricas que son transformados automáticamente en unidades lineales sin umbral. En Weka este método tiene su propia interfaz de usuario, que permite alterar la estructura de la red. También se pueden ajustar otros parametros como: velocidad de aprendizaje (learning rate), momentum y épocas.

2.11.6 RBFNetwork

Este algoritmo implementa una red neuronal de función de base radial, derivando el centro y la distancia de las neuronas ocultas usando k-means y combinando las salidas obtenidas de la capa oculta usando regresión logística, en caso que la clase de salida sea nominal y regresión lineal si la clase de salida es numérica. La activación de las funciones base, se encuentra normalizada para sumar 1 antes de ser pasadas a los modelos lineales. Es posible especificar k, el número de cluster que se desea utilizar. Si la clase es nominal, k-means es aplicado por separado a cada clase para obtener k clusters en cada clase.

2.11.7 IBk

El IBk es un clasificador de k vecinos más cercanos. Una variedad de algoritmos de búsqueda pueden ser usa-dos para agilizar la tarea de encontrar los vecinos más cercanos: búsqueda lineal (método por defecto), kD-trees, ball-tress y cover-trees. La función de distancia es un parámetro que depende del método de búsqueda, el usado por defecto es la distancia euclidiana, pero existen otros distancias: Chebyshev, Manhattan y Minkowski. Por defecto el número de vecinos es $k = 1$.

2.11.8 LWL

LWL es un algoritmo general para aprendizaje localmente ponderado. Este algoritmo asigna pesos usando un método basado en instancias y construye un clasificador usando las intancias ponderadas. La interfaz para el algoritmo LWL permite seleccionar qué clasificador usar, de los disponibles en weka. Es posible elegir el número de vecinos, esto determina la forma del kernel que se utiliza para asignar los pesos.

2.11.9 ConjuntiveRule

Este algoritmo basado en reglas, aprende una simple regla que puede predecir el valor de una clase, ya sea nominal o numérica. La regla está compuesta por antecedentes unidos por el operador "and" y el consecuente (valor de la clase) para la clasificación/regresión. En este caso, el consecuente es la distribución de las clases (o

la media para valores numéricos). Si la instancia de prueba no se encuentra cubierta por la regla, el valor es predicho usando la distribución o valor de los datos de entrenamiento que tampoco son cubiertos por la regla. La ganancia de información (clase nominal) o reducción de la varianza (clase numérica) se calcula para cada antecedente y las reglas se podan usando el método de reducción del error.

2.11.10 ZeroR

ZeroR es aúun más simple que el algoritmo anterior, pues el valor que predice es la moda (clase nominal) y la media (clase numérica).

Capítulo 3

Materiales y Métodos

En este capítulo se describen los métodos y materiales que se requieren para el desarrollo del modelo que se presentará en el capitulo 4. En la primera sección se describen los operadores Alfa y Beta, ya que son fundamentales para la implementación del clasificador Gamma. La segunda sección describe el operador $\mathcal{U}_\beta$, que también es utilizado por el clasificador Gamma. En la tercera sección se describe la operación de módulo ya que es utilizada por el operador Gamma. La cuarta sección describe el algoritmo del código Johnson-Möbius modificado, ya que los patrones a usar para la clasificación deben estar codificados con este método. En la quinta sección se describe el clasificador Gamma, que se utiliza como propuesta en este trabajo de tesis. Finalmente, en la sección 6 se describe brevemente el banco de datos utilizado para realizar los experimentos.

3.1 Operadores Alfa y Beta

En esta sección se presentan los operados Alfa (α) y Beta (β), ya que son parte fundamental para la implementación del clasificador Gamma, que será utilizado en la propuesta de este trabajo de tesis. Los operadores descritos en esta sección son la base para las memorias asociativas Alfa-Beta, presentadas en [44]. A continuación se presentan las tablas con la definición de los operadores, dados los conjuntos:

$$A = \{0, 1\} \qquad B = \{0, 1, 2\}$$

Tabla 3.1: Definición del operador Alfa

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

Tabla 3.2: Definición del operador Beta

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

3.2 Operador $\mathcal{U}_\beta$

En esta sección se presenta el operador $\mathcal{U}_\beta$, definido en [45], y que también es utilizado por el clasificador Gamma.

Definición 3.2.1 *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 $\mathcal{U}_\beta(x)$ de la siguiente manera: $\mathcal{U}_\beta(x)$ tiene como argumento de entrada un vector binario n-dimensional x y la salida es un número entero no negativo que se calcula así:*

$$\mathcal{U}_\beta(x) = \sum_{i=1}^{n} \beta(x_i, x_i)$$

Ejemplo 3.2.1 *Sea* $x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$*; obtener* $\mathcal{U}_\beta(x)$

Dada la definición 3.2.1, $\mathcal{U}_\beta(x) = \sum_{i=1}^{6} \beta(x_i, x_i)$, por lo que $\mathcal{U}_\beta(x) = \beta(1,1) + \beta(1,1) + \beta(0,0) + \beta(1,1) + \beta(0,0) + \beta(1,1) = 1 + 1 + 0 + 1 + 0 + 1 = 4$. Entonces $\mathcal{U}_\beta(x) = 4$.

Ejemplo 3.2.2 *Sea* $x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$*; obtener* $\mathcal{U}_\beta(x)$

en este caso, $\mathcal{U}_\beta(x) = \sum_{i=1}^{8} \beta(x_i, x_i)$, por lo que $\mathcal{U}_\beta(x) = \beta(1,1) + \beta(0,0) + \beta(0,0) + \beta(1,1) + \beta(1,1) + \beta(1,1) + \beta(0,0) + \beta(1,1) = 1+0+0+1+1+1+0+1 = 5$. Entonces $\mathcal{U}_\beta(x) = 5$.

3.3 Módulo

El contenido de esta sección fue tomado de [50]. El concepto de módulo, junto con el operador de la sección anterior, son relevantes en el desarrollo del clasificador Gamma.

Hay situaciones en las que, al utilizar el operador de división acostumbrado, resulta de más interés el residuo (entero) de dicha operación que el resultado (fraccional) mismo. Para resolver este problema existe el operador *módulo*, denotado por *mod*.

Definición 3.3.1 *Sean a un número entero y m un número entero positivo. Se denota por $a \mod m$ al residuo de dividir a por m.*

Dicho de otra manera, $a \mod m$ es el número entero r tal que $a = qm + r$ y $0 \leq r < m$.

Ejemplo 3.3.1 *Se puede ver que* $17 \mod 5 = 2$ *y* $-133 \mod 9 = 2$.

3.4 Código binario Johnson-Möbius modificado

Dado que los vectores con que trabaja el clasificador Gamma, son codificados usando el código Johnson-Möbius modificado, en esta sección se describe este método, que fue presentado por primera vez en [46].

Algoritmo 3.4.1 *Algoritmo del Código Johnson-Möbius modificado:*

1. Sea un conjunto de números reales

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

   donde n es un número entero positivo fijo.

2. Si uno de los números del conjunto (por ejemplo r_i) es negativo, crear un nuevo conjunto transformado a través de la operación "restar r_i a cada uno de los n números"

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

   donde $t_j = r_j - r_i \forall j \in \{1, 2, ..., n\}$ y particularmente $t_i = 0$. Nota: si hay más de un negativo, se trabaja con el menor.

3. Escoger un número fijo d de decimales y truncar cada uno de los números del conjunto transformado (los cuales son no negativos) precisamente a d decimales.

4. Realizar un escalamiento de $10d$ en el conjunto del paso 3, para obtener un conjunto de n enteros no negativos

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

donde e_m es el número mayor.

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

Ejemplo 3.4.1 *Sea el conjunto* $r = \{2.4, -0.7, 1.826, 1.5\}\,; r \in \mathbb{R}$.

Paso 1: $r = \{2.4, -0.7, 1.826, 1.5\}$.

Paso 2: Existe un número negativo (-0.7)

2.4 - (-0.7) = 3.1
-0.7 - (-0.7) = 0
1.826 - (-0.7) = 2.526
1.5 - (-0.7) = 2.2

por lo que se obtiene el conjunto transformado $t = \{3.1, 0.0, 2.526, 2.2\}$.

Paso 3: Se escoge el número fijo d = 1 para obtener $t = \{3.1, 0.0, 2.5, 2.2\}$.

Paso 4: Se realiza el escalamiento de 10d para obtener $e = \{31, 0, 25, 22\}$, donde $e_m = 31$.

Paso 5: Para cada número e_i del conjunto e, se generan $e_m - e_i$ ceros concatenados con e_i unos.

1. Cada número será codificado con 31 bits.

2. Para el número $e_1 = 31$, se tendrán 31 - 31 = 0 ceros concatenados de 31 unos.

3. Para el número $e_2 = 0$, se tendrán 31 - 0 = 31 ceros concatenados de 0 unos.

4. Para el número $e_3 = 25$, se tendrán 31 - 25 = 6 ceros concatenados de 25 unos.

5. Para el número $e_4 = 22$, se tendrán 31 - 22 = 9 ceros concatenados de 22 unos.

Los códigos correspondientes se muestran en la siguiente tabla.

Tabla 3.3: Ejemplos del código Johnson-Möbius modificado

Número	Códigos Johnson-Möbius modicado
31	11111111111111111111111111111111
0	00000000000000000000000000000000
25	00000011111111111111111111111111
31	00000000011111111111111111111111

3.5 Clasificador Gamma

El clasificador Gamma es un clasificador de patrones de alto desempeño, cuyo algoritmo original fue presentado en [45], y que utiliza el operador Gamma de similitud (γ). El clasificador Gamma también hace uso de los operadores Alfa y Beta, descrito en la sección 3.1; el operador $\mathcal{U}_\beta$, descrito en la sección 3.2 y los patrones codificados en código Johnson-Möbius modicado, cuyo algoritmo fue descrito en la sección 3.4. En esta sección se introduce primero la definición del operador Gamma de similitud y a continuación se presenta el algoritmo del clasificador Gamma modificado, que fue presentado en [48].

3.5.1 Operador Gamma de similitud

El operador Gamma de similitud está basado en las operaciones Alfa y Beta de las memorias asociativas Alfa-Beta; este operador indica si dos vectores son parecidos o no, dado un grado de disimilitud θ; que indica la tolerancia para que dos vectores, al compararlos, sean considerados similares, no obstante que son diferentes [45]. A continuación se define el operador Gamma de similitud.

Definición 3.5.1 *Sean: el conjunto $A = \{0, 1\}$, un número $n \in \mathbb{Z}+, x \in A^n$ y $y \in A^n$ dos vectores binarios n-dimensionales, con la i-ésima componente representada por x_i y y_i, respectivamente, y además, θ 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 θ, y la salida es un número binario que se calcula así:*

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

Ejemplo 3.5.1 *Sea* $x = \begin{pmatrix}1\\0\\1\\1\\0\\1\end{pmatrix}$, $y = \begin{pmatrix}1\\1\\1\\0\\0\\1\end{pmatrix}$ $y\ \theta = 3$ *; calcular* $\gamma(x, y, \theta)$

Paso 1: Para este caso $n = 6$.

Paso 2: Calcular $\alpha(x, y)$ se obtiene $\begin{pmatrix}1\\0\\1\\2\\1\\1\end{pmatrix}$.

Paso 3: Aplicar el módulo 2 a cada componente del vector que obtuvimos en el paso 1, se obtiene el vector $\begin{pmatrix}1\\0\\1\\0\\1\\1\end{pmatrix}$.

Paso 4: Aplicar el operador $\mathcal{U}_\beta$ al vector obtenido en el paso 3. $\mathcal{U}_\beta \begin{pmatrix}1\\0\\1\\0\\1\\1\end{pmatrix} = 4$; que al restarlo de $n = 6$ da como resultado 6 - 4 = 2; $2 \leq \theta$ por tanto $\gamma(x, y, \theta) = 1$

Ejemplo 3.5.2 *Sea* $x = \begin{pmatrix}1\\0\\0\\0\\1\\1\\0\\1\end{pmatrix}$, $y = \begin{pmatrix}1\\0\\1\\1\\1\\0\\1\\0\end{pmatrix}$ $y\ \theta = 1$ *; calcular* $\gamma(x, y, \theta)$

Paso 1: Para este caso $n = 8$.

Paso 2: Calcular $\alpha(x, y)$ se obtiene $\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 2 \\ 0 \\ 2 \end{pmatrix}$.

Paso 3: Aplicar el módulo 2 a cada componente del vector que obtuvimos en el paso 1, se obtiene el vector $\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$.

Paso 4: Aplicar el operador $\mathcal{U}_\beta$ al vector obtenido en el paso 3. $\mathcal{U}_\beta \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = 3$; que al restarlo de $n = 8$ da como resultado 8 - 3 = 5; $5 \notin \theta$ por tanto $\gamma(x, y, \theta) = 0$

3.5.2 Algoritmo del Clasificador Gamma

En el trabajo presentado en [48], se caracteriza la operación del clasificador Gamma y se define el conjunto fundamental ideal para este clasificador.

Definición 3.5.2 *Sea el conjunto fundamental del clasificador Gamma el conjunto de patrones asociados a una clase, de la forma* $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \,|\mu = 1, 2, \ldots, p\}$*; donde* $\mathbf{x}^\mu$ *es un patrón y* $\mathbf{y}^\mu$ *es su clase correspondiente. Además, para este conjunto fundamental se cumplen las siguientes tres afirmaciones:*

$$\mathbf{x}^i \neq \mathbf{x}^j \ \forall i, j \in \{1, 2, \ldots, p\} tal\ que\ i \neq j$$

Esto implica que no hay patrones repetidos.

$$\mathbf{x}^i = \mathbf{x}^j \implies \mathbf{y}^i = \mathbf{y}^j \ \forall i, j \in \{1, 2, \ldots, p\}$$

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

$$\mathbf{y}^i \neq \mathbf{y}^j \implies \mathbf{x}^i \neq \mathbf{x}^j \ \forall i, j \in \{1, 2, \ldots, p\}$$

Clases diferentes tienen asociados patrones diferentes.

Dicho de otra manera, el conjunto fundamental debe inducir 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.

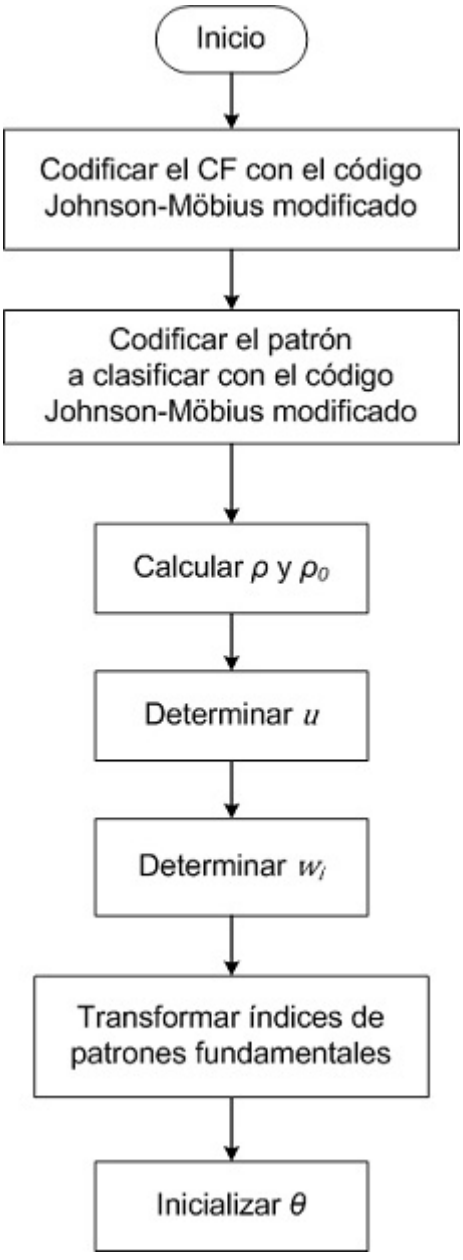

Figura 3.1: Diagrama de bloques del algoritmo del clasificador Gamma, primera parte

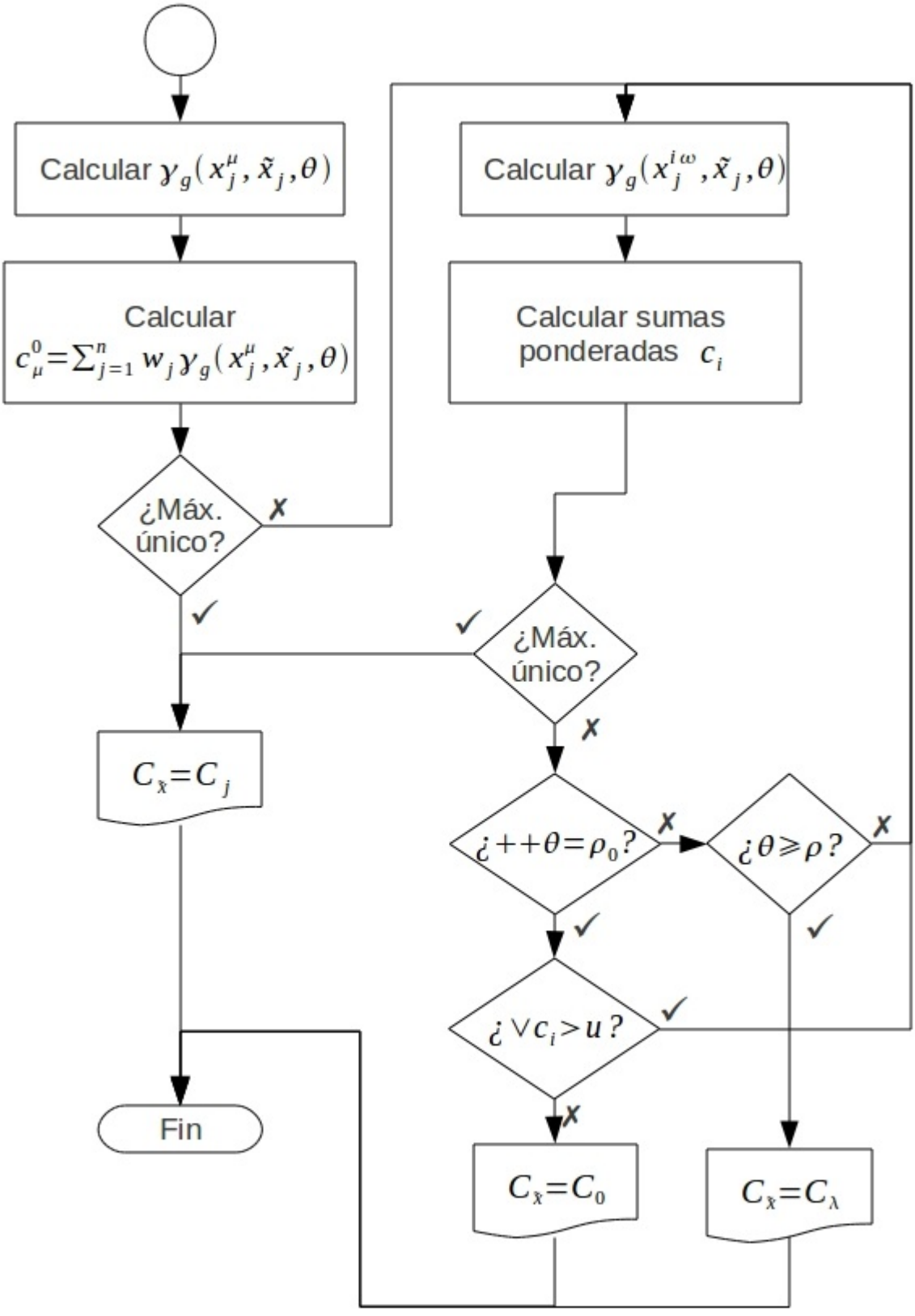

Figura 3.2: Diagrama de bloques del algoritmo del clasificador Gamma, segunda parte

Algoritmo 3.5.1 *Sea el conjunto fundamental del clasificador Gamma de acuerdo con la definición 3.5.2. Al presentarse un patrón a clasificar* $\tilde{\mathbf{x}}$*, donde* $\tilde{\mathbf{x}}$ *es un vector real n-dimensional* $\tilde{\mathbf{x}} \in \mathbb{R}^n$*, con* $n \in \mathbb{Z}^+$*, se realiza lo siguiente:*

1. Codificar cada componente de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado, restando el valor menor a todos los valores por cada componente; asimismo, se obtiene un valor $e_m = \bigvee_{i=1}^{p} x_j^i$ por cada componente. Con lo anterior se desplaza el rango de cada componente para que vaya de 0 a e_m. Así, la componente x_j^i se transforma en un vector binario de dimensión $e_m(j)$.

2. Codificar cada componente del patrón a clasificar con el código Johnson-Möbius modificado, utilizando las mismas condiciones que se utilizaron para codificar las componentes de los patrones fundamentales. En caso de que alguna componente del patrón a clasificar sea mayor al e_m correspondiente $(\tilde{\mathbf{x}}_\xi > e_m(\xi))$, 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 $e_m(\xi) + |\tilde{\mathbf{x}}_\xi|$ a $mgamma_\xi$.

3. Calcular el parámetro de paro ρ y el parámetro de pausa ρ_0. Dependiendo del problema a tratar, algunas posibilidades sugeridas para estos parámetros son las siguientes:

   - $\rho = \bigwedge_{j=1}^{n} \left(\bigvee_{i=1}^{p} x_j^i \right)$.
   - $\rho = \frac{1}{n} \sum_{j=1}^{n} \left(\bigvee_{i=1}^{p} x_j^i \right)$.
   - $\rho = \bigvee_{j=1}^{n} \left(\bigvee_{i=1}^{p} x_j^i \right)$.
   - $\rho_0 = \bigwedge_{j=1}^{n} \left(\bigvee_{i=1}^{p} x_j^i \right)$, sobre todo si $\rho = \bigvee_{j=1}^{n} \left(\bigvee_{i=1}^{p} x_j^i \right)$.
   - $\rho_0 > \rho$, 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 $w_i \in \mathbb{R}^+ | i = 1, 2, \ldots, n$. 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 transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo $\mathbf{x}^\mu$, se convierta en dos índices: uno para la clase (por ejemplo la clase i) y otro para el orden que le corresponde a ese patrón dentro de esa clase (por ejemplo ω). Bajo estas condiciones ejemplificadas, la notación para el patrón $\mathbf{x}^\mu$ será ahora, con la transformación, $\mathbf{x}^{i\omega}$. Lo anterior se realiza para todos los patrones del conjunto fundamental.

7. Inicializar θ a 0.

8. Realizar la operación $\gamma_g\left(x_j^\mu, \tilde{x}_j, \theta\right)$ para cada componente de cada uno de los patrones fundamentales y del patrón a clasificar, considerándose $mgamma_\xi$ como la dimensión del patrón binario $\tilde{\mathbf{x}}_\xi$, en caso necesario.

9. Calcular la suma ponderada inicial c_i^0 de los resultados obtenidos en el paso 8, para cada patrón fundamental $\mu = 1, 2, \ldots, p$:

$$c_\mu^0 = \sum_{j=1}^{n} w_j \cdot \gamma_g\left(x_j^\mu, \tilde{x}_j, \theta\right)$$

10. Si existe un máximo único, cuyo valor es además igual a n, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{\mathbf{y}} = \mathbf{y}^j \text{ tal que } \bigvee_{\mu=1}^{p} c_\mu^0 = c_j^0 = n$$

    De lo contrario, continuar.

11. Realizar la operación $\gamma_g\left(x_j^{i\omega}, \tilde{x}_j, \theta\right)$ para cada clase y para cada componente de cada uno de los patrones fundamentales que corresponden a esa clase, y del patrón a clasificar, considerándose $mgamma_\xi$ como la dimensión del patrón binario $\tilde{\mathbf{x}}_\xi$ si es necesario.

12. Calcular la suma ponderada c_i de los resultados obtenidos en el paso 11, para cada clase $i = 1, 2, \ldots, m$:

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} w_j \cdot \gamma_g \left(x_j^{i\omega}, \tilde{x}_j, \theta \right)}{k_i}$$

13. Si existe más de un máximo entre las sumas ponderadas por clase, incrementar θ 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 paro: $\theta \geq \rho$.

14. 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:

$$C_{\tilde{\mathbf{x}}} = C_0$$

    (b) Si $\bigvee_{i=1}^{m} c_i > u$ entonces se continua en el paso 11.

15. Si existe un máximo único, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{\mathbf{y}} = \mathbf{y}^j \text{ tal que } \bigvee_{i=1}^{m} c_i = c_j$$

16. En caso contrario: si λ es el índice más pequeño de clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase $\tilde{\mathbf{y}} = \mathbf{y}^{\lambda}$.

3.6 Conjunto de Datos

En la seccion 5.1 se brindará una descripción más detallada del banco de datos que será utilizado para los experimentos de este trabajo de tesis. Podemos observar en la tabla 5.2, que los datos de las emisiones de contaminantes son capturados en la forma de una serie de tiempo; estos datos deberán ser codificados en forma de vectores que puedan ser utilizados por el clasificador Gamma.

Para evitar problemas potenciales con respecto a la no estacionalidad de los datos capturados, se trabaja con las diferencias entre las muestras consecutivas [55] - [57]. Las diferencias son calculadas después de aplicar los pasos 3 y 4 del código Johnson-Möbius modificado, descrito en la sección 3.4. La tabla 3.4 muestra un ejemplo de este procedimiento.

Tabla 3.4: Ejemplo del cálculo de diferencias

	Datos Originales	Datos Escalados	Datos Truncados	Diferencias
1	11.53	115.3	115	
2	11.63	116.3	116	1
3	11.69	116.9	116	0
4	11.76	117.6	117	1
5	11.88	118.8	118	1
6	11.91	119.1	119	1
7	10.80	108.0	108	-11
8	11.34	113.4	113	5
9	11.96	119.6	119	6
10	11.71	117.1	117	-2
11	11.84	118.4	118	1

Los datos tomados de una serie de tiempo serán codificados en k patrones de dimensión n representados de la siguiente forma:

$$x^k = \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \\ . \\ . \\ . \\ x_n^k \end{pmatrix}$$

donde

x^k es el k-ésimo patrón
x_n^k es el n-ésimo valor del patrón x^k

3.6.1 Conjunto de Entrenamiento

El conjunto de entrenamiento estará formado por patrones de dimensión n, que contendrán los valores de una serie de tiempo correspondiente a una prueba de manejo, en la cual se tomaron p muestras de emisiones. Si se desea codificar una serie de tiempo en vectores de dimensión n = 5, el primer patrón contendrá los valores de los índices 0 al 4, el segundo patrón los valores de los índice 1 al 5 y así sucesivamente hasta llegar al último patrón que contendrá los valores de los índices x_{p-4} al x_p. La tabla 3.5 muestra la estructura de una serie de datos, mientras que la tabla 3.6 muestra la forma en que se organizan los patrones correspondientes a la serie de datos de la tabla 3.5.

Tabla 3.5: Serie de datos para el conjunto de entrenamiento

índice	valor
0	x_0
1	x_1
2	x_2
3	x_3
$\vdots$	$\vdots$
$p-4$	x_{p-4}
$p-3$	x_{p-3}
$p-2$	x_{p-2}
$p-1$	x_{p-1}
p	x_p

Tabla 3.6: Codificación de patrones de entrenamiento

$$x^0 = \begin{pmatrix} x_0^0 \\ x_1^0 \\ x_2^0 \\ x_3^0 \\ x_4^0 \end{pmatrix} \qquad x^1 = \begin{pmatrix} x_1^1 \\ x_2^1 \\ x_3^1 \\ x_4^1 \\ x_5^1 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} x_2^2 \\ x_3^2 \\ x_4^2 \\ x_5^2 \\ x_6^2 \end{pmatrix} \qquad x^k = \begin{pmatrix} x_{p-4}^k \\ x_{p-3}^k \\ x_{p-2}^k \\ x_{p-1}^k \\ x_p^k \end{pmatrix}$$

Al codificar una serie de datos de la forma antes mencionada, el número de patrones para el conjunto de entrenamiento, está dado por:

$$\textit{Total de Patrones} = p - n$$

donde

p es el número de datos de la serie de tiempo
n es la dimensión de los patrones

3.6.2 Conjunto de Prueba

Los patrones en el conjunto de prueba tendrán la misma dimensión n que los patrones del conjunto de entrenamiento y contendrán datos del mismo contaminante que se utilizó en el conjunto de entrenamiento, pero de una prueba de manejo distinta a la utilizada durante la fase de entrenamiento. La codificación del conjunto de prueba será igual que la del conjunto fundamental pero los patrones se denotarán por $\tilde{x}$. La tabla 3.7 muestra la estructura de una serie de datos, mientras que la tabla 3.8 muestra la forma en que se organizan los patrones correspondientes a la serie de datos de la tabla 3.7.

Tabla 3.7: Serie de datos para el conjunto de prueba

índice	valor
0	$\tilde{x}_0$
1	$\tilde{x}_1$
2	$\tilde{x}_2$
$\vdots$	$\vdots$
$p-4$	$\tilde{x}_{p-4}$
$p-3$	$\tilde{x}_{p-3}$
$p-2$	$\tilde{x}_{p-2}$
$p-1$	$\tilde{x}_{p-1}$
p	$\tilde{x}_p$

Tabla 3.8: Codificación de patrones de prueba

$$\tilde{x}^0 = \begin{pmatrix} \tilde{x}_0^0 \\ \tilde{x}_1^0 \\ \tilde{x}_2^0 \\ \tilde{x}_3^0 \\ \tilde{x}_4^0 \end{pmatrix} \qquad \tilde{x}^1 = \begin{pmatrix} \tilde{x}_1^1 \\ \tilde{x}_2^1 \\ \tilde{x}_3^1 \\ \tilde{x}_4^1 \\ \tilde{x}_5^1 \end{pmatrix}$$

$$\tilde{x}^2 = \begin{pmatrix} \tilde{x}_2^2 \\ \tilde{x}_3^2 \\ \tilde{x}_4^2 \\ \tilde{x}_5^2 \\ \tilde{x}_6^2 \end{pmatrix} \qquad \tilde{x}^k = \begin{pmatrix} \tilde{x}_{p-4}^k \\ \tilde{x}_{p-3}^k \\ \tilde{x}_{p-2}^k \\ \tilde{x}_{p-1}^k \\ \tilde{x}_p^k \end{pmatrix}$$

3.6.3 Clases

Para la aplicación del clasificador Gamma a la tarea de predicción, las clases estarán codificadas como un vector unidimensional. Consideremos el i-ésimo patrón x^i, que se forma de una serie de datos; el patrón estará formado desde el valor i hasta el valor $i+n-1$ de la serie de datos. La clase correspondiente a x^i, denotada por C_i, será el valor que sigue al último elemento de x^i, es decir que C_i es igual al valor $i+n$. La figura 3.3 muestra un ejemplo de la forma en que son asignadas las clases.

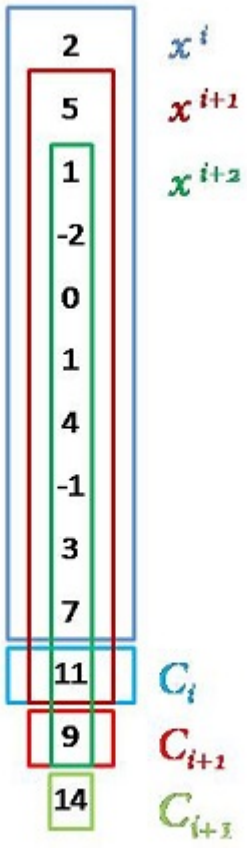

Figura 3.3: Ejemplo de asignación de clases

Capítulo 4

Solución Propuesta

En este capítulo se describe la metodología utilizada para realizar la predicción de los contaminantes emanados a través del escape de un vehículo automotor. En la primera sección se realizan algunas consideraciones generales sobre el modelo. En la segunda sección se presentan los aportes y simplificaciones, que este trabajo de tesis introduce al modelo del clasificador Gamma.

4.1 Consideraciones Iniciales

El algoritmo del clasificador Gamma incluye de 2 condiciones de paro: parámetro de paro y parámetro de pausa. El parámetro de paro ρ garantiza la convergencia del algoritmo; garantizando que después de un determinado número de iteraciones el proceso de clasificación se detenga y el patrón a clasificar sea asignado a una clase que corresponda a uno de los máximos. El parámetro de pausa permite detener el proceso de clasificación y asignar el patrón a clasificar a la clase desconocida, cuando no existe suficiente similitud con patrones conocidos.

Ambas condiciones de paro se obtienen de forma automática a partir de los patrones de aprendizaje y no toman en cuenta el comportamiento que puede tener al clasificador ante ciertos tipos de datos.

4.2 Enriquecimiento y Simplificación del Clasificador Gamma

4.2.1 Condiciones de paro

Existen bancos de datos en donde la gran similitud entre sus patrones, provoca una ambigüedad que hace difícil la tarea de los clasificadores. En este caso en particular vamos a analizar el comportamiento del clasificador Gamma ante esta clase de datos y la solución propuesta para esta situación, que representa un aporte original de este trabajo de tesis. Consideremos el siguiente ejemplo:

Ejemplo 4.2.1 *Sea el conjunto fundamental integrado por los patrones*

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

donde $C_1 = \{x^1\}$, $C_2 = \{x^2\}$, $C_3 = \{x^3\}$, $C_4 = \{x^4\}$ *y* $C_5 = \{x^5\}$.

Clasificar el patrón $y = \begin{pmatrix} 2 \\ 0 \\ -2 \\ 0 \\ -1 \\ 0 \\ 2 \\ -1 \\ -2 \\ 1 \end{pmatrix}$.

Para $\theta = 0$

$c_1 = 2$, $c_2 = 4$, $c_3 = 1$, $c_4 = 4$ y $c_5 = 2$, suma máxima no es única, incrementar θ.

Para $\theta = 1$

$c_1 = 4$, $c_2 = 9$, $c_3 = 10$, $c_4 = 9$ y $c_5 = 10$, suma máxima no es única, incrementar θ.

Como se puede observar en este ejemplo, c_3 y c_5 alcanzaron el valor máximo posible que es n, es decir el número de rasgo de los patrones; en este caso el patrón a clasificar se parece tanto a x_3 como a x_5 en todos sus rasgos, presentándose una situación ambigua para la clasificación. A partir de este punto la ambigüedad sólo irá en aumento, pues las clases cuyas sumas c_i ya alcanzaron el valor de n mantendrán dicho valor y es posible que al seguir incrementando θ la suma c_i de otras clases también alcancen este máximo, dificultando aún más la decisión de a qué clase asignar el patrón a clasificar.

Dado que la suma máxima no es única, el clasificador Gamma continuará incrementando θ, hasta alcanzar el parámetro de pausa o el parámetro de paro y

dependiendo del valor del umbral u, el patrón será clasificado en la clase desconocida o se clasificará en alguna de las clases para la cual c_i es máxima al alcanzar el parámetro de paro. Cualquiera de las 2 situaciones antes mencionadas son no deseables pues afectan la eficiencia y/o eficacia del clasificador.

Para atacar esta situación y tratar de minimizar su efecto sobre la clasificación, se propone introducir una nueva condición de paro cuando se cumplan las siguientes condiciones:

1. El máximo de las sumas ponderadas c_i no es único.
2. El máximo de las sumas ponderadas $c_i = n$.

Si ambas condiciones se cumplen, entonces se procede a realizar la clasificación del patrón con la información disponible, es decir usando las clases para las cuales el máximo de las sumas ponderadas $c_i = n$. Durante la fase de experimentación de este trabajo de tesis se realizaron pruebas con 2 formas diferentes de utilizar las clases antes mencionadas para la clasificación:

1. Sea σ el índice de clases más pequeño que corresponde a uno de los máximos, asignar al patrón a clasificar la clase $\tilde{\mathbf{y}} = \mathbf{y}^{\sigma}$.
2. Dado que la aplicación del clasificador Gamma, que se estudia en el presente trabajo es la predicción; no resulta indispensable asignar el patrón a una sola clase, por lo que también se realizaron pruebas sacando el promedio de todas las clases para las cuales el máximo de las sumas ponderadas $c_i = n$

Se realizaron experimentos utilizando ambas posibilidades, siendo la segunda opción la que presentó mejores resultados. La gráfica 4.1 muestra la comparación de los valores predichos contra los datos reales, utilizando el clasificador Gamma tal como se define en la seccion 3.5.2; mientras que la gráfica 4.2 muestra la comparación de los valores predichos contra los datos reales, utilizando el clasificador con el ajuste propuesto en la presente sección. Se puede observar que la curva de la gráfica 4.2 muestra un mejor ajuste a los datos reales, que la curva de la gráfica 4.1. La tabla 4.1 muestra algunas comparaciones de los resultados del clasificador Gamma sin modificaciones y del mismo clasificador con el ajuste propuesto en esta sección.

Tabla 4.1: Comparación del clasificador Gamma, con y sin ajustes

	C.F.	C.P.	Gamma Original (RMSE)	Gamma modificado (RMSE)
1	CARB_ICTx_test1	CARB_ICTx_test2	2.47	0.73
2	CARB_ICTx_test3	CARB_ICTx_test4	2.40	1.90
3	CARB_ICTx_test 5	CARB_ICTx_test6	1.10	0.86

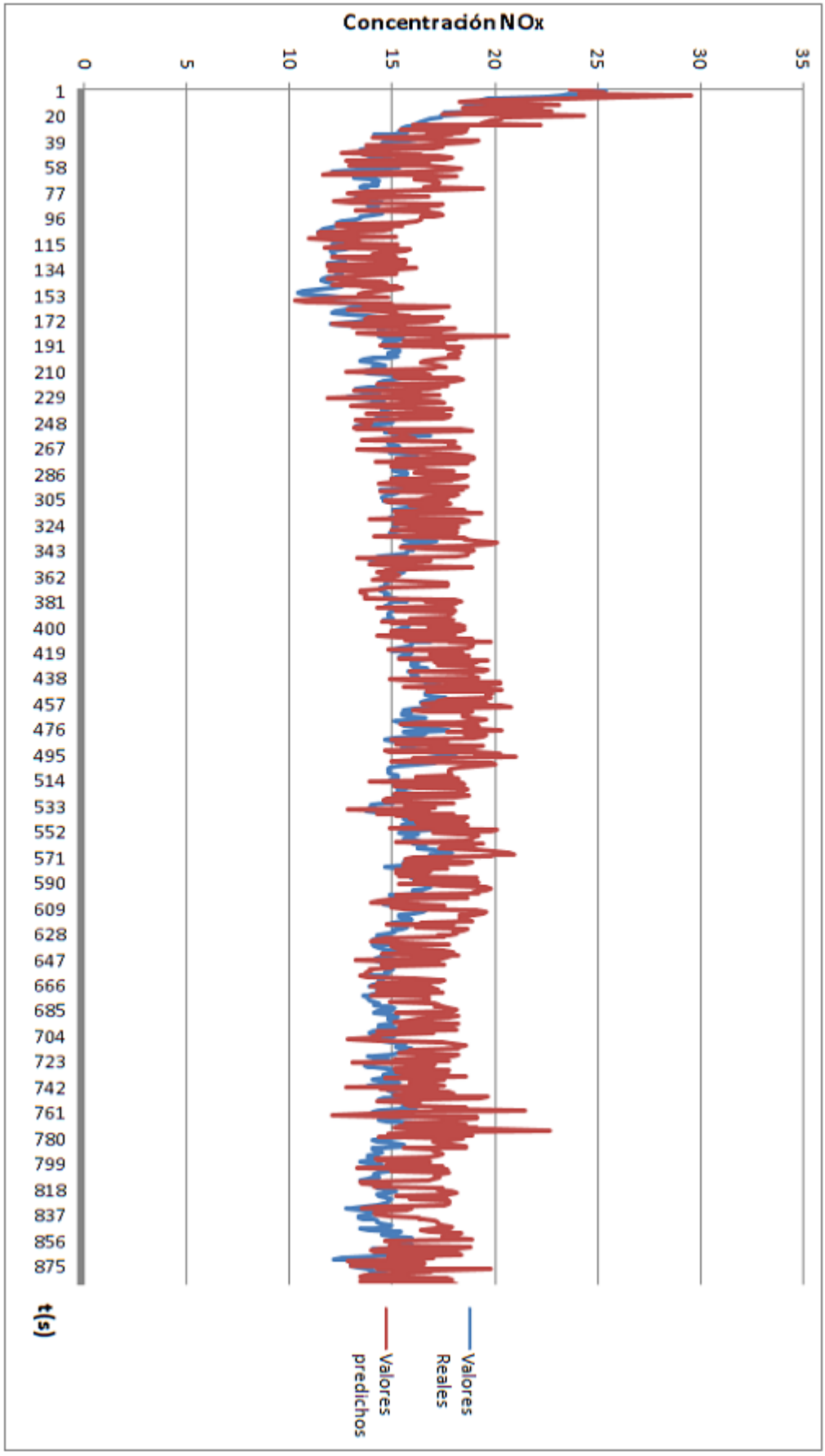

Figura 4.1: Clasificación de NO_x con el clasificador Gamma original

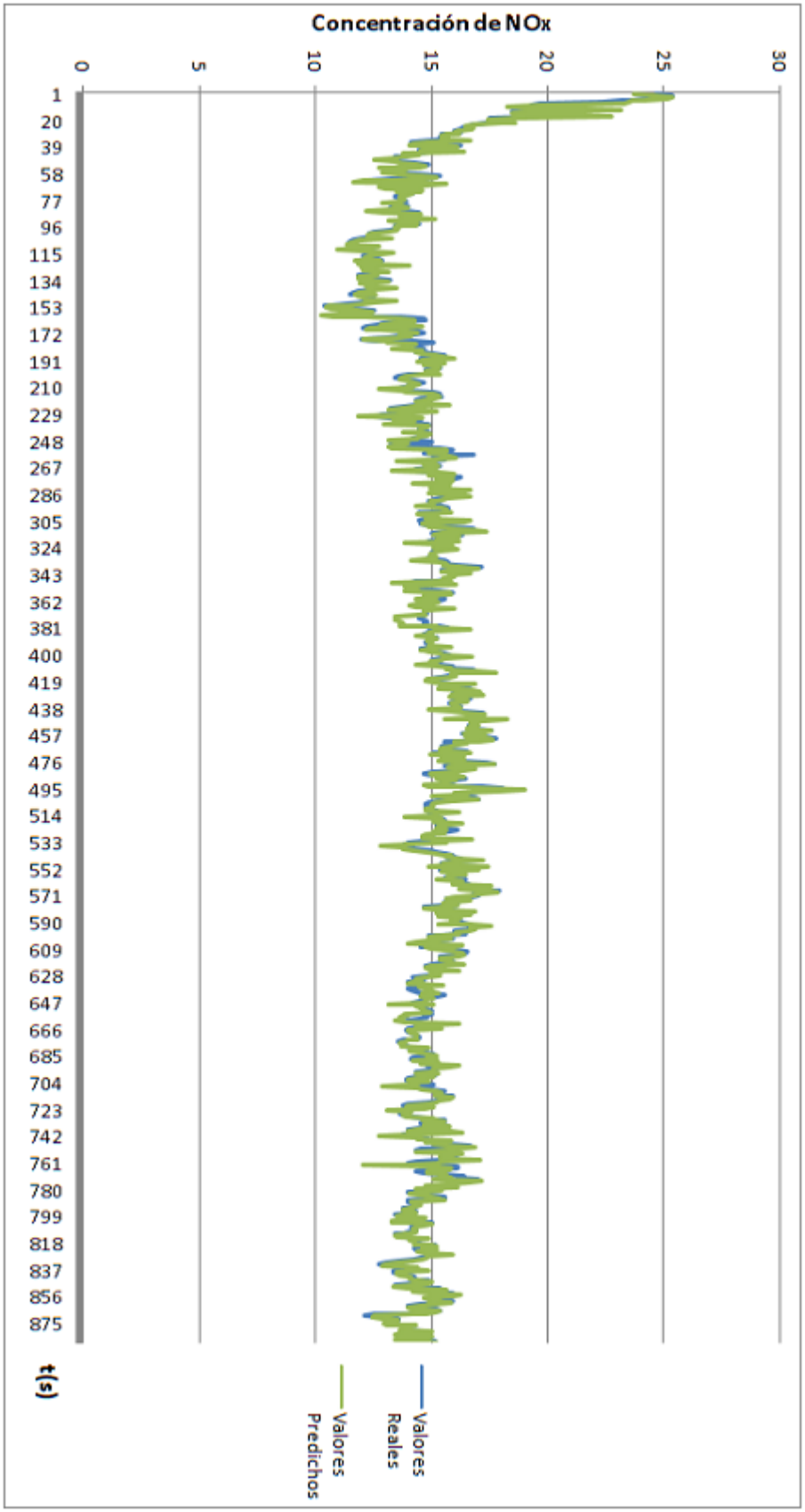

Figura 4.2: Clasificación de NO_x con el clasificador Gamma modificado

4.2.2 Simplificación y Parámetros para la Solución Propuesta

El paso 5 del algoritmo del clasificador Gamma, que se define en la sección 3.5.2, presenta el uso de pesos que deben ser asignados a cada dimensión del conjunto fundamental. El objetivo de estos pesos es darle un mayor valor a aquellos rasgo que permiten separar linealmente las clases; mientras que a los rasgos que no cumplan con esta condición se les asigna un valor menor. Esta asignación de pesos tiene sentido cuando cada una de las dimensiones aporta información diferente sobre el problema al que se aplica el clasificador; en el caso particular de la aplicación presentada en este trabajo de tesis, los datos de cada dimensión provienen de una serie de datos y por tanto representan la misma información de los elementos que están siendo clasificados, pero medida en un intervalo de tiempo distinto.

Debido a esta característica, propia de la forma en que se codificaron los patrones a partir de una serie de tiempo, se utiliza el valor de 1 para todos los pesos que son utilizados en el cálculo de las sumas ponderadas c_i para cada clase $i = 1, 2, \ldots, m$. dada la fórmula original para este cálculo:

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} w_j \cdot \gamma_g\left(x_j^{i\omega}, \tilde{x}_j, \theta\right)}{k_i}$$

La fórmula simplificada para el cálculo de las sumas ponderadas queda de la siguiente manera:

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} \gamma_g\left(x_j^{i\omega}, \tilde{x}_j, \theta\right)}{k_i}$$

Se seleccionó el número de rasgos $n = 10$; esta selección fue realizada de forma experimental; una serie de pruebas con patrones de diferentes tamaños fueron llevadas a cabo. Las pruebas en que el número de rasgo era igual a 10 presentaron mejores resultados.

Los valores propuestos para el parámetro de paro ρ, dados en el paso 3 del algoritmo del clasificador Gamma sugieren el uso del mínimo, máximo o promedio de los máximos de cada dimensión. Por la forma en que se codificaron los patrones, el máximo de los datos de la serie de tiempo original aparace en todas las dimensiones de los patrones del conjunto fundamental; por lo que el parámetro de paro usado fue simplemente el máximo de la primera dimensión de los patrones en el conjunto fundamental:

$$\rho = \left(\bigvee_{i=1}^{p} x_1^i\right)$$

Durante la fase de experimentos, se pudo observar que los bancos de datos utilizados contienen algunos patrones atípicos; al clasificar dichos patrones presentan valores muy alejados de los patrones conocidos en cualquiera de sus dimensiones, por lo que no tiene sentido tratar de forzar la clasificación y estos patrones serán asignados a la clase desconocida C_0. Esto significa que se debe seleccionar un valor para el parámetro de pausa menor o igual al valor del parámetro de paro. Se realizaron pruebas con diferentes valores de ρ_0: $\rho/4$, $\rho/2$, ρ y menores a $\rho/4$. Para valores $\rho_0 < \rho/4$, existían patrones que eran asignados a la clase desconocida, pero que al aumentar el valor del parámetro de pausa eran asignados a una clase conocida; esto indica que un valor de $\rho_0 >= \rho/4$ permitiría una mejor clasificación. Con los valores $\rho/4$, $\rho/2$ y ρ los patrones asignados a la clase desconocida son siempre los mismos, por lo que cualquiera de ellos es un valor válido para el parámetro de pausa. Por razones de eficiencia se seleccionó $\rho_0 = \rho/4$.

El valor de umbral seleccionado fue cero, de forma que si se alcanza el parámetro de pausa y el patrón a clasificar no muestra al menos similitud en uno de sus rasgos con alguno de los patrones del conjunto fundamental, será enviado a la clase desconocida pues no existe suficiente información para realizar la clasificación.

4.3 Clasificador Gamma Enriquecido

Incluyendo los ajustes y simplificaciones descritas en la sección anterior, el algoritmo del clasificador Gamma propuesto se presenta a continuación. Las figuras 4.3 y 4.4 muestran el diagrama de flujo de la solución propuesta.

Algoritmo 4.3.1 *Sea el conjunto fundamental del clasificador Gamma de acuerdo con la definición 3.5.2. Al presentarse un patrón a clasificar* $\tilde{\mathbf{x}}$*, donde* $\tilde{\mathbf{x}}$ *es un vector real n-dimensional* $\tilde{\mathbf{x}} \in \mathbb{R}^n$*, con* $n \in \mathbb{Z}^+$*, se realiza lo siguiente:*

1. Codificar cada componente de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado, restando el valor menor a todos los valores por cada componente; asimismo, se obtiene un valor $e_m = \bigvee_{i=1}^{p} x_j^i$ por cada componente. Con lo anterior se desplaza el rango de cada componente para que vaya de 0 a e_m. Así, la componente x_j^i se transforma en un vector binario de dimensión $e_m(j)$.

2. Codificar cada componente del patrón a clasificar con el código Johnson-Möbius modificado, utilizando las mismas condiciones que se utilizaron para codificar las componentes de los patrones fundamentales. En caso de que alguna componente del patrón a clasificar sea mayor al e_m correspondiente ($\tilde{\mathbf{x}}_\xi > e_m(\xi)$), 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 $e_m(\xi) + |\tilde{\mathbf{x}}_\xi|$ a $mgamma_\xi$.

3. Calcular el parámetro de paro ρ y el parámetro de pausa ρ_0. :

$$\rho = \left(\bigvee_{i=1}^{p} x_1^i \right)$$

$$\rho_0 = \rho/4$$

4. Determinar el umbral de pausa u.

$$u = 0$$

5. Se asignan los pesos para cada dimesión con el valor de 1.

6. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo $\mathbf{x}^\mu$, se convierta en dos índices: uno para la clase (por ejemplo la clase i) y otro para el orden que le corresponde a ese patrón dentro de esa clase (por ejemplo ω). Bajo estas condiciones ejemplificadas, la notación para el patrón $\mathbf{x}^\mu$ será ahora, con la transformación, $\mathbf{x}^{i\omega}$. Lo anterior se realiza para todos los patrones del conjunto fundamental.

7. Inicializar θ a 0.

8. Realizar la operación $\gamma_g\left(x_j^\mu, \tilde{x}_j, \theta\right)$ para cada componente de cada uno de los patrones fundamentales y del patrón a clasificar, considerándose $mgamma_\xi$ como la dimensión del patrón binario $\tilde{\mathbf{x}}_\xi$, en caso necesario.

9. Calcular la suma ponderada inicial c_i^0 de los resultados obtenidos en el paso 8, para cada patrón fundamental $\mu = 1, 2, \ldots, p$:

$$c_\mu^0 = \sum_{j=1}^{n} w_j \cdot \gamma_g\left(x_j^\mu, \tilde{x}_j, \theta\right)$$

10. Si existe un máximo único, cuyo valor es además igual a n, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{\mathbf{y}} = \mathbf{y}^j \text{ tal que } \bigvee_{\mu=1}^{p} c_\mu^0 = c_j^0 = n$$

De lo contrario, continuar.

11. Realizar la operación $\gamma_g\left(x_j^{i\omega}, \tilde{x}_j, \theta\right)$ para cada clase y para cada componente de cada uno de los patrones fundamentales que corresponden a esa clase, y del patrón a clasificar, considerándose $mgamma_\xi$ como la dimensión del patrón binario $\tilde{\mathbf{x}}_\xi$ si es necesario.

12. Calcular la suma ponderada c_i de los resultados obtenidos en el paso 11, para cada clase $i = 1, 2, \ldots, m$:

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} w_j \cdot \gamma_g\left(x_j^{i\omega}, \tilde{x}_j, \theta\right)}{k_i}$$

13. Si existe más de un máximo entre las sumas ponderadas por clase y el máximo es igual al número de rasgos n, entonces: si σ es el índice más pequeño de clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase $\tilde{\mathbf{y}} = \mathbf{y}^{\sigma}$; de lo contrario, continuar.

14. Si existe más de un máximo entre las sumas ponderadas por clase, incrementar θ en 1 y repetir los pasos 11, 12 y 13, hasta que:

    (a) exista un máximo único;

    (b) no existe un máximo único, pero $\bigvee_{i=1}^{m} c_i = n$;

    (c) o se cumpla con la condición de pausa: $\theta = \rho_0$;

    (d) o se cumpla con la condición de paro: $\theta \geq \rho$.

15. 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:

$$C_{\tilde{\mathbf{x}}} = C_0$$

    (b) Si $\bigvee_{i=1}^{m} c_i > u$ entonces se continua en el paso 11.

16. Si existe un máximo único, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{\mathbf{y}} = \mathbf{y}^j \text{ tal que } \bigvee_{i=1}^{m} c_i = c_j$$

17. En caso contrario: si λ es el índice más pequeño de clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase $\tilde{\mathbf{y}} = \mathbf{y}^{\lambda}$.

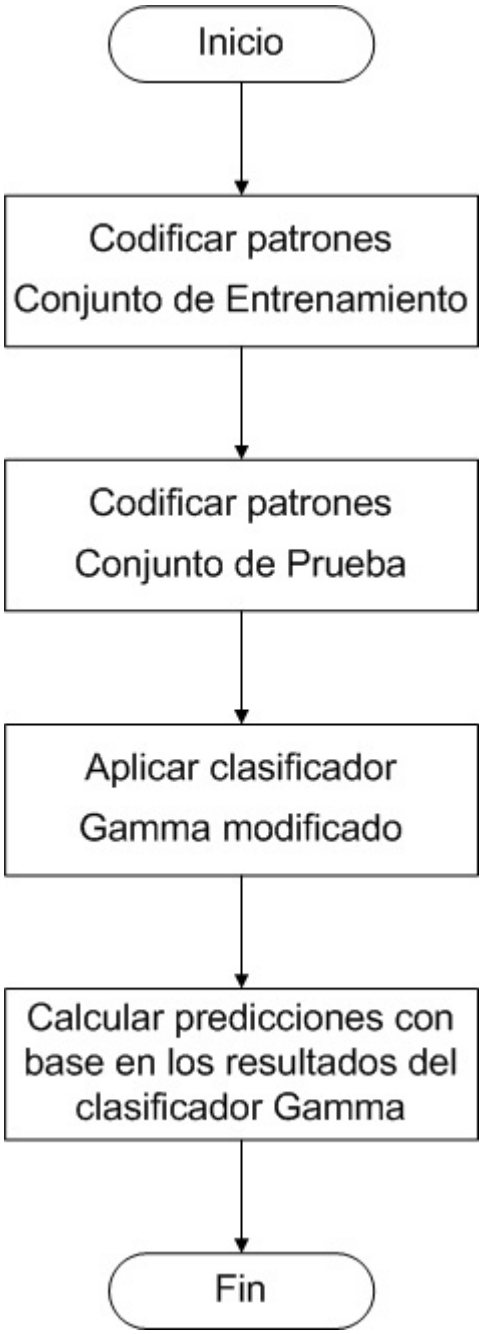

Figura 4.3: Diagrama de bloques de la solución propuesta, parte 2

Habiendo definido todas las herramientas necesarias para la implementación de la solución propuesta, sólo resta plantear los pasos generales del modelo:

1. Codificar los patrones para el conjunto de entrenamiento de acuerdo con el procedimiento descrito en la sub-sección 3.6.1 y asignar las clases de acuerdo con lo descrito en la sub-sección 3.6.3.

2. Codificar los patrones para el conjunto de prueba de acuerdo con el procedimiento descrito en la sub-sección 3.6.2 y asignar las clases de acuerdo con lo

descrito en la sub-sección 3.6.3.

3. Aplicar el clasificador Gamma, con los ajustes descritos en la sección 4.2.

4. Calcular valores predichos de las concentraciones del contaminante con el que se trabaja. Dado que los datos que se le pasan al clasificador Gamma son diferencias, como se explica en la sección 3.6, se deberá aplicar el proceso inverso para obtener los valores de las concentraciones de contaminante que se predicen.

La figura 4.4 muestra el diagrama de flujo del algoritmo del clasificador Gamma propuesto; los ajustes realizados se muestran en un color diferente.

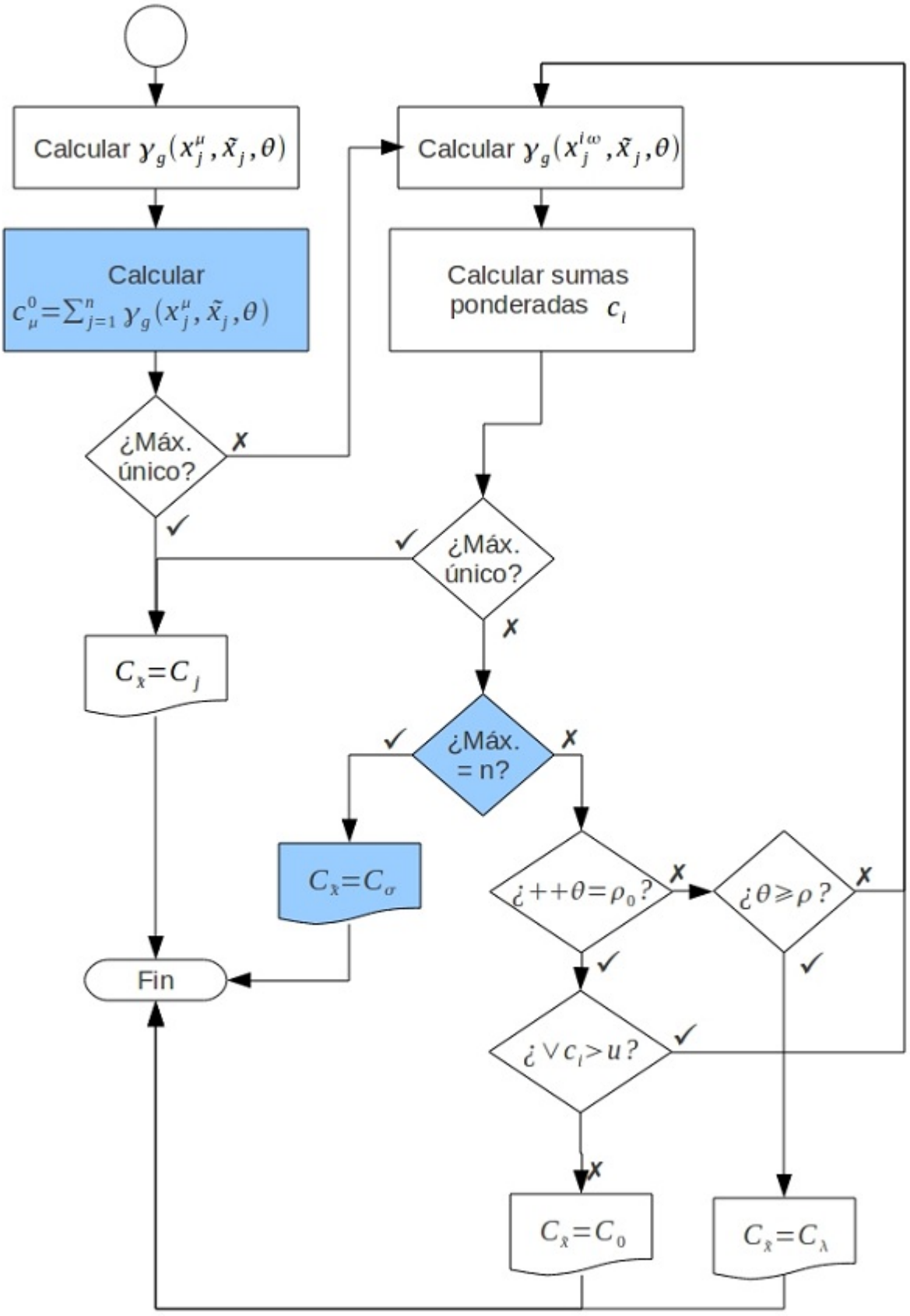

Figura 4.4: Diagrama de bloques de la solución propuesta, parte 2

Capítulo 5

Resultados y Discusión

En el presente capítulo se muestran los resultados de algunos de los experimentos realizados con el clasificador Gamma, para la clasificación y predicción de los contaminantes emitidos por el escape de un vehículo automotor.

5.1 Emisiones de Motores Diesel

5.1.1 Banco de datos

El contenido de esta sección se encuentra basado fuertemente en el reporte [60]. Los bancos de datos utilizados para las pruebas realizadas en esta sección fueron obtenidos de `http://www.crcao.com/publications/emissions/index.html`. Estos bancos de datos forman parte de un estudio realizado por el Coordinate Research Council (CRC) con el objetivo de caracterizar las emisiones de motores diesel y determinar si cumplen con las regulaciones establecidas en 2007 por la EPA (Environmental Protection Agency).

Se utilizaron 6 bancos de datos para las pruebas realizadas en esta sección, todos ellos conteniendo datos capturados durante un ciclo de manejo CARBx-ICT; correspondiente a los ciclos idle, creep y transient del ciclo de manejo CARB HHDDE-5 desarrollado por CARB (California Air Resources Board). Los bancos de datos corresponden a pruebas de manejo efectuadas en 3 motores diferentes, donde la prueba de manejo fue repetida 2 veces para cada motor.

Los datos de las concentraciones fueron medidos a una temperatura de referencia de 20 ℃y una presión de 101.3 kPa. Cada banco de datos consta de 900 mediciones. Los motores utilizados son modelo 2007, de las compañias Caterpillar, Cummins y Volvo. El combustible utilizado es diesel con bajo contenido de azufre. La tabla 5.1 muestra la relación de las pruebas con los motores y las mediciones realizadas en cada prueba.

Tabla 5.1: Relación de datos por prueba de manejo

	Banco de Datos	Motor	Cantidad de Datos
1	Test1	Caterpillar	900
2	Test2	Caterpillar	900
3	Test3	Cummins	900
4	Test4	Cummins	900
5	Test5	Volvo	900
6	Test6	Volvo	900

Las mediciones de emisiones incluidas en los bancos de datos son: monóxido de carbono (CO), óxidos nitrosos (NOx), hidrocarburos (HC), partículas suspendidas (PM), metano ($CH4$), dióxido de carbono (CO_2); entre otros.

La tabla 5.2 presenta una muestra de uno de los bancos de datos utilizados para los experimentos realizados; se muestran las columnas con los datos de los contaminantes que son de interés para este trabajo de tesis, ya que los bancos de datos incluyen mediciones de otros contaminantes.

Para cada uno de los experimentos realizados en esta sección, se utilizan series de tiempo de 2 pruebas de manejo diferentes. Ambas series de tiempo corresponden a las emisiones de un mismo contaminante; es decir que por cada experimento realizado se hace la predicción de un solo contaminante. Una de las series de datos es usada como conjunto fundamental, mientras que la otra conforma el conjunto de prueba.

La serie de tiempo utilizada para formar el conjunto fundamental, es codificada de acuerdo con la metodología presentada en la sección 3.6.1. La serie de tiempo utilizada para formar el conjunto de prueba, es codificada de acuerdo con la metodología presentada en la sección 3.6.2.

Ya que el objetivo para el cual fueron creados los bancos de datos usados en los experimentos, no tiene que ver con tareas de de clasificación o predicción, no es posible comparar los resultados obtenidos con otras publicaciones; por lo que la estrategia a seguir para poder comparar resultados, es usar los mismos datos que se le proporcionan al clasificador Gamma con algunos algoritmos implementados en WEKA. Para medir el desempeño de las predicciones se utilizará el *Root Mean Square Error (RMSE)* que se calcula utilizando la ecuación 5.1

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2} \tag{5.1}$$

Tabla 5.2: Ejemplo de banco de datos de emisiones

	Temperatura del escape	NO_x	CO	CO_2
1	95.7	6.2531	2.8448	5073.2536
2	96.5	9.8239	3.0639	5077.0433
3	97.4	13.1489	3.3809	5070.3595
4	98.4	13.1233	2.8982	6126.0778
5	99.3	12.2108	2.4929	6095.6059
6	100.1	11.4226	2.8759	5014.1326
7	100.8	11.5398	3.1931	5023.0873
8	101.4	10.5570	2.9221	5051.1999
9	102.0	9.4300	2.5583	5031.8932
10	102.5	9.0879	2.1058	6085.5071
11	103.1	10.4206	2.3920	6080.5923
12	103.6	13.8831	2.5498	6100.3542
13	104.2	19.1086	3.0385	6094.6572
14	104.7	18.8824	3.5944	5024.4836
15	105.2	17.2240	3.0137	6104.7021
16	105.7	16.4368	2.8995	6094.9596
17	106.1	15.4366	2.5666	6093.7078
18	106.6	14.6581	1.0063	6094.3922
19	107.0	14.5496	1.9526	6096.2619
20	107.5	13.7676	3.1834	6091.7107
21	107.8	13.5490	3.5680	6099.7337
22	108.2	13.7670	2.6455	5025.4874
23	108.5	14.4409	2.4910	5033.6430
24	108.7	13.9949	2.5570	5033.8357
25	109.0	13.3290	2.2719	5038.6636
26	109.5	13.0918	2.7287	6082.0821
27	110.0	12.9862	3.4667	6087.0657
28	110.5	13.5393	3.4064	6083.3494
29	110.9	14.3169	2.25644	6082.0082
30	111.2	15.2093	2.6601	5019.4909
31	111.6	15.8865	3.1020	5032.9252
32	111.8	15.6835	2.7469	5041.2432
33	112.1	15.6703	2.4748	5040.6585
34	112.5	17.1139	2.4138	5037.3087
35	112.9	18.7797	2.43026	5030.2632

5.1.2 Predicción de Óxidos Nitrosos

En esta sección se presentan los resultados de las predicciones realizada para óxidos nitrosos, utilizando el banco de datos descrito en la seccion 5.1.1. Los conjutos de datos usados para cada experimento se detallan en la tabla 5.3. Las figuras 5.1, 5.2 y 5.3 presentan las gráficas de los valores reales contra los valores predichos.

Tabla 5.3: Experimentos para predicción de NO_x

Experimento	Conjunto de entrenamiento	Conjunto de prueba
1	test 2	test 1, test 2
2	test 5	test 5, test 6
3	test 4	test 3, test 4

Las tablas 5.4, 5.5 y 5.6 muestran la comparación de los resultados obtenidos por el clasificador Gamma con los resultados obtenidos por algunos de los algoritmos implementados en WEKA.

Tabla 5.4: Comparación de resultados para la predicción de NO_x, prueba 1

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
Gamma	**900**	**1,800**	**0.78**
IBk1	900	1,800	0.79
RepTree	900	1,800	0.80
SimpleLinearRegression	900	1,800	0.81
DecisionStump	900	1,800	0.81
ConjutiveRule	900	1,800	0.81
LWL	900	1,800	0.82
RBFNetwork	900	1,800	0.84
ZeroR	900	1,800	0.84
MLP	900	1,800	1.20
LeastMedSq	900	1,800	0.79

Tabla 5.5: Comparación de resultados para la predicción de NO_x, prueba 2

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
IBk1	900	1,800	1.02
Gamma	**900**	**1,800**	**1.15**
MLP	900	1,800	1.18
RepTree	900	1,800	1.24
LWL	900	1,800	1.27
DecisionStump	900	1,800	1.28
SimpleLinearRegression	900	1,800	1.27
ConjutiveRule	900	1,800	1.37
ZeroR	900	1,800	1.37
LeastMedSq	900	1,800	1.39
RBFNetwork	900	1,800	1.39

Tabla 5.6: Comparación de resultados para la predicción de NO_x, prueba 3

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
MLP	900	1,800	1.76
IBk1	900	1,800	1.78
Gamma	**900**	**1,800**	**2.00**
RepTree	900	1,800	2.10
LeastMedSq	900	1,800	2.42
RBFNetwork	900	1,800	2.96
SimpleLinearRegression	900	1,800	2.43
LWL	900	1,800	2.44
ConjutiveRule	900	1,800	2.61
ZeroR	900	1,800	3.38
DecisionStump	900	1,800	2.65

De las tablas anteriores podemos observar que los resultados presentados por el clasificador Gamma son muy competitivos, en relación con otros algoritmos tradicionalmentes utilizados para la realizar predicciones; como son las redes neuronales y la regresión. Las predicciones realizadas con este contaminante (NO_x), fueron las que presentaron los mejores resultados de todos los experimentos.

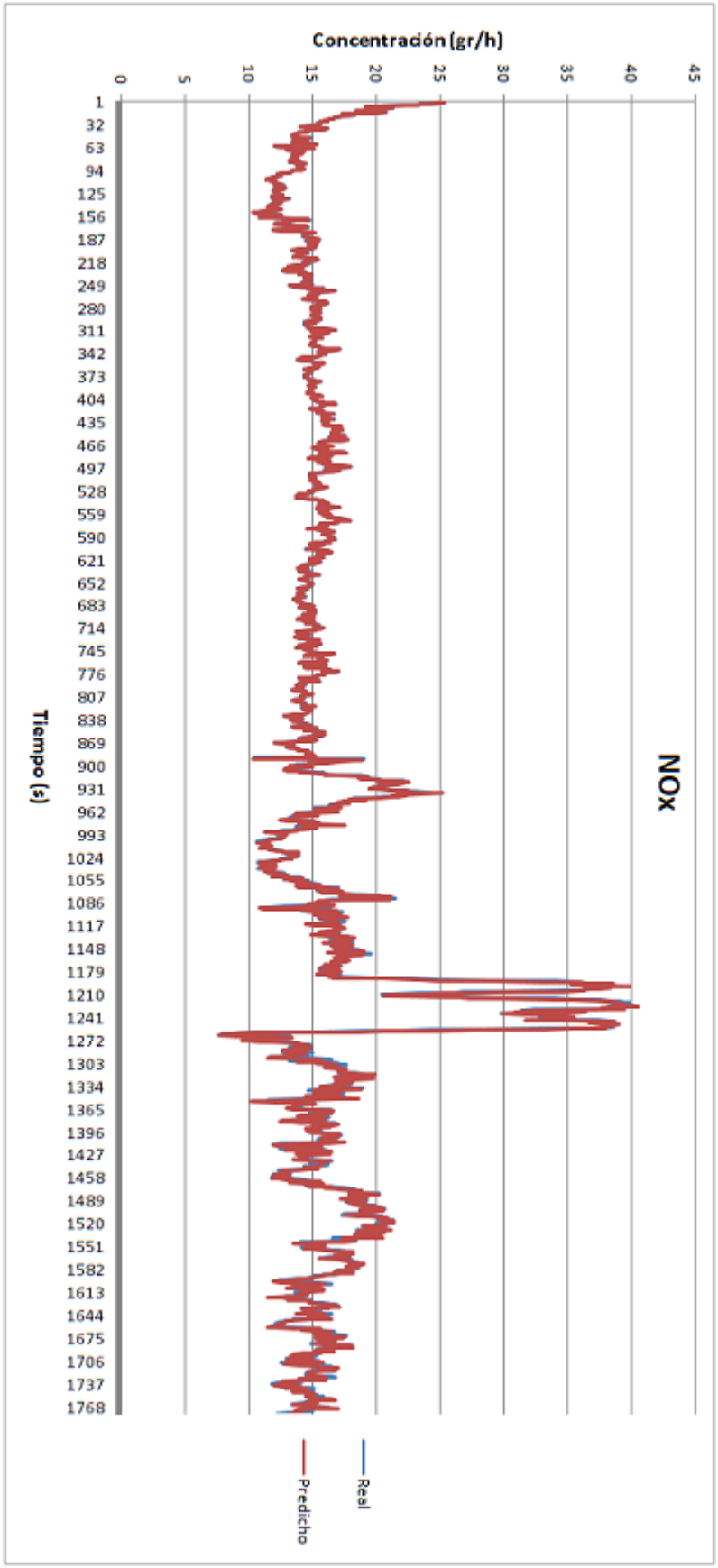

Figura 5.1: Predicción de óxidos nitrosos, prueba 1

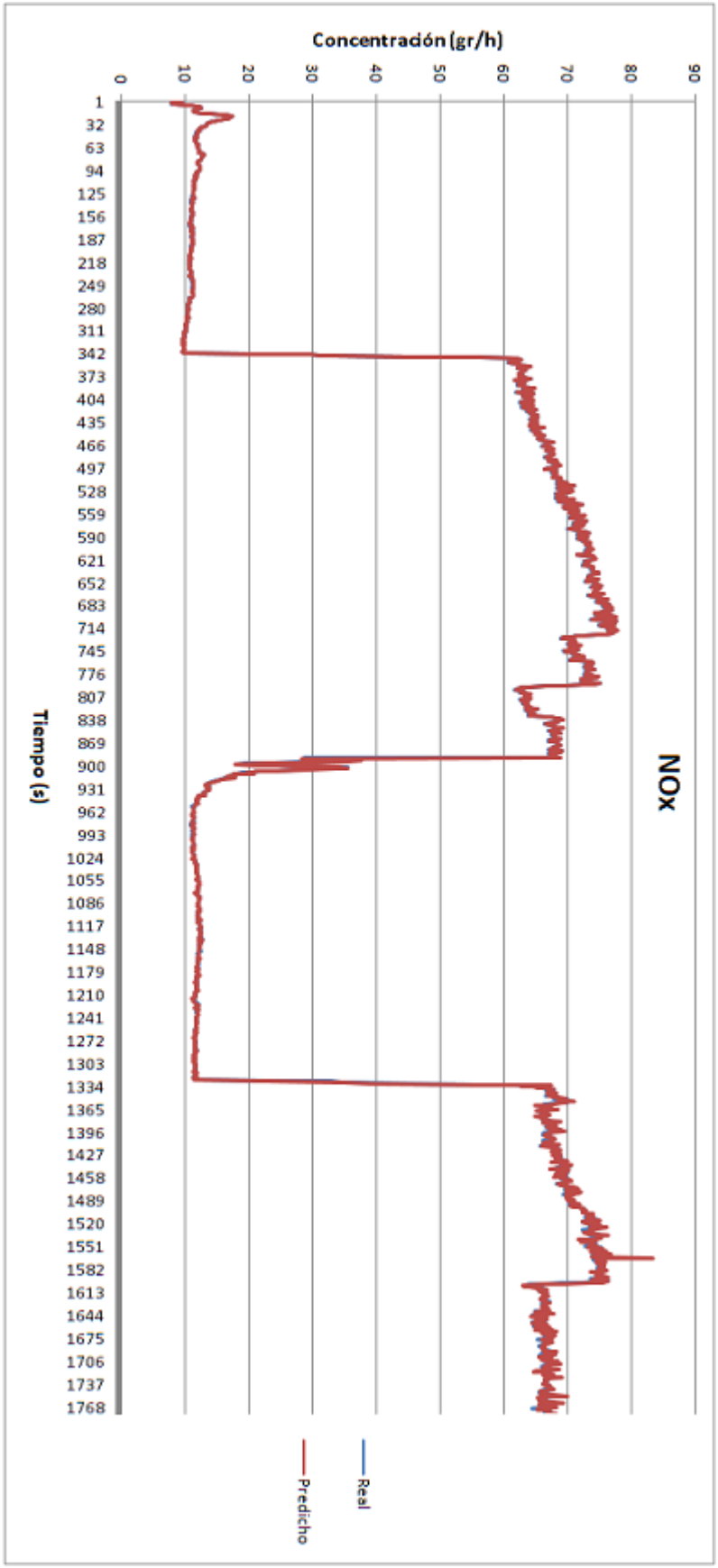

Figura 5.2: Predicción de óxidos nitrosos, prueba 2

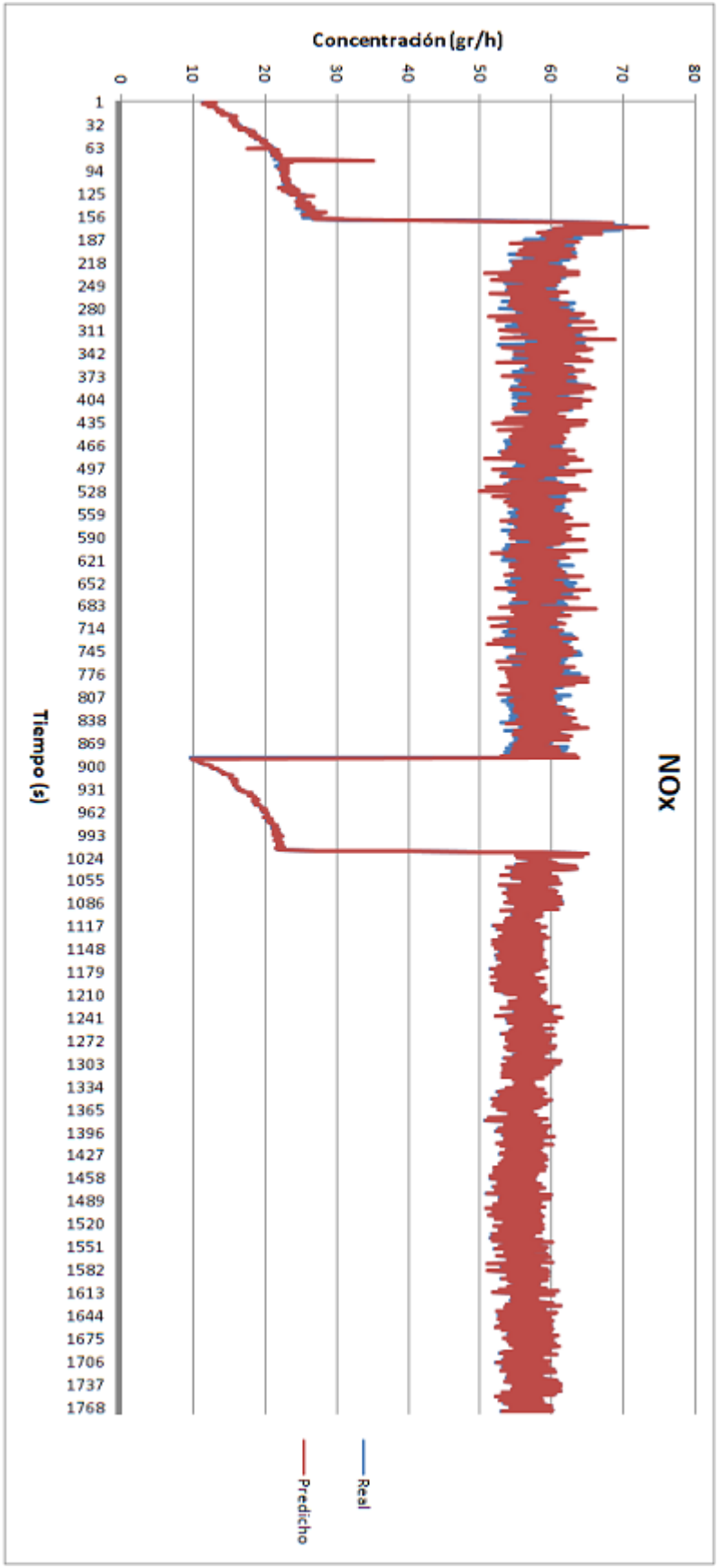

Figura 5.3: Predicción de óxidos nitrosos, prueba 3

5.1.3 Predicción de Dióxido de Carbono

En esta sección se presentan los resultados de las predicciones realizada para dióxido de carbono, utilizando el banco de datos descrito en la seccion 5.1.1. Los conjutos de datos usados para cada experimento se detallan en la tabla 5.7. Las figuras 5.4, 5.5 y 5.6 presentan las gráficas de los valores reales contra los valores predichos.

Tabla 5.7: Experimentos para predicción de CO_2

Experimento	Conjunto de entrenamiento	Conjunto de prueba
1	test 2	test 1, test 2
2	test 4	test 3, test 4
3	test 6	test 5, test 6

Las tablas 5.8, 5.9 y 5.10 muestran la comparación de los resultados obtenidos por el clasificador Gamma con los resultados obtenidos por algunos de los algoritmos implementados en WEKA.

Tabla 5.8: Comparación de resultados para la predicción de CO_2, prueba 1

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
Gamma	**900**	**1,800**	**367.77**
SimpleLinearRegression	900	1,800	381.66
DecisionStump	900	1,800	383.16
ZeroR	900	1,800	383.82
RBFNetwork	900	1,800	383.83
LeastMedSq	900	1,800	383.91
RepTree	900	1,800	384.29
LWL	900	1,800	384.84
ConjutiveRule	900	1,800	386.18
IBk1	900	1,800	387.85
MLP	900	1,800	507.08

Tabla 5.9: Comparación de resultados para la predicción de CO_2, prueba 2

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
MLP	900	1,800	412.47
SimpleLinearRegression	900	1,800	420.11
Gamma	**900**	**1,800**	**446.65**
ZeroR	900	1,800	461.63
LeastMedSq	900	1,800	461.73
IBk1	900	1,800	463.57
RBFNetwork	900	1,800	464.52
ConjutiveRule	900	1,800	474.36
LWL	900	1,800	475.81
DecisionStump	900	1,800	476.41
RepTree	900	1,800	499.36

Tabla 5.10: Comparación de resultados para la predicción de CO_2, prueba 3

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
IBk1	900	1,800	212.55
LWL	900	1,800	252.54
Gamma	**900**	**1,800**	**276.99**
DecisionStump	900	1,800	284.78
MLP	900	1,800	289.05
ConjutiveRule	900	1,800	303.03
SimpleLinearRegression	900	1,800	303.24
RBFNetwork	900	1,800	307.72
RepTree	900	1,800	307.81
ZeroR	900	1,800	307.81
LeastMedSq	900	1,800	307.85

En los resultados presentados para CO_2, llama la atención que el valor del RMSE es bastante elevado si lo comparamos con los errores presentados en los resultados de los otros 2 contaminantes (CO y NO_x). Esto se debe en gran medida a que los valores de las concentraciones de CO_2 son bastante más grandes que las concentraciones que se presentan de CO y NO_x, por lo que los valores de error también se incrementan. A pesar de esta situación los resultados presentados por el clasificador Gamma en la predicción de CO_2 son muy competitivos.

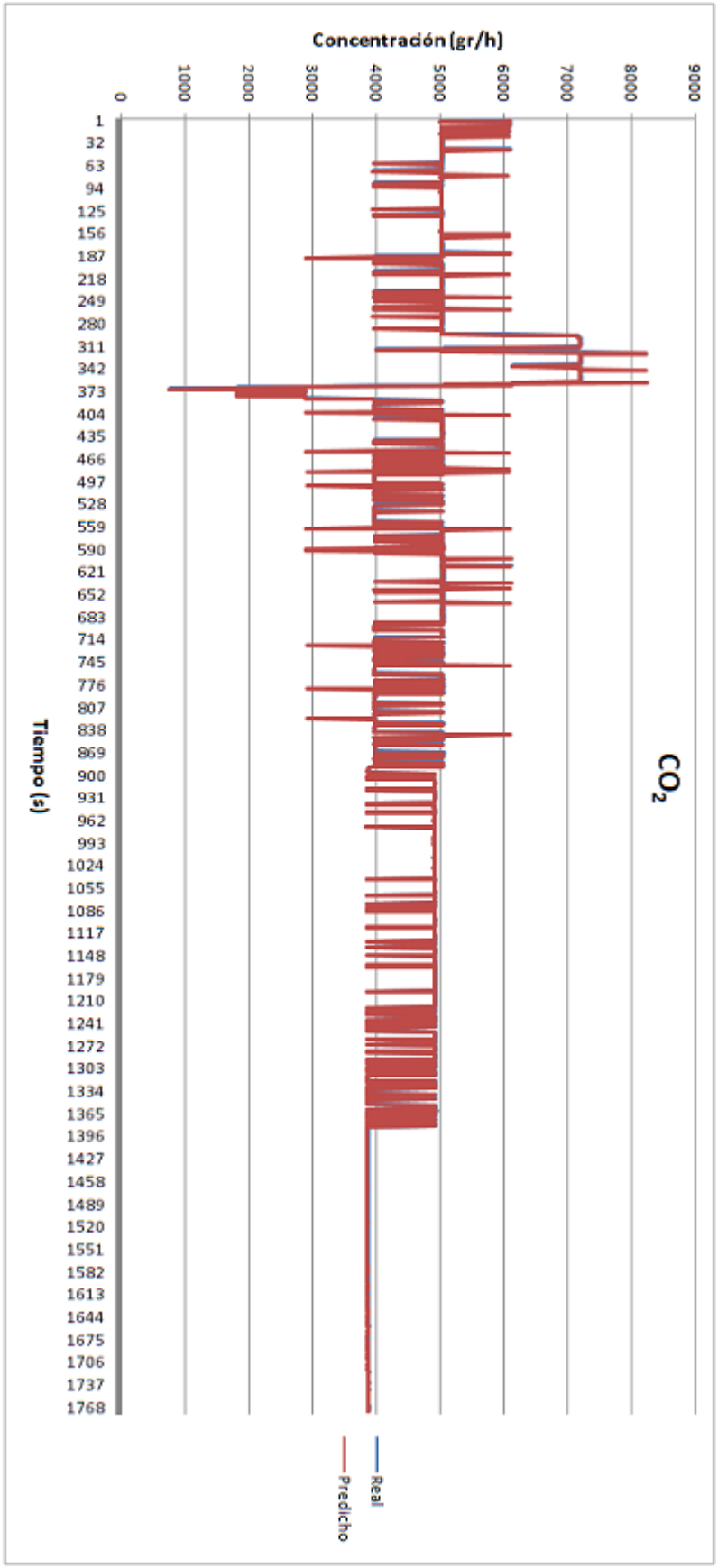

Figura 5.4: Predicción de dióxido de carbono, prueba 1

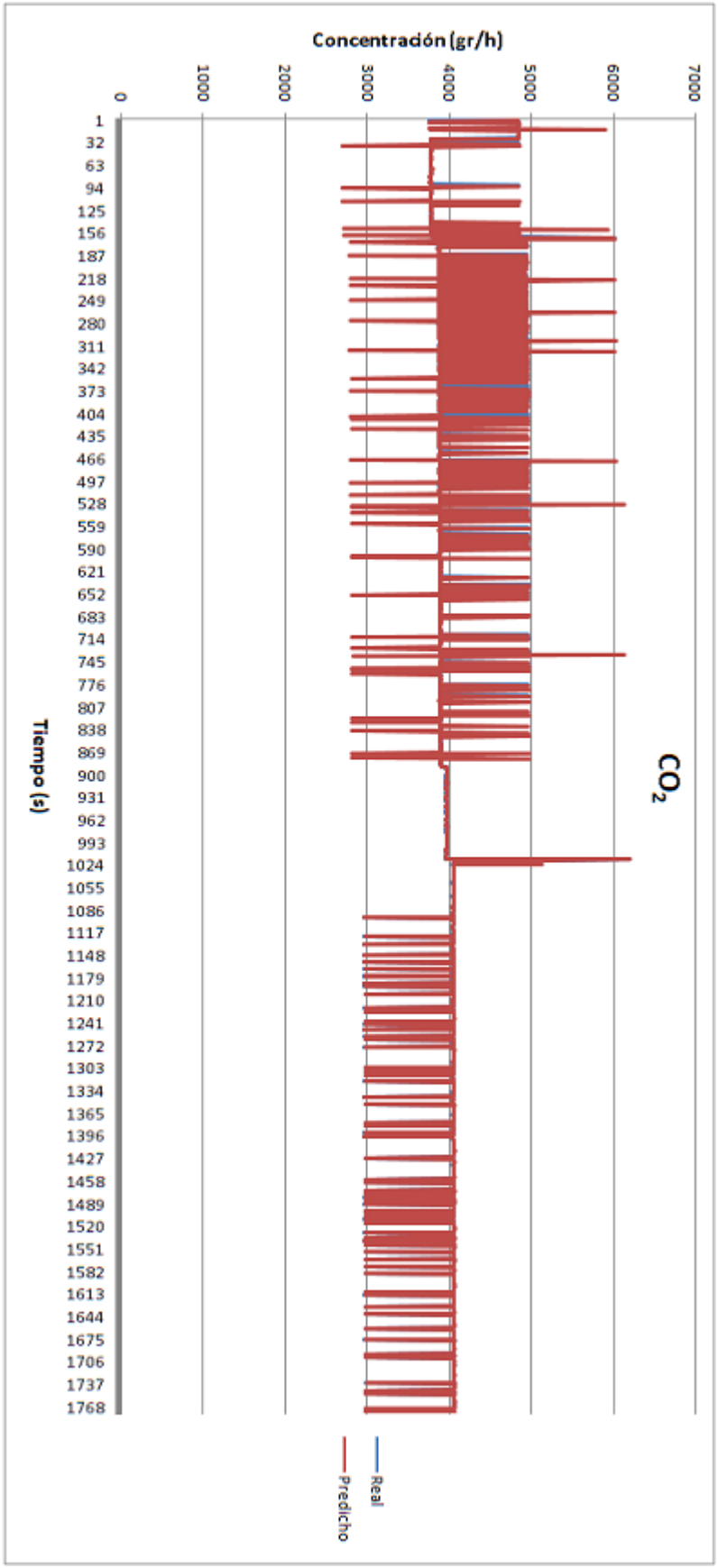

Figura 5.5: Predicción de dióxido de carbono, prueba 2

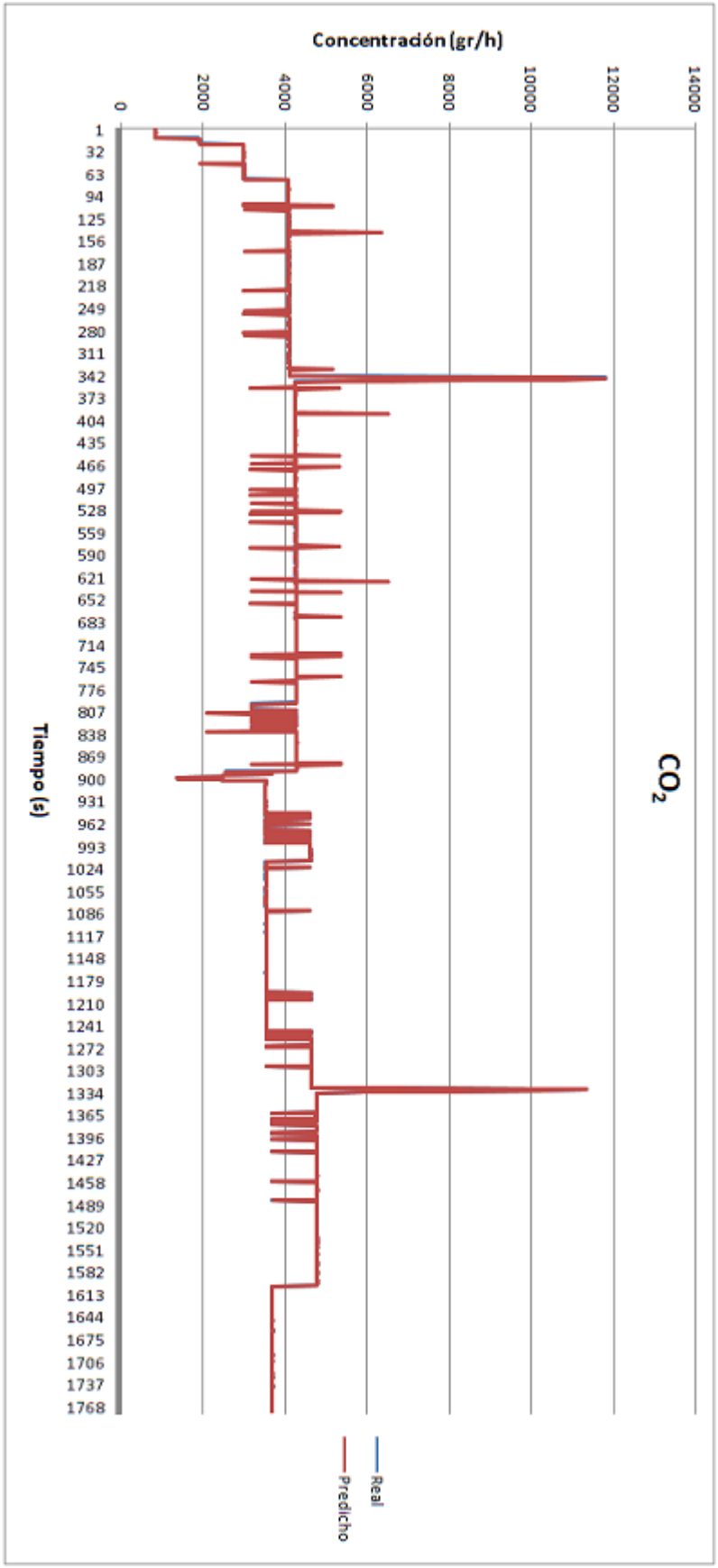

Figura 5.6: Predicción de dióxido de carbono, prueba 3

5.1.4 Predicción de Monóxido de Carbono

En esta sección se presentan los resultados de las predicciones realizada para monóxido de carbono, utilizando el banco de datos descrito en la seccion 5.1.1. Los conjutos de datos usados para cada experimento se detallan en la tabla 5.11. Las figuras 5.7, 5.8 y 5.9 presentan las gráficas de los valores reales contra los valores predichos, en el caso del contaminante monóxido de carbono.

Tabla 5.11: Experimentos para predicción de CO

Experimento	Conjunto de entrenamiento	Conjunto de prueba
1	test 1	test 1, test 2
2	test 5	test 5, test 6
3	test 3	test 3, test 4

Las tablas 5.12, 5.13 y 5.14 muestran la comparación de los resultados obtenidos por el clasificador Gamma con los resultados obtenidos por algunos de los algoritmos implementados en WEKA.

Tabla 5.12: Comparación de resultados para la predicción de CO, prueba 1

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
MLP	900	1,800	2.22
LeastMedSq	900	1,800	2.24
IBk1	900	1,800	2.30
RepTree	900	1,800	2.32
SimpleLinearRegression	900	1,800	2.41
Gamma	**900**	**1,800**	**2.43**
LWL	900	1,800	2.52
DecisionStump	900	1,800	2.60
ConjutiveRule	900	1,800	2.61
RBFNetwork	900	1,800	2.73
ZeroR	900	1,800	2.74

Tabla 5.13: Comparación de resultados para la predicción de CO, prueba 2

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
IBk1	900	1,800	0.88
LWL	900	1,800	0.98
MLP	900	1,800	1.00
LeastMedSq	900	1,800	1.18
Gamma	**900**	**1,800**	**1.22**
SimpleLinearRegression	900	1,800	1.33
RepTree	900	1,800	1.49
DecisionStump	900	1,800	1.49
ConjutiveRule	900	1,800	1.53
RBFNetwork	900	1,800	1.77
ZeroR	900	1,800	1.79

Tabla 5.14: Comparación de resultados para la predicción de CO, prueba 3

Algoritmo usado	Tamaño del Conjunto Fundamental	Tamaño del Conjunto de Prueba	Desempeño (RMSE)
Gamma	**900**	**1,800**	**0.23**
IBk1	900	1,800	0.23
LeastMedSq	900	1,800	0.24
MLP	900	1,800	0.25
RepTree	900	1,800	0.29
SimpleLinearRegression	900	1,800	0.29
LWL	900	1,800	0.29
DecisionStump	900	1,800	0.30
ConjutiveRule	900	1,800	0.31
RBFNetwork	900	1,800	0.34
ZeroR	900	1,800	0.34

Los resultados para la predicción de CO, fueron los que presentaron el desempeño más pobre de los experimentos realizados. Puede observarse en la tablas anteriores, que aunque en 1 de los experimentos resultó ser el mejor en las predicciones, en las otras 2 no alcanza a situarse en los 3 primeros lugares, a diferencia de los resultados para NO_x y CO_2, en donde siempre se mantiene entre los primeros 3.

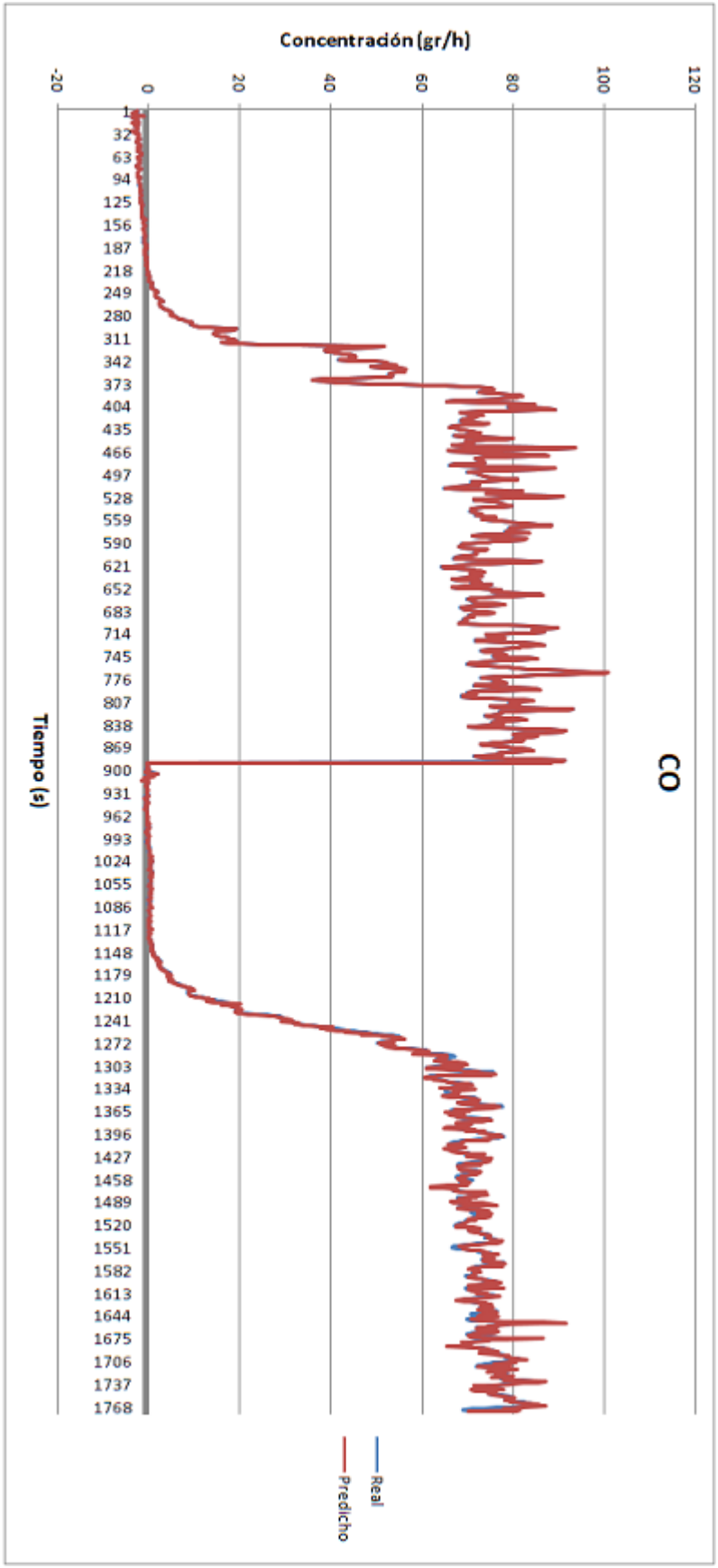

Figura 5.7: Predicción de monóxido de carbono, prueba 1

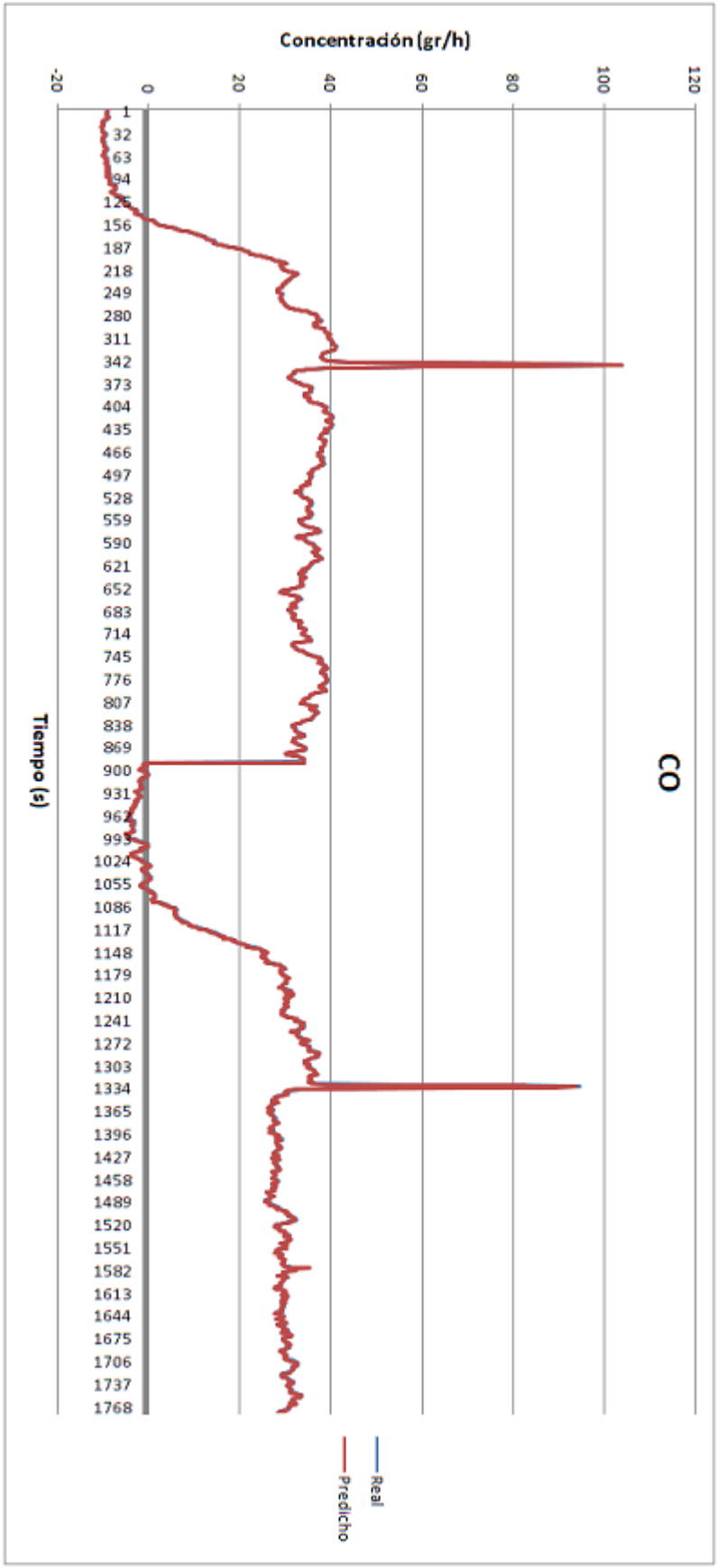

Figura 5.8: Predicción de monóxido de carbono, prueba 2

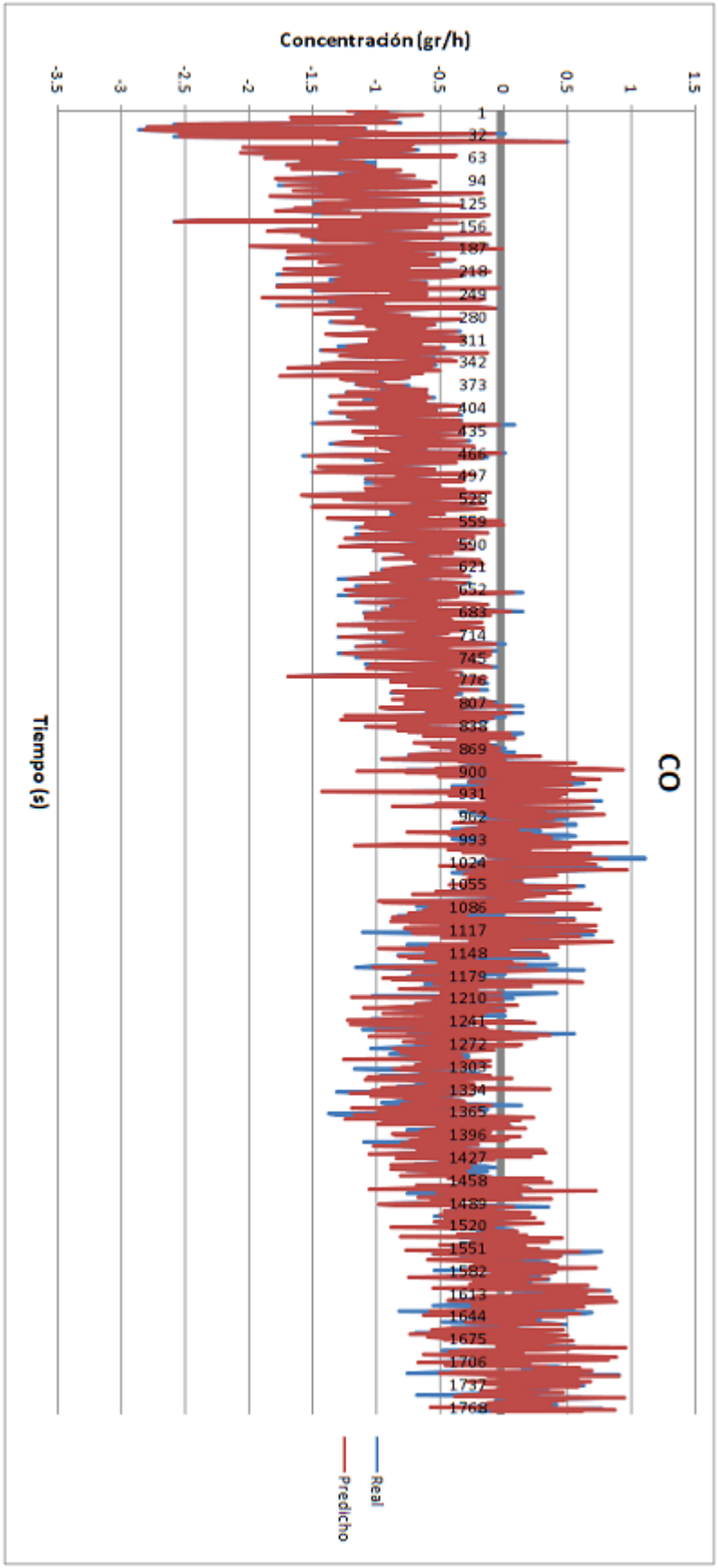

Figura 5.9: Predicción de monóxido de carbono, prueba 3

A pesar del buen desempeño que mostró el clasificador Gamma durante todos los experimentos realizados, éste aún presenta algunos errores en las predicciones con respecto a los valores reales. Al realizar un análisis detallado de los datos para los cuales se presentan dichos errores, se pudo observar que las series de tiempo usadas, exhiben algunos valores atípicos; es decir valores para la concentración de contaminantes que se encuentran muy alejados de la media de cada una de las series de datos. Estos valores atípicos degradan el desempeño en las predicciones realizadas por el clasificador Gamma, por lo que consideramos conveniente la aplicación de algún método de pre-procesamiento que permita suavizar estos valores.

Capítulo 6

Conclusiones y Trabajo Futuro

En este capítulo se presentan las conclusiones derivadas de los experimentos y resultados presentados en este trabajo de tesis. Además se proponen algunas ideas de trabajos futuros que se podrían realizar, que den la pauta para desarrollar nuevos trabajos de investigación que abarquen puntos no cubiertos en la presente tesis.

6.1 Conclusiones

Al realizar la investigación documental del estado del arte, fue notorio que a pesar de la amplia variedad de métodos aplicados a la predicción de contaminantes, su aplicación para predecir las emisiones específicamente generadas por automóviles no es tan frecuente como las aplicaciones dedicadas a predecir contaminantes en el medioambiente en general.

Durante el análisis de los diferentes métodos del estado del arte, se hizo evidente que el modelo matemático del clasificador Gamma, es en general mucho más sencillo que la mayoría de los métodos usados para predicción de contaminantes. Otra ventaja de la solución propuesta, es el uso de los datos en la forma de una serie de tiempo; lo cual permite que las predicciones realizadas no se vean afectadas por las diversas variables que intervienen en el complejo proceso de formación de los contaminantes.

A pesar de que el clasificador Gamma fue originalmente diseñado para la tarea de clasificación, los resultados presentados en este trabajo de tesis demuestran que puede ser aplicado a la tarea de predicción de manera eficiente y eficaz.

Es la primera vez que se aplica con éxito el clasificador Gamma a la predicción de contaminantes emitidos a través del escape de un vehículo automotor.

El comportamiento de cada contaminante analizado en este trabajo de tesis fue muy diferente con respecto al de los otros contaminates también analizados, aun cuando las muestras de cada uno de ellos fueran tomadas durante un mismo evento. Esto provocó diferencias en los resultados obtenidos por el clasificador Gamma; el contaminante para el cual el Gamma presentó las mejores predicciones fue el NO_x,

mientras que los resultados para el CO fueron los que presentaron la eficacia más pobre.

Los resultados obtenidos por el clasificador Gamma, mostraron ser competitivos al ser comparados con otros algoritmos tradicionalmente usados para realizar predicciones, como son las redes neuronales y la regresión.

A pesar del buen desempeño del clasificador Gamma, este modelo aún exhibe algunas limitaciones:

- La presencia de valores atípicos en los datos del conjunto fundamental, principalmente valores máximos o mínimos, degradan la precisión en las predicciones realizadas por el clasificador.
- Cuando los valores del conjunto fundamental inducen una relación que no es una función, ocasiona problemas al realizar la clasificación.
- Los valores para el tamano de los patrones, ρ, ρ_0 y u, son determinados de forma empírica, basado en el conocimiento acumulado durante los experimentos realizados.

El algoritmo original del clasificador Gamma, presenta dificultades al clasificar patrones para los cuales la suma ponderada c_i es igual al tamaño de los patrones (n) y además c_i no es un máximo único. El principal aporte del presente trabajo de tesis, es la introducción de un parámetro de paro cuando la situación mencionada anteriormente se presenta y que se realice la clasificación con la información disponible en ese momento.

6.2 Trabajo Futuro

Desarrollo de una metodología que permita seleccionar la dimensión óptima de los patrones que se codifican a partir de un serie de tiempo.

Modificar el algoritmo del clasificador Gamma, para mejorar su eficiencia.

Realizar pruebas con diferentes ciclos de manejo, que permitan analizar el comportamiento del clasificador Gamma ante una variedad aún más amplia de situaciones.

Incluir en el modelo alguna forma de pre-procesamiento de los datos, que permita tratar algunos valores atípicos de los bancos de datos; esto permitiría mejorar la precisión en los resultados de las predicciones. Entre métodos que se podrían utilizar para esta tarea de pre-procesamiento se encuentran: la interpolación de Lagrange o la extrapolación de Richardson.

Referencias

[1] Secretaría del Medio Ambiente. (2009). Calidad del aire en la Ciudad de México: Informe 2009. Disponible en http://www.calidadaire.df.gob.mx.

[2] Secretaría de Salud. (2007). Programa Nacional de Salud 2007-2012. México. ISBN 978-970-721-414-9. Disponible en http://portal.salud.gob.mx/descargas/pdf/pns_version_completa.pdf.

[3] Secretaría del Medio Ambiente y los Recursos Naturales (SEMARNAT). (2007). Y el medio ambiente? Problemas en México y todo el mundo. México. ISBN 978-968-817-877-5. Disponible en http://www.semarnat.gob.mx/informacionambiental/publicaciones/Pages /publicaciones.aspx

[4] Secretara del Medio Ambiente. (2010). Inventario de emisiones de contaminantes criterio de la ZMVM 2008. Disponible en http://www.sma.df.gob.mx/sma/index.php.

[5] Alonso, J. M., Alvarruiz, F., Desantes, J. M., Hernandez, L., Hernandez, V. & Molto, G. (2007). Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions. IEEE Transactions on Evolutionary Computation, vol. 11, no. 1, pp. 46 - 55.

[6] Brunelli, U., Piazza, V., Pignato, L., Sorbello, F. & Vitabile S. (2007). Two-days ahead prediction of daily maximum concentrations of SO_2, O_3, PM_{10}, NO_2, CO in the urban area of Palermo, Italy. Atmospheric Environment, vol. 41, no. 14, pp. 2967 - 2995.

[7] Deleawe, S., Kusznir, J., Lamb, B. & Cook, D. J. (2010). Predicting air quality in smart environments. Journal of Ambient Intelligence and Smart Environments, vol. 2, no. 2, pp. 145 - 154.

[8] Ganapathy, T., Gakkhar, R. P. & Murugesan, K. (2009). Artificial neural network modeling of jatropha oil fueled diesel engine for emission predictions. Thermal Science, vol. 13, no. 3, pp. 91 - 102.

[9] Ghazikhani, M. & Mirzaii, I. (2011). Soot emission prediction of a wastegated turbo-charged DI diesel engine using artificial neural network. Neural Computing & Applications, vol. 20, no. 2, pp. 303 - 308.

[10] Karri, V. & Ho, T. (2009). Predictive models for emission of hydrogen powered car using various artificial intelligent tools. Neural Computing & Applications, vol. 18, no. 5, pp. 469 - 476.

[11] Kiani, M. K. D., Ghobadian, B., Tavakoli, T., Nikbakht, A. M. & Najafi, G. (2010). Application of artificial neural networks for the prediction of performance and exhaust emissions in SI engine using ethanol - gasoline blends. Energy, vol. 35, no. 1, pp. 65 - 69.

[12] Kurt, A. & Oktay, A. B. (2010). Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. Expert Systems with Applications, vol. 37, no. 12, pp. 7986 - 7992.

[13] Ozdemir, H., Demir, G., Altay, G., Albayrak, S. & Bayat, C. (2008). Prediction of Tropospheric Ozone Concentration by Employing Artificial Neural Networks. Environmental Engineering Science, vol. 25, no. 9, pp. 1249 - 1254.

[14] Shakil, M., Elshafei, M., Habib, M. A. & Maleki, F. A. (2009). Soft sensor for NO_X and O_2 using dynamic neural networks. Computers & Electrical Engineering, vol. 35, no. 4, pp. 578 - 586.

[15] Tzima, F. A., Mitkas, P. A., Voukantsis, D. & Karatzas, K. (2011). Sparse episode identification in environmental datasets: The case of air quality assessment. Expert Systems with Applications, vol. 38, no. 5, pp. 5019 - 5027.

[16] Yetilmezsoy, K., Ozkaya, B. & Cakmakci, M. (2011). Artificial intelligence-based prediction models for environmental engineering. Neural Network World, vol. 21, no. 3, pp. 193 - 218.

[17] Jang J. S. R. (1993). ANFIS - Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man & Cybernetics, vol. 23, no. 3, pp. 665 - 685.

[18] Noori R., Hoshyaripour G., Ashrafi K. & Araabi B. N. (2010). Uncertainty analysis of developed ANN and ANFIS models in prediction of carbon monoxide daily concentration. Atmospheric Environment, vol. 44, no. 4, pp. 476 - 482.

[19] Zito, P., Chen, H. & Bell, M. C. (2008). Predicting Real-Time Roadside CO and NO_2 Concentrations Using Neural Networks. IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 514 - 522.

[20] Chen, X., Schmid, N. A., Wang, L. & Clark, N. N. (2010). Regression-Based Oxides of Nitrogen Predictors for Three Diesel Engine Technologies. Journal of the Air & Waste Management Association, vol. 60, no. 1, pp. 72 - 90.

[21] Pisoni, E., Farina, M., Carnevale, C. & Piroddi, L. (2009). Forecasting peak air pollution levels using NARX models. Engineering Applications of Artificial Intelligence, vol. 22, no. 4 - 5, pp. 593 - 602.

[22] Polat, K. & Durduran, S. (2011). Usage of output-dependent data scaling in modeling and prediction of air pollution daily concentration values (PM10) in the city of Konya. Neural Computing & Applications, pp. 1 - 10.

[23] Lughofer, E., Macin, V., Guardiola, C. & Klement, E. P. (2011). Identifying static and dynamic prediction models for NO_X emissions with evolving fuzzy systems. Applied Soft Computing, vol. 11, no. 2, pp. 2487 - 2500.

[24] Osowski, S. & Garanty, K. (2007). Forecasting of the daily meteorological pollution using wavelets and support vector machine. Engineering Applications of Artificial Intelligence,vol. 20, no. 6, pp. 745 - 755.

[25] Wang, W., Men, C. & Lu, W. (2008). Online prediction model based on support vector machine. Neurocomputing, vol. 71, no. 4 - 6, pp. 550 - 558.

[26] 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.

[27] Dong, M., Yang, D., Kuang, Y., He, D., Erdal, S. & Kenski, D. (2009). $PM_{2.5}$ concentration prediction using hidden semi-Markov model-based times series data mining. Expert Systems with Applications, vol. 36, no. 5, pp. 9046 - 9055.

[28] López-Yañez, I., Argüelles-Cruz, A. J., Camacho-Nieto, O. & Yañez-Marquez, C. (2011). Polluants Time-Series Prediction Using the Gamma Classifier. International Journal of Computational Intelligence Systems, vol. 4, no. 4, pp. 680 - 711.

[29] Pires, J. C. M., Alvim-Ferraz, M. C .M., Pereira, M. C. & Martins, F. G. (2011). Prediction of tropospheric ozone concentrations: Application of a methodology based on the Darwin's Theory of Evolution. Expert Systems with Applications, vol. 38, no. 3, pp. 1903 - 1908.

[30] Tinaut, F. V., Melgar, A., Gimnez, B. & Reyes, M. (2011). Prediction of performance and emissions of an engine fuelled with natural gas/hydrogen blends. International Journal of Hydrogen Energy, vol. 36, no. 1, pp. 947 - 956.

[31] Cruz-Meza, M. E. (2006). Aprendizaje y recuperación de imágenes en color mediante memorias asociativas Alfa-Beta. Tesis de Maestría en Ciencias de la Computación, Centro de Investigación en Computación, IPN, México.

[32] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., López-Yáñez, I. (2006). Complexity of Alpha-Beta Bidirectional Associative Memories. MICAI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 4293, pp. 357-366.

[33] Aldape-Pérez, M., Yáñez-Márquez, C., López -Leyva, L. O. (2006). Feature Selection using a Hybrid Associative Classifier with Masking Technique. Fifth Mexican International Conference on Articial Intelligence, MICAI 2006, pp. 151-160.

[34] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., López-Yáñez, I. (2006). Alpha-Beta Bidirectional Associative Memories Based Translator. International Journal of Computer Science and Network Security, vol. 6, no. 5A, pp. 190-194.

[35] Steinbuch, K. (1961). Die Lernmatrix. Kybernetik, vol. 1, no. 1, pp. 36 - 45.

[36] Willshaw, D., Buneman, O. & Longuet-Higgins, H. (1969). Non-holographic associative memory. Nature, no. 222, pp. 960 - 962.

[37] Anderson, J. A. (1972). A simple neural network generating an interactive memory. Mathematical Biosciences, vol. 14, no. 3 - 4, pp. 197 - 220.

[38] Kohonen, T. (1972). Correlation matrix memories. IEEE Transactions on Computers, vol. C - 21, no. 4, pp. 353 - 359.

[39] Nakano, K. (1972). Associatron - A model of associative memory. IEEE Transactions on Systems, Man, and Cybernetics, vol. 2, no. 3, pp. 380 - 388.

[40] Amari, S. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. IEEE Transactions on Computers, vol. C - 21, no. 11, pp. 1197 - 1206.

[41] 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.

[42] Kosko, B. (1988). Bidirectional associative memories. IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, no. 1, pp. 49 - 60.

[43] 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.

[44] 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.

[45] López-Yáñes, I. (2007). Clasificador automático de alto desempeño. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.

[46] Flores-Carapia, R. (2006). Memorias Asociativas Alfa-Beta basadas en el código Johnson-Mobius modificado. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.

[47] Sáenz-Morales, G. L. (2010). Predicción de contaminantes atmosféricos mediante el clasificador Gamma. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.

[48] López-Yáñes, I. (2011). Teoría y aplicaciones del clasificador Gamma. Tesis de Doctorado en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.

[49] Brapenning, P. J., Thuijsman, F. & Weijters, A. J. M. M. (1991). Artificial Neural Network: An introduction to ANN theory and practice. New York: Springer-Verlag.

[50] Rosen, K. H. (1995). Discrete Mathematics and Its Applications. Third Edition. McGraw-Hill.

[51] Duda, R. O., Hart, P. E. & Stork, D. G. (2001). Pattern Classification. Second Edition. McGraw-Hill.

[52] Marques de Sá, J. P. (2001). Pattern Recognition: Concepts, Methods and Applications. Springer.

[53] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009). The WEKA data mining software: an update, SIGKDD Explorations, vol. 11, no. 1, pag. 1018.

[54] Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing. New Jersey: Prentice Hall.

[55] Schelter, B., Winterhalder, M. & Timmer, J. (2006) Handbook of Time Series Analysis. Germany: Wiley.

[56] Palit,A. K. & Popovic, D. (2005). Computational Intelligence in Time Series Forecasting. London, U. K: Springer-Verlag.

[57] Pollock, D. S. G. (1999). A Handbook of Time-Series Analysis, Signal Processing and Dynamics. London, U. K: Academic Press.

[58] Secretaría del Medio Ambiente y los Recursos Naturales (SEMARNAT). (2007). NOM-041-SEMARNAT-2006. Disponible en `http://www.semarnat.gob.mx/leyesynormas/Pages/fuentesmoviles.aspx`.

[59] Witten, I. H., Frank, E., Hall, M. A. (2011) Data Mining Practical Machine Learning Tools and Techniques. Elsevier.

[60] Coordinate Research Council, INC. (2009). CRC Report: ACES Phase 1. Disponible en `http://www.crcao.com/publications/emissions/index.html`.

[61] Adamatzky, A., De Lacy Costello, B., Bull, L., Stepney, S. & Teuscher, C. (2007). Unconventional Computing. United Kingdom: Luniver Press.

[62] International Journal of Unconvetional Computing. http://www.oldcitypublishing.com/IJUC/IJUC.html