



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

**PREDICCIÓN DE LOS GASES CONTAMINANTES
EMITIDOS POR VEHÍCULOS AUTOMOTORES
EMPLEANDO LAS TECNOLOGÍAS DEL CÓMPUTO
MÓVIL Y LA COMPUTACIÓN EN LA NUBE**

T E S I S

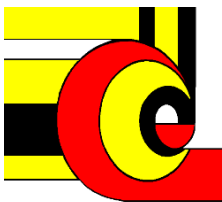
**QUE PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN
SISTEMAS DIGITALES**

PRESENTA:

MORENO AGUILERA PAULINA ELIZABETH

DIRECTORES DE TESIS:

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



MÉXICO, D.F.

JUNIO DEL 2014



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

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 29 del mes de abril de 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Predicción de los gases contaminantes emitidos por vehículos automotores empleando las tecnologías del cómputo móvil y la computación en la nube"

Presentada por el alumno:

MORENO
Apellido paterno

AGUILERA
Apellido materno

PAULINA ELIZABETH
Nombre(s)

Con registro:

A	1	2	0	5	9	4
---	---	---	---	---	---	---

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

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

LA COMISIÓN REVISORA
Directores de Tesis

Dr. Olexsiy Pogrebnyak

Dr. Amadeo José Argués Cruz

Dr. Sergio Suárez Guerra

Dr. Cornelio Pérez Márquez

Dr. Oscar Camacho Nieto

Dr. José Giovanni Guzmán Lugo

PRESIDENTE DEL COLEGIO DE PROFESORES

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



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 09 del mes Mayo del año 2014, el (la) que suscribe Moreno Aguilera Paulina Elizabeth alumno (a) del Programa de Maestría en ciencias en ingeniería de cómputo con opción en sistemas digitales, con número de registro A120594, adscrito a 5º semestre, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Amadeo José Argüelles Cruz y el Dr. Oleksiy Pogrebnyak y cede los derechos del trabajo intitulado Predicción de los gases contaminantes emitidos por vehículos automotores empleando las tecnologías del cómputo móvil y la computación en la nube, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección(es) pmoreno_a12@sagitario.cic.ipn.mx, jamadeo@cic.ipn.mx y olek@cic.ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.


Paulina Elizabeth Moreno Aguilera

Nombre y firma

Resumen

En este trabajo de tesis se emplea el algoritmo de clasificación Gamma, el cual forma parte del enfoque asociativo Alfa-Beta, mismo que emplea los operadores Alfa-Beta como base de sus operaciones fundamentales y que sirvieron de inspiración para el desarrollo de las memorias asociativas Alfa-Beta. Con base en este modelo se diseña, desarrolla e implementa un sistema informático que muestra a los usuarios los resultados obtenidos de la predicción de gases contaminantes provenientes de las emisiones del escape de vehículos en comparación con la norma oficial mexicana vigente.

La predicción se realiza tomando como base un conjunto de bancos de datos de vehículos con características distintas. Cada banco de datos proviene directamente de datos obtenidos al circular por el arroyo vehicular de la ciudad de México a través de un analizador de gases portátil; los resultados son comparados con la norma oficial mexicana NOM-041-SEMARNAT-2006, que establece los límites máximos permisibles de emisión de gases contaminantes provenientes del escape de vehículos automotores en circulación.

Con este trabajo se pretende contribuir de manera significativa con el desarrollo de alternativas que permitan la toma de decisiones tempranas que favorezcan a la mitigación de los niveles de emisiones de los vehículos a través del uso de tecnologías, modelos y paradigmas de computación.

Abstract

This thesis is based on the Gamma classification algorithm, which is part of the Alpha-Beta associative approach, which uses the same Alpha-Beta operators as the basis for fundamental operations and served as inspiration for the development of Alpha-Beta associative memories. Based on this model is designed, developed and implemented a computer system that shows users the results of the prediction of gaseous pollutants from exhaust emissions of vehicles compared to the current official Mexican standard.

The prediction uses a set of data banks from vehicles with different characteristics. Each data bank used in the prediction is proceeded directly from data thrown by vehicles in Mexico City carrying a portable gas analyzer; the obtained results are compared with the Official Mexican Norm NOM-041-SEMARNAT-2006, which establish the maximum allowed limits for each pollutant gas from the vehicle exhaust.

This work has the goal of contributing significantly in the development of different alternatives that allow earlier decision making favoring the mitigation of pollutant gas emissions from vehicles through the use of technologies, models and computing paradigms.

Agradecimientos

A mis padres que hicieron todo lo posible para que llegaré hasta donde hoy me encuentro, por motivarme y apoyarme cuando sentí que el camino se hacía más oscuro, a ustedes por siempre mi cariño y mi agradecimiento.

*Gracias por tu paciencia, amor y comprensión, estuviste a mi lado aún en los peores momentos.
Por inspirarme a dar lo mejor de mí, esta tesis también es tuya, gracias por todo, Eric.*

A mis directores de tesis, que en este andar por la vida influyeron con sus lecciones y experiencias para afrontar los retos que pone la vida.

Contenido

Resumen.....	4
Abstract.....	5
Agradecimientos.....	6
Contenido.....	7
Glosario de Términos.....	11
Introducción.....	12
1.1 Antecedentes	12
1.2 Justificación	14
1.3 Objetivo general	14
1.4 Objetivos específicos	14
1.5 Contribuciones	15
1.6 Organización del documento	15
Estado del arte	16
2.1 Técnicas para la predicción de contaminantes.	16
2.1.1 Redes neuronales artificiales.	16
2.1.2 Lógica difusa	17
2.1.3 Enfoque probabilístico – estadístico.	17
2.1.4 Enfoque asociativo.	18
2.2 Cómputo móvil.	18
2.2.1 Comunicación inalámbrica.	19
2.2.2 Modelo cliente móvil – servidor fijo.	19
2.2.3 Restricciones del cómputo móvil.	20
2.2.4 Cómputo en la nube móvil.	21
2.2.5 Modelos de programación mCloud.	21
2.3 Cómputo en la nube.	22
2.3.1 Servicios en la nube.	23
2.3.2 Modelos de implementación de la nube.	24
2.3.3 Características del cómputo en la nube.	24
2.4 Modelos y aplicaciones existentes.	27
2.4.1 Modelos.	27

2.4.2	Aplicaciones.....	27
	Materiales y métodos.....	29
3.1	Operadores Alfa (α) y Beta (β).....	29
3.2	Operador \mathcal{U}_β	30
3.3	Módulo.....	31
3.4	Código binario Johnson- Möbius modificado.....	31
3.5	Clasificador Gamma.....	34
3.5.1	Operador Gamma de similitud (γ).....	34
3.5.2	Algoritmo del clasificador Gamma.....	37
3.6	Descripción del banco de datos.....	42
3.7	Plataforma Android.....	42
3.7.1	Fundamentos de las aplicación en Android.....	43
3.8	Amazon Web Services AWS.....	43
3.8.1	AWS EC2.....	44
3.8.2	AWS Elastic Beanstalk.....	44
	Solución propuesta.....	45
4.1	Modificación del banco de datos.....	45
4.1.1	Descripción de los patrones.....	46
4.1.2	Descripción de las clases.....	47
4.2	Valores iniciales para el clasificador Gamma.....	47
4.2.1	Condiciones de paro.....	47
4.2.2	Umbral de pausa.....	48
4.2.3	Grado de similitud.....	48
4.2.4	Asignación de pesos.....	48
4.3	Criterio de validación del clasificador.....	49
4.4	Sistema de verificación.....	49
4.4.1	Arquitectura general de la aplicación.....	49
4.4.2	Servicios web.....	50
4.4.3	Aplicación móvil.....	52
4.4.3.1	Interfaz de usuario.....	52
4.4.4	Interacción de los componentes del sistema.....	56
	Resultados y discusión.....	58

5.1 Prueba 1	58
5.2 Prueba 2	61
5.3 Prueba 3	63
Conclusiones y trabajo futuro.....	67
6.1 Conclusiones	67
6.2 Trabajos futuros.	68
Referencias.	69

Índice de Tablas

Tabla 1-Características de la nube.....	25
Tabla 2 Plataformas Emergentes.....	26
Tabla 3 Definición del Operador Alfa.....	29
Tabla 4 Definición del operador Beta	30
Tabla 5- Números Codificados mediante Johnson-Möbius modificado. Ejemplo 3.4.1.....	33
Tabla 6-Números Codificados mediante Johnson-Möbius modificado. Ejemplo 3.4.2.....	34
Tabla 7- Operador Alfa, Ejemplo 3.5.1	35
Tabla 8- Operador Alfa, Ejemplo 5.2	36
Tabla 9- Niveles máximos permisibles de emisión de gases contaminantes.	42
Tabla 10- Tabla de los sistemas operativos más vendidos en el 2013	43
Tabla 11-Estructura del banco de datos.	45
Tabla 12-Clases a emplear en el proceso de predicción.....	47
Tabla 13-Pesos asignados a cada rasgo	48
Tabla 14- Funciones wsSavePatterns	51
Tabla 15- Funciones wsPredictionProcess.....	51
Tabla 16- Patrones por cada clase	59
Tabla 17- Comparación del rendimiento entre el clasificador Gamma y los algoritmos del entorno de análisis Weka para la prueba de predicción 1.....	60
Tabla 18-Resultados del vehículo respecto a la NOM-041-SEMARNAT-2006.....	61
Tabla 19-Patrones por cada clase	61
Tabla 20- Comparación del rendimiento entre el clasificador Gamma y los algoritmos del entorno de análisis Weka para la prueba de predicción 2.....	63
Tabla 21-Resultados del vehículo respecto a la nom-041-semarnat-2006	63
Tabla 22-Patrones por cada clase	64
Tabla 23- Comparación del rendimiento entre el clasificador Gamma y los algoritmos del entorno de análisis Weka para la prueba de predicción 3.....	65
Tabla 24-Resultados del vehículo respecto a la nom-041-semarnat-2006	66

Índice de Figuras

Figura 1-Parte 1 del clasificador Gamma: Pre-procesamiento.....	40
Figura 2- Parte 2 del clasificador Gamma: Funcionamiento del clasificador Gamma.....	41
Figura 3- Perfil de prueba de verificación en condiciones controladas.....	46
Figura 4- Generación de Patrones	46
Figura 5- Arquitectura del sistema	50
Figura 6- Pantalla de captura de datos	52
Figura 7- Estructura de la pantalla de captura.....	53
Figura 8-Explorador de archivos	53
Figura 9- Cargar archivo.....	54
Figura 10- Procesamiento de la información.....	54
Figura 11- Pantallas de inicio de proceso de verificación (Predicción)	55
Figura 12-Pantalla de Resultados	55
Figura 13- Pantalla de Resultados PDF	56
Figura 14-Resultados PDF	56
Figura 15-Diagrama de secuencia de los componentes del sistema.....	57
Figura 16- Valores reales	59
Figura 17-Valores predichos por el clasificador gamma.....	60
Figura 18- Valores reales	62
Figura 19-Valores predichos por el clasificador Gamma	62
Figura 20- Valores reales	64
Figura 21-Valores predichos por el clasificador Gamma	65

Glosario de Términos.

3G - Abreviación de tercera generación de transmisión de voz y datos mediante UMTS (*Universal Mobile Telecommunications System*).

AWS- Amazon Web Service

BFTree- Árbol de Decision Best-First

CO - Monóxido de Carbono

CO2- Dióxido de carbono

Decision Table – Tabla de decision

EC2- Elastic Compute Cloud

HC- Hidrocarburos

MCC- Mobile Cloud Computing

Multilayer Perceptron – Perceptrón Multicapa

NO_x- Óxido de Nitrógeno

PM2.5 - Partículas suspendidas menores de 2.5 micrómetros

RBFNetwork- Red Neuronal de Funciones de Base Radial

S3- Simple Storage Service

Capítulo 1

Introducción

En este trabajo de tesis se presenta el empleo del clasificador Gamma a un entorno de predicción de contaminantes atmosféricos, emitidos a través del escape de vehículos automotores, capturados y alojados en un banco de datos y utilizando como referente a la normativa oficial mexicana, provenientes de fuentes móviles. El proceso de aplicación del clasificador sobre los datos se lleva a cabo empleando cómputo móvil y cómputo en la nube.

1.1 Antecedentes

La contaminación del aire se describe como la contaminación de la atmósfera por cualquier agente físico, químico o biológico que modifique sus características naturales, los principales contaminantes relacionados con la calidad del aire son el bióxido de azufre (SO₂), el monóxido de carbono (CO), los óxidos de nitrógeno (NO_x) las partículas suspendidas, compuestos orgánicos volátiles (COV) y el ozono (O₃). [1]

Los contaminantes del aire emitidos por diversas fuentes perjudican en la salud de personas y animales, además ocasionan daño a las plantas. La calidad del aire ha sido una de las principales problemáticas en todo el mundo a causa de más de 2 millones de muertes al año ocasionadas por la exposición a contaminantes atmosféricos nocivos; con base en lo anterior, el desarrollo de sistemas de supervisión de la calidad del aire se ha hecho imprescindible para la protección de la salud pública. [2]

Anticiparse a las condiciones variantes de la contaminación atmosférica, puede ayudar a crear soluciones que reduzcan el problema. La habilidad de predecir es extremadamente benéfica, la información obtenida a través de ello puede apoyar en la toma de decisiones sustanciales, puesto que podemos conocer con un margen de error una perspectiva del comportamiento tomando como base la información del histórico; es por esto que la predicción concisa de los niveles de contaminación puede colaborar en la planificación social y el crecimiento industrial. Existe una variedad de técnicas utilizadas en la actualidad para la predicción de contaminantes atmosférica, algunas de las principales son:

- Redes neuronales: [3]- [7].
- Lógica difusa: [7]-[9].
- Enfoque probabilístico - estadístico: [10]- [12].
- Enfoque asociativo: [13]-[23].

De las técnicas mencionadas anteriormente, el enfoque asociativo ha demostrado su gran potencial para la solución de una variedad de tareas [22], [23], [24]-[28]. Dentro de este enfoque se encuentra el clasificador Gamma. En el año 2007 investigadores del Centro de Investigación en Computación del Instituto Politécnico Nacional, presentan un nuevo clasificador de alto desempeño [22], [23], el cual utiliza los operadores Alfa y Beta, pertenecientes a las memorias asociativas Alfa-Beta [28]. El clasificador Gamma ha sido probado recientemente en la predicción de emisiones de contaminantes a través del escape de vehículos automotores, utilizando diversos bancos de datos 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 el año 2007 por la EPA (Environmental Protection Agency) [23].

Derivado de la preocupación que embarga desde años atrás y actualmente la calidad del aire en todo el mundo y a las técnicas de predicción de contaminantes que cada día son más extensas y precisas, se han desarrollado una diversidad de proyectos relacionados con el monitoreo de la calidad del aire [29]-[34].

En la Ciudad de México y su área metropolitana, una de las principales fuentes contaminantes antropogénicas son los vehículos automotores, que producen la mayor cantidad de contaminantes, siendo el monóxido de carbono (CO) el gas que más se emite [1]. Desde 1989, en la Ciudad de México se implementó el programa de verificación vehicular, el cual tiene como objetivo detectar unidades con emisiones por encima de los límites máximos permisibles establecidos por la norma oficial mexicana [36]. En enero del año 2000 se publicó la Ley Ambiental para la conservación y el mejoramiento en la calidad del aire [37]. Sin embargo, estos programas que buscan mejorar la calidad del aire y aplicar las acciones para hacer que las unidades detectadas como contaminantes se ajusten a la normatividad vigente no han sido suficiente; por lo que es necesario perfeccionar estos mecanismos de medición y control de emisiones de gases contaminantes; principalmente, los mecanismos relacionados con las principales fuentes contaminantes como lo son los vehículos automotores. El introducir mejoras a los mecanismos existentes, está ligado al uso en conjunto, de las tecnologías, modelos y paradigmas de computación; la agrupación de lo anterior, permitirá desarrollar sistemas que lleven a cabo un uso eficiente de los recursos de dichos modelos y paradigmas. Alguno de los sistemas que tienen como propósito general el monitoreo de ciertas actividades, han utilizado tecnologías como la telefonía móvil [38]-[40] y paradigmas de computación como el cómputo en la nube [41]-[45] y el cómputo móvil [46], [47].

Además del anteriormente mencionado, otro trabajo involucrado en la problemática de la ciudad de México con respecto a la calidad del aire, es el proyecto PIUTE 10-77 “*Sensor*

Inteligente para el monitoreo de los Niveles de Emisión de contaminantes en un Vehículo Automotor”, cuyo desarrollo toma lugar en el Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN) y está orientado al parque vehicular de la Ciudad de México. Uno de los objetivos de este trabajo es la de contribuir de manera directa con los programas de mejora de la calidad del aire de la ciudad de México a través de un sistema de monitoreo de emisiones “reales” a bordo del vehículo. La idea de dicho proyecto es que se puede mejorar el PVV al combinar distintos tipos de tecnología, desde las relacionadas con detección de emisiones contaminantes provenientes del escape de los automóviles hasta las tecnologías móviles con el fin de tener un mejor manejo y administración de las emisiones contaminantes automotrices.

1.2 Justificación

La contaminación del aire es hoy en día un problema grave que aumenta exponencialmente; ambientalistas y organizaciones no gubernamentales de todo el mundo han comenzado a realizar acciones que colaboren a contener y disminuir dicho problema. Sabiendo que los automóviles son una de las principales fuentes de contaminantes, es importante el desarrollo de un sistema de predicción de contaminantes, que produzca datos fiables sobre la concentración de gases contaminantes provenientes de las emisiones del escape de vehículos automotores. El sistema propuesto en este trabajo de tesis podrá mostrar a los usuarios del mismo, mediante los datos reales de operación de un vehículo, las circunstancias en las que se encuentra su vehículo en la actualidad, dando pauta a la toma de decisiones tempranas que favorezcan a la mitigación de los niveles de emisiones contaminantes.

1.3 Objetivo general

Diseñar un sistema informático que permita procesar los datos obtenidos de los niveles de emisión de contaminantes provenientes de un vehículo automotor, e incorporar un algoritmo de predicción que permita predecir el comportamiento de la información recibida, para posteriormente ser comparada con las normas oficiales mexicanas vigentes.

1.4 Objetivos específicos

- Analizar y determinar las características del algoritmo de clasificación para la predicción de contaminantes atmosféricos a emplear en el sistema.
- Integrar y desarrollar, mediante una plataforma de código abierto, una aplicación móvil que tenga como objetivo realizar la predicción de los niveles de emisión de contaminantes provenientes de un vehículo automotor y mostrar los resultados de la misma.

- Consolidar en una base de datos la adquisición de la información y el resultado de su procesamiento para su interpretación, con la finalidad de revisar los niveles de contaminación emitida por un vehículo automotor.
- Emplear el cómputo en la nube para el procesamiento de la información adquirida por medio de la aplicación móvil y para la comparación con las normas oficiales mexicanas vigentes de dicho procesamiento.
- Realizar experimentos para probar el sistema en su totalidad y validar la implementación del modelo de clasificación propuesto.

1.5 Contribuciones

Un sistema basado en el cómputo móvil y cómputo en la nube utilizado para la obtención, procesamiento, almacenamiento y predicción de la información de contaminantes atmosféricos provenientes de un vehículo automotor, haciendo uso del clasificador Gamma.

1.6 Organización del documento

El contenido de los siguientes capítulos para el presente trabajo de tesis se describe a continuación:

El capítulo 2 presenta el estado del arte, donde se abordan de manera puntual los temas esenciales en la investigación realizada, como lo son: predicción de contaminantes, sistemas de monitoreo de contaminantes atmosféricos, cómputo móvil y cómputo en la nube.

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

El capítulo 4 es la parte relevante de la tesis, puesto que se describe y se sustenta teóricamente el desarrollo de la solución para el trabajo de tesis.

En el capítulo 5 se muestran los resultados de los experimentos realizados, y finalmente el capítulo 6 incluye las conclusiones, las aportaciones y recomendaciones para trabajos futuros.

Capítulo 2

Estado del arte

El presente capítulo está estructurado en 3 secciones: la primera aborda los principales métodos empleados para la predicción de contaminantes, la segunda sección expone el crecimiento y las aportaciones obtenidas hasta hoy en día del cómputo móvil y el cómputo en la nube, y finalmente se presentan determinados sistemas enfocados a la predicción, generados a partir de la interacción entre ambos paradigmas anteriormente mencionados.

2.1 Técnicas para la predicción de contaminantes.

La contaminación del aire en años recientes se ha convertido en uno de los problemas más críticos en relación con el medio ambiente y la calidad de vida de los seres humanos; debido a esto se han desarrollado herramientas robustas que ayuden a prevenir el aumento en los niveles de contaminación.

2.1.1 Redes neuronales artificiales.

Las redes neuronales artificiales (RNA) fueron diseñadas con el objetivo de generar máquinas inteligentes para resolver problemas, tales como la predicción y optimización, imitando la forma en la que funciona u operan las neuronas en el cerebro humano [3].

Las redes neuronales artificiales consisten por lo general en un gran número de neuronas; i.e., elementos simples de cómputo lineales o no lineales interconectados generalmente de maneras complejas y organizados en capas. Las RNA tienen la capacidad de procesar un extenso conjunto de datos y realizar predicciones con extraordinaria exactitud. Debido a las capacidades antes mencionadas, las RNA han sido utilizadas para la predicción de ciertos contaminantes atmosféricos [3] – [7]. Se han utilizado para predecir emisiones de NO_x, PM, CO, HC procedentes de un motor diésel [6].

Respecto a la predicción de contaminantes atmosféricos, se han definido diferentes modelos de RNA: Ming Cai et al. [4] presentan en su trabajo el diseño de modelos basados en el perceptrón multicapa y la comparación con otros modelos como regresión

lineal múltiple y el modelo de origen de línea de California (CALINE, por sus siglas en inglés); en [5] Dahe Jiang et al. muestran el desarrollo de una RNA feed-forward perceptron multicapa para auxiliar al sistema de monitoreo del índice de contaminación del aire en la ciudad de Shanghai.

2.1.2 Lógica difusa.

En este enfoque se crean funciones de pertenencia, las cuales convierten un parámetro medible objetivamente en una pertenencia sujeta 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 [8].

Debido a que un modelo basado en la lógica difusa no utiliza fórmulas tediosas empíricas y expresiones matemáticas complejas, proporciona una manera transparente y un análisis sistemático de la interpretación del comportamiento dinámico de un problema ambiental basado en un conjunto de conectores lógicos [7].

N. Bose et al. [9], presenta en su trabajo la predicción de smog y NOx a través de técnicas de lógica difusa.

2.1.3 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, naive 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 [10]. Cheon et al. [10], comparan una red bayesiana con un árbol de decisión, para predecir valores máximos de O₃; los resultados muestran que las redes bayesianas obtienen mejores resultados que los modelos basados en arboles de decisión. Las redes bayesianas brindan una ventaja adicional, permiten realizar un análisis de causas para determinar qué factores pueden influir en el incremento en la concentración de O₃ y poder desarrollar planes de contingencia.

Otro método que se ha aplicado a la predicción de la calidad del aire es el clasificador naive 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. [11], muestran la aplicación de este clasificador a la predicción de los niveles de CO₂ 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 [12]. Dong et al. [12], proponen el uso de un modelo oculto de Markov para la predicción de la concentración de PM2.5. El modelo presentado en este estudio, incorpora una modificación con respecto a los modelos tradiciones de Markov; esta

modificación le permite incorporar estructuras de tiempo adecuadas para problemas de predicción.

2.1.4 Enfoque asociativo.

El primer modelo matemático de memoria asociativa fue propuesto por Karl Steinbuch en 1961, la Lernmatrix [13], que es una memoria heteroasociativa que puede funcionar como clasificador de patrones binarios. En 1969 Willshaw, Buneman y Longuet-Higgins, presentan el Correlograph [14], dispositivo óptico capaz de funcionar como una memoria asociativa.

En el año 1972 se realizaron diversas investigaciones sobre memorias asociativas; James A. Anderson con su Interactive Memory [15], Teuvo Kohonen presentó sus Correlation Matrix Memories [16], Kaoru Nakano dio a conocer su Associatron [17] y Amari realizó algunos aportes teóricos [18], el cual resulta de gran importancia ya que establece antecedentes para uno de los modelos más distinguido de memorias asociativas, la memoria de Hopfield [19]. En 1982 el físico John J. Hopfield publicó su artículo, que representa un punto de referencia en el desarrollo de las memorias asociativas y redes neuronales.

Es en 1998 cuando Ritter et al. [20], 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 entonces.

El enfoque Alfa-Beta fue creado en el centro de investigación en computación del IPN en 2002 y hace uso de las memorias asociativas morfológicas; el objetivo principal de estas memorias es recuperar patrones binarios completos a partir de patrones de entrada que pueden estar alterados.

En el trabajo que se presenta en [21], se propone el uso del clasificador Gamma; este clasificador de alto desempeño fue presentado en el año 2007 [22], para la predicción de la concentración de contaminantes en el medio ambiente. El clasificador utiliza los operadores Alfa y Beta, que son a su vez la base de las memorias asociativas Alfa-Beta. En el año 2011 se diseñó e implementa un método para el procesamiento y predicción de datos atmosféricos generados por el escape de vehículos automotores utilizando el clasificador Gamma [23].

2.2 Cómputo móvil.

El cómputo móvil describe el uso de dispositivos de cómputo, los cuales usualmente interactúan de cierta manera con un sistema de información central. La tecnología referente al cómputo móvil permite a los usuarios generar, consultar, almacenar procesar y comunicar información, sin limitarse a una ubicación fija [38].

La capacidad de un entorno de computación móvil está basada en ciertos factores de movilidad como: el consumo de energía, portabilidad, poder de cómputo, espacio libre en memoria, entradas de usuario y características de visualización, ancho de banda y persistencia [39].

Los teléfonos inteligentes se han convertido en dispositivos con importantes capacidades de cómputo, que continúan mejorando en cuanto a potencia en el procesamiento y funcionalidades, teniendo así el potencial de convertirse en un dispositivo de computación personal [40].

2.2.1 Comunicación inalámbrica.

Internet y la World Wide Web han dado lugar a la demanda de acceso a internet desde terminales móviles. Debido a esto, se ha producido el desarrollo de nuevos sistemas de acceso inalámbrico y tarjetas de acceso inalámbrico incorporadas en terminales móviles. Los productos que operan bajo el estándar de una red de área local inalámbrica o WLAN (del inglés Wireless Local Area Network) definido en IEEE 802.11 [48], han sido aceptados exitosamente por los usuarios debido a que se ha asegurado la interoperabilidad de los productos.

Un sistema de cómputo móvil es un conjunto de dispositivos móviles y transceptores estáticos conectados por una red que también es estática. Los dispositivos móviles se comunican entre ellos a través de una comunicación inalámbrica, a la vez que los transceptores se comunican entre ellos a través de la red alámbrica.

El cómputo en la nube móvil utiliza una variedad de protocolos de conexión para la comunicación, entre estos están WiFi, Bluetooth, 3G y LTE. WiFi (wireless Ethernet 802.11b) y Bluetooth operan en la banda de 2,4 GHz. WiFi tiene un alcance con un radio de aproximadamente 300m en un espacio abierto y soporta velocidades de hasta 11Mbps [49].

2.2.2 Modelo cliente móvil - servidor fijo.

Una de las arquitecturas más utilizadas en la computación es el modelo cliente-servidor, en el cual el cliente solicita un servicio en otro ordenador y el servidor para el cual se generó dicha solicitud realiza los cálculos, gestiona los datos y todos los recursos compartidos sobre una red fija.

Este modelo ha sido adaptado a un entorno de cómputo móvil, en el cual cualquiera de las peticiones realizadas desde el dispositivo móvil es enviada a través de un host móvil a la red fija.

Este modelo puede tener variaciones en el mecanismo de comunicación de la información, como [51]:

- I. Las solicitudes y respuestas directas entre el cliente móvil y el servidor no son recomendables, debido a la lentitud o falta de conexión que pueden ocurrir si la red no es confiable*
- II. Encolamiento de las solicitudes si es que no se ha establecido una conexión, hasta que ocurra alguna entre el host móvil y el servidor fijo para que así la*

cola de mensajes sea transmitida al servidor para su procesamiento.

III. *Suministrar varios servidores que están particionados o duplicados para aumentar tanto la disponibilidad del servidor como su rendimiento.*

2.2.3 Restricciones del cómputo móvil.

La red inalámbrica tiene como características bajo ancho de banda, una alta tasa de error y desconexiones frecuentes debido a que el ambiente que la rodea interactúa con la señal bloqueándola e introduciendo ruido y eco. Estos factores pueden incrementar la latencia en la comunicación [50].

Algunas de las limitaciones más importantes son las siguientes:

I. *Ancho de banda.*

La red inalámbrica proporciona un ancho de banda menor que las conexiones cableadas y este puede variar de 1 a 4 órdenes de magnitud dependiendo del tipo de conexión.

II. *Riesgos de seguridad.*

Cuando un dispositivo móvil necesita realizar una tarea que depende de las redes públicas, la seguridad puede ser comprometida debido al fácil acceso a las redes inalámbricas, especialmente si la transmisión se extiende sobre una gran área.

III. *Consumo de energía.*

Los dispositivos móviles dependen en gran parte de una batería bastante limitada (debido a su tamaño compacto) como fuente de energía. En los componentes dinámicos, el consumo de energía es proporcional a CV^2F [51], donde C es la capacitancia del circuito, V es la oscilación del voltaje y F es la frecuencia de reloj.

IV. *Capacidades de cómputo*

Algunas aplicaciones requieren de procesamiento intensivo para realizar tareas como el procesamiento de imágenes, minería de datos, realidad aumentada o síntesis de voz, todas estas aplicaciones exigen capacidades de cómputo altas, por lo cual es difícil implementar dichas aplicaciones en dispositivos móviles, debido a que dichas actividades son realmente costosas en términos de energía y almacenamiento para un dispositivo móvil [54].

2.2.4 *Cómputo en la nube móvil.*

Los dispositivos móviles están propiamente limitados por el procesamiento, almacenamiento y la energía limitada de una batería [53]. Las aplicaciones en tiempo real son uno de los diversos tipos de aplicaciones móviles que demanda recursos intensivos de cómputo [52]. Debido a la abundancia y el fácil acceso a los recursos públicos del cómputo en la nube, la nube móvil tiene la capacidad de extender sus capacidades; el cómputo intensivo y el almacenamiento de grandes cantidades de datos deben migrar hacia la nube [53], [54].

El cómputo en la nube, las limitaciones de los dispositivos móviles, la infraestructura ubicua inalámbrica, y los servicios de localización, han conllevado a la formación de un nuevo paradigma de computación llamado cómputo en la nube móvil o *MCC* (del inglés *Mobile Cloud Computing*). *MCC* puede ser definido como un servicio que permite a los usuarios móviles, ajustar las capacidades de procesamiento y almacenamiento según se requiera, mediante la descarga del cómputo intensivo y el almacenamiento exhaustivo generado por ciertas tareas, en recursos de la nube proporcionando acceso inalámbrico en cualquier momento [54].

La descarga de datos y la ejecución de procesos en la nube, son utilizadas por otros proveedores de recursos (distintos al dispositivo propio en sí) para albergar la ejecución de las aplicaciones móviles. Esta infraestructura en la cual el almacenamiento y el procesamiento de los datos suceden extrínsecamente al dispositivo móvil se le puede llamar *nube móvil* o *mCloud* [52].

2.2.5 *Modelos de programación mCloud.*

Uno de los modelos más recientes *CloneCloud* [54], propone la descarga de una parte de la ejecución de una aplicación móvil en clones del dispositivo móvil que operan dentro de la nube.

El generar particiones apropiadas de la aplicación móvil es resuelto mediante la combinación de un análisis estático y dinámico de perfiles; al mismo tiempo que minimiza el tiempo de ejecución y el consumo de energía

En este trabajo se emplean máquinas virtuales de aplicación, como *DalvikVM* la máquina virtual de Java; debido a que son ampliamente utilizadas en plataformas móviles y es relativamente sencillo manipular ejecutables de aplicaciones y migrar piezas a dispositivos computacionales de arquitecturas diferentes.

MAUI [56] soporta la descarga de código para minimizar el consumo de energía y cambios requeridos para las aplicaciones. Estos beneficios han sido obtenidos de varias propiedades adquiridas de los entornos de código gestionado como Microsoft *.NET Common Language Runtime*.

MAUI hace uso de la portabilidad del código para generar dos versiones de un dispositivo móvil, en donde una de las versiones se ejecuta localmente en el dispositivo móvil y la otra es ejecutada remotamente en la nube. Además, *MAUI* utiliza técnicas de

reflexión combinadas con la seguridad de tipos para identificar de manera automática los métodos de uso remoto, y obtener solo el estado del programa que necesitan dichos métodos.

MAUI perfila cada método de la aplicación y mediante la serialización determinan el coste de la descarga de cada método. Todo lo anterior es empleado para determinar que métodos deben ser ejecutados en la nube y que debe ejecutarse localmente en el dispositivo móvil.

2.3 *Cómputo en la nube.*

El cómputo en la nube puede y ha sido definido de diferentes formas dependiendo del punto de vista de cada espectador, desarrollador, investigador, usuario, etc., se puede ver como aplicaciones basadas en la web, o como una poderosa herramienta de cómputo, o una red en la que se cobran ciertas tarifas por tiempo de procesamiento; incluso los líderes en este tema difieren en ciertos aspectos en la visión que cada uno tiene de ello. En la siguiente mesa redonda de discusión acerca del cómputo en la nube [41], se definen al cómputo en la nube de diversas formas:

- I.** Es más que solo una tendencia en la prestación de servicios que ocurren en la actualidad, es el movimiento de los servicios de aplicación en el Internet y el aumento del uso de Internet para acceder a una amplia variedad de servicios que tradicionalmente se localizaban en un centro de datos.
- II.** La nube es una plataforma para el cómputo general y / o servicios.
- III.** Es la unión de tres capas: software como un servicio o SaaS (software-as-a-service, por sus siglas en inglés), que ofrece aplicaciones como Google Apps y Salesforce.com, plataforma como un servicio o PaaS (platform-as-a-service), que proporciona elementos fundamentales para el desarrollo de nuevas aplicaciones; e infraestructura como un servicio o IaaS (*Infrastructure* -as-a-service), donde se muestra que la infraestructura también se puede acceder a través de la nube.
- IV.** Es el espacio de cómputo de internet disponible para el sector público o privado.

Aunque estas definiciones son ciertas; el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) en la publicación presentada por Peter Mell et al.[42] definen al cómputo en la nube como un modelo que permite el acceso a la red de manera ubicua y conveniente a un conjunto compartido de recursos de computación configurables (ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser suministrados rápidamente y liberados con el menor esfuerzo de gestión o interacción del proveedor del servicio. El modelo de la nube se compone de 3 modelos de servicios y 2 modelos de implementación.

2.3.1 Servicios en la nube.

Con base en los servicios ofrecidos en la nube, se pueden clasificar en tres tipos:

I. *Software como servicio.*

El software como servicio o (*SaaS por sus siglas en inglés*), es probablemente el servicio más común que ofrece la nube. Con el software como servicio, los usuarios pueden acceder a una aplicación—o servicio—ofrecida por el proveedor de la nube; el cliente no paga por la adquisición del software, sino por el uso. La aplicación está disponible para el usuario en cualquier momento a través de internet [43], [44], [45].

II. *Plataforma como servicio.*

La plataforma como servicio consiste en ofrecer un entorno de desarrollo total o parcial en la nube, al cual se puede acceder y utilizar en línea. El desarrollador utiliza los bloques de construcción del entorno de desarrollo del proveedor para generar sus propias aplicaciones. Las plataformas proporcionadas por cada proveedor no suelen ser compatibles.

III. *Infraestructura como servicio.*

La infraestructura como servicio consiste en ofrecer servicios relacionados con hardware, utilizando los principios del cómputo en la nube. Este servicio proporciona a los clientes un amplio conjunto de recursos informáticos, como las máquinas virtuales, a través de internet.

El servicio de infraestructura está dirigido a las empresas de TI y desarrolladores de software, permitiéndoles un incremento o decremento el número de máquinas virtuales en ejecución dependiendo de la carga de trabajo, de esta manera se promueve la eficiencia en el uso de recursos de TI [43], [44], [45].

Cada uno de estos tipos de servicios posee diferentes propuestas y objetivos, pero todos comparten un modelo de negocios; la idea de este modelo reside en rentar el uso de sus recursos de cómputo (infraestructura, aplicaciones y plataformas) a los clientes [44].

2.3.2 Modelos de implementación de la nube.

El cómputo en la nube con base en su implementación puede dividirse en los siguientes modelos:

I. Nube pública

La infraestructura informática es colocada en cualquier parte de las instalaciones del proveedor de la nube– el cliente no tiene control sobre donde se puede localizar –y puede ser compartida entre varias empresas. Los servicios y recursos ofrecidos por el proveedor de la nube son alcanzables para el público vía Internet; el cliente realiza un pago por los servicios o recursos que utiliza [43], [44].

La mayor preocupación en este modelo es el tema de la seguridad debido a que la manera en que el sistema es organizado es determinada por el proveedor de la nube y es accesible al público [44].

II. Nube privada

La infraestructura informática es destinada a una organización en específico, i.e., no es compartida entre empresas. Los servicios y recursos son alcanzables desde un organismo privado.

Este modelo de implementación hace énfasis en los beneficios de la integración, optimización del hardware y la escalabilidad. Por su naturaleza, la nube privada tiene un costo más elevado que la nube pública, sin embargo el nivel de seguridad es mayor [43], [44].

2.3.3 Características del cómputo en la nube.

El cómputo en la nube posee atributos específicos y capacidades como un gran soporte para la virtualización, servicios con interfaces de servicios web, almacenamiento y servicios de aplicaciones [45].

TABLA 1-CARACTERÍSTICAS DE LA NUBE

Características de cómputo en la nube	
Composición	Computadoras comerciales, servidores de gama alta, almacenamiento conectado a la red.
Tamaño/Escalabilidad	100 a 1000
Sistema operativo	Un Hipervisor (Máquina Virtual) en el cual múltiples sistemas operativos se ejecutan.
Red/Velocidad	Dedicada, de gama alta con baja latencia y ancho de banda alto
Seguridad/Privacidad	<ul style="list-style-type: none"> ✓ Cada usuario o aplicación esta provista de una máquina virtual. ✓ Alto nivel de seguridad y privacidad garantizado ✓ Soporte para la creación de listas de control de Acceso
Servicio de negociación	Basado en los acuerdos de nivel de servicio, que son negociados entre los proveedores y consumidores.
Gestión de recursos	Centralizado / Distribuido
Distribución/ Planificación	Centralizada y descentralizada
Estándares / Interoperabilidad	Servicios web (SOAP y REST)
Gestión de fallos	Proporciona un fuerte apoyo para la tolerancia a fallos y la duplicación de contenido. Las máquinas virtuales pueden migrar con facilidad de un nodo a otro.

2.3.4 Plataformas emergentes.

TABLA 2 PLATAFORMAS EMERGENTES

Característica	Sistema				
	Amazon Elastic Compute Cloud (EC2)	Google App Engine	Microsoft Azure	Sun Network.com (Sun Grid)	GRIDS Lab Aneka
Se enfoca en	Infraestructura	Plataforma	Plataforma	Infraestructura	Plataforma de software para empresas
Tipo de servicio	Cómputo, Almacenamiento (Amazon S3)	Aplicaciones Web	Aplicaciones Web y aplicaciones no basadas	Cómputo	Cómputo
Virtualización	A nivel de sistema operativo ejecutándose en un hipervisor Xen	Contenedor de aplicación	A nivel de sistema operativo a través de Azure Fabric	Sistema de encolamiento de procesos distribuido Sun Grid Engine	Administrador y planificador de Recursos
Negociación dinámica de los parámetros de la Calidad del Servicio	Ninguna	Ninguna	Ninguna	Ninguna	Acuerdo de nivel de servicios basado en reservas de
Interfaz de acceso a los usuarios	Herramientas de línea de comandos Amazon EC2	Consola de administración basada en la Web	Portal de Windows Azure	Portal Web Sun Grid	Workbech, Portal Web
Web APIS	SI	SI	SI	SI	SI
Framework de programación	Amazon Machine Image (AMI) basada en linux	Python	Microsoft.NET	OS Solaris, Java, C, C++ ,FORTRAN	APIs que soportan diferentes modelos de programación en C# y otros lenguajes soportados

2.4 Modelos y aplicaciones existentes.

2.4.1 Modelos.

En el trabajo presentado por M. Di Francesco [46] se habla de los dispositivos móviles como fuentes enriquecidas de recolección de datos por usuarios. Se propone un framework para almacenar, procesar y entregar datos de ciertos sensores a aplicaciones alojadas en la nube.

El framework propuesto consta de diversos módulos, los cuales hacen uso de servicios como, *Amazon Simple Notification System (SNS)* y *Simple Queue Service (SQS)*. *SNS* se encarga del envío de notificaciones desde la nube, las notificaciones pueden tomar la forma de un SMS, un correo electrónico, entre otros. *SQS* es un mecanismo asíncrono el cual permite la transferencia de mensajes o datos entre elementos distribuidos, sin pérdida de mensajes, aun cuando los elementos no estén disponibles. Además de los servicios mencionados anteriormente, el framework hace uso de *Amazon SimpleDB* y *Amazon Simple Storage Service (S3)* para el almacenamiento de la información.

X. Zhang et al. [47], proponen un modelo de aplicación elástica que mediante el aprovechamiento de los recursos de la nube, incrementa la capacidad de los recursos limitados del dispositivo móvil. El propósito de este trabajo consiste en diseñar una arquitectura y un middleware que permita utilizar aplicaciones elásticas que constan de múltiples elementos a los cuales se les da el nombre de *weblets*. Para interactuar con este nuevo modelo de aplicación, desarrollaron un framework de referencia el cual contiene la aplicación, la arquitectura y ejemplos de aplicaciones elásticas; el cual funciona con Amazon Elastic Compute Cloud (EC2) y *Amazon Simple Storage Service (S3)*.

2.4.2 Aplicaciones.

2.4.2.1 SensorScope.

SensorScope [33], es un proyecto basado en una red de sensores inalámbricos que tiene por objeto proporcionar un sistema de monitoreo ambiental abierto de bajo costo y confiable y mejorar las técnicas de recolección de datos con la última tecnología. Este sistema de monitoreo abarca desde el hardware (estaciones sensoras) y software cliente/servidor, hasta el software de base de datos y una interfaz web conveniente. Las estaciones sensoras periódicamente toman una muestra de sus sensores y transmiten los valores recolectados a través de la red inalámbrica a un punto de recopilación o estación base. La estación base envía estos datos a un servidor central y el cual está a disposición del usuario en tiempo real.

SensorScope permite a los científicos ambientales utilizar un portal para estudiar, analizar y controlar el proceso de medición a través de una interfaz web [33].

2.4.2.2 *OpenSense Open Community Driven Sensing of Enviroment.*

OpenSense [31], es un proyecto el cual tiene como objetivo explorar una comunidad de sensores que utiliza redes de sensores inalámbricos en el contexto de la supervisión de la contaminación atmosférica. En la comunidad de sensores, sensores y consumidores involucran personas y a sus dispositivos (teléfonos celulares, relojes y termómetros en vehículos), unidades sensoras personalizadas para medir polvo o gases tóxicos los cuales son trasportados por la comunidad (personas, autobuses, automóviles, etc.). Lo anterior da pie a un paradigma de sensado no estructurado, donde la comunidad participa recolectando datos; y de esta manera la propia comunidad esperaría un panorama más exacto del medio ambiente que el actual. Dentro de sus objetivos de optimización se encuentran: (1) reducir el consumo de recursos y por consecuente el costo de la infraestructura, mientras que aumenta la exactitud de la información obtenida; (2) optimizar la exactitud de los datos, (3) minimizar la trasmisión, análisis y almacenamiento de los datos obtenidos.

La comunidad de sensores se enfrenta a grandes retos técnicos debido a la infraestructura abierta y escalable que posee; por ejemplo, los sensores son públicos o privados, por lo tanto pueden ser utilizado por el propietario durante la actividad de sensado; pueden tener diferentes capacidades respecto a batería, precisión o comunicación; los sensores de la calidad del aire en objetos en movimiento provocan perturbaciones moderadas y además con el paso del tiempo es necesario calibrarlos nuevamente [31].

2.4.2.3 *EcoDrive.*

Es una aplicación GPS móvil para ayudar a reducir la huella de carbono, la cual sugiere la ruta más eficiente respecto a la cantidad de combustible que se emplea. La aplicación utiliza el framework UCFC (Ubiquitous personal Carbon Footprint Calculation Platform). El usuario proporciona información acerca de su vehículo (modelo, tipo de combustible, etc.) para generar un perfil localmente para uso futuro. El framework resuelve el punto de partida y llegada mediante una API de codificación geográfica. Esta información en conjunto con el perfil del vehículo es utilizada para calcular las rutas [32].

Arquitectura del sistema.

Módulo de sensor: Se utiliza el sensor GPS del dispositivo móvil, el cual permite determinar la posición actual del usuario.

Modulo del dispositivo móvil: Este módulo almacena temporalmente los datos obtenidos del sensor. Puede enviar datos como la localización y destino del usuario a un servidor en forma de XML o servicio web.

Módulo de servidor: Realiza la función principal de calcular la huella de carbono para cada usuario por medio del modelo cuantitativo integrado basado en el perfil del vehículo. El servidor contiene una base de datos de varios tipos de automóviles y su correspondiente consumo milla por galón [32].

Capítulo 3

Materiales y métodos

En este capítulo se describen los materiales, métodos y conceptos necesarios para fundamentar teóricamente y realizar el desarrollo del modelo que se presenta en el capítulo 4. Este capítulo cuenta con 6 secciones, en la primera sección se describen los operadores Alfa (α) y Beta (β) [28], esenciales en la implementación del algoritmo de clasificación Gamma [22]. En la segunda sección se describe un tercer operador empleado por el clasificador Gamma llamado *operador de Gamma de similitud*. En la sección 3, se describe el código Johnson-Möbius modificado, debido a que los patrones con los que trabaja el clasificador deben estar codificados con este método. La descripción de la operación módulo, operación utilizada por el clasificado Gamma se encuentra en la sección 4. La sección 5 está dedicada a la descripción del clasificador Gamma, el cual es empleado como propuesta en el desarrollo de este trabajo de tesis. En la sección 6 se detalla el banco de datos utilizado para realizar los experimentos.

3.1 Operadores Alfa (α) y Beta (β).

Debido a que estos operadores son fundamentales para el clasificador Gamma, es necesario dar una descripción detallada de los mismos. La definición de estos dos operadores y sus propiedades fueron demostradas en [28] y retomadas en los trabajos de tesis [56], [22], [23]. A continuación se exponen las tablas que representan a los operadores α y β . Dados los conjuntos:

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

La operación binaria α está definida en la siguiente tabla:

TABLA 3 DEFINICIÓN DEL OPERADOR ALFA.

x	y	$\alpha(x,y)$
0	0	1
0	1	0
1	0	2
1	1	1

La operación binaria β está definida en la siguiente tabla:

TABLA 4 DEFINICIÓN DEL OPERADOR BETA

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}_β .

En esta sección se detalla el operador \mathcal{U}_β , definido en el trabajo de tesis [22], para su utilización en el desarrollo del clasificador Gamma.

Definición 3.2.1 [22]: Sean: el conjunto $A = \{0,1\}$, un número $n \in \mathbb{Z}^+$ y $\mathbf{x} \in A^n$ un vector binario de dimensión n , con la i -ésima componente representada por x_i . Se define el operador $\mathcal{U}_\beta(\mathbf{x})$ de la siguiente manera: $\mathcal{U}_\beta(\mathbf{x})$ tiene como argumento de entrada un vector binario n -dimensional \mathbf{x} y la salida es un número entero no negativo que se calcula de la siguiente manera:

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

Ejemplo 3.2.1 [22]: Sea $x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$; obtener $\mathcal{U}_\beta(x)$

Dada la definición 3.2.1, $\mathcal{U}_\beta(x) = \sum_{i=1}^4 \beta(x_i, x_i)$, por lo que $\mathcal{U}_\beta(x) = \beta(1,1) + \beta(0,0) + \beta(1,1) + \beta(0,0) = 1 + 0 + 1 + 0 = 2$. Entonces, $\mathcal{U}_\beta(x) = 2$.

Ejemplo 3.2.2 [22]: Sea $x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$; obtener $\mathbf{u}_\beta(x)$

Para el ejemplo 3.2.2, $\mathcal{U}_\beta(x) = \sum_{i=1}^6 \beta(x_i, x_i)$, por lo tanto $\mathcal{U}_\beta(x) = \beta(1,1) + \beta(0,0) + \beta(1,1) + \beta(1,1) + \beta(0,0) + \beta(0,0) = 1+0+1+1+0+0 = 4$. Entonces $\mathbf{u}_\beta(x) = 4$.

3.3 Módulo.

El concepto de módulo es importante en el desarrollo del clasificador Gamma. El contenido de esta sección se obtuvo de [57].

En algunas situaciones es de nuestro interés sólo el residuo de un número entero cuando es dividido por un número entero positivo. Debido a lo anterior, se cuenta con el operador *módulo* especial para ello, denotado como *mod*.

Definición 3.1.3 [57]: Sea a un número entero y m un número entero positivo. Se denota por $a \bmod m$ al residuo de dividir a por m .

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

Ejemplo 3.3.1 [57]: Se tiene que $101 = 11 * 9 + 2$, por lo tanto el cociente cuando 101 es dividido por 11 es 9 y el residuo es 2, que es igual a: $101 \bmod 11 = 2$.

3.4 Código binario Johnson- Möbius modificado.

Los vectores con los que trabaja el clasificador Gamma deben ser codificados con el código Johnson-Möbius modificado [58], el cual es descrito en la presente sección.

Algoritmo 3.4.1 [58]: Algoritmo utilizado para la obtención del código Johnson-Möbius modificado:

1. Sea un conjunto de números reales.

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

donde n es un número entero positivo fijo.

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

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

donde $t_j = r_j - r_i \forall j \in \{1, 2, \dots, n\}$ y particularmente $t_i = 0$. Nota: 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 no son negativos) exactamente a d decimales.
4. Realizar un escalamiento de 10^d en el conjunto del paso 3, para finalmente se tenga un conjunto de n enteros negativos.

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

donde e_m es el número mayor.

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

Ejemplo 3.4.1 [58]: Sea el conjunto $r = \{0.6, 3.209, 0.456, -1.5\}$ donde $r \subset \mathbb{R}$, se tiene lo siguiente:

Paso 1: $r = \{0.6, 3.209, 0.456, -1.5\}$

Paso 2: Existe un número negativo (-1.5), por lo que se obtiene el conjunto transformado

$$r = \{0.6 - (-1.5), 3.209 - (-1.5), 0.456 - (-1.5), -1.5 - (-1.5)\}$$

$$t = \{2.1, 4.709, 1.956, 0.0\}$$

Paso 3: Se escoge el número fijo $d=1$ para obtener:

$$t = \{2.1, 4.7, 1.9, 0.0\}$$

Paso 4: Se realiza el escalamiento de 10^d para obtener $e = \{21, 47, 19, 0\}$, donde $e_m = 47$, 47 es el número mayor del conjunto e .

Paso 5: Para cada número e_i del conjunto e , se generan $(e_m - e_j)$ ceros concatenados con e_i unos. Los resultados se muestran en la siguiente tabla.

1. Cada número se codifica con e_m bits.
2. $e_1 = 21$, por lo tanto: $47 - 21 = 26$ ceros concatenados de 21 unos.
3. $e_2 = 47$, por lo tanto: $47 - 47 = 0$ ceros concatenados de 47 unos.

Tabla 6-NÚMEROS CODIFICADOS MEDIANTE JOHNSON-MÖBIUS MODIFICADO. EJEMPLO 3.4.2

Númer	Código Johnson-Möbius modificado
39	111
39	11
30	0000000001111111111111111111111111111111111111
0	000

3.5 Clasificador Gamma.

En esta sección se describe el clasificador de patrones de alto desempeño Gamma, el cual fue propuesto en [22]. Dicho clasificador, emplea el operador Gamma de similitud (γ); el cual es detallado en esta sección, y los operadores Alfa (α), Beta (β), \mathcal{U}_β previamente descritos. Los patrones utilizados por el clasificador están codificados mediante el código Johnson-Möbius modificado, del cual se expusieron ejemplos en el apartado anterior.

3.5.1 Operador Gamma de similitud (γ).

El operador Gamma de similitud está basado en los operadores Alfa (α) y Beta (β) de las memorias asociativas Alfa-Beta, este operador indica si dos vectores son parecidos o no, dado un grado de disimilitud θ ; por ello, este operador es de similitud. El argumento θ , indica la tolerancia para que dados dos vectores, al compararlos sean considerados similares, no obstante que son diferentes. A continuación se define el operador gama de similitud.

Definición 4.1 [22]: 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 gama de similitud $\gamma(x, y, \theta)$ de la siguiente manera: $\gamma(x, y, \theta)$, tiene como argumento de entrada dos vectores binarios n -dimensionales x 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 [22]: Sea $x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$, $y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ y $\theta = 3$; calcular $\gamma(x, y, \theta)$

1. En este caso $n = 5$
2. Calcular $\alpha(x, y)$:

TABLA 7- OPERADOR ALFA, EJEMPLO 3.5.1

x	y	$\alpha(x, y)$
1	0	2
0	0	1
0	1	0
1	0	2
1	1	1

3. Aplicar el módulo 2 a cada componente del vector que obtuvimos en 1.

$$\alpha(x, y) = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}, \text{ modulo } 2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

4. Aplicar el operador \mathcal{U}_β al vector obtenido en 3.

$$\mathcal{U}_\beta = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 2$$

Al restarlo de $n = 5$, se obtiene $5-2=3$; $3 \leq \theta$, por lo tanto: $\gamma(x, y, \theta) = 1$

Ejemplo 3.5.2 [22]: Sea $x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$, $y = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ y $\theta = 2$; calcular $\gamma(x, y, \theta)$

1. En este caso $n = 7$

2. Calcular $\alpha(x, y)$:

TABLA 8- OPERADOR ALFA, EJEMPLO 5.2

x	y	$\alpha(x, y)$
1	0	2
1	1	1
0	0	1
0	0	1
1	1	1
0	1	0
1	0	2

3. Aplicar el módulo 2 a cada componente del vector que obtuvimos en 1.

$$\alpha(x,y) = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 2 \end{pmatrix}, \text{ modulo } 2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

4. Aplicar el operador \mathcal{U}_β al vector obtenido en 3.

$$\mathcal{U}_\beta = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 4$$

Al restarlo de $n = 7$, se obtiene $7-4=3$; $3 \not\leq \theta$, por lo tanto: $\gamma(x, y, \theta) = 0$

3.5.2 Algoritmo del clasificador Gamma.

En el trabajo de tesis [22], se demuestra el algoritmo del clasificador Gamma al caracterizarse su operación, además de definirse el conjunto fundamental ideal con el cual el clasificador realiza una recuperación correcta.

Definición 3.2.2 [22]: *Sea el conjunto fundamental del clasificador Gamma, el conjunto de patrones asociados a una clase, de la forma $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$; donde x^μ es un patrón y y^μ es su clase correspondiente. Además, para este conjunto fundamental se cumplen las siguientes tres afirmaciones:*

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

Esto implica que no hay patrones repetidos.

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

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

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

Clases diferentes tienen asociados patrones diferentes.

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

Desde la propuesta del clasificador Gamma en [22], se han realizado varias aportaciones para mejorar su desempeño. En este trabajo de tesis, se utiliza el clasificador Gamma enriquecido y simplificado propuesto en [23] como el modelo a utilizar.

Algoritmo 3.2.2 [23]: *Sea el conjunto fundamental del clasificador Gamma de acuerdo con la definición 3.2.2. Al presentarse un patrón a clasificar \tilde{x} , donde \tilde{x} es un vector real n -dimensional $\tilde{x} \in \mathbb{R}^n$, con $n \in \mathbb{Z}^+$, se realiza lo siguiente:*

1. Codificar las componentes de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado. Se resta el menor valor a todos los componentes de cada patrón y se obtiene un elemento trasladado $e_m = \bigvee_{i=1}^p x_j^i$ por cada componente esto con la finalidad de trabajar en un rango de 0 a e_m . Así, la componente x_j^i se transforma en un vector binario de dimensión $e_m(j)$.
2. Codificar cada componente del patrón a clasificar con el código Johnson-Möbius modificado, utilizando las condiciones del paso anterior. En caso de que alguna componente del patrón a clasificar sea mayor al elemento e_m correspondiente, esto

es ($\tilde{x}_\xi > e_m(\xi)$), igualar esa componente a $e_m(\xi)$ y guardar su valor anterior en la variable $mgamma_\xi$. Por lo tanto, si alguna componente da un valor negativo una vez desplazada, igualar esa componente a 0 y asignar el valor a $e_m(\xi) + |\tilde{x}_\xi|$ a $mgamma_\xi$.

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

$$p = \left(\bigvee_{i=1}^p x_j^i \right) \text{ y } p_0 = p/4$$

4. Determinar el umbral de pausa u : $u = 0$.
5. Asignar pesos para cada dimensión con el valor de 1.
6. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice que tenía un patrón originalmente en el conjunto fundamental, por ejemplo x^μ , se convierta en dos índices: uno para la clase a la que pertenece (ejemplo clase i) y otro para el orden que le corresponde a dicho patrón dentro de esa clase (ejemplo orden ω). Bajo estas condiciones, la notación para el patrón x^μ será ahora $x^{i\omega}$, esto es el patrón x pertenece a la clase i en la posición ω . Lo anterior se realiza para todos los patrones del conjunto fundamental.
7. Iniciar θ a 0
8. Realizar la operación $\gamma(x_j^\mu, \tilde{x}_j, \theta)$ para las componentes de cada uno de los patrones del conjunto fundamental y del patrón a clasificar, considerándose $mgamma_\xi$ como la dimensión del patrón binario \tilde{x}_ξ en caso necesario.
9. Calcular la suma ponderada inicial c_i^0 de los resultados obtenidos en el paso anterior, para cada patrón fundamental $\mu = 1, 2, \dots, p$:

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

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

$$\tilde{y} = y^i \text{ tal que } \bigvee_{\mu=1}^p c_\mu^0 = c_i^0 = n.$$

En otro caso, continuar.

11. Realizar la operación $\gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)$ para cada clase y para cada componente de cada uno de los patrones del conjunto fundamental que corresponden a esa clase, y del

patrón a clasificar, considerándose $m\gamma_{\xi}$ como la dimensión del patrón binario \tilde{x}_{ξ} si es necesario.

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

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

13. Evaluar, si existe más de un máximo entre las sumas ponderadas por clase 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 $\tilde{y} = y^{\sigma}$; de lo contrario, continuar.
14. Evaluar, 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 pausa: $\theta \geq \rho$.

15. Evaluar, si se cumple con la condición de pausa $\theta = \rho_0$, se compara el valor máximo de las sumas ponderadas con el umbral de pausa.

- a. Si $\bigvee_{i=1}^m c_i \leq u$ entonces se asigna la clase desconocida al patrón a clasificar:

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

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

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

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

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

$$\tilde{y} = y^{\lambda}$$

A continuación se presentan dos diagramas de bloques. El primero describe el pre-procesamiento (pasos 1 a 7) que debe realizarse y el segundo describe el resto del algoritmo del clasificador Gamma.

FIGURA 1-PARTE 1 DEL CLASIFICADOR GAMMA: PRE-PROCESAMIENTO

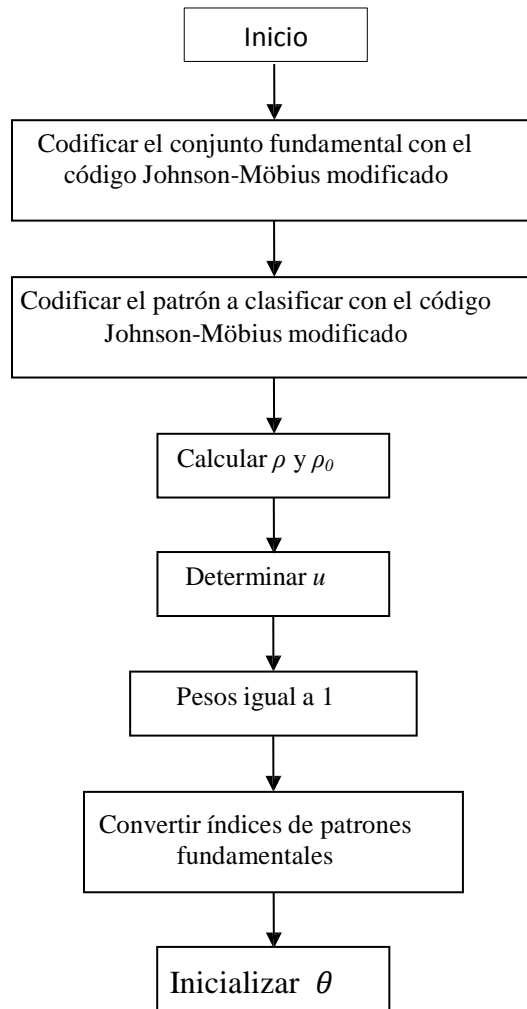
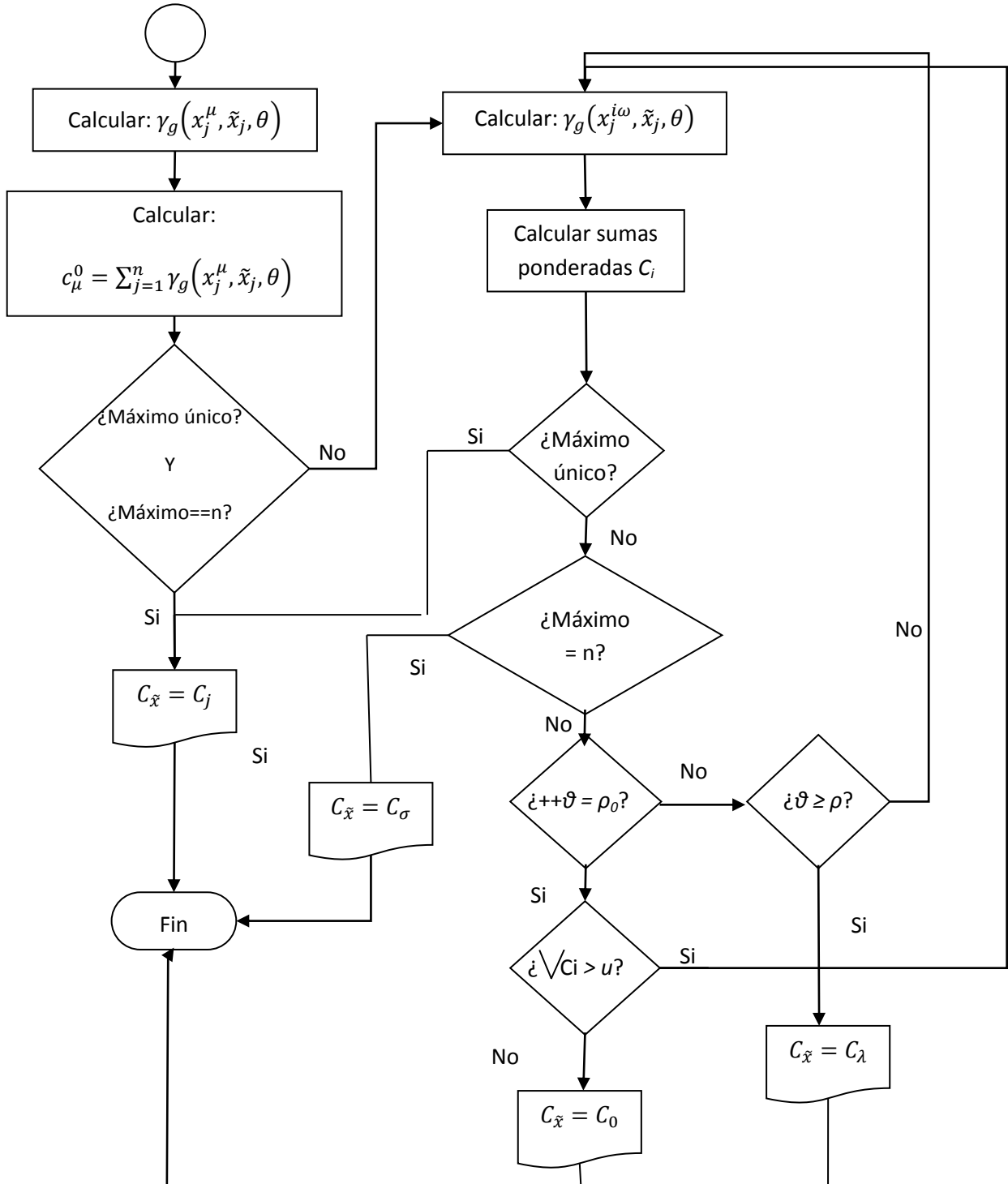


FIGURA 2- PARTE 2 DEL CLASIFICADOR GAMMA: FUNCIONAMIENTO DEL CLASIFICADOR GAMMA.



3.6 Descripción del banco de datos.

Se cuenta con tres bancos de datos, de tres vehículos con características distintas. Cada banco de datos a emplear es el resultado de un procedimiento diseñado para realizar una prueba de emisiones similar a la prueba de verificación vehicular, considerando los gases y las concentraciones que se especifican en la normativa; los datos han sido obtenidos directamente del arroyo vehicular para diferentes automóviles bajo distintos regímenes de carga, durante situaciones de tráfico diversas [59]. El banco de datos contiene 5 atributos: HC(Hidrocarburos), CO(Monóxido de Carbono) , O₂(Oxígeno) , CO₂(Dióxido de carbono), NO(Óxido Nítrico); los cuales serán utilizados para ser comparados con el estándar de la norma oficial mexicana NOM-041-SEMARNAT-2006[35], que establece los límites máximos permisibles de emisión de gases contaminantes provenientes de los escapes de vehículos automotores en circulación.

Tabla 9- Niveles máximos permisibles de emisión de gases contaminantes de los gases de escape de los vehículos que utilizan gasolina como combustible. Aplicable en la ciudad de México y su área metropolitana.

TABLA 9- NIVELES MÁXIMOS PERMISIBLES DE EMISIÓN DE GASES CONTAMINANTES.

Año-Modelo vehículo	Hidrocarburos HC (ppm)	Monóxido de Carbono CO (%VOL)	Oxígeno O ₂ (%VOL)	Óxido de Nitrógeno NO (ppm)	Dilución		Lambda λ
					Min CO + CO ₂ (%VOL)	Max	
1990 y anteriores	150	1.5	3.0	2500	13	16.5	1.1
1991 y anteriores	100	1.0	3.0	1500	13	16.5	1.05

El banco de datos contiene las muestras por segundo de entre aproximadamente 50 y 60 minutos, considerando un viaje estándar de oriente a poniente de la ciudad en condiciones estándar de tráfico para la ciudad de México.

3.7 Plataforma Android.

Android es un sistema operativo de código abierto basado en un sistema Linux, creado por Google y la Open Handset Alliance para una amplia gama de dispositivos móviles. Android se ha colocado rápidamente como el sistema operativo de móviles más utilizado del mundo, situación que se alcanzó en el último trimestre de 2010 [61].

TABLA 10- TABLA DE LOS SISTEMAS OPERATIVOS MÁS VENDIDOS EN EL 2013

Worldwide Smartphone Sales to End Users by Operating System in 3Q13 (Thousands of Units)				
Operating System	3Q13 Units	3Q13 Market Share (%)	3Q12 Units	3Q12 Market Share (%)
Android	205,022.7	81.9	124,552.3	72.6
iOS	30,330.0	12.1	24,620.3	14.3
Microsoft	8,912.3	3.6	3,993.6	2.3
BlackBerry	4,400.7	1.8	8,946.8	5.2
Bada	633.3	0.3	4,454.7	2.6
Symbian	457.5	0.2	4,401.3	2.6
Others	475.2	0.2	683.7	0.4
Total	250,231.7	100.0	171,652.7	100.0

Source: Gartner (November 2013)

3.7.1 Fundamentos de las aplicación en Android.

Las aplicaciones de Android están escritas en el lenguaje de programación Java. Las herramientas del SDK de Android compilan el código dentro de un paquete de Android, un archivo con la extensión .apk. Este archivo es el que los dispositivos utilizan para instalar la aplicación. Una vez que la aplicación ha sido instalada en el dispositivo, trabaja dentro de su propio entorno [60]:

- Cada aplicación es un usuario diferente en el sistema operativo.
- Por defecto, cada aplicación tiene un ID de usuario único asignado.
- Cada proceso tiene su propia máquina virtual (VM), por lo tanto el código de una aplicación se ejecuta independiente de otras aplicaciones.

De esta manera, el sistema Android implementa el principio de privilegios mínimos. Es decir, cada aplicación, por defecto, sólo tiene acceso a los componentes que necesita para realizar sus actividades y solamente eso. Esto genera un ambiente seguro en el que una aplicación no puede tener acceso a otras partes del sistema para el cual no cuenta con los permisos [60].

3.8 Amazon Web Services AWS.

Amazon Web Services ofrece un grupo de servicios que en conjunto forman una plataforma de computación confiable, escalable y de bajo costo "en la nube".

AWS provee bloque de construcción para soportar virtualmente cualquier tipo de carga de trabajo, se puede ajustar a las necesidades del problema y solo se paga por los recursos que se utilizan. AWS cuenta con un conjunto completo de herramientas para construir aplicaciones

escalables, se tiene acceso a almacenamiento de alta duración como *Amazon S3*, computo de bajo costo como *Amazon EC2*, bases de datos de alto rendimiento como *Amazon RDS* y las herramientas para administrar dichos recursos como *AWS Elastic Beanstalk* [62].

Con AWS se puede tener una gestión fiable de la base de datos ejecutándose en minutos a través de múltiples centros de datos geográficamente aislados e implementar aplicaciones utilizando las herramientas y los lenguajes más utilizados [62].

3.8.1 AWS EC2.

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web en la nube, que proporciona capacidad informática virtual la cual puede ser modificada y facilita a los desarrolladores recursos informáticos escalables basados en la web.

Este servicio proporciona varios tipos de instancia, sistemas operativos y paquetes de software, también permite seleccionar una configuración de memoria, CPU y almacenamiento de instancias, así como el tamaño de la partición de arranque óptimo para su sistema operativo y su aplicación.

Para hacer uso de Amazon EC2, se necesita elegir una imagen de máquina de Amazon (AMI) con ciertas características preestablecidas o crear una AMI dependiendo de las necesidades del usuario [62].

3.8.2 AWS Elastic Beanstalk.

AWS Elastic Beanstalk es un servicio el cual permite administrar e implementar aplicaciones de manera rápidamente en la nube.

Cuando una aplicación es cargada en Elastic Beanstalk, se gestionan de manera automática el equilibrio de la carga de trabajo, el auto escalamiento y el estado de la aplicación. Elastic Beanstalk utiliza los servicios de AWS, como Amazon Elastic Cloud Compute (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon Simple Notification Service (Amazon SNS), Elastic Load Balancing y Auto Scaling para ofrecer la misma infraestructura de alta fiabilidad, escalable y rentable con la que miles de empresas trabajan hoy en día [62].

Capítulo 4

Solución propuesta

En este capítulo se presenta el modelo propuesto para realizar la predicción de los niveles de emisión de contaminantes provenientes de un vehículo automotor y el desarrollo del sistema informático que permite la obtención de los datos de los niveles de emisión de contaminantes provenientes de un vehículo automotor y muestra los resultados de la predicción realizada por el clasificador Gamma.

4.1 Modificación del banco de datos.

Tal como se mencionó en la sección 3.6, se cuenta con un banco de datos de emisión de gases contaminantes provenientes de los escapes de vehículos automotores en circulación, el cual debe ser modificado para realizar el procedimiento de predicción que se busca.

TABLA 11-ESTRUCTURA DEL BANCO DE DATOS.

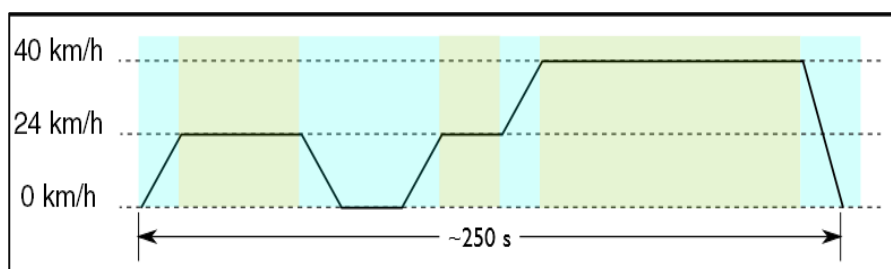
N° Muestra	CO	CO ₂	NO	HC	O ₂
1	0.011378	12.85784	11.44	2.552	10.1222
2	0.00776	14.10008	5.864	10.048	2.9228
3	0.011304	13.78372	5.748	8.812	3
4	0.011304	13.78392	5.748	8.788	3
5	0.012556	13.76444	20.02	5.26	3
⋮	⋮	⋮	⋮	⋮	⋮
3000	0.001356	14.10872	14.072	3.592	2.81216

En primera instancia se tiene que el banco de datos original contiene muestras por segundo de los gases analizados, como se puede ver en la tabla 11, cada muestra está compuesta de los valores pertenecientes a cada uno de los 5 gases; pero con la finalidad de utilizar el banco de datos en la predicción, es necesario generar la cantidad de n número de patrones y de las clases a las que serán distribuidos taxonómicamente.

4.1.1 Descripción de los patrones.

Los procedimientos de medición para la verificación de los límites de emisión de contaminantes provenientes de los vehículos automotores en circulación especificados en la norma oficial mexicana NOM-047-SEMARNAT-1999[36], nos dicen que los métodos de prueba de dinamómetro para medir las emisiones vehiculares tienen una duración de 250 segundos.

FIGURA 3- PERFIL DE PRUEBA DE VERIFICACIÓN EN CONDICIONES CONTROLADAS



Los resultados de estas pruebas son comparados con los niveles máximos permisibles de emisión de gases contaminantes en NOM-041-SEMARNAT-2006[35]; por lo tanto, obteniendo el promedio por cada gas analizado de un conjunto de 250 muestras del banco de datos anteriormente descrito, se puede generar un patrón comparable con la norma.

FIGURA 4- GENERACIÓN DE PATRONES

Nº Muestra	CO	CO ₂	NO	HC	O ₂
1	0.011378	12.85784	11.44	2.552	10.1222
2	0.00776	14.10008	5.864	10.048	2.9228
3	0.011304	13.78372	5.748	8.812	3
4	0.011304	13.78392	5.748	8.788	3
5	0.012556	13.76444	20.02	5.26	3
⋮	⋮	⋮	⋮	⋮	⋮
250	0.001309	14.10098	5.934	8.599	2.81216

Nº Patrón	CO-AVG	CO ₂ -AVG	NO_AVG	HC_AVG	O ₂ _AVG
1	4.016872	11.37196	273.492	446.524	7.22344

4.1.2 Descripción de las clases.

Para el proceso de predicción que se presenta en esta tesis, es importante determinar si cada uno de los patrones generados está entre o sobre los niveles permisibles de emisión de gases contaminantes presentes en la norma NOM-041-SEMARNAT-2006[35]; por lo tanto, de acuerdo con el criterio anterior, solo se generaron 2 clases.

TABLA 12-CLASES A EMPLEAR EN EL PROCESO DE PREDICCIÓN

Clases	Descripción
1	Patrones sobre los límites permisibles
2	Patrones entre los límites permisibles

4.2 Valores iniciales para el clasificador Gamma

El clasificador Gamma incluye ciertas condiciones iniciales para su correcto funcionamiento, como lo son las condiciones de paro: parámetro de paro y pausa, la asignación de pesos a cada uno de los rasgos y el umbral de pausa u .

4.2.1 Condiciones de paro.

El parámetro de paro ρ garantiza la convergencia del algoritmo; esto quiere decir, que después de un número determinado 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 ρ_0 permite detener el proceso de clasificación y asignar el patrón a clasificar a la clase desconocida, cuando no existe suficiente similitud con patrones conocidos.

Los valores propuestos para el parámetro de paro ρ en el trabajo de tesis [23], sugieren el uso del mínimo, máximo o promedio de los máximos de cada rasgo. En este caso en particular, el comportamiento de los patrones dicta que siempre existe un máximo entre las sumas ponderadas por clase, por lo tanto el valor de ρ y ρ_0 utilizados en este trabajo son:

$$\rho = \left[\bigwedge_{j=1}^n \left(\bigvee_{i=1}^p x_j^i \right) \right]$$
$$\rho_0 = \rho/4$$

4.2.2 Umbral de pausa.

Considerando que el valor del umbral depende fuertemente de las características del problema y las propiedades del conjunto fundamental, en el trabajo de tesis [23] se proponen 2 sugerencias para valores iniciales, En este trabajo se ha seleccionado que el valor inicial de u sea 0.

4.2.3 Grado de similitud.

En el trabajo de tesis [23], se establece que el grado de similitud θ debe ser igual a cero e ir aumentando su valor en uno en cada iteración, hasta que el patrón que se está procesando sea clasificado. Después de varias pruebas realizadas con el banco de datos, se observó que con $\theta = 0$, el proceso de similitud entre los patrones está entre 0% y 20%, esto quiere decir que de los 5 rasgos del banco de datos, no se parecen en ninguno o solo en un rasgo. Estos resultados reflejan el bajo desempeño del algoritmo tomando como valor inicial de θ el propuesto anteriormente.

Si lo que se necesita es tener un mayor grado de similitud, el valor de θ debe aumentar para tener una tolerancia considerable entre los patrones. Lo anteriormente planteado dio como resultado que existen diversos valores iniciales idóneos para θ ; exclusivamente en el presente trabajo, el valor inicial para θ utilizado en pruebas posteriores es el siguiente:

$$\theta = 5$$

4.2.4 Asignación de pesos.

En el trabajo de tesis [23], para asignar los pesos a cada uno de los rasgos pertenecientes al banco de datos, se sugieren diferentes rangos como valores iniciales empíricos; para el proceso de predicción, exclusivamente en este trabajo, el rango a utilizar es el siguiente:

$$\text{rango de [1.5 ,2]}$$

De acuerdo con los experimentos realizados, se obtuvo que los pesos con los cuales el clasificador Gamma presenta un excelente desempeño son los siguientes:

TABLA 13-PESOS ASIGNADOS A CADA RASGO

Rasgo	Peso asignado
CO	1.6
CO ₂	1.8
NO	1.5
HC	1.5
O ₂	2

4.3 Criterio de validación del clasificador.

Para estimar la probabilidad de que el modelo de predicción, al presentarle un nuevo caso, sea capaz de clasificarlo correctamente, es necesario utilizar un método de validación. En el presente trabajo se ha seleccionado el método *hold- out*, el cual divide el banco de datos en dos subconjuntos: el conjunto fundamental, que está formado por las 2/3 partes del conjunto original y el conjunto de prueba, que contempla el 1/3 restante del conjunto.

Ejemplo 4.3: Dado un conjunto de 500 patrones, calcular el tamaño del conjunto fundamental y el conjunto de prueba.

Conjunto Original: 500 patrones

Conjunto Fundamental= 2/3 del conjunto original = 67% del conjunto= 335 patrones

Conjunto de Prueba= 1/3 del conjunto original=33% del conjunto=165 patrones

4.4 Sistema de verificación.

Con el propósito de contar con un entorno, en el cual se pueda enviar información y solicitar una respuesta en tiempo real para realizar la predicción de contaminantes atmosféricos, se ha desarrollado una aplicación móvil la cual interactúa con ciertos servicios web que se encuentran alojados en la nube, los cuales contienen la lógica de negocio y los procesos que se deben realizar con los datos enviados desde el cliente; con lo anterior se logra extender las capacidades del dispositivo móvil para realizar el cómputo intensivo y el almacenar grandes cantidades de datos.

Otra de las características por las que se hace uso del cómputo en la nube es debido a que se cuenta con herramientas que permiten que la infraestructura pueda aumentar y disminuir automáticamente como sea requerido, sin tener que adecuar el sistema que reside en esa infraestructura.

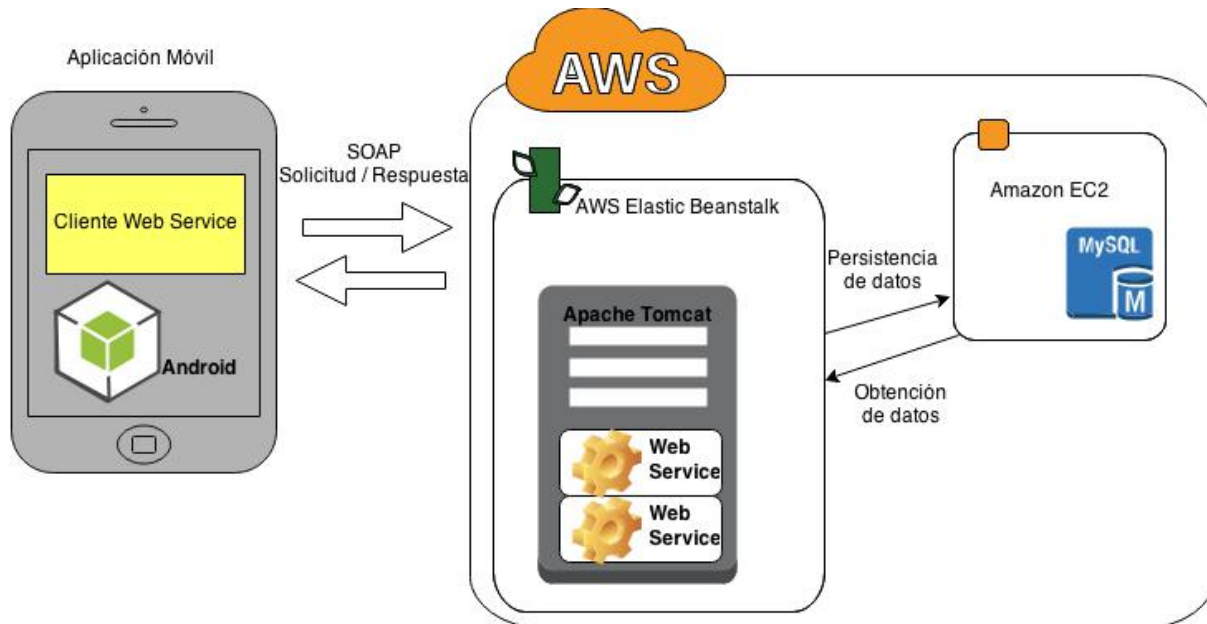
Los servicios web hacen posible que diferentes aplicaciones se comuniquen entre si y compartan servicios entre ellas sin importar la plataforma que se utilice, y mejoran el flujo de la información entre las aplicaciones haciendo que su información y su lógica de negocios este a disposición de otras aplicaciones a través de una interfaz de programación de aplicaciones (API).

4.4.1 Arquitectura general de la aplicación

La aplicación está dividida en tres partes esenciales: la aplicación móvil (cliente), los servicios web que contienen la lógica del sistema y se encuentran alojados en la nube, y la base de datos que de igual manera se encuentra en la nube a través de la plataforma AWS (Amazon Web Services). La aplicación móvil está desarrollada para el sistema operativo Android, también se

cuenta con un entorno que incluye las características necesarias para contener los servicios web a través de AWS Elastic Beanstalk y se utiliza el servicio de Amazon EC2 para crear una microinstancia virtual con MySQL para soportar la base de datos.

FIGURA 5- ARQUITECTURA DEL SISTEMA



Cuando hablamos de un “entorno”, nos referimos a una versión del sistema que se implementa en los recursos de AWS. Cada entorno ejecuta solamente una versión, no obstante, puede ejecutar la misma versión o diferentes versiones en muchos ambientes al mismo tiempo. Cuando un entorno es creado, AWS Elastic Beanstalk proporciona los recursos necesarios para ejecutar la versión del sistema que se requiera.

Una vez iniciado el entorno en AWS Elastic Beanstalk, se selecciona una plataforma; para el sistema se ha seleccionado un servidor web Apache Tomcat, el cual contendrá los servicios web utilizados por la aplicación.

4.4.2 Servicios web

Se han diseñado 2 servicios web basados en el protocolo *SOAP* mediante el entorno de desarrollo para Java *Eclipse*, cada uno es componentes de software independientes del sistema de verificación, he interactúan con la aplicación móvil y la base de datos para presentar información dinámica al usuario. Cada uno de los servicios web contiene un conjunto de funciones concretas de las cuales es responsable.

Como el componente que realiza una petición a nuestro servicio web puede estar implementado con cualquier tecnología, el proveedor del servicio pone a disposición un archivo XML con extensión *wsdl* en el que describa los detalles del servicio que provee. *WSDL* (Lenguaje de Descripción de Servicios Web) representa una especie de contrato entre el proveedor y el que solicita, además define la sintaxis y los mecanismos de intercambio de mensajes. Los archivos *wsdl* de los servicios han sido creados a través de una clase que contiene la lógica y métodos que deseamos para nuestro servicio web con ayuda del motor de SOAP Apache Axis2.

Como se mencionó anteriormente, cada uno de los servicios web tiene funciones específicas. A continuación se presenta cada servicio con sus respectivas funciones y se describe el comportamiento de cada una de ellas:

1- **wsSavePatterns.**

Este servicio es el encargado de almacenar en la base de datos los patrones o muestras que se reciben, las cuales serán analizadas posteriormente para la realizar la predicción.

TABLA 14- FUNCIONES WSSAVEPATTERNS

Nombre	Descripción
insertPattern	-- Realiza una llamada a la función ConectTo e inserta en la base de datos cada uno de los patrones obtenidos mediante gatherData
ConectTo	--Realiza la conexión con la base de datos
gatherData	--Recibe lo patrones enviados desde el componente externo que realiza la petición y realiza una invocación a la función insertPattern

De las funciones anteriores, *gatherData* es la función disponible o visible, por la cual es llamado al servicio web, y es la encargada de invocar a los métodos que contienen la lógica del servicio.

2- **wsPredictionProcess**

Es el servicio web principal del sistema debido a que engloba todo el proceso del algoritmo de clasificación empleado en este trabajo de tesis, el clasificador Gamma. Se encarga de realizar la predicción de los contaminantes atmosféricos que se encuentran en forma de patrones dentro de la base de datos, y de retornar los resultados de dicha predicción a quien lo invoca.

TABLA 15- FUNCIONES WSPREDICTIONPROCESS

Nombre	Descripción
predictionClass	--Recibe la clase correspondiente al patrón actual y almacena el resultado en la base de datos.
generateClasses	--Realiza el proceso de creación de las clases a las que pertenecerán los patrones con base en los valores de sus atributos.

gammaSimilitudProcess	-- Ejecuta la funciones del operador gama de similitud (γ) e invoca al función weightSums
parametersPauseStopUmb	-- Calcula el parámetro de paro ρ , el parámetro de pausa ρ_0 y establece el valor del umbral
predictionClassMaxNoUnique	--Encuentra la clase correspondiente al patrón actual y almacena el resultado en la base de datos.
johnsonMCoding	-- Realiza la codificación de las componentes de cada patrón dentro de la base de datos que pertenecen al conjunto fundamental y al conjunto de prueba con el código Johnson-Möebius modificado
generateDataPrediction	--Obtiene los patrones almacenados en la base de datos del conjunto fundamental y de prueba para realizar el pre- procesamiento de los mismos mediante la invocación de la función johnsonMCoding
weightSums	-- Calcula la suma ponderada inicial o la suma ponderada por cada clase, dependiendo de las condiciones estipuladas por el proceso.

4.4.3 Aplicación móvil

La aplicación móvil diseñada para el sistema de verificación, como se mencionó anteriormente, ha sido desarrollada para el sistema operativo Android. Se eligió trabajar sobre este sistema operativo porque es gratuito, es distribuido bajo licencia de software libre, y en los últimos años ha ganado una enorme popularidad que va aumentando con el paso del tiempo. La aplicación es soportada en todos los dispositivos que tengan una versión igual o mayor a la 2.3 *Gingerbread*.

Se tienen 3 módulos principales, los cuales son 1) ingreso de los datos que serán procesados, 2) el proceso de predicción, 3) mostrar y descargar resultados. Cada uno de estos módulos es mostrado a continuación en la siguiente sección

4.4.3.1 Interfaz de usuario

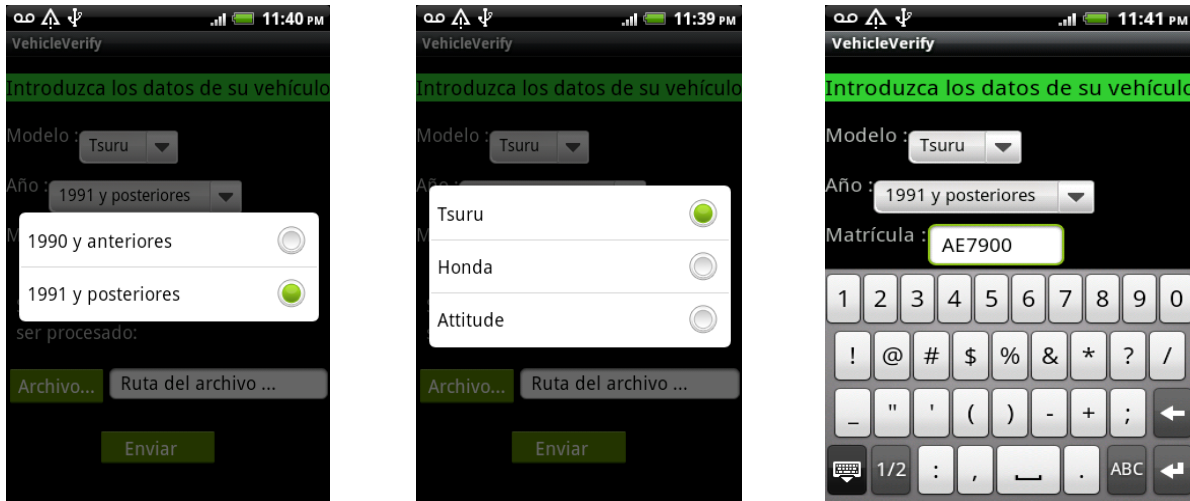
Al iniciar la aplicación, el usuario visualizará la siguiente pantalla:

FIGURA 6- PANTALLA DE CAPTURA DE DATOS



En esta pantalla se solicitan al usuario los datos del vehículo, y se selecciona el archivo el cual contiene los datos de cada contaminante provenientes del escape del vehículo, por medio de los cuales se realizará la predicción

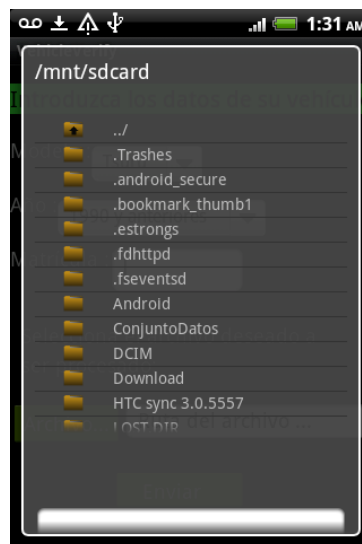
FIGURA 7- ESTRUCTURA DE LA PANTALLA DE CAPTURA



Se cuenta con 3 diferentes opciones de modelo del vehículo, debido a que los bancos de datos utilizados en este trabajo de tesis provienen de dichos modelos; se tiene también 2 opciones para el año del vehículo, solo se cuentan con dichas opciones porque de acuerdo con la información sobre los límites permisibles de cada uno de los gases que se analizan en la norma oficial mexicana; si un vehículo es de un año posterior o igual a 1991, los límites permitidos son diferentes a los de un vehículo del año 1990 o anterior.

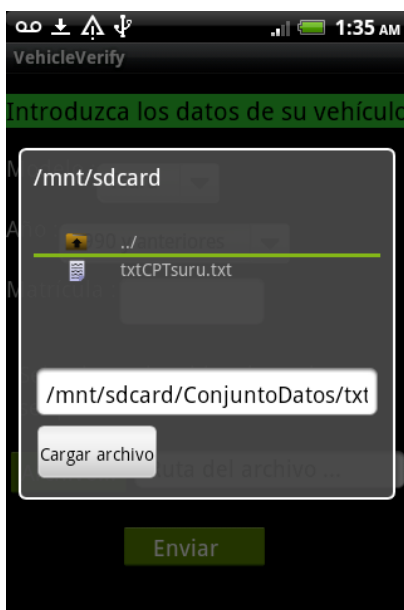
Para seleccionar el archivo con el cual se realizará la prueba, el usuario deberá dar clic en el botón de "Archivo...", el cual abrirá un explorador de archivos como se muestra en la siguiente figura.

FIGURA 8-EXPLORADOR DE ARCHIVOS



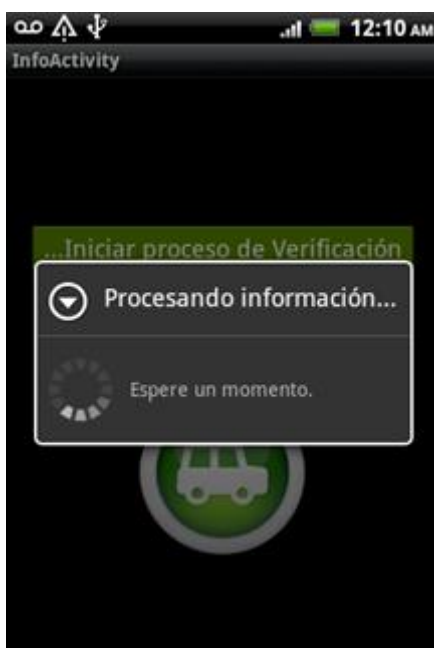
Después de seleccionar el archivo de prueba, el usuario deberá seleccionar el botón “Cargar archivo”, y automáticamente se cerrará el explorador de archivos.

FIGURA 9- CARGAR ARCHIVO



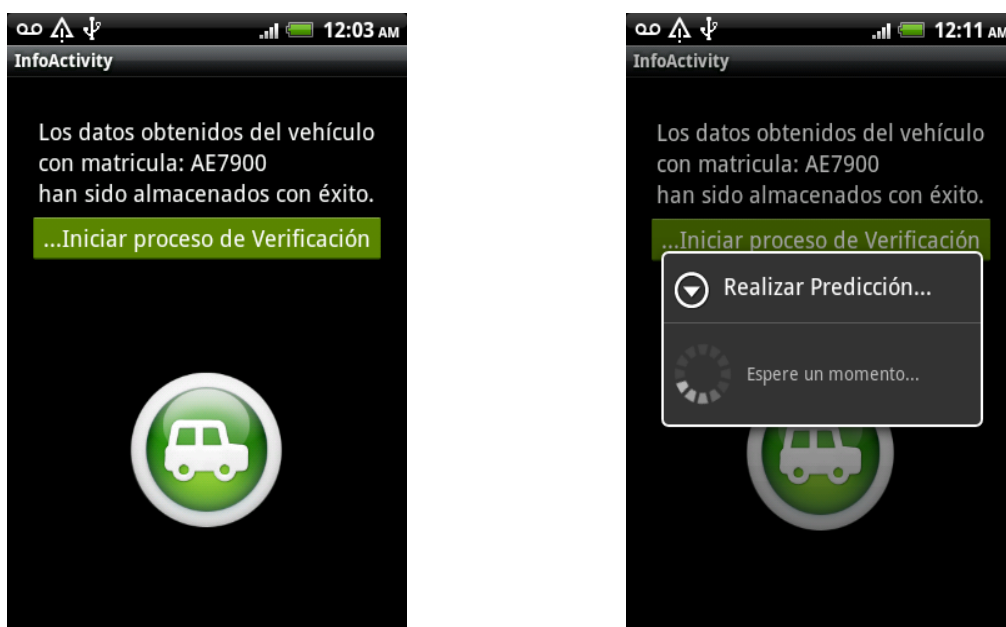
El proceso necesita almacenar los datos del archivo de prueba, por lo que al presionar el botón “Enviar”, se realizan las tareas necesarias para persistir la información en la base de datos y se realiza un pre-procesamiento a dicha información para ser utilizada posteriormente para el proceso de predicción.

FIGURA 10- PROCESAMIENTO DE LA INFORMACIÓN



Una vez realizado lo anterior, se procederá a iniciar con el proceso de predicción seleccionando el botón “... Iniciar proceso de Verificación”.

FIGURA 11- PANTALLAS DE INICIO DE PROCESO DE VERIFICACIÓN (PREDICCIÓN)



Cuando el proceso de predicción termine se desplegará la siguiente pantalla con el resultado obtenido para el vehículo en cuestión.

FIGURA 12-PANTALLA DE RESULTADOS



Esta figura indica si el vehículo cumple con lo estipulado en la norma mencionada; no obstante si el usuario desea conocer a detalle los resultados obtenidos, puede descargar los

resultados en un documento PDF seleccionando la opción del menú desplegable “*Resultados PDF*”, el cual se muestra a continuación.

FIGURA 13- PANTALLA DE RESULTADOS PDF

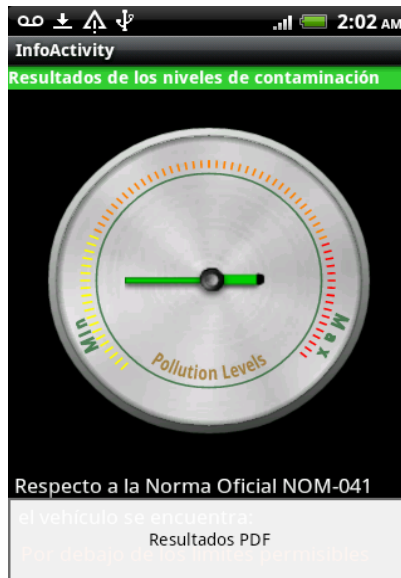


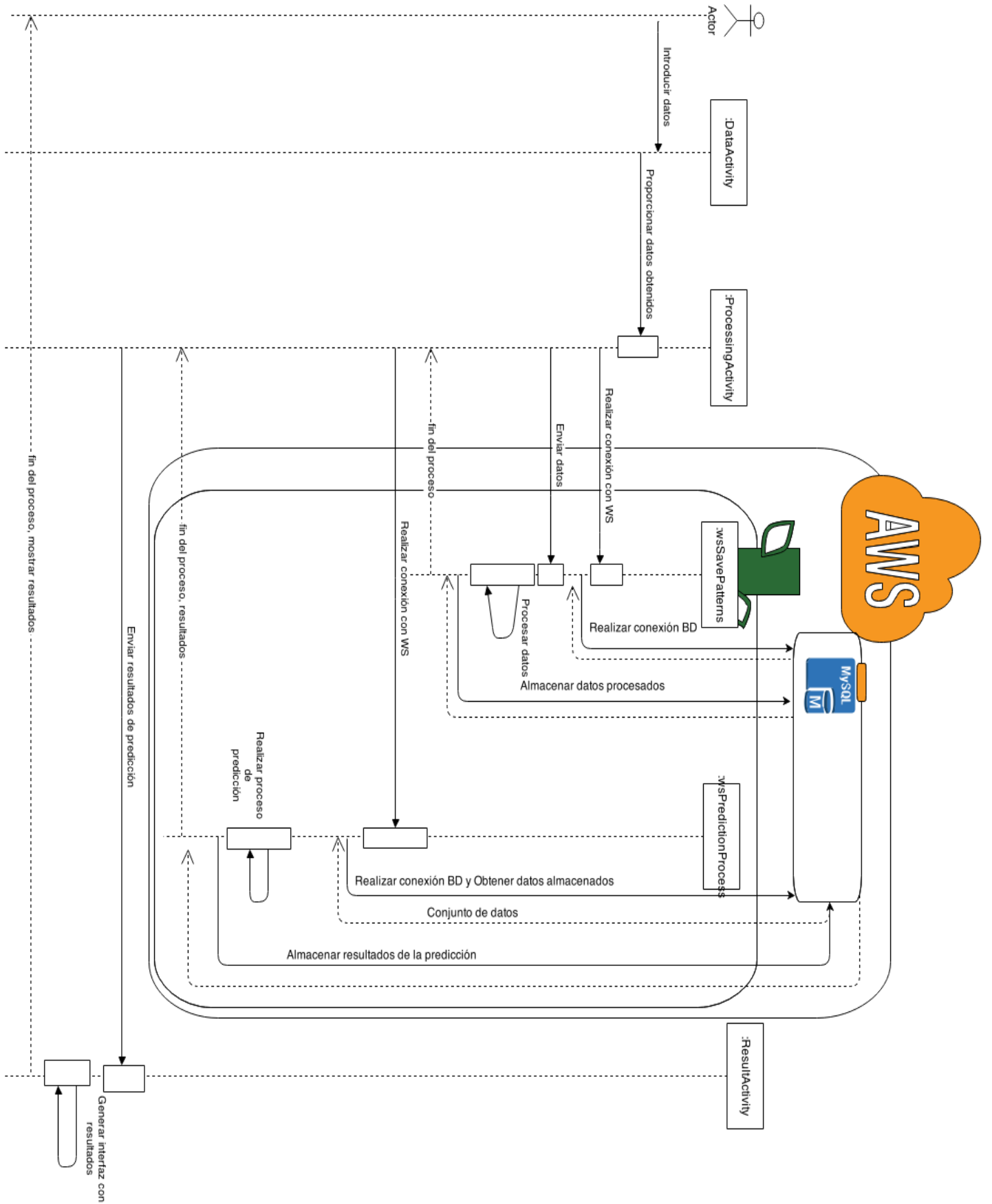
FIGURA 14-RESULTADOS PDF

Modelo-Año del vehículo	Límites máximos permisibles				
	Hidrocarburos (HC) ppm*	Monóxido de carbono (CO) % Vol	Oxígeno (O2) % Vol	Oxido de Nitrógeno (NOX) ppm*	Dilución (CO +CO2) % Vol
	Max	Max	Max	Max	Max
Vehículos 1991 y posteriores	100	1	3	1500	16.5
Tsuru - 1991	152	2.6	3	1540	12.7

4.4.4 Interacción de los componentes del sistema.

Como se han definido previamente los componentes que interviene en el sistema, lo siguiente es mostrar cómo interactúan en conjunto. A continuación se muestra el diagrama de secuencia entre los componentes del sistema.

FIGURA 15-DIAGRAMA DE SECUENCIA DE LOS COMPONENTES DEL SISTEMA



Capítulo 5

Resultados y discusión

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

Para cada experimento realizado en esta sección, se emplea un banco de datos de emisión de gases contaminantes provenientes de los escapes de vehículos automotores en circulación con la estructura mencionada anteriormente en el capítulo 4, el cual pertenece a un determinado vehículo con ciertas características (dígase del año y modelo del vehículo).

Cada banco de datos a emplear es el resultado de un procedimiento diseñado para obtener valores en tiempo real de la variación de las emisiones contaminantes en un ciclo real de trabajo.

La operación de cada vehículo en condiciones normales de operación está sujeta a la variabilidad incontrolable del medio ambiente y de las diversas condiciones del tráfico; por lo tanto, para estimar correctamente las emisiones vehiculares que resultan de las pruebas basadas en este procedimiento se consideraron rutas y ambientes similares.

5.1 Prueba 1

La primera prueba contempla un total de 34 patrones correspondientes a un vehículo modelo *Honda CR-V*, año 2012. El total de patrones ha sido dividido para conformar el conjunto fundamental y el conjunto de prueba de la siguiente manera:

1. Conjunto fundamental: 21 patrones, de los cuales 5 pertenecen a la clase C1 y 16 a la clase C2, aproximadamente el 60% del total de patrones.
2. Conjunto de prueba: 13 patrones, de los cuales 10 pertenecen a la clase C1 y 3 a la clase C2, aproximadamente el 40% del total de patrones.

TABLA 16- PATRONES POR CADA CLASE

Clase	Descripción	Patrones
C1	Fuera de los límites permitidos	15
C2	Dentro de los límites permisibles	19

La predicción realizada mediante el clasificador Gamma, de manera gráfica, se muestra en las siguientes figuras:

FIGURA 16- VALORES REALES

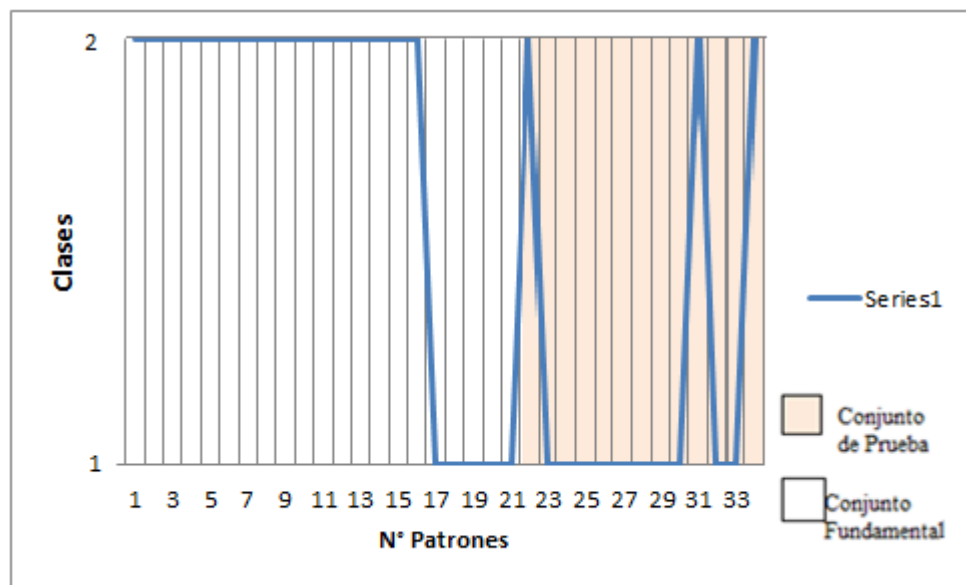
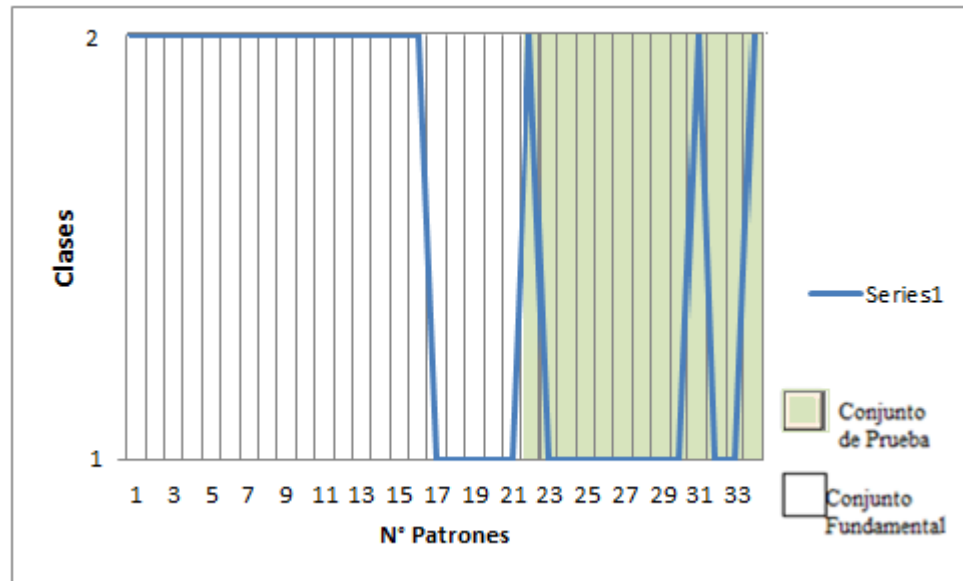


FIGURA 17-VALORES PREDICHOS POR EL CLASIFICADOR GAMMA



Como se puede observar en las figuras anteriores, el comportamiento de los valores predichos por el clasificador Gamma es exactamente igual al comportamiento de los valores reales; por lo tanto, se han clasificado correctamente el 100% de los patrones a clasificar.

Una de las formas de conocer el rendimiento de predicción con cualquiera de los algoritmos de clasificación empleados, es calculando la raíz del error medio cuadrático (RMSE, Root Mean Square Error).

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

TABLA 17- COMPARACIÓN DEL RENDIMIENTO ENTRE EL CLASIFICADOR GAMMA Y LOS ALGORITMOS DEL ENTORNO DE ANÁLISIS WEKA PARA LA PRUEBA DE PREDICCIÓN 1

Clasificador	Patrones correctamente predichos (%)	Patrones incorrectamente predichos por (%)	Rendimiento de la predicción (RMSE)
Gamma	100%	0%	0
BFTree	76.9231 %	23.769 %	0.4530
Decision Table	76.9231 %	23.769 %	0.4530
RBFNetwork	69.2308 %	30.7692%	0.4547

Multilayer perceptron	46.1538 %	53.8462%	0.6591
-----------------------	-----------	----------	--------

Al realizar el proceso de predicción y comparar los resultados obtenidos con la norma oficial mexicana NOM-041-SEMARNAT-2006, se tiene la siguiente tabla, la cual es generada a partir del sistema propuesto en esta tesis.

TABLA 18-RESULTADOS DEL VEHÍCULO RESPECTO A LA NOM-041-SEMARNAT-2006

Modelo-Año del vehículo	Límites máximos permisibles				
	Hidrocarburos (HC) ppm*	Monóxido de carbono (CO) % Vol	Oxígeno (O2) % Vol	Oxido de Nitrógeno (NOX) ppm*	Dilución (CO +CO2) % Vol
	Max	Max	Max	Max	Max
Vehículos 1991 y posteriores	100	1	3	1500	16.5
HondaCRV 2012	3.86	0.002	9.2	11.83	13.212

5.2 Prueba 2

La segunda prueba contempla un total de 24 patrones correspondientes a un vehículo modelo *Attitude*, año 2010. El total de patrones ha sido dividido para conformar el conjunto fundamental y el conjunto de prueba de la siguiente manera:

1. Conjunto fundamental: 14 patrones, de los cuales 12 pertenecen a la clase C1 y 2 a la clase C2, aproximadamente el 60% del total de patrones.
2. Conjunto de prueba: 10 patrones, de los cuales 7 pertenecen a la clase C1 y 3 a la clase C2, aproximadamente el 40% del total de patrones.

TABLA 19-PATRONES POR CADA CLASE

Clase	Descripción	Patrones
C1	Fuera de los límites permitidos	19
C2	Dentro de los límites permisibles	5

La predicción realizada mediante el clasificador Gamma, de manera gráfica, se muestra en las siguientes figuras:

FIGURA 18- VALORES REALES

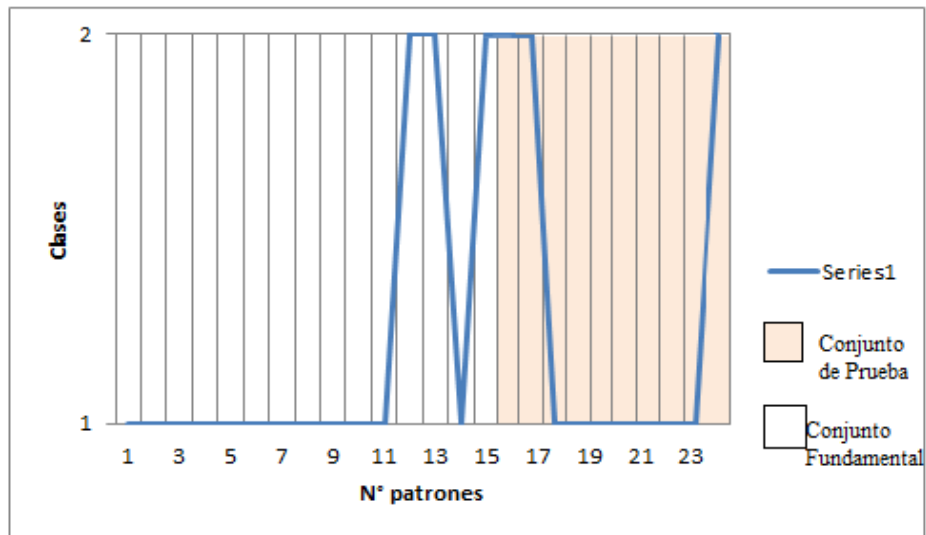
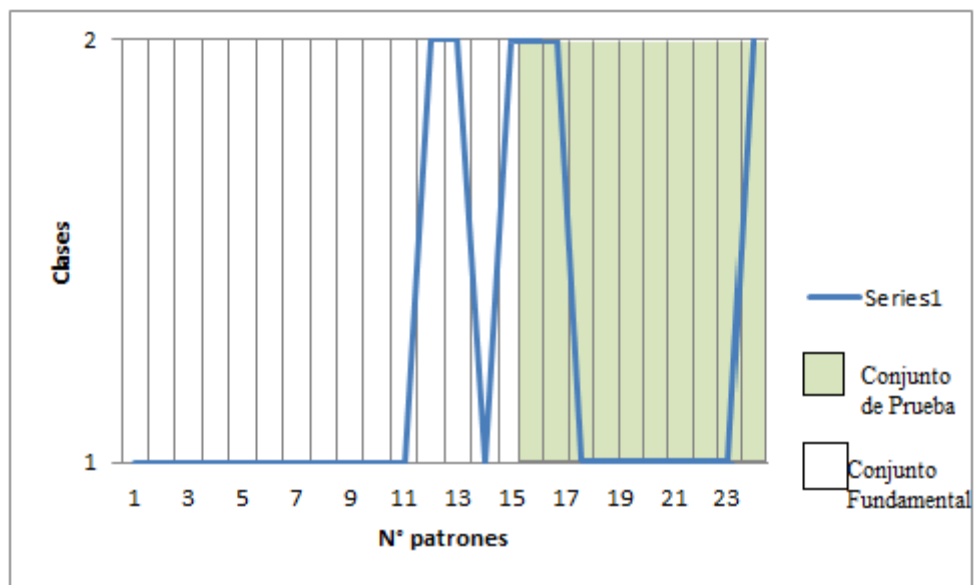


FIGURA 19-VALORES PREDICHOS POR EL CLASIFICADOR GAMMA



Como se puede observar en las figuras anteriores, el comportamiento de los valores predichos por el clasificador Gamma es exactamente igual al comportamiento de los valores reales; por lo tanto, se han clasificado correctamente el 100% de los patrones a clasificar.

La siguiente tabla muestra a detalle los resultados obtenidos para esta prueba, tanto del clasificador Gamma como de los algoritmos de clasificación del entorno de análisis *Weka* empleados previamente en la prueba 1.

TABLA 20- COMPARACIÓN DEL RENDIMIENTO ENTRE EL CLASIFICADOR GAMMA Y LOS ALGORITMOS DEL ENTORNO DE ANÁLISIS WEKA PARA LA PRUEBA DE PREDICCIÓN 2

Clasificador	Patrones correctamente predichos (%)	Patrones incorrectamente predichos por (%)	Rendimiento de la predicción (RMSE)
Gamma	100%	0%	0
Decision Table	90.47 %	9.53 %	0.31
BFTree	90.47 %	9.53 %	0.29
Multilayer perceptron	80.95 %	19.04%	0.4364
RBFNetwork	80.95 %	19.04%	0.3806

Al realizar el proceso de predicción y comparar los resultados obtenidos con la norma oficial mexicana NOM-041-SEMARNAT-2006, se tiene la siguiente tabla, la cual es generada a partir del sistema propuesto en esta tesis.

TABLA 21-RESULTADOS DEL VEHÍCULO RESPECTO A LA NOM-041-SEMARNAT-2006

Modelo-Año del vehículo	Límites máximos permisibles				
	Hidrocarburos (HC) ppm*	Monóxido de carbono (CO) % Vol	Oxígeno (O2) % Vol	Oxido de Nitrógeno (NOX) ppm*	Dilución (CO +CO2) % Vol
	Max	Max	Max	Max	Max
Vehículos 1991 y posteriores	100	1	3	1500	16.5
Attitude 2010	3.86	0.032	4.5	29.17	11.86

5.3 Prueba 3

La tercera prueba contempla un total de 63 patrones correspondientes a un vehículo modelo *Tsuru*, año 1991. El total de patrones ha sido dividido para conformar el conjunto fundamental y el conjunto de prueba de la siguiente manera:

1. Conjunto fundamental: 44 patrones, de los cuales 35 pertenecen a la clase C1 y 7 a la clase C2, aproximadamente el 60% del total de patrones.

2. Conjunto de prueba: 21 patrones, de los cuales 19 pertenecen a la clase C1 y 2 a la clase C2, aproximadamente el 40% del total de patrones.

TABLA 22-PATRONES POR CADA CLASE

Clase	Descripción	Patrones
C1	Fuera de los límites permitidos	54
C2	Dentro de los límites permisibles	9

La predicción realizada mediante el clasificador Gamma, de manera gráfica, se muestra en las siguientes figuras:

FIGURA 20- VALORES REALES

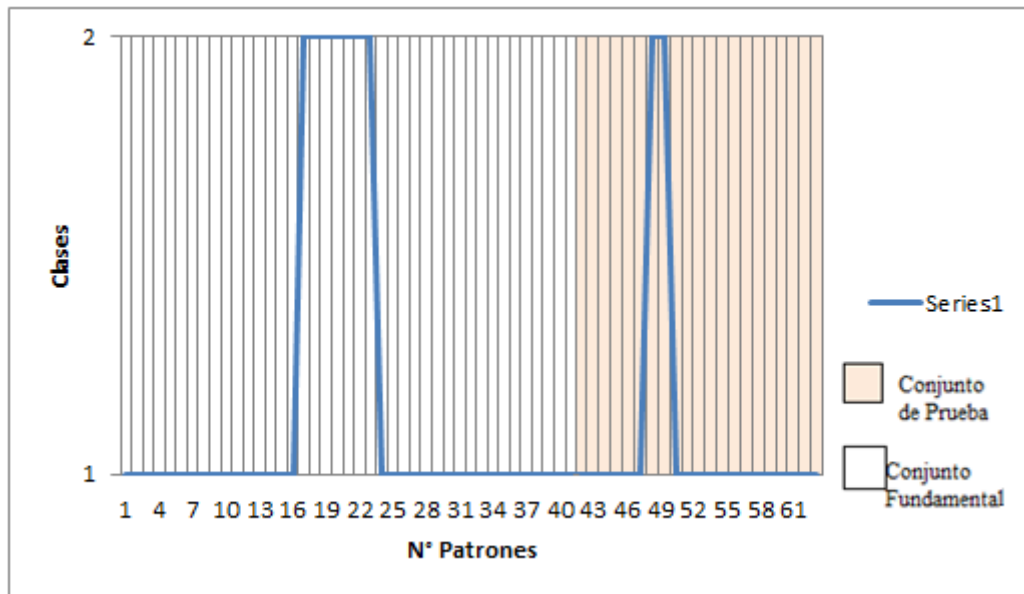
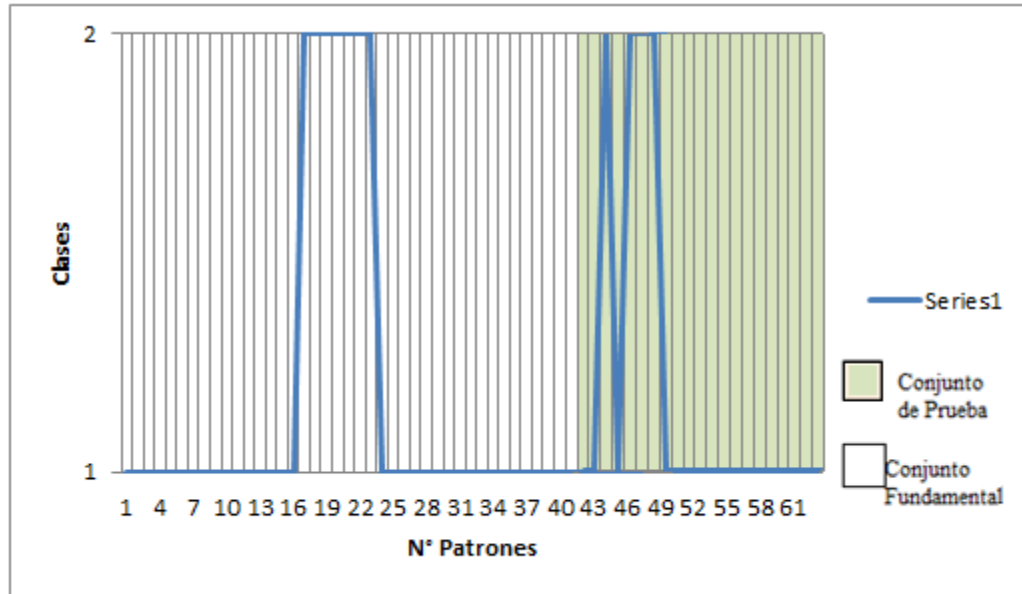


FIGURA 21-VALORES PREDICHOS POR EL CLASIFICADOR GAMMA



Como se puede observar en las figuras anteriores, el comportamiento de los valores predichos por el clasificador Gamma es ligeramente diferente a los valores reales, se han clasificado correctamente solo el 90.47% de los patrones a clasificar.

La siguiente tabla muestra a detalle los resultados obtenidos para esta prueba, tanto del clasificador Gamma como de algunos de los algoritmos de clasificación del entorno de análisis *Weka*.

TABLA 23- COMPARACIÓN DEL RENDIMIENTO ENTRE EL CLASIFICADOR GAMMA Y LOS ALGORITMOS DEL ENTORNO DE ANÁLISIS WEKA PARA LA PRUEBA DE PREDICCIÓN 3

Clasificador	Patrones correctamente predichos (%)	Patrones incorrectamente predichos por (%)	Rendimiento de la predicción (RMSE)
Gamma	90.47%	9.53%	0.31
BFTree	90.47%	9.53%	0.31
Decision Table	90.47%	9.53%	0.29
RBFNetwork	80.95 %	19.05%	0.4364
Multilayer perceptron	80.95 %	19.05%	0.3806

Al realizar el proceso de predicción y comparar los resultados obtenidos con la norma oficial mexicana NOM-041-SEMARNAT-2006, se tiene la siguiente tabla para este vehículo, la cual es generada a partir del sistema propuesto en esta tesis.

TABLA 24-RESULTADOS DEL VEHÍCULO RESPECTO A LA NOM-041-SEMARNAT-2006

Modelo-Año del vehículo	Límites máximos permisibles				
	Hidrocarburos (HC) ppm*	Monóxido de carbono (CO) % Vol	Oxígeno (O2) % Vol	Oxido de Nitrógeno (NOX) ppm*	Dilución (CO +CO2) % Vol
	Max	Max	Max	Max	Max
Vehículos 1991 y posteriores	100	1	3	1500	16.5
Tsuru - 1991	152	2.6	3	1540	12.7

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo son presentadas las conclusiones obtenidas derivadas del trabajo realizado. Posteriormente se proponen algunos trabajos a futuro que podrían realizarse con el fin de continuar y mejorar lo propuesto en este documento.

6.1 Conclusiones

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

- Se analizó a conciencia la estructura del clasificador Gamma con la finalidad de obtener los mejores resultados mediante su funcionamiento. Se detectó, al menos en el desarrollo de trabajo de esta tesis, que es necesario adaptar el algoritmo con base en la información del banco de datos a emplear, esto es, algunos valores iniciales deben ser considerados para obtener resultados satisfactorios.
- Se generó una aplicación móvil para el sistema operativo Android, la cual cumple con el propósito de ser la interfaz gráfica entre el usuario y los componentes que realizan el procesamiento de predicción de contaminantes atmosféricos provenientes del escape de un vehículo automotor a través de la implementación del clasificador Gamma.
- Se realizó la implementación del algoritmo del clasificador Gamma mediante el desarrollo de un componente de software que reside en la nube.
- Derivado del proceso de predicción, se generó una base de datos, la cual también se encuentra alojada en la nube, en la cual se almacenaron los resultados obtenidos, los cuales son comprados con la norma oficial mexicana NOM-041-SEMARNAT-2006.
- Al evaluar el desempeño del clasificador Gamma, se observaron desempeños más altos que los obtenidos por los algoritmos de clasificación contenidos en el entorno

Weka con los que se le comparó. El promedio del rendimiento de todas las pruebas realizadas está por encima del 90 % de predicción.

Se cumple el objetivo general ya que se diseñó, desarrolló, implementó un sistema informático, el cual utiliza el clasificador Gamma para la predicción de contaminantes atmosféricos.

En general el clasificador Gamma, y de acuerdo con los resultados del capítulo anterior, el clasificador Gamma tiene un desempeño de clasificación competitivo para el banco de datos utilizado en este trabajo de tesis.

6.2 Trabajos futuros.

1. Incorporar al proceso de predicción información los trabajos previamente realizados con respecto a la calidad del aire de la ciudad de México, de esta forma se podría tener un análisis a fondo de cómo afecta la calidad del aire a los niveles de contaminantes.
2. Generar una aplicación web que acceda al sistema informático desarrollado en este trabajo de tesis, en la cual se puedan observar estadísticas de un periodo definido del comportamiento de cada vehículo.

Referencias.

- [1] 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://biblioteca.semarnat.gob.mx/janium/Documentos/Ciga/libros2009/CG007297.pdf>
- [2] Ravishankara, a. R., Dawson, J. P., & Winner, D. a. (2012). New Directions: Adapting air quality management to climate change: A must for planning. Atmospheric Environment, vol. 50, pp. 387–389,doi:10.1016/j.atmosenv.2011.12.048
- [3] Anil K. Jain, J. M., K. Mohiuddin (1996) Artificial Neural Networks: A Tutorial. IEEE Computer, vol. 29, pp. 31-44, doi: 10.1109/2.485891
- [4] Cai, M., et al. (2009). Prediction of hourly air pollutant concentrations near urban arterials using artificial neural network approach. Transportation Research Part D: Transport and Environment vol.14, pp. 32-41, doi:10.1016/j.trd.2008.10.004
- [5] Jiang, D., et al. (2004). Progress in developing an ANN model for air pollution index forecast. Atmospheric Environment, vol. 38, pp.7055-7064, doi: 10.1016/j.atmosenv.2003.10.066
- [6] 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, issue 1, pp. 46 – 55, doi:10.1109/TEVC.2006.876364
- [7] Yetilmezsoy, K., Ozkaya, B., & Cakmakci, M. (2011) Artificial Intelligence-based Prediction Models for Environmental Engineering. Neural Network World, vol. 3, no. 11, pp. 193-218
- [8] Duda, R. O., Hart, P. E. & Stork, D. G. (2001). Pattern Classification, Wiley & Sons, Incorporated, USA.
- [9] N. Bose, a. N. S. K. (2009). Prediction of emissions Using Combustion Parameters in a Diesel Engine through Fuzzy Logic Techniques. International J. Comp. Intel. Appl., vol. 8 issue 2, pp. 195-207, doi: 10.1142/S1469026809002497
- [10] Cheon, S.-P., et al. (2009) Bayesian networks based rare event prediction with sensor data. Knowledge-Based Systems, vol. 22, issue 5, pp. 336-343, doi: 10.1016/j.knosys.2009.02.004.

- [11] Deleawe, S., et al. (2010) Predicting Air Quality in Smart Environments. J Ambient Intell Smart Environ, vol. 2, no. 2, pp. 145-152, doi: 10.3233/AIS-2010-0061
- [12] Dong, M., et al. (2009) PM2.5 concentration prediction using hidden semi-Markov model-based times series data mining. Expert Systems with Applications, vol. 36, issue 5, pp. 9046-9055, doi: 10.1016/j.eswa.2008.12.017
- [13] Von K. STEIN BUCH, K. (1961) . Die Lernmatrix. Kybernetik, vol. 1, issue 1, pp. 36-45, doi:10.1007/BF00293853
- [14] Willshaw, D. J. B., O. P, Longuet-Higgins, H (1969) Non -holographic associative memory. Nature, vol. 222, pp. 960-962, doi: 10.1038/222960a0
- [15] Anderson, J. A. (1972). A simple neural network generating an interactive memory. Mathematical Biosciences, vol. 14, no. 4, pp. 197-220, doi: 10.1109/TC.1972.5008975
- [16] Kohonen, T. (1972). Correlation matrix memories. IEEE TRANSACTIONS ON COMPUTERS, vol. C-21, issue 4, pp. 353-359, doi: 10.1109/TC.1972.5008975
- [17] Nakano, K. (1972). Associatron - A model of associative memory. IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-2, issue 3, pp. 380-388, doi: 10.1109/TSMC.1972.4309133
- [18] Amari, S.-I. (1972) Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements. IEEE Transactions on Computers, vol. C-21, issue 11, pp. 1197-1206, doi: 10.1109/T-C.1972.223477
- [19] 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, doi: 10.1073/pnas.79.8.2554
- [20] Ritter, G. X., et al. (1998) Morphological Associative Memories. IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 9, issue 2, pp. 281-293, doi: <http://dx.doi.org/10.1109/72.661123>
- [21] López, I., et al. (2011). Polluants Time Series Prediction Using the Gamma Classifier. International Journal of Computational Intelligence Systems, vol. 4, issue 4, pp. 680-711, doi: <http://dx.doi.org/10.2991/ijcis.2011.4.4.23>
- [22] López, I. (2007). Clasificador automático de alto desempeño. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México
- [23] Uriarte, A. (2012). Procesamiento de datos de monitoreo atmosférico usando clasificación no convencional. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación

- [24] 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.
- [25] 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, doi: 10.1007/11925231_34
- [26] 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 Artificial Intelligence, MICAI 2006, ISBN: 0-7695-2722-1, pp. 151-160, doi: 10.1109/MICAI.2006.15
- [27] 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, doi: 10.1007/s11063-007-9040-2
- [28] Yáñez-Márquez, C. (2002). Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios. Tesis de Doctorado en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.
- [29] Lumbreras, J., et al. (2008). Assessment of vehicle emissions projections in Madrid (Spain) from 2004 to 2012 considering several control strategies. Transportation Research Part A: Policy and Practice, vol. 42, issue 4, pp. 646-658, doi: 10.1016/j.tra.2008.01.026
- [30] Barrón-Adame, J. M., et al. (2012) Unsupervised system to classify SO2 pollutant concentrations in Salamanca, Mexico. Expert Systems with Applications, vol. 39, issue 1, pp. 107-116, doi: 10.1016/j.eswa.2011.05.083
- [31] Aberer, K., et al. (2010) OpenSense: open community driven sensing of environment. Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming, ISBN: 978-1-4503-0431-3, pp. 39-42, doi: 10.1145/1878500.1878509
- [32] Rahman, F., et al. (2012) Let EcoDrive be your guide: development of a mobile tool to reduce carbon footprint and promote green transport. Proceedings of the 27th Annual ACM Symposium on Applied Computing, ISBN: 978-1-4503-0857-1, pp. 519-524, doi:10.1145/2245276.2245376
- [33] Barrenetxea, G., et al. (2008) SensorScope: Out-of-the-Box Environmental monitoring. Proceedings of the 7th international conference on Information processing in sensor networks, ISBN: 978-0-7695-3157-1, pp. 332-343, doi:

10.1109/ipsn.2008.28

- [34] Honicky, R., et al. (2008) N-smarts: networked suite of mobile atmospheric real-time sensors. Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions, ISBN: 978-1-60558-180-4, pp. 25-30, doi: 10.1145/1397705.1397713
- [35] SEMARNAT, NOM-041-SEMARNAT-2006. SEMARNAT, 2007, pp. 35–41, 2006.
- [36] SEMARNAT, NOM-047-SEMARNAT-1999. SEMARNAT, 2007, pp. 1–29, 2006.
- [37] Gaceta, Oficial. Del, Distrito Federal (2002). “Asamblea legislativa del distrito federal ley ambiental del distrito federal,” vol. 2000, pp. 1–105.
- [38] Deepak G , D. P. B. S. (2012) Challenging Issues and Limitations of Mobile Computing. International Journal of Computer Technology & Applications vol. 3, issue 1, pp. 177-181
- [39] Venkatraman, S. (2005) Mobile Computing Models - Are they Meeting the Mobile Computing Challenges? Association for Computing Machinery New Zealand Bulletin, vol. 1, no. 1, ISSN 1176-9998
- [40] Terrenghi, L., et al. (2009) Elastic mobility: stretching interaction. Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, ISBN: 978-1-60558-281-8, pp. 1-4, doi: 10.1145/1613858.1613916
- [41] Creeger, M. (2009) DISTRIBUTED COMPUTING CTO Roundtable : Cloud Computing. Magazine Queue - Distributed Computing, vol. 7, issue 5, pp. 18, doi: <http://dx.doi.org/10.1145/1551644.1551646>
- [42] Peter Mell and T. Grance (2011) The NIST Definition of Cloud Computing. National Institute of Standards and Technology, NIST Special Publication 800-145, pp. 1-7
- [43] Mohiuddin Ahmed, A. S. M. R. C., Mustaq Ahme, Md. Mahmudul Hasan Rafee (2012) An Advanced Survey on Cloud Computing and State-of-the-art Research Issues. International Journal of Computer Science, vol. 9, issue 1, pp. 201
- [44] Leavitt, N. (2009) Is Cloud Computing Really Ready for Prime Time? IEEE Computer, vol. 42, issue 1, pp. 15-20, doi: 10.1109/MC.2009.20
- [45] Buyya, R., et al. (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. , vol. 25, issue 6, pp. 599-616, doi: 10.1016/j.future.2008.12.001
- [46] Fakoor, R., et al. (2012) An integrated cloud-based framework for mobile phone sensing. Proceedings of the first edition of the MCC workshop on Mobile cloud

computing 47-52 DOI: 10.1145/2342509.2342520

- [47] Zhang, X., et al. (2011) Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing. Mobile Networks and Applications, vol. 16, issue 3, pp. 270-284, doi: 10.1007/s11036-011-0305-7
- [48] IEE Standard Association (2012) Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Computer Society 2793
- [49] Wi-Fi Alliance. (1999). "Wi-Fi." <http://www.wi-fi.org/>.
- [50] Forman, G. H. and J. Zahorjan (1994) The challenges of mobile computing. Computer, vol. 27, issue 4, pp. 38-47, doi: 10.1109/2.274999
- [51] Sinha, P. K. (1997). Distributed Operating Systems: Concepts and Design, Wiley, pp. 743.
- [52] Fernando, N., et al. (2013). "Mobile cloud computing: A survey." Future Generation Computer Systems, vol. 29, issue 1, pp. 84-106, doi:10.1016/j.future.2012.05.023
- [53] Khan, A. N., et al. (2012) Towards secure mobile cloud computing: A survey. Future Generation Computer Systems, vol. 29, issue 5, pp. 1278-1299, doi: 10.1016/j.future.2012.08.003
- [54] Ihm, S., et al. (2011) CloneCloud: Elastic Execution between Mobile Device and Cloud. EuroSys' 11 Proceedings of the sixth conference on Computer systems , ISBN: 978-1-4503-0634-8, pp. 301-314, doi: [10.1145/1966445.1966473](https://doi.org/10.1145/1966445.1966473)
- [55] Cuervo, E., et al. (2010) MAUI: making smartphones last longer with code offload. MobiSys' 10 Proceedings of the 8th international conference on Mobile systems, applications, and services, ISBN: 978-1-60558-985-5, pp. 49-62, doi: 10.1145/1814433.1814441
- [56] López-Yáñez, I. (2011). Teoría y aplicaciones del clasificador Gamma. Tesis de Doctorado en Ciencia de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México
- [57] Rosen, K. H. (2004). Matemática discreta y sus aplicaciones. (5th Ed.). España. McGraw-Hill
- [58] Flores-Carapia, R. (2006). Memorias asociativas Alfa-Beta basadas en el código Johnson- Möbius modificado. Tesis de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, Centro de Investigación en Computación, México D.F., México.

- [59] Amadeo José Argüelles Cruz, Mario Matamoros de Luis, Paulina Elizabeth Moreno Aguilera, Cornelio Yáñez Márquez (2013), “Mobile system surveillance for vehicular pollutants emission, based on Wi-Fi ad-hoc network”. 7th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2013), Costa Rica, 2013.
- [60] Connie Guglielmo (2013). Smartphones Dominate Mobile Phone Sales in Q3, Lenovo Joins Samsung, Apple In Top Three. Forbes, <http://www.forbes.com/sites/connieguglielmo/2013/11/14/smartphones-dominate-mobile-phone-sales-in-q3-lenovo-joins-samsung-apple-in-top-three/>
- [61] Application Fundamentals. Android Developers. <http://developers.android.com>
- [62] Amazon Web Services AWS, <http://aws.amazon.com/>