INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Verificación de síndromes mediante un sistema Web interactivo

T E S I S

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

PRESENTA:

SERGIO CERÓN FIGUEROA

DIRECTORES DE TESIS:

DR. CORNELIO YÁÑEZ MÁRQUEZ
DR. ITZAMÁ LÓPEZ YÁÑEZ

MÉXICO, D.F. NOVIEMBRE DE 2013

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 14:00 horas del día 12 del mes de noviembre de 2013 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Verificación de síndromes mediante un sistema Web interactivo"

Presentada por el alumno:

CERÓN	**FIGUEROA**	**SERGIO**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

A	1	2	0	3	7	4

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

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. Cornelio Yáñez Márquez

Dr. Itzamá López Yáñez

Dr. Marco Antonio Moreno Ibarra

Dr. Amadeo José Argüellez Cruz

Dr. Oscar Camacho Nieto

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

ESTADOS UNIDOS MEXICANOS
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México D.F. el día 26 del mes Noviembre del año 2013, el que suscribe Sergio Cerón Figueroa alumno del Programa de Maestría en Ciencias de la Computación con número de registro A120374, adscrito a Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de Dr. Cornelio Yáñez Márquez y Dr. Itzamá López Yáñez y cede los derechos del trabajo intitulado Verificación de síndromes mediante un sistema Web interactivo, 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 sceronf@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Sergio Cerón Figueroa

Nombre y firma

Resumen

En el presente trabajo de tesis se desarrolla e implementa un sistema web interactivo para la verificación de síndromes utilizando algoritmos de reconocimientos de patrones basados en cómputo no convencional. El algoritmo de clasificación utilizando es un híbrido entre las Memorias Asociativas Morfológicas, las cuales pertenecen al enfoque asociativo, y el clasificador kNN.

Se realiza la clasificación de pacientes con tumores mamográficos, problemas de columna vertebral y parkinsonismo. Los bancos de datos utilizados se encuentran en el repositorio del UCI para las tareas de clasificación.

El sistema web se presenta como una herramienta que permite un diagnóstico remoto a pacientes con algún tipo de síndrome, brindando una alta tasa de verdaderos positivos en la verificación de los síntomas.

Abstract

In the current document of thesis, is developed and implemented a web-based interactive system for syndromes verification, using pattern recognition algorithms based on unconventional computation. The classification algorithm used is a hybrid between Morphological Associative Memories, which belong to the associative approach, and the kNN classifier.

Classification of mammographic masses patients, spinal problems and parkinsonism is performed. The UCI databases are used for classification tasks.

The web system is presented as a tool that allows remote patients diagnosis with some kind of syndrome, providing a high rate of true positives in symptoms verification.

Índice

Índice de figuras

Índice de tablas

Capítulo 1

1. Introducción

En este trabajo de tesis se presentan los fundamentos teóricos y prácticos para el desarrollo de un sistema web interactivo que permite la verificación de síndromes haciendo uso de algoritmos de reconocimiento de patrones sobre aplicaciones web.

1.1. Antecedentes

La palabra síndrome proviene del griego antiguo syndrome que significa “concurrencia de síntomas". De manera general un síndrome se entiende como el conjunto de características que definen a una enfermedad.

A través de la historia del ser humano se han desarrollado una serie de técnicas para el diagnóstico de enfermedades, tomando como base el cuadro clínico del paciente. Algunas veces éstas técnicas han tenido un mal desempeño, proporcionando falsos negativos que han costado en algunos casos la vida de las personas. De esta forma, se ha despertando el interés de la comunidad científica para mejorar y automatizar las técnicas de diagnóstico y verificación de síndromes, a fin de que éstas sean más confiables y precisas [1, 2, 3].

El reconocimiento de patrones (RP) es una herramienta poderosa para la clasificación de síndromes, permitiendo al experto confirmar y verificar los síntomas del paciente, asociando estos a una enfermedad específica.

Las técnicas de clasificación basadas en el RP pueden ser automatizadas por medio de sistemas de cómputo. El sistema podrá operar previa una fase de entrenamiento, donde se ajustará al tipo de patrones y posteriormente actuará sobre un conjunto de síndromes cuyo tipo se desconoce.

También, existen otras tareas que pertenecen al reconocimiento de patrones, como la recuperación, regresión, clustering, sistemas de recomendación, por mencionar algunos. Sin embargo, la clasificación esta asociada en gran parte de la literatura al reconocimiento de patrones.

Actualmente existen varias técnicas y herramientas para el diagnóstico y verificación de síndromes, algunas de ellas pertenecientes al reconocimiento de patrones:

- *Enfoque probabilístico-estadístico* [4, 5, 6, 7].
- *Enfoque neuronal* [8, 9, 10, 11, 12].
- *Enfoque lógico-difuso* [13].
- *Enfoque asociativo* [14].
- *Enfoque basado en árboles* [15, 16, 17].
- *Máquinas de soporte vectorial* [18, 19, 20, 21, 22, 23, 24].
- *Enfoque neuro-difuso* [25, 26].
- *Clustering* [27, 28].
- *Morfología Matemática* [29, 30].
- *Aplicaciones Web [31, 32, 33].*

Si bien, el uso del enfoque asociativo para la verificación de síndromes ha sido casi nulo; en el trabajo de Yáñez *et al.* [14], se presenta un sistema para la detección automática de fracturas en cráneos haciendo uso de técnicas de morfología matemática junto a los modelos asociativos Alfa-Beta, logrando así un rendimiento competitivo con respecto a los clasificadores presentados en la literatura actual.

El primer modelo del enfoque asociativo fue desarrollado por Karl Steinbuch con su *Lernmatrix* en 1961, actuando como una memoria heteroasociativa para la clasificación de patrones binarios. Posteriormente, Willshaw, Buneman y Longuet-Higgins presentaron su modelo llamado *Correlograph* basado en las propiedades ópticas de la luz a través de un dispositivo capaz de actuar como una memoria asociativa [34].

Durante varios años a la postre, el desarrollo del enfoque asociativo mostró avances significativos: las *Correlation Matrix Memories* de Teuvo Kohonen, el *Linear Associator* de Anderson y Kohonen, y el *Associatron* de Kaoru Nakano son algunos ejemplos. Sin embargo, fue hasta 1982 cuando John J. Hopfield marcó un avance significativo, con la memoria asociativa que lleva su apellido. Tal avance no se repetiría hasta 1998 con las *Morphological Associative Memories* de Ritter *et al.*, las cuales permitieron superar las limitaciones en la capacidad de aprendizaje de los modelos conocidos en esa época [35].

Aplicaciones Web

Las aplicaciones web tienen su historia hasta hace poco, cuando las necesidades de sistemas más complejos sobre el protocolo HTTP (HiperText Transfer Protocol) eran requeridos. Este, es el protocolo estándar utilizado en la WWW (Word Wide Web) que Tim Berners-Lee creó en la Organización Europea para la Investigación Nuclear (CERN) en el año de 1989 [36, 37, 38].

Las primeras aplicaciones web eran mas bien simples páginas que servían contenido estático, y el usuario no podía interactuar directamente con ellas. Dado que la naturaleza misma del protocolo HTTP es unidireccional, es decir, solo se envían paquetes hacia el cliente, fue necesario crear protocolos alternativos para cada tipo de problema [39].

Protocolos como el FTP (File Transfer Protocol) para la transferencia de archivos, SMTP (Simple Mail Transfer Protocol) y POP3 (Post Office Protocol 3) para correo, Usenet para noticias, etc. tuvieron su auge hasta principios del siglo XXI, cuando la WWW se apoyó sobre un conjunto de tecnologías conocidas como AJAX (Asynchronous Javascript And XML) para simular una comunicación bidireccional entre el cliente y el servidor.

Posteriormente el término Web2.0 [40, 41, 42] se empezó a volver popular para referirse a aquellos sitios de Internet que generaban su contenido a partir de la participación de sus usuarios. Este concepto se popularizó gracias al auge de los blogs, redes sociales y plataformas colaborativas como Wikipedia. El conjunto de tecnologías que caracterizaron la Web 2.0 son: AJAX, RSS, JSON, XML por mecionar algunas.

Actualmente existen tecnologías más sofisticadas, como los WebSockets [43], que permiten crear canales de acceso bidireccionales entre el cliente y el servidor sobre aplicaciones web, para llevar a cabo tareas complejas como: resultados en tiempo real, envío y actualización de archivos, generación dinámica de imágenes, etc.

Recientemente la versión 5 del lenguaje HTML se ha vuelto popular en Internet [44, 45, 46], prometiendo dar un salto en los paradigmas de aplicaciones que existen actualmente. Esta nueva versión representa una mejora a los navegadores de Internet, así como a los servidores HTTP que permitirán actuar conjuntamente para sincronizar contenido entre ambas partes, acceder a archivos vía remota y localmente, funcionar de manera offline, etc.

El día de hoy existen tecnologías web que permiten implementar algoritmos de reconocimiento de patrones para formar sistemas de clasificación. La propuesta de este trabajo es utilizar las memorias asociativas morfológicas sobre una aplicación web que permita llevar a cabo tareas de clasificación de síndromes.

1.2. Justificación

El diagnóstico remoto de pacientes es importante en lugares donde no se tiene acceso a costosos aparatos y a médicos especialistas. Existen también casos donde el tiempo para confirmar una enfermedad es importante, y un resultado oportuno puede se decisivo en la vida del paciente. De esta forma, el sistema propuesto es una herramienta que podrá resolver los problemas mencionados anteriormente.

Debido a que el uso del enfoque asociativo para la clasificación de síndromes ha sido casi nulo, y que las memorias morfológicas han demostrado un rendimiento considerable sobre diferentes tareas de reconocimiento de patrones, se presenta un modelo competitivo con algunos clasificadores presentados en el estado del arte, basado en las memorias asociativas morfológicas para la verificación de síndromes.

1.3. Objetivo General

Desarrollar e implementar un sistema web interactivo que permita la verificación de síndromes con una alta eficiencia y confiabilidad, haciendo uso de algoritmos de reconocimiento de patrones sobre una aplicación web.

1.4. Objetivos específicos

- Realizar una investigación sobre estado del arte de los algoritmos de clasificación relacionados al diagnóstico y verificación de síndromes.
- Desarrollar una aplicación web que permita ejecutar el algoritmo de clasificación propuesto.
- Analizar las ventajas de implementar el sistema web sobre una plataforma de cómputo en la nube.

- Realizar experimentos sobre varios bancos de datos, para probar el sistema web y el algoritmo propuesto de clasificación.
- Describir un método para usar las Memorias Autoasociativas Morfológicas como clasificador de patrones.

1.5. Contribuciones

- Aplicación web que permite la clasificación de síndromes
- Algoritmo de clasificación basado en las Memorias Autoasociativas Morfológicas
- Aplicación web para comparar el algoritmo propuesto con otros clasificadores
- Graficas de estadísticas del rendimiento
- Modelo de implementación en la nube

1.6. Organización del Documento

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

En el capítulo 2 se presenta el estado del arte referente al panorama actual de investigación de las diferentes técnicas y herramientas para la verificación de síndromes.

En el capítulo 3 se presentan aquellas herramientas matemáticas y conceptuales necesarias para el desarrollo de este trabajo.

En el capítulo 4 se desarrolla la parte principal de esta tesis, dado que aquí se presenta el algoritmo de clasificación utilizado en el sistema web. Asimismo, se describe la arquitectura, diseño y flujo de trabajo de la aplicación web.

En el capítulo 5 se realiza un análisis comparativo de los resultados obtenidos con el algoritmo de clasificación propuesto sobre el sistema web desarrollado.

Finalmente, en el capítulo 6 se exponen las conclusiones, aportaciones y trabajo a futuro propuesto para desarrollar.

Capítulo 2

2. Estado del Arte

Un diagnóstico correcto de síndromes en la actualidad es un tema importante que requiere atención, así como soluciones que permitan una alta confiabilidad. En este capítulo se detallan los últimos trabajos realizados respecto a la detección y verificación de síndromes, tomando como base diferentes enfoques del reconocimiento de patrones.

2.1. Enfoque Neuronal

Las Redes Neuronales Artificiales (RNA) son modelos computarizados simples del sistema nervioso biológico, que posibilitan el reconocimiento de patrones complejos en un conjunto de datos. Pueden ser aplicadas a problemas que tengan un gran número de variables lineales, así como en los casos donde no se tiene conocimiento de la relaciones entre dichas variables.

El uso de las RNA para la detección y verificación de síndromes es bastante común. Tagluk y Sezgin [8], Tagluk *et al.* [11], Güneş *et al.* [12], proponen en sus trabajos el uso de redes neuronales con características biespectrales (QPC) para la identificación de pacientes con *Síndrome de Apnea Obstructivo de Sueño (OSAS)*, el cual es una situación que se presenta cuando la persona esta durmiendo y repetidamente se detiene la vía aérea superior del tracto mientras se sigue ejerciendo un esfuerzo para la respiración durante al menos 10 segundos.

Yang *et al.* [9], presentan en su trabajo una aplicación del Perceptron Multicapa (MLP) para la clasificación de patrones del *Síndrome de Dolor Regional Complejo*, el cual es una condición que causa un dolor fuerte en una extremidad (o parte de ella) y que puede ocasionar severos deterioros en el rendimiento físico.

Asimismo, se han desarrollado modelos de redes neuronales que combinados con otros métodos mejoran los resultados de las predicciones, como es el caso de Hirose *et al.* [10], donde se hace uso de Regresión Logística Múltiple (MLR) basado en factores clínicos como el *Indice de Resistencia a la Insulina Calculado por Homeostasis (HOMA-IR)*,

para predecir un posible *Síndrome Metabólico* (Diabetes, Hipertensión, Dislipdemia ó Trastornos Ateroscleróicos).

De forma similar, en el trabajo de Ahmad *et al.* [47] se muestra un método de optimización de las RNA basado en algoritmos genéticos para mejorar el diagnóstico de cáncer de mama, logrando resultados bastante competitivos en la literatura actual.

2.2. Enfoque Probabilístico Estadístico

Los modelos Probabilísticos Estadísticos nacen de uno de los teoremas más importantes de la probabilidad: el teorema de Bayes, del cual se derivan una gran variedad de métodos como las redes bayesianas y el clasificador naïve Bayes.

La importancia de este teorema radica en la posibilidad de cambiar el orden de las probabilidades, es decir, una vez conociendo la probabilidad a *priori*, se puede calcular la probabilidad a *posteriori*, según la siguiente fórmula:

Teorema 1. *Teorema de Bayes. Sean $A_1, A_2, ..., A_n$ eventos correspondientes a la partición del espacio muestral X, y B un evento dentro del mismo espacio, entonces:*

$$P(A_i \mid B) = \frac{P(A_i)P(B|A_i)}{P(B)}$$

donde, $P(A_i)$ es la probabilidad de que suceda A_i antes de saber algo de B, y $P(B \mid A_i)$ es la probabilidad de que suceda B dentro de la partición A_i. De esta forma:

$$P(w_j \mid x) = \frac{P(w_j)p(x|w_j)}{p(x)}$$

con la cual podemos calcular teóricamente la pertenencia a una clase de un patrón de entrada.

En el trabajo de Aussem *et al.* [4], se hace uso de una Redes Bayesiana (RB) para realizar el análisis de los factores de riesgo del *Carcinoma de Cavum*, uno de los tipos de cáncer que se da en la parte superior de la faringe y que según un estudio realizado por la Agencia Internacional de Investigación en Cáncer se da en tazas de incidencia bajas, pero que en ocasiones puede darse en forma endémica. El trabajo mencionado

anteriormente toma como base las *fronteras de Markov* para el *feature selection* de la RB. El limite de Markov es el mínimo conjunto de variables condicionales en las cuales todas las demás variables medidas se vuelven independientes [4].

Como se mencionó anteriormente un Síndrome Metabólico define una variedad de riesgos que incluyen obesidad, resistencia a la insulina, hipertensión, etc., y afecta a más del 25 % de los adultos simplemente en los Estados Unidos. Se ha vuelto un problema muy serio en los países Asiáticos debido a los hábitos comunes de dieta y estilos de vida. El trabajo de Park y Cho [5] propone una forma de ordenamiento evolutivo de atributos en las redes bayesianas para la predicción de algún Síndrome Metabólico.

Un método que se ha aplicado a la detección del *Síndrome de Fatiga Crónico (CFS)*, es el naïve Bayes, asumiendo que las probabilidades de cada rasgo en el problema son independientes. Huang *et al.* [6] muestran que su modelo puede lidiar con la complejidad de las asociaciones entre el CFS usando los factores genéticos como Polimorfismos de un Solo Nucleotido (SNPs).

2.3. Enfoque basado en Arboles de Decisión

Una manera intuitiva de clasificar patrones es a través e una secuencia de preguntas, donde la pregunta siguiente será hecha dependiendo de la respuesta a la pregunta actual.

Este método es particularmente usado para datos no numéricos, dado que todas las preguntas tienen como respuesta un "si/no", "verdadero/falso" o de manera general un conjunto de valores definidos, de tal forma que no es necesario convertirlos a números.

Tal secuencia de preguntas es mostrada en forma de un grafo dirigido conocido como árbol de decisión, donde por convención el primero nodo ó nodo raíz esta en la parte superior, conectado con los nodos sucesivos.

La fase de clasificación inicia en el nodo raíz, el cual pregunta por el valor de una característica especifica del patrón. Los diferentes nodos a los que apunta son un posible valor de respuesta y con base en ella seguirá el recorrido del árbol en forma descendiente.

Es importante mencionar que los valores de los nodos para cada pregunta son mutuamente excluyentes, es decir, que sólo se podrá tomar uno y sólo un camino a seguir en el recorrido del árbol.

Un ejemplo de árbol de decisión simple se muestra en la figura 2.1.

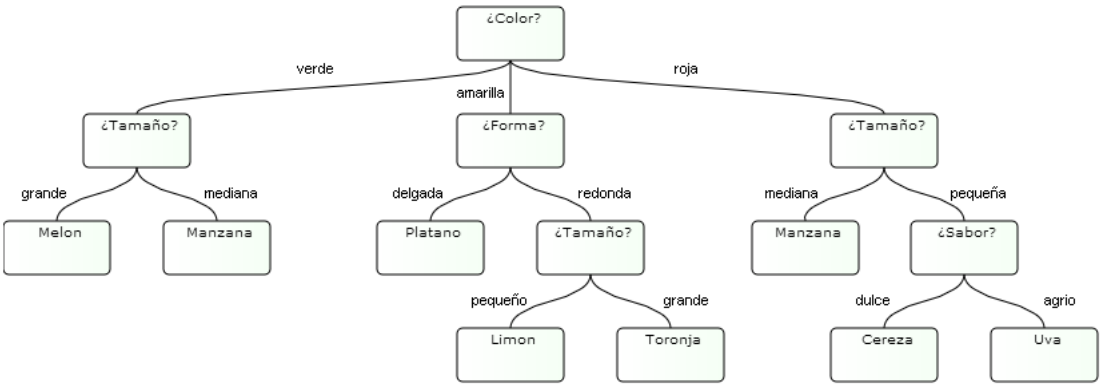


Figura 2.1: Representación de un árbol de decisión para la clasificación de frutas según su color, forma, tamaño y sabor

Uno de los trabajos con respecto al enfoque basado en árboles, es un sistema de soporte a decisiones realizado por Worachartcheewan *et al.* [15] para identificar individuos con Síndrome Metabólico sobre la población de Tailandia.

Sin embargo, no todos los síndromes ostentan el titulo como tal; las enfermedades del corazón son un tema importante que requiere un diagnóstico preciso en las salas de emergencia. Son *et al.* [16], proponen un modelo para el diagnóstico temprano de la *Insuficiencia Cardíaca Congestiva (ICC)* haciendo uso de arboles de decisión y *Rough Sets*.

La ICC es considerada un síndrome y define como el estado en el cual hay anomalías cardíacas que pueden causar un mal funcionamiento del corazón, tales como no cumplir la demanda de circulación de sangre en el cuerpo y elevando los niveles de presión en la misma [48].

En el trabajo de Scalzo *et al.* [17], se presenta una forma de hacer predicciones de *Hipertensión Intra-Craneal (IH)* usando arboles de decisión aleatorizados; un nuevo modelo llamado (Extra-Trees) para la clasificación que demuestra una alta eficiencia en comparación con algunos clasificadores como el Linear y AdaBoost.

La IH es una forma específica de *Lesiones Cerebrales Traumaticas (TBI)* que necesita de un tratamiento constante a fin de evitar daños secundarios, y sus posibles causas son:

expansión del volumen de sangre en las arterias ó crecimiento de una masa de lesión cerebral [17].

2.4. Enfoque Lógico Difuso

Existen problemas donde hay de conocimiento informal de un problema, y necesitamos construir un clasificador. Las funciones de pertenencia a una categoría son la base de la lógica difusa, permitiendo convertir una medida objetiva en una categoría de pertenencia subjetiva que posteriormente y con base en una regla de conjunción, transformará los valores en una función de discriminación [49].

Existen muchas maneras de definir una función de discriminación, pero la más común es:

$$1 - Min[\mu_x(x), \mu_y(y)]$$

donde su extensión obvia es cuando hay más de dos características:

$$1 - Min[\mu_{x_1}(x_1), \mu_{x_2}(x_2), \mu_{x_3}(x_3), ..., \mu_{x_n}(x_n)]$$

Algunas limitaciones de este enfoque son:

- Es difícil trabajar con vectores de dimensiones altas o en problemas demasiado complejos
- Los métodos puramente lógico difusos, no hacen uso del conjunto de entrenamiento

Es por ello que son pocos los trabajos recientes sobre síndromes relacionados con este enfoque, Álvarez-Estévez y Moret-Bonillo [13] muestran un sistema automático que aplica lógica difusa para detectar eventos "Apneicos" y clasificarlos en alguna forma del síndrome de Apnea (Apena normal ó Hipo-apnea). Este trabajo utiliza técnicas de Lógica Difusa con Pesos, junto con los operadores difusos AND y OR para mejorar el rendimiento de la clasificación.

2.5. Enfoque basado en Máquinas de Soporte Vectorial

Las SVM (Support Vector Machines) funcionan como modelos de aprendizaje supervisado para diferentes tareas del reconocimiento de patrones. Pueden efectuar de manera eficiente una clasificación no lineal, elevando los datos de entrada a un espacio de dimensión superior (n) lo suficientemente grande, de tal forma que los patrones pertenecientes a dos categorías puedan ser separados por un hiperplano de dimensión $n-1$[49].

Para lograr encontrar el hiperplano se requiere maximizar la distancia de los *vectores de soporte* al hiperplano; dichos vectores representan a los patrones que se encuentran en la frontera entre clases, es decir, al hiperplano buscado.

Un ejemplo de SVM se puede apreciar en la figura 2.2, donde los vectores de soporte están marcados con un recuadro y el hiperplano con una línea punteada.

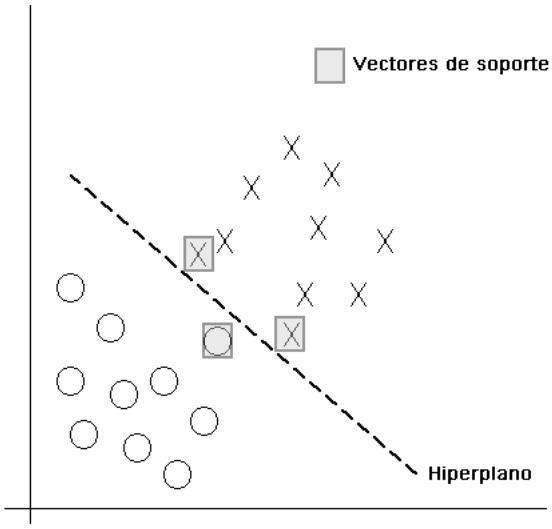


Figura 2.2: Ejemplo de Máquina de Soporte Vectorial

También es cierto que en la actualidad las SVMs son uno de los modelos más precisos para la regresión, recuperación y clasificación de patrones.

En los trabajos de Dobrowolski *et al.* [18], y Subasi [19], se muestra el uso de SVMs para la clasificación y diagnóstico de *Desórdenes Neuromusculares (ND)*. El primer trabajo [18], propone un nuevo método basado en el análisis de escalogramas determinado por una técnica llamada Symlet 4. Mientras que en el segundo [19], las SVMs son

combinadas con un modelo de Optimización de Enjambres de Partículas (PSO) para mejorar la precisión en el diagnóstico de los ND.

Los ND son un síndrome que se origina en el sistema nervioso, específicamente en las uniones neuromusculares, y causa diferentes daños en las fibras musculares que van desde la pérdida de resistencia a la amputación [19].

Por otro lado, Dukart *et al.* [20] proponen una nueva técnica de clasificación basada en SVMs e *Imágenes de Resonancia Magnética* para mejorar la diferenciación ó relación entre el Síndrome de Alzheimer y la *Degeneración del Lóbulo Frontotemporal*. De manera similar, el uso de imágenes neurológicas es aplicado en el trabajo de Orrù *et al.* [23] para identificar otros tipos de desordenes psicológicos como la esquizofrenia, depresión, bipolaridad y autismo.

Existen a su vez, para este mismo enfoque, trabajos relacionados con el *Síndrome de Apnea Obstructivo de Sueño (OSAS)* mencionado al principio de este capítulo. Khandoker *et al.* [24], usan SVMs para un sistema de detección automática del *OSAS*, analizando registros de electrocardiogramas y permitiendo así estimar la severidad del mismo.

Recientes trabajos de investigación han utilizado las SVMs para detección de cáncer de mama; en el artículo publicado por Azar y El-Said [50] se muestra un análisis comparativo de seis tipos diferentes de SVMs para medir el rendimiento, utilizando como referencia el banco de datos clásico de Wisconsin Breast Cancer ubicado en el repositorio del UCI.

2.6. Clustering

El clustering es un procedimiento que consiste en realizar agrupaciones de patrones con base en sus rasgos, mediante el uso de algún tipo de métrica. Dentro del reconocimiento de patrones se considera una modelo de aprendizaje no supervisado.

Este enfoque ha sido usado recientemente para tareas de clasificación; en el trabajo de Nakayama *et al.* [27], se combina con *Mapas Auto Organizados (SOM)*, para predecir los resultados y complicaciones en pacientes con *Insuficiencia Hepática Aguda* según el intervalo desde los primeros síntomas hasta que se desarrolla *Encefalopatía*.

Dentro de este enfoque, uno de los algoritmos más utilizados es el *k-means*. El *k-means* consiste de forma iterativa en ir calculando los centroides de un agrupamiento que

minimicen la distancia entre los datos y el centroide más cercano. De esta manera se obtienen particiones que tengan una varianza intra-clase relativamente baja.

De manera formal, es un algoritmo para poner N puntos de datos en un espacio $I-dimensional$ dentro de K agrupamientos. Donde cada agrupamiento es parametrizado por un vector $m^{(k)}$ llamado su vecino (mean).

Los puntos de datos se denotan por $x^{(n)}$, donde n va desde 1 hasta N y cada vector x tiene I componentes x_i. De esta forma se asume que el espacio x esta dentro de otro espacio real y se pueden definir métricas de distancia entre puntos.

A continuación se muestra el algoritmo para el *k-means* [51]:

Algoritmo 1 Algoritmo estándar del *k-means*

Inicialización Establecer los K vecinos $m^{(k)}$ a valores aleatorios

Asignación Cada punto de datos n es asignado al vecino más cercano, de esta forma, denotamos por $\hat{k}^{(n)}$ el punto $x^{(n)}$ que pertenece a $k^{(n)}$, como se muestra a continuación:

$$\hat{K}^{(n)} = argmin\{d(m^{(k)}, x^{(n)})\}$$

donde $d(m^{(k)}, x^{(n)})$ es una métrica dentro del espacio x entre los puntos $m^{(k)}$ y $x^{(n)}$.

Actualización Este paso consiste en ajustar los vecinos con base en la siguiente fórmula:

$$m^{(k)} = \frac{\sum_n r_k^{(n)} x^{(n)}}{R^{(k)}}$$

donde $R^{(k)}$ es conocido como la responsabilidad total del vecino k y se calcula de la siguiente manera:

$$R^{(k)} = \sum_n r_k^{(n)}$$

Repetir los pasos de asignación y actualización Se repiten ambos pasos hasta que no haya cambios en las asignaciones, es decir, hasta que $\hat{k}^{(n)}$ se mantenga constante.

El algoritmo de *k-means* fue visto originalmente como un intento de buscar un método computable para buscar una partición óptima de un conjunto de datos. De manera

general, el procedimiento no converge a esa solución óptima, aunque hay casos donde sí lo hace [52].

Dada la naturaleza heurística del algoritmo, es útil para resolver problemas con alto coste computacional, en un menor tiempo y con resultados bastante competitivos. Ha sido empleado no solo en clasificación de patrones, sino también en la detección de similitudes. Tal es el caso de Lee *et al.* [28], donde se muestra el uso de *Clustering* para generar ciertas hipótesis sobre las diferencias entre la *Esquizofrenia Relacionada con la Edad Paterna (PARS)* y otros tipos de esquizofrenia.

2.7. Morfología Matemática

La morfología matemática es una herramienta poderosa para segmentar imágenes de manera eficaz, útil para encontrar la forma de una región, y determinar sus fronteras y esqueletos.

La morfología matemática es utilizada para el procesamiento de imágenes, y sus operaciones principales son la dilatación y la erosión. Sin embargo, para comprender estos conceptos, antes es necesario definir algunas propiedades del conjunto $A \subseteq Z^2$, es decir, una región finita y no vacía.

Dado que $A \subseteq Z^2$, el operador + es conmutativo, asociativo, existe un elemento neutro y un inverso aditivo. Tales propiedades están definidas en la adición de conjuntos de Minkowski y algunas operaciones básicas son las siguientes:

Elemento de estructura

Un elemento de estructura es un subgrafo de una imagen que permite investigar la morfología de los objetos n-dimensionales de otra imagen.

Erosión

La erosión es una operación morfológica que permite reducir el tamaño de objetos oscuros sobre fondo claro (dentro de una imagen) quitando las partes que no caben en el elemento de estructura.

Para un conjunto X, la erosión con un elemento de estructura B es denotado por $\varepsilon_\beta(X)$ y se define como el lugar geométrico de los puntos x tales que B esta incluido en X cuando su origen esta en x:

$$\varepsilon_\beta(X) = \{x|B_x \subseteq X\}$$

Aunque también puede ser escrita la fórmula en términos de una intersección de traslaciones del elemento de estructura:

$$\varepsilon_\beta(X) = \bigcap_{b\epsilon B} X_{-b}$$

Lo anterior esta definido para imágenes binarias, sin embargo, para imágenes en tonos de gris, basta con cambiar el conjunto X por una función f, y el operador intersección por el operador mínimo, como se muestra a continuación:

$$\varepsilon_\beta(f) = \bigwedge_{b\epsilon B} f_{-b}$$

Donde $f(x)$ es el valor asignado en tono de gris (normalmente de 0-255) y el mínimo, será el menor valor para la función en la ventana definida por el elemento de estructura cuyo origen está en x.

Dilatación

La dilatación es la operación morfológica dual de la erosión, que permite aumentar el tamaño de los objetos oscuros sobre fondo claro, dentro de una imagen, agregando las partes del elemento de estructura que no están incluidas en el objeto.

Para un conjunto X, la dilatación con un elemento de estructura B es denotado por $\delta_\beta(X)$ y se define como el lugar geométrico de los puntos x tales que B toca a X cuando su origen coincide en x:

$$\delta_\beta(X) = \{x|B_x \cap X \neq \emptyset\}$$

La fórmula anterior indica que al menos una parte del elemento de estructura debe esta dentro de B, de tal forma que lo restante del SE se añadirá a la imagen dilatada.

De manera similar a la erosión, la definición anterior se puede escribir en términos de la unión de un conjunto de traslaciones del elemento de estructura:

$$\delta_{\beta}(X) = \bigcup_{b \epsilon B} X_{-b}$$

Así podemos expandir la dilatación a imágenes en tonos de gris cambiando el conjunto X por una función f y el operador unión por el operador máximo:

$$\delta_{\beta}(f) = \bigvee_{b \epsilon B} f_{-b}$$

Donde $f(x)$ es el valor asignado en tono de gris (normalmente de 0-255) y el máximo, será el valor mayor para la función en la ventana definida por el elemento de estructura cuyo origen esta en x.

Apertura

La apertura morfológica es una operación que permite resaltar las partes claras de una imagen, reduciendo consigo las partes oscuras de la misma.

La idea detrás de la apertura es hacer una dilatación a una imagen erosionada, con el mismo elemento de estructura para recuperar tanto como sea posible la imagen original.

La apertura γ de una imagen f con un elemento de estructura B se denota por $\gamma_B(f)$ y esta definida como la erosión de f con B, seguida de una dilatación con el mismo elemento de estructura reflejado ($\check{B}$):

$$\gamma_B(f) = \delta_{\check{B}}[\varepsilon_{\beta}(f)]$$

Algunas de las propiedades que cumple la apretura son:

- Es idempotente, es decir, aperturas adicionales con el mismo elemento de estructura no afectan el resultado

- La apertura dará como resultado siempre una imagen más pequeña que la original

En general, la apertura es útil para remover pequeñas partes de la imagen que sean menores al elemento de estructura, por ejemplo ruido.

Dado que las operaciones son relativamente fáciles de computar, la morfología matemática es usada como herramienta para facilitar el reconocimiento de patrones, a manera de pre procesamiento.

En el trabajo de Naegel [30], se hace uso de un operador morfológico para detectar espacios oscuros correspondientes a espacios intervertebrales, a fin de etiquetar vertebras humanas sobre imágenes tridimensionales.

En el caso de Elmoataz *et al.* [29], se propone un método de segmentación basado en morfología matemática para la detección de tejidos tumorales en imágenes biomédicas.

2.8. Aplicaciones Web

Las aplicaciones web son esencialmente diferentes de las páginas de internet convencionales, ya que las primeras generan su contenido dinámicamente [53], es decir, no existe una versión física del recurso solicitado.

Una de las ventajas más significativas de una aplicación web, es que éstas corren directamente sobre el navegador, pero su lógica de negocio y la mayor parte del procesamiento de datos se realiza del lado del servidor [54].

En el trabajo de Michaelson *et al.* [31], se hace uso de calculadoras web programadas sobre PHP y Javascript, para estimar el riesgo de muerte por cáncer de mama, la reducción de la vida del paciente, y el impacto de diferentes opciones de tratamiento.

Lin *et al* [33], muestran en su trabajo el desarrollo de un sistema web en tiempo real para soporte a decisiones basado en la arquitectura MVC (Modelo-Vista-Controlador) aplicado al tratamiento de cáncer de próstata. El sistema permite a los médicos, por medio de diagramas interactivos, dar seguimiento a los registros de cada paciente de manera instantánea y eficiente. La arquitectura MVC puede ser fácilmente adaptada a para el tratamiento de otras enfermedades y sobre la información de diferentes hospitales.

Otro sistema web para el diagnóstico remoto de patologías en hospitales rurales de Japón fue desarrollado y presentado en el 5to Congreso Internacional de Microscopia Virtual (2012) por Mori *et al.*[32]. El sistema hace uso de diapositivas virtuales almacenadas en formato PDF (Portable Document Format) para ser compartidas a través de un servidor FTP para su acceso remoto.

Capítulo 3

3. Materiales y Métodos

En el presente capítulo se describen los materiales y métodos necesarios para el desarrollo del sistema interactivo que se presentará en el capitulo 4. Dentro de la primera sección se describen las memorias asociativas de manera general, para posteriormente, en la segunda sección abordar las memorias asociativas morfológicas, las cuales forman la base del algoritmo propuesto. La tercera sección describe de manera detallada algunas operaciones morfológicas necesarias para el pre procesamiento de imágenes con el fin de generar patrones. En la cuarta sección se describen las tecnologías de internet que se utilizarán en el sistema web.

3.1. Memorias Asociativas

Una memoria asociativa es un sistema de entrada-salida que puede actuar como un recuperador o clasificador de patrones [35]. El esquema básico de una memoria asociativa se puede ver en la figura 3.1:

Figura 3.1: Memoria Asociativa

Los patrones de entrada (x) y salida (y) están representados por un vector columna cada uno, de tal manera que, pueden formar una asociación entre ellos similar a un par ordenado: (x, y).

Con la finalidad de realizar una manipulación adecuada de los patrones de entrada y de salida, se denotarán con las mismas letras, pero agregándoles un superíndice para representar cada asociación. Por ejemplo, un patrón de entrada x^3 es correspondiente con un patrón de salida y^3, formando la asociación (x^3, y^3).

Una memoria asociativa M es una matriz bidimensional que se genera con base en un conjunto de asociaciones conocidas previamente, este conjunto es llamado conjunto fundamental o de entrenamiento. De esta manera, los elementos de la matriz M cuyos índices sean ij, se representarán como m_{ij}.

El conjunto fundamental de asociaciones es representado con la siguiente notación:

$$\{(x^\mu, y^\mu)|\mu = 1, 2, \ldots, p\}$$

Dentro de este conjunto, existe un caso particular para el cual cada patrón de entrada es el mismo que el de salida, es decir, $x^\mu = y^\mu \; \forall \mu \epsilon \{1, 2, \ldots, p\}$. Si se cumple la condición mencionada anteriormente, se dice que la memoria es *autoasociativa*; en otro caso se dice que la memoria es *heteroasociativa* [35].

Una de las ventajas más significativas de las memorias asociativas es la capacidad de trabajar con patrones de entrada ruidosos, es decir, versiones alteradas de x, cuya notación es una tilde sobre el patrón de la siguiente manera: $\tilde{x}^k$, que representaría la versión alterada del patrón original x^k.

Los tipos de ruido más comunes para patrones binarios son: aditivo, sustractivo y combinado. Cuando se le presenta un patrón ruidoso $\tilde{x}^\omega$ como entrada a una memoria M, y esta responde con el correspondiente patrón de salida fundamental y^ω, se puede decir que la recuperación es correcta; de esta manera, si se cumple para todos los patrones del conjunto fundamental, se dice que la memoria tiene recuperación correcta [35].

3.2. Memorias Morfológicas Autoasociativas

Las memorias morfológicas son un tipo de memorias asociativas que pueden funcionar como recuperador o clasificador. Utilizan las operaciones de dilatación y erosión en la fase de aprendizaje y, máximos y mínimos para la fase de recuperación [34].

Existen a su vez dos tipos de memorias asociativas morfológicas: las memorias max simbolizadas por la letra M y las memorias *min* simbolizadas con la letra W; en ambos casos pueden funcionar de modo heteroasociativo ó autoasociativo.

La única consideración sobre las memorias autoasociativas, es que el conjunto fundamental tiene la forma $\{(x^\mu, x^\mu)|\mu = 1, 2, \ldots, p\}$.

Debido a que las operaciones utilizadas para la fase de aprendizaje y recuperación involucran productos máximos y mínimos, es preciso definirlos.

Producto máximo

Sea D una matriz de $m \times p$ y H otra matriz de $p \times n$, entonces el producto máximo de D y H se denota por $C = D \nabla H$ y se define de la siguiente manera:

$$C_{ij} = \bigvee_{k=1}^{p} (d_{ik} + h_{kj})$$

Producto mínimo

De manera similar al producto máximo, el producto mínimo se denota por $C = D \triangle H$ y se define de la siguiente manera:

$$C_{ij} = \bigwedge_{k=1}^{p} (d_{ik} + h_{kj})$$

De esta manera, las expresiones anteriores se encuentran relacionados profundamente con las dos operaciones básicas de la morfología matemática: dilatación y erosión.

El conjunto fundamental de las memorias morfológicas es $(x^\mu, y^\mu)|\mu = 1, 2, \ldots, p$, donde

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \epsilon A^n$$

Memorias morfológicas max

Las memorias morfológicas max utilizan como base el producto mínimo para las fases de aprendizaje y recuperación [34]. Sin embargo, el nombre de max proviene del operador máximo utilizado en la fase de aprendizaje, como se muestra a continuación:

Algoritmo 2 Algoritmo de las memorias morfológicas max autoasociativas

---

Fase de aprendizaje La fase de aprendizaje consta de dos etapas:

1. Calcular la matriz $x^\mu \triangle (-x^\mu)^t$, donde $(-x^\mu)^t = (-x_1^\mu, x_2^\mu, \ldots, x_n^\mu)$ y $\mu = 1, 2, \ldots, p$
2. Aplicar el operador máximo $\vee$ a las p matrices calculadas anteriormente

$$M = \bigvee_{\mu=1}^{p} [x^\mu \triangle (-x^\mu)^t]$$

Fase de recuperación La fase de recuperación de las memorias morfológicas max consiste en operar la memoria asociativa M con un patrón de entrada x^ω para obtener el patrón recuperado, de la siguiente forma:

$$x = M \triangle x^\omega$$

Así, la $i - ésima$ componente del vector recuperado x es:

$$x_i = \bigwedge_{j=1}^{n} (m_{ij} + x_j^\omega)$$

3.3. k-Nearest Neighbor (kNN)

Es un de los algoritmos de clasificación más significativo y ampliamente usado en el reconocimiento de patrones debido a su simplicidad y eficacia[55]. Sin embargo, su rendimiento esta influenciado por varios factores, como la métrica de distancia usada para obtener los vecinos más cercanos (Nearest Neighbors) y el tamaño de los patrones [56].

En los últimos años se han desarrollado mejoras a este algoritmo [57], desde la adaptación local de las métricas hasta la forma más óptima de elegir el valor de k.

El algoritmo depende fuertemente de la métrica utilizada, sin embargo, lo más práctico y común es usar la distancia euclidiana. A continuación se describe el algoritmo $k-NN$ general, para cualquier valor de k.

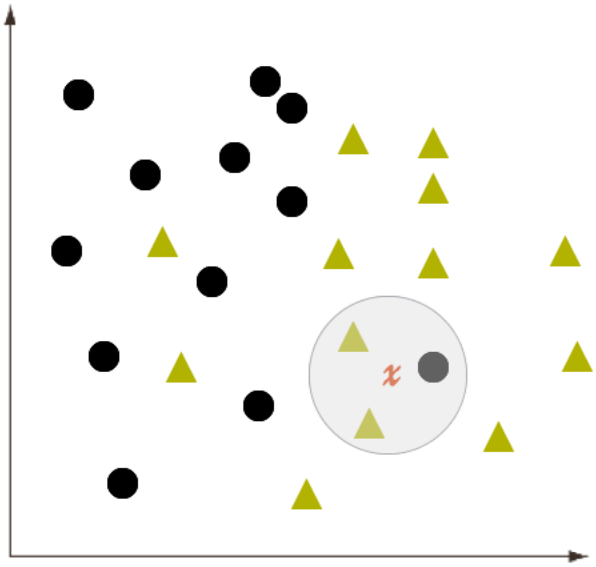
Algoritmo 3 Algoritmo general del $k-NN$

1. Seleccionar la métrica a utilizar
2. Calcular la distancia del patrón x del conjunto de prueba, a cada patrón del conjunto de entrenamiento, utilizando la métrica elegida
3. Ordenar las distancias en orden ascendente
4. Seleccionar los k primeros valores de las distancias ordenadas
5. Usar *majority* en los k valores para asignar la clase al patrón x

Es claro que para el caso donde $k = 1$, no se necesitan ordenar los valores ni tampoco usar la regla de *majority*.

En la figura 3.2 se muestra un ejemplo del clasificador 3-NN, donde el patrón x sería clasificado en la clase de los triángulos (dos triángulos contra un circulo).

Figura 3.2: Ejemplo de clasificación usando $k-NN$



3.4. Distancia Euclidiana

La distancia euclidiana es una norma de Minkowsky de orden 2, y es la más común de las métricas ya que equivale a medir el tamaño del segmento de recta que une a dos puntos. Normalmente se utiliza en geometría analítica y en análisis vectorial.

La distancia euclidiana entre dos vectores x y y de dimensión $n\epsilon\mathbb{Z}^+$, con componentes x_i y y_i respectivamente, donde $i\epsilon\{1, 2, \ldots, n\}$, se denota por $d_2(x, y)$ y se calcula de la siguiente forma:

$$d_2(x, y) = \left[\sum_{i=1}^{n} (x_i - y_i)^2\right]^{\frac{1}{2}}$$

Y cumple con las mismas propiedades que cualquier métrica:

1. $d_2(x, y) \geq 0\ \forall x, y\epsilon\mathbb{Z}^+\ y\ d_2(x, y) = 0 \longleftrightarrow x = y$
2. $d_2(x, y) = d_2(y, x)\ \forall x, y\epsilon\mathbb{Z}^+$
3. $d_2(x, z) \leq d_2(x, y) + d_2(y, z)\ \forall x, y, z\epsilon\mathbb{Z}^+$

3.5. Análisis ROC

El análisis ROC es una métrica del rendimiento para clasificadores binarios, que representa gráficamente a la Razón de Verdaderos Positivos (RVP) contra la Razón de Falsos Positivos (RFP) [58, 59]. Otra forma de representarla es mediante el par ordenado (1-especificidad, sensibilidad).

Las fórmulas para obtener las diferentes razones son las siguientes:

$$RVP = sensibilidad = \frac{VerdaderosPositivos}{VerdaderosPositivos + FalsosNegativos}$$

$$RFP = 1 - especificidad = \frac{FalsosPositivos}{FalsosPositivos + VerdaderosNegativos}$$

El espacio donde se grafican las Curvas ROC, se denomina Espacio ROC y se muestra en la figura 3.3.

Figura 3.3: Espacio ROC

Una valor comprendido sobre la diagonal representa un buen clasificador, llegando a su máximo en la coordenada (0,1) donde no existe ningún falso negativo y ningún falso positivo. Por el contrario, cualquier valor sobre de la linea roja, representa mas bien una clasificación aleatoria. Es importante mencionar que un valor debajo de la linea roja, se considera una clasificación pobre y puede ser invertido para obtener buenos resultados.

En el caso de un clasificador que asigna valores numéricos a cada clase como etiqueta binaria, el rendimiento proporciona un solo punto en el espacio ROC. Para otros clasificadores, tales como el Bayesiano ó una Red Neuronal, que regresan valores de probabilidad se puede definir un umbral de asignación.

Las curvas ROC se generan a partir de múltiples matrices de confusión obtenidas al definir un umbral en el clasificador binario. De esta forma, se genera una matriz para cada valor del umbral y por consiguiente un punto en el Espacio ROC. La curva consiste en dibujar todos los puntos en el espacio ROC comprendido entre (0,0) y (1,1).

El área bajo la curva ROC (AUC) es uno de los indicadores más utilizados para predecir la capacidad del clasificador al momento de discriminar una instancia positiva más alto que una instancia negativa [60].

En el ámbito sanitario se denominan Curvas de Rendimiento Diagnóstico, y se recurre a ellas para elegir entre dos pruebas diagnósticas distintas. Para ello se compara el área bajo la curva ROC de cada prueba y se elige la que tenga la mayor área, de tal forma que, el sistema tenga más probabilidades de clasificar a un paciente enfermo que cualquier otro elegido al azar [59].

3.6. Cómputo en la Nube

El cómputo en la nube hace referencia al hecho de distribuir y consumir recursos a través de una conexión a internet con el fin de proporcionar aplicaciones o servicios. Como resultado se ha hecho popular la virtualización de centros de datos minimizando los tiempos de despliegue de aplicaciones en producción y con ello los costos [61].

Uno de los proveedores de cómputo en la nube más conocido recientemente es Amazon, tanto de sus servicios de cómputo, como de almacenamiento. Los costos que manejan están basados en el tiempo de ejecución, la capacidad de almacenamiento usados, las características de la maquina a utilizar y el tráfico generado por la aplicación.

Un sistema de clasificación es un caso muy práctico para hacer uso de recursos en la nube, y cumple los requisitos necesarios para ser considerado una Aplicación Web. En este contexto es importante definir una arquitectura que permita realizar un desarrollo ágil y de tipado fuerte para garantizar que los resultados de la clasificación serán siempre consistentes.

IaaS

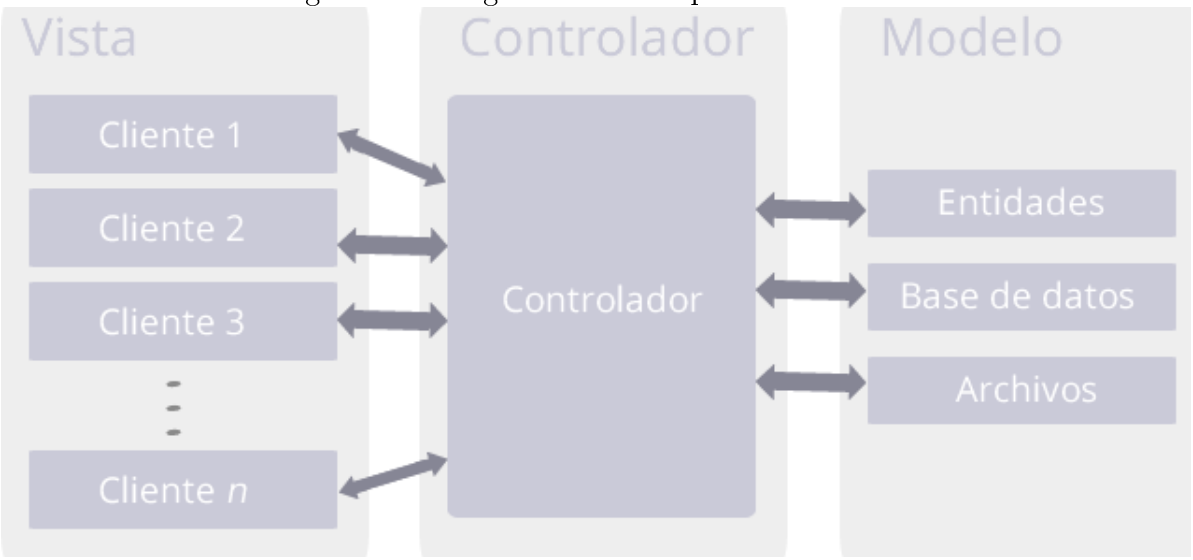
IaaS es un concepto asociado al cómputo en la nube que se refiere al hecho de consumir recursos computacionales (normalmente virtualizados)[62] y montar sobre ellos los servicios que sean necesarios para la ejecución de la aplicación [63, 64]. Amazon ofrece servicios para consumir recursos de IaaS [65], entre los que destacan EC2 (Elastic Cloud 2) y S3 (Storage 3), para los cuales se detallará su uso más adelante, dependiendo de la capa a la que corresponda el servicio.

3.7. Arquitectura MVC

Modelo-Vista-Controlador (MVC) es una arquitectura de software que separa los datos de una aplicación, de la interfaz de usuario (UI) y la lógica de control, en tres componentes distintos. Esta arquitectura se ve frecuentemente en aplicaciones web, donde la vista representa las páginas HTML, las entidades del dominio como el modelo, y el controlador es la lógica de negocio.

Normalmente la interacción entre los tres componentes, se da de la siguiente forma: múltiples vistas acceden al controlador para ejecutar operaciones sobre las entidades del modelo. La figura 3.4 muestra el diagrama de interacción entre capas del modelo MVC.

Figura 3.4: Diagrama de la arquitectura MVC



Aunque MVC fue pensado originalmente para ser usado sobre aplicaciones de escritorio, en la actualidad se han desarrollado un conjunto de frameworks que adaptan esta arquitectura hacia aplicaciones web. Algunos de los frameworks *open-source* que implementan esta tecnología se encuentran:

- SpringMVC
- JSF
- Struts

3.8. Bases de datos NoSQL

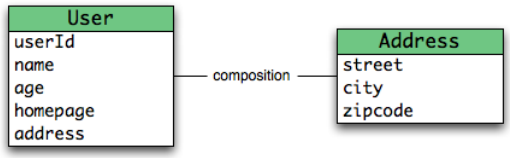
Como su nombre lo indica, NoSQL es una familia de Sistemas Manejadores de Bases de Datos (DBMS) que no usan el lenguaje SQL, y muchas de las veces no implican el uso del modelo relacional. El almacenamiento también es diferente ya que no utiliza esquemas fijos, y permiten un escalamiento horizontal. Este tipo de bases de datos están optimizadas para operaciones de lectura y agregación de datos, y son capaces de trabajar con grandes cantidades de datos, haciéndolas ideales para resolver ciertos tipos de problemas [66].

Aunque existen diferentes tipos de bases de datos NoSQL, la forma más común es la que permite almacenar los datos mediante el modelo de *clave-valor* codificados en algún formato estándar como XML o JSON. Este modelo permite almacenar entidades con diferente número de atributos y diferentes tipos de datos.

Un ejemplo práctico para conocer las bases de datos NoSQL de clave-valor se presente a continuación:

Ejemplo 2. Ejemplo de conversión de una base de datos relacional a una base de datos NoSQL clave-valor

Suponga que se tiene el siguiente diagrama Entidad-Relación:



Dentro de una base NoSQL quedaría en formato JSON de la siguiente manera:

```
{
        useId: 1,
        name: "sergio",
        age: 24,
        homepage: "www.google.com",
        address.street: "constituyentes",
        addrees.city: "Mexico",
        address.zipcode: 54942
}
```

Una de las diferencias más notables de las bases de datos NoSQL, es que **no** garantizan completamente ACID(Atomicidad, Coherencia, Aislamiento y Durabilidad). Sin embargo, esto es intercambiado por la alta velocidad en lectura de datos.

En resumen, este tipo de DBMS están orientadas a problemas muy específicos, donde se cumplan los siguientes aspectos:

- Se requiera trabajar con grandes cantidades de datos
- Se requiera un acceso veloz a los datos
- Se trabajen tablas con columnas variables
- No se realicen operaciones de escritura y eliminación frecuentes
- No se requieran operaciones transaccionales

Capítulo 4

4. Solución Propuesta

En el presente capítulo se detalla la arquitectura y algoritmos utilizados para la verificación de síndromes mediante un sistema web interactivo. En la primera sección se describen las consideraciones sobre el modelo de clasificación utilizado, así como los aportes y modificaciones. En las siguientes secciones se abordan los temas referentes a las tecnologías utilizadas para la implementación del sistema y su despliegue sobre la plataforma de cómputo en la nube de Amazon.

4.1. Algoritmo de Clasificación

El algoritmo utilizado para la clasificación (y por consiguiente verificación) de síndromes es la Memoria Morfológica Autoasociativa *max*. Las consideraciones que llevaron a la elección de dicho algoritmo residen en el hecho de su tolerancia al ruido aditivo.

Por un lado, ya que el algoritmo de la memoria morfológica autoasociativa max permite el uso de valores reales como componentes de los vectores de entrada y salida, no es necesario realizar un pre procesamiento de los bancos de datos. Sin embargo, debido a que dicha memoria funciona mas bien como un recuperador, es necesario realizar ciertas adecuaciones al mismo que permitan comportarse como clasificador.

De esta forma, el algoritmo para la clasificación de síndromes se comporta de manera idéntica a una MAM max en la fase de aprendizaje. Aunque para la fase de recuperación no es tan sencillo.

En esta última fase se introduce el algoritmo kNN para la clasificación de los patrones recuperados de la memoria asociativa, logrando así aprovechar las ventajas de las MAM referentes al ruido y la eficacia del clasificador kNN. Para ello considere el siguiente par ejemplos:

Ejemplo 3. Sea el conjunto fundamental integrado por los patrones:

$$x^1 = \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix},\ x^2 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\ x^3 = \begin{pmatrix} 0 \\ -4 \\ 2 \end{pmatrix} \longrightarrow Clase\,1$$

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

Clasificar el patrón ruidoso $\tilde{x}^2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$ perteneciente a la $Clase1$.

Utilizando el kNN de manera convencional con $k = 3$ se obtienen las siguientes distancias:

Tabla 4.1: Distancias calculadas para el Ejemplo 2 usando el patrón original

	distancia euclidiana
$d(x^1, \tilde{x}^2)$	2
$d(x^2, \tilde{x}^2)$	1
$d(x^3, \tilde{x}^2)$	5.8310
$d(x^4, \tilde{x}^2)$	1.7321
$d(x^5, \tilde{x}^2)$	1
$d(x^6, \tilde{x}^2)$	2.4495

Como se puede observar, $d(x^2, y),\ d(x^4, y),\ d(x^5, y)$ son las distancias menores, lo cual nos indica que el patrón pertenece a la Clase 2, ya que tanto x^4 como x^5 pertenecen a dicha clase.

Ahora, utilizando el Algoritmo 2, se obtiene la siguiente memoria asociativa M para la fase de aprendizaje:

$$M = \begin{pmatrix} 0 & 4 & 1 \\ 1 & 0 & 1 \\ 2 & 6 & 0 \end{pmatrix}$$

De igual forma para la fase de recuperación, se obtiene el siguiente resultado:

$$x^2 = M \triangle \tilde{x}^2 = \begin{pmatrix} 0 & 4 & 1 \\ 1 & 0 & 1 \\ 2 & 6 & 0 \end{pmatrix} \triangle \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Permitiendo recuperar de manera correcta el patrón x^2.

Finalmente al utilizar el 3NN se obtienen las siguientes distancias:

Tabla 4.2: Distancias calculadas para el Ejemplo 2 usando el patrón recuperado

	distancia euclidiana
$d(x^1, x^2)$	1
$d(x^2, x^2)$	0
$d(x^3, x^2)$	5
$d(x^4, x^2)$	1.4142
$d(x^5, x^2)$	1.4142
$d(x^6, x^2)$	2.2361

Lo cual nos indica que $d(x^1, x^2),\ d(x^2, x^2),\ d(x^4, x^2)$ son las menores, clasificando finalmente al patrón $\tilde{x}^2$ en la $Clase1$, ya que tanto x^1 como x^2 pertenecen esta clase.

Para el ejemplo anterior es importante observar que el patrón $\tilde{x}^2$ tiene ruido aditivo, ya que es donde la MAM max tiene mejor desempeño.

Ejemplo 4. Sea el conjunto fundamental integrado por los patrones:

$$x^1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix},\ x^2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix} \longrightarrow Clase\ 1$$

$$x^3 = \begin{pmatrix} 5.0 \\ 3.6 \\ 4.9 \\ 2.0 \end{pmatrix},\ x^4 = \begin{pmatrix} 6.5 \\ 3.0 \\ 5.2 \\ 2.0 \end{pmatrix} \longrightarrow Clase\ 2$$

Clasificar el patrón $\tilde{x}^2 = \begin{pmatrix} 1.7 \\ 3.0 \\ 5.2 \\ 0.3 \end{pmatrix}$ perteneciente a la $Clase\,1$.

Utilizando el kNN de manera convencional con $k = 3$ se obtienen las siguientes distancias:

Tabla 4.3: Distancias calculadas para el Ejemplo 2 usando el patrón original

	distancia euclidiana
$d(x^1, \tilde{x}^2)$	5.12
$d(x^2, \tilde{x}^2)$	4.96
$d(x^3, \tilde{x}^2)$	3.77
$d(x^4, \tilde{x}^2)$	5.09

Como se puede observar, $d(x^3, \tilde{x}^2),\ d(x^2, \tilde{x}^2),\ d(x^4, \tilde{x}^2)$ son las distancias menores, lo cual nos indica que el patrón pertenece a la $Clase 2$, ya que tanto x^3 como x^4 pertenecen a dicha clase.

Ahora, utilizando el Algoritmo 2, se obtiene la siguiente memoria asociativa M para la fase de aprendizaje:

$$M = \begin{pmatrix} 0.0 & 3.5 & 3.7 & 4.9 \\ -1.4 & 0.0 & 2.1 & 3.3 \\ -0.1 & 2.2 & 0.0 & 3.2 \\ -3.0 & -1.0 & -1.2 & 0.0 \end{pmatrix}$$

De igual forma para la fase de recuperación, se obtiene el siguiente resultado:

$$z^2 = M \triangle \tilde{x}^2 = \begin{pmatrix} 0.0 & 3.5 & 3.7 & 4.9 \\ -1.4 & 0.0 & 2.1 & 3.3 \\ -0.1 & 2.2 & 0.0 & 3.2 \\ -3.0 & -1.0 & -1.2 & 0.0 \end{pmatrix} \triangle \begin{pmatrix} 1.7 \\ 3.0 \\ 5.2 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 1.7 \\ 0.3 \\ 1.6 \\ -1.3 \end{pmatrix}$$

En este caso, el patrón recuperado no corresponde a ninguno del conjunto fundamental, por eso se denota como z^2.

Finalmente al utilizar el 3NN se obtienen las siguientes distancias:

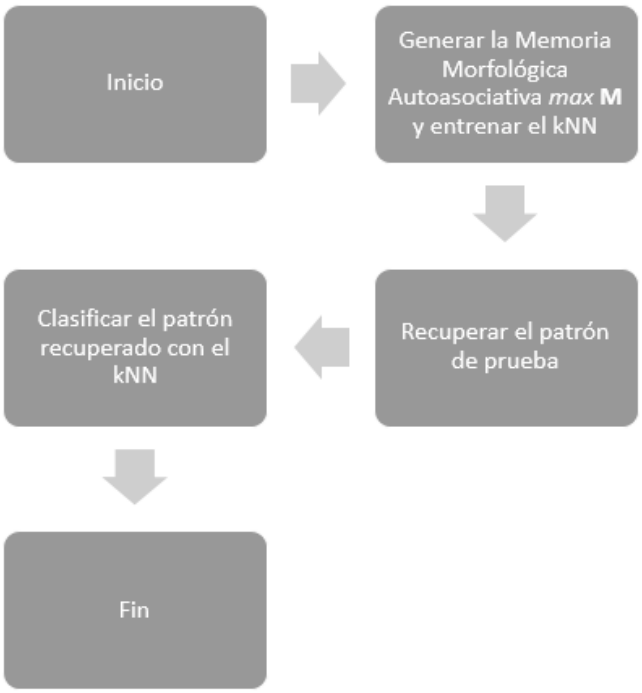
Tabla 4.4: Distancias calculadas para el Ejemplo 2 usando el patrón recuperado

	distancia euclidiana
$d(x^1, z^2)$	4.90
$d(x^2, z^2)$	4.45
$d(x^3, z^2)$	6.60
$d(x^4, z^2)$	7.36

Lo cual nos indica que $d(x^1, z^2),\ d(x^2, z^2),\ d(x^3, z^2)$ son las distancias menores, clasificando finalmente al patrón $\tilde{x}^2$ en la $Clase\,1$, ya que tanto x^1 como x^2 pertenecen a esta clase.

De esta forma, el diagrama de flujo para el algoritmo de verificación de síndromes (en adelante conocido como mNN) se muestra en la figura 4.1:

Figura 4.1: Diagrama de flujo del algoritmo de clasificación *mNN*



4.2. Aplicación Web

Con el fin de desarrollar un sistema interactivo que mantenga una comunicación directa entre el cliente y el servidor, el sistema hace uso de la arquitectura MVC (Model-View-Controller) sobre la plataforma de desarrollo J2EE, permitiendo distribuir diferentes tecnologías de este estándar en el desarrollo de una aplicación web basada en capas.

Dada la flexibilidad del patrón MVC, es bastante práctico implementarlo sobre una arquitectura de cómputo en la nube utilizando una IaaS (Infrastructure as a Service). Las ventajas de implementar una aplicación web sobre cómputo en la nube son muchas, entre ellas destacan:

- Alta disponibilidad - Gracias a la virtualización, el sistema no se encuentra físicamente en algún lugar específico y puede ser migrado en caso de fallos.
- Consumo bajo demanda - Solo se paga lo que se consume, en otro caso la plataforma es compartida por otros sistemas, mejorando así la eficiencia en consumo energético.
- Acceso inmediato - Útil en proyectos donde no se tiene un *datacenter* y se requiere un acceso inmediato a los recursos de cómputo.
- Escalable - El sistema puede se escalado vertical y horizontalmente de manera transparente, aumentando virtualmente el numero de núcleos y memoria RAM.

Para un sistema de reconocimiento de patrones, la tolerancia a fallos es importante, sobre todo para pruebas diagnósticas. La alta disponibilidad también es vital para cumplir los objetivos de este trabajo de tesis. Y debido al consumo elevado de recursos, se requiere que la aplicación pueda ser escalada en cualquier momento sin tener que adecuar el algoritmo.

Con base en las ventajas del cómputo en la nube, a continuación se describen cada una de las capas del modelo MVC implementadas sobre la plataforma de Amazon AWS.

Modelo

Representa las instancias que se almacenan en la base de datos NoSQL, permitiendo acceder a las mismas de una forma eficiente para el algoritmo.

En el sistema propuesto, con base en las prestaciones de la arquitectura *cloud computing* de Amazon y a través de su servicio EC2, se instaló MongoDB como motor de base de datos NoSQL dentro de una microinstancia.

MongoDB es sistema base de datos NoSQL, open-source y escrito en el lenguaje de programación C++ [67]. Algunas de las ventajas de MongoDB son:

- Almacenamiento orientado a documentos » Para esquemas dinámicos con estilo JSON.
- Fácil escalamiento » Escalamiento horizontal transparente.
- Map/Reduce » Listo para trabajar con operaciones de agregación y procesamiento de datos.

Debido a que cada banco de datos tiene diferentes tipos de patrones, cada uno de ellos con diferentes rasgos; un DBMS NoSQL es el caso ideal para usarlo, debido a no se realizan operaciones frecuentes de eliminación, y que el sistema de clasificación requiere una baja latencia.

El acceso al modelo se realiza a través de una capa de servicios (*service layer*) que es implementada para conectarse a MongoDB desde java. El patrón de diseño DAO (Data Access Object) permite abstraer operaciones de consulta y actualización de datos, para que se realicen de manera transparente al controlador.

En el caso del sistema propuesto, se almacena el banco de datos con algunas propiedades y todas sus instancias. Un ejemplo de como se persiste un banco de datos en MongoDB se muestra a continuación:

```
{                                                                    1
        "_id" : ObjectId("52772cfe45c1f14e8b7732fa"),                2
        "name" : "iris.arff",                                        3
        "instances" : [                                              4
                {                                                    5
                        "class" : 0,                                 6
                        "features" : [                               7
                                5.1,                                 8
                                3.5,                                 9
                                1.4,                                 10
                                0.2                                  11
```

```
                    ]
                },
                ...
            ],
            "features" : [
                "sepallength",
                "sepalwidth",
                        "petallength",
                    "petalwidth"
            ],
            "classes" : {
                "Iris-virginica" : 2,
                "Iris-setosa" : 0,
                "Iris-versicolor" : 1
            }
}
```

Vista

Básicamente la vista representa la interfaz de la aplicación, cuyos componentes permiten interactuar con el usuario de una forma que le resulte accesible. PrimeFaces de JBoss es una librería para el desarrollo de interfaces web, cuenta un conjunto que componentes que facilitan el diseño de la capa de vista (*view layer*) e implementa de manera nativa y transparente la tecnología AJAX.

El sistema de verificación de síndromes, PrimeFaces permite sincronizar la ejecución del algoritmo y renderizar en el cliente solo las porciones de código HTML referentes al resultado de la clasificación y rendimiento obtenidos. Los componentes utilizados para subir los bancos de datos y mostrar las tablas de resultados se enlazan directamente con el controlador facilitando la tarea escritura y lectura en el servicio de Amazon S3.

Controlador

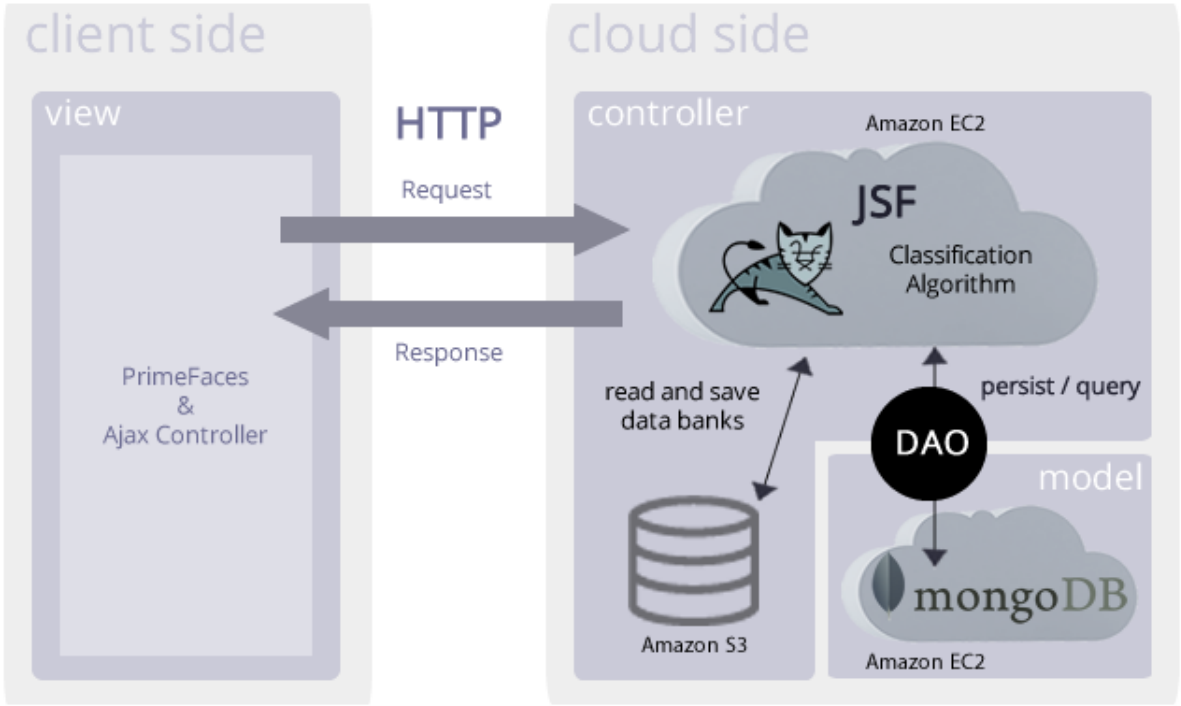
Conocido comúnmente por alojar la mayor parte de la lógica de negocio, JSF es un framework basado completamente en MVC, nos permite abstraer las acciones realizadas por el usuario dentro de las vistas para poder ejecutar los procesos que requieren acceso a los datos del Modelo y a los algoritmos alojados en la arquitectura de Amazon EC2.

En el caso del sistema de clasificación, dentro de esta capa residen los algoritmos de reconocimiento de patrones (Memorias Morfológicas Auto Asociativas *max* y *kNN*). Sin embargo, estos algoritmos no interactúan directamente con la vista, para ello se utiliza un componente propio de JSF conocido como *ManagedBean*. La función de esta clase es capturar los eventos de los componentes de la vista, específicamente cuando el usuario sube un banco de datos o inicia la clasificación, y proporcionar un método que permita la ejecución del algoritmo.

Adicionalmente la clase *ManagedBean* de JSF permite mantener, si es necesario, los datos de la operación en memoria durante la sesión HTTP, para poder configurar el sistema y ejecutar la clasificación de manera independiente.

En la figura 4.2 se muestra la arquitectura general de la aplicación web implementada sobre Amazon AWS, haciendo uso de los servicios EC2 para la virtualización del Servidor Apache Tomcat y MongoDB, y el servicio S3 de Amazon para el almacenamiento masivo de los bancos de datos usados en el entrenamiento y clasificación. Así mismo, se puede ver que la comunicación HTTP se realiza a través del Controlador AJAX de PrimeFaces hacia la Capa de Negocio en el servidor basada en JSF, donde se encuentra el algoritmo de reconocimiento de patrones.

Figura 4.2: Arquitectura general de la aplicación web



4.3. Flujo del Proceso

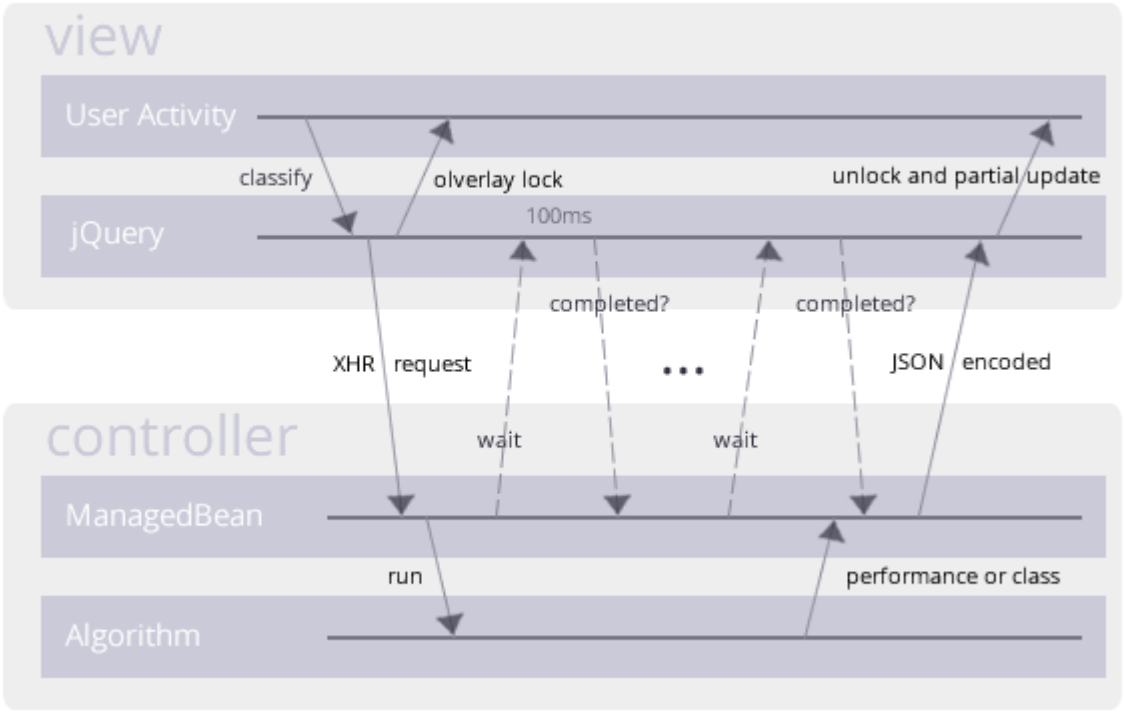
El proceso de clasificación inicia cuando el cliente sube un banco de datos al sistema por medio del componente *fileupload* de PrimeFaces, en ese momento el cliente envía al servidor por medio de AJAX el archivo, y este lo amacena dentro del servicio S3 de Amazon con la finalidad de poder utilizarlo en futuras predicciones. Una vez que se ha realizado el *file uploading*, se permite configurar algunos parámetros del clasificador, la clase que representa un verdadero positivo y la clase que representa un falso negativo.

El procedimiento para adjuntar el archivo de prueba es similar al del banco de entrenamiento, y una vez que se proceda a clasificar, el sistema leerá los datos en formato ARFF (Attribute-Relation File Format) de los archivos de entrenamiento y prueba, y los convertirá en entidades del modelo para *poder persistirlos* en la base de datos.

Cabe mencionar que las acciones basadas en AJAX se ejecutan de manera asíncrona, pero de alguna forma es necesario bloquear la ejecución en la vista para mantener la consistencia de los resultados que se le mostrarán al usuario, forzando una comunicación sincronizada entre el navegador y el servidor.

En la figura 4.3, se muestra el diagrama de tiempos que sigue la comunicación cuando el cliente pide ejecutar la clasificación de patrones. El flujo comienza cuando se ejecuta asíncronamente la acción de clasificar, por medio del objeto XHR (XmlHTTPRequest) hacia el servidor donde es necesario bloquear la vista con una superposición de algún elemento y recurrir a la técnica de *pooling* cada 100ms para verificar cuando esté listo el resultado, en cuyo caso, el servidor enviará el resultado de la clasificación en formato JSON (JavaScript Object Notation) para minimizar el tamaño de los datos. De esta forma se envía la señal de desbloqueo al cliente, para actualizar solo la parte de la página HTML que muestra el resultado.

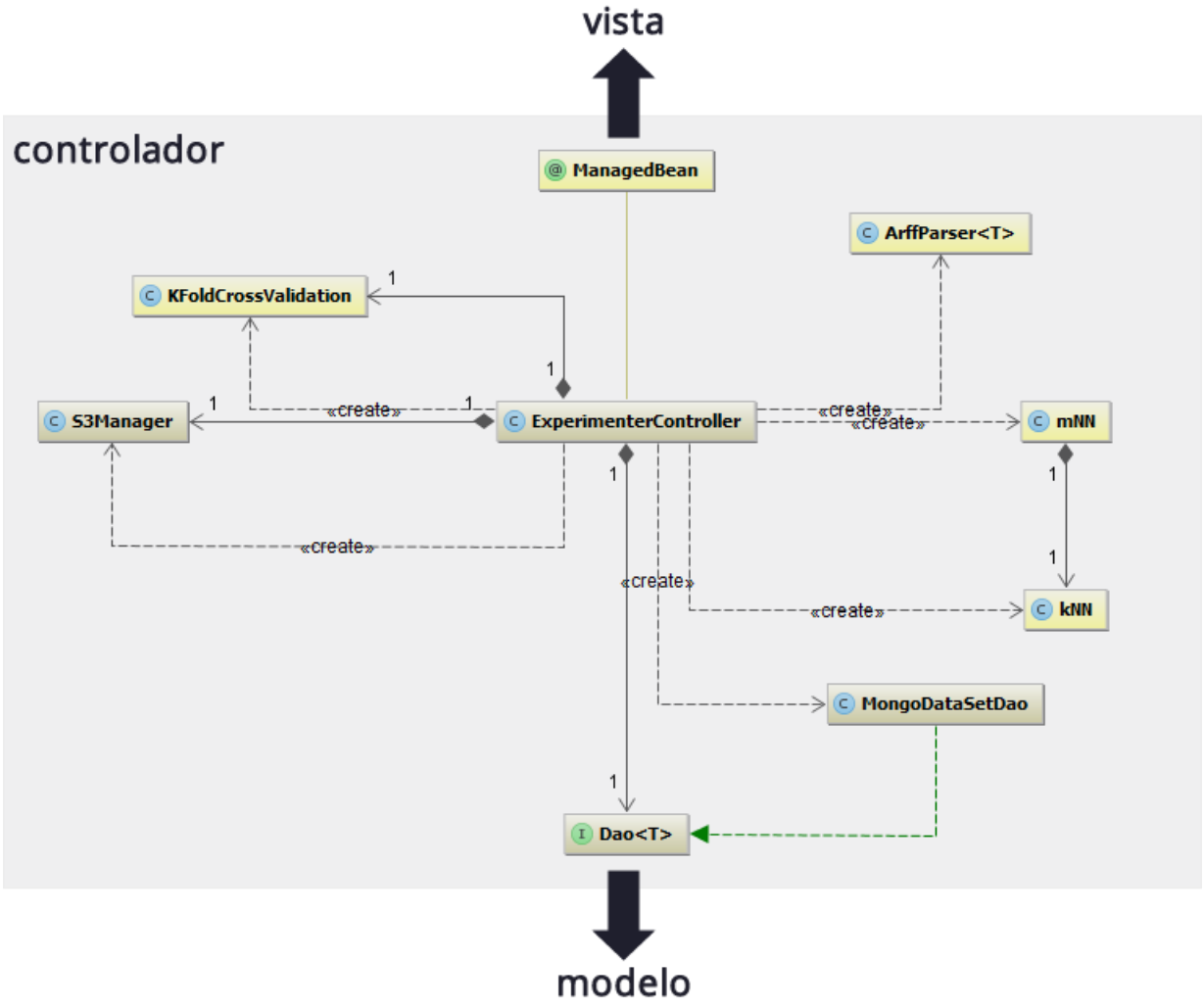
Figura 4.3: Diagrama de tiempos para la sincronización de la respuesta en la clasificación de patrones



4.4. Diseño

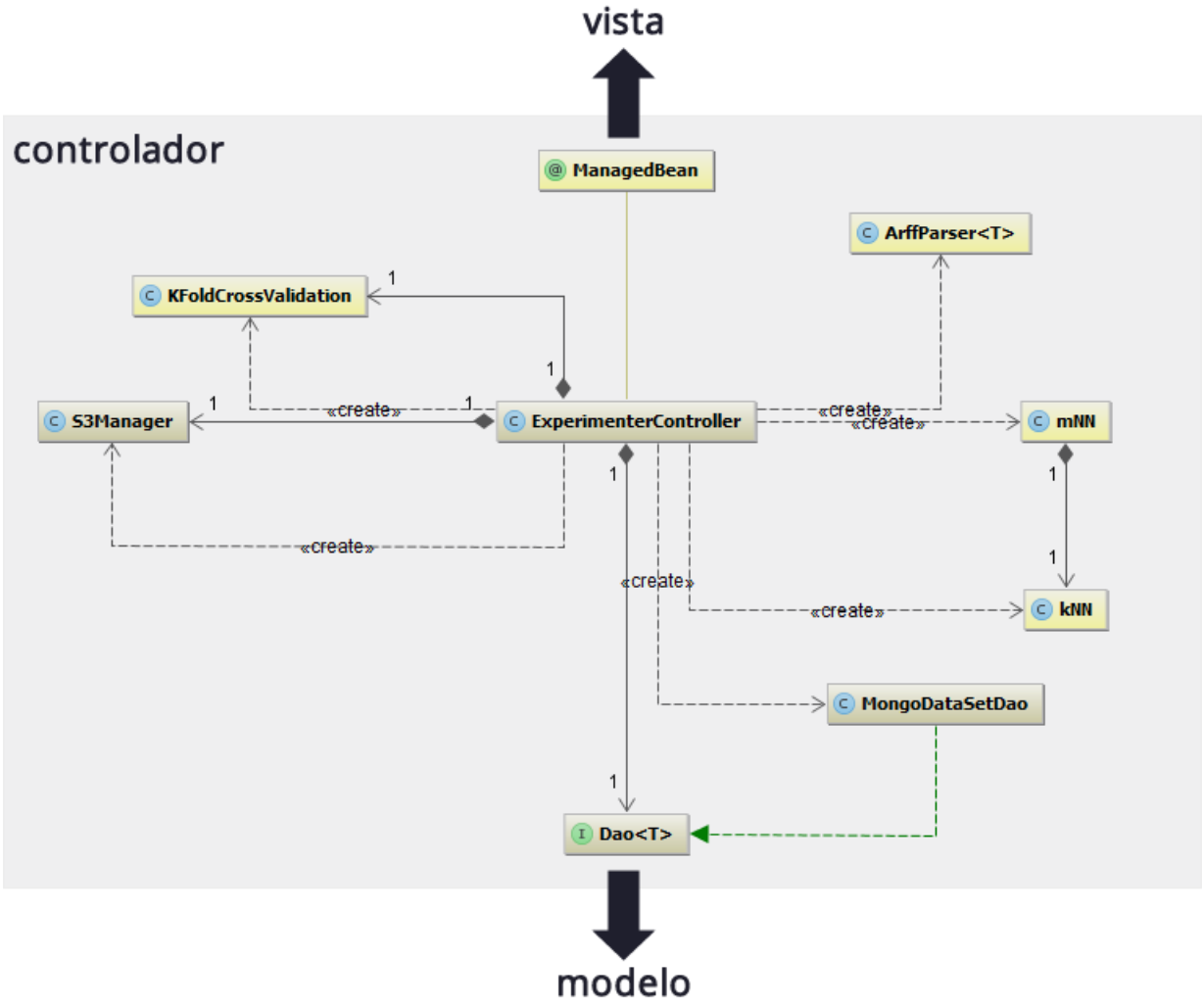
Como se ha mencionado anteriormente, la mayor parte de la lógica de negocio está relacionada con la capa del Controlador. En el diagrama de clases UML de la figura 4.4 se puede apreciar los objetos que interactúan para la validación de los algoritmos de clasificación, y cuales de ellos enlazan con las otras dos capas de patrón MVC.

Figura 4.4: Diagrama de clases UML del Controlador de la aplicación web para la validación del clasificador



También es importante mostrar el diagrama de clases para la clasificación de síndromes, es decir, la parte más importante del sistema. En la figura 4.5 se visualiza los objetos que interactúan para la clasificación de patrones, y cuales de ellos enlazan con la Vista y el Modelo.

Figura 4.5: Diagrama de clases UML del Controlador de la aplicación web para la clasificación de síndromes



En ambos casos, la vista es manejada por medio de la *interface* ManagedBean de JSF, y por lo tanto todos los componentes de la vista se encuentran ligados a ella.

Cuando se realiza la validación del algoritmo de clasificación, ExperimentController hace uso de la clase KFoldCrossValidation para obtener las métricas de rendimiento. Por otro lado, si se pide clasificar patrones, se entrena con todo el banco de datos y se devuelve la clase resultante de la clasificación.

La función que realizan las otras clases mostradas en los diagramas son las siguientes:

- **ArffParser:** Lee un banco de datos ARFF y lo convierte en un arreglo de ins-

tancias

- **S3Manager:** Es la encargada de las operaciones de lectura y escritura en la plataforma de almacenamiento S3 de Amazon.
- **MongoDataSetDao:** Implementa los métodos de acceso a los patrones (lectura y persistencia) en la base de datos MongoDB, a través de la API de Java para establecer la conexión.

Capítulo 5

5. Resultados y Discusión

5.1. Bancos de Datos

El contenido de esta sección esta fuertemente basado en los bancos de prueba utilizados para tareas de clasificación, del repositorio de Machine Learning del Departamento de Información y Ciencias de la Computación de la University of California, y fueron obtenidos de http://archive.ics.uci.edu/ml/.

Con la finalidad de realizar comparaciones sobre otros clasificadores que no se implementaron en el sistema propuesto, se uso los mismos bancos de datos que se le proporcionan al sistema web con algunos algoritmos ya existentes en el software de minería de datos llamado WEKA [68].

5.2. Entorno de Prueba

El entorno donde se implementó el sistema web interactivo presentado en esta tesis, es una micro-instancia del servicio de Amazon EC2 con las siguientes características:

- Tipo: t1.micro
- Sistema Operativo: Ubuntu Amazon Custom
- Almacenamiento: 5GB
- vCPU: 1
- Memoria RAM: 615MB

Los algoritmos implementados en WEKA se ejecutaron con las configuraciones por *default* y en una computadora portátil con las siguientes características:

- Modelo: Dell XPS LX502

- Sistema Operativo: Windows 7 x64
- Software: Weka 3.7.9-SNAPSHOT
- Procesador: Intel i5-2410M @ 2.30 GHZ
- Memoria RAM: 4GB
- Almacenamiento: 500GB

Así mismo, se utilizó el método de validación K-Fold Cross-Validation con *k=10*, debido a que este valor es utilizado en gran parte de la literatura y artículos publicados en revistas JCR como un estándar en la medida de rendimiento para sistemas de clasificación.

5.3. Iris-Plant

La *Iris-Plant* Database es un banco de datos de referencia para cualquier clasificador, que permite medir su desempeño con respecto a otros clasificadores.

Es uno de los bancos más prácticos para pruebas debido a su tamaño, balanceo de clases y valores que maneja en los rasgos de cada patrón. El conjunto de datos contiene 3 clases de 50 instancias cada una, donde cada clase indica un tipo de planta (Iris Setosa, Iris Veriscolor e Iris Virginica). Cada patrón a su vez tiene 4 atributos, más 1 que representa la clase. La tabla 5.1 muestra los rasgos presentes en el banco de datos *Iris-Plant*.

Tabla 5.1: Información de rasgos del banco de datos Iris Plant

Rasgo	Unidad	Descripción
Sepal Lenght	cm	Largo del sépalo
Sepal Width	cm	Ancho del sépalo
Petal Legth	cm	Largo del pétalo
Petal Width	cm	Ancho del pétalo

De este banco de datos se saben muchos detalles de su estructura, por ejemplo, que la Iris Setosa es linealmente separable de las otras dos clases, la Iris Versicolor y la Iris Virginica, pero que éstas a su vez no son linealmente separables entre sí.

Para las pruebas, se utilizó el método de validación *K-Fold Cross-Validation* con *K=10*, en un set de 10 pruebas cada uno, formando un total de 100 pruebas.

En la tabla 5.2 se presentan los resultados de la clasificación de patrones realizada para el banco de datos Iris-Plant. Los resultados solo corresponden solamente al rendimiento ya que no se realizó ningún análisis de curvas ROC debido al poco interés de un análisis diagnóstico sobre este banco de datos.

Tabla 5.2: Rendimientos de Clasificación para el banco de datos Iris Plant

	mNN(k=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Rendimiento (%)	96.0000	96.0000	92.6667	96.0000	92.6667	92.0000

5.4. Parkinsons

Este banco de datos fue creado por Max Little de la Universidad de California, en colaboración con el Centro Nacional para la Voz y el Habla. Esta compuesto por un conjunto de mediciones biomédicas basadas en la voz de 31 personas, de las cuales 23 padecen la enfermedad de Parkinson.

El conjunto de datos consta de 195 instancias distribuidas en 2 clases (147 con Parkinson y 48 sin Parkinson). Los rasgos de cada instancia se muestran en la tabla 5.3.

Tabla 5.3: Información de rasgos del banco de datos Parkinson

Rasgo	Unidad	Descripción
Nombre	ASCII	El nombre del paciente
MDVP:Fo	Hz	Promedio de la frecuencia fundamental vocal
MDVP:Fhi	Hz	Máximo de la frecuencia fundamental vocal
MDVP:Flo	Hz	Mínimo de la frecuencia fundamental vocal
MDVP:Jitter	%	Múltiples medidas de la variación en la frecuencia fundamental
MDVP:Jitter(Abs)	Número	
MDVP:RAP	Número	
MDVP:PPQ	Número	
Jitter:DDP	Número	
MDVP:Shimmer	Número	Múltiples medidas de la variación en amplitud
MDVP:Shimmer(dB)	Número	
Shimmer:APQ3	Número	
Shimmer:APQ5	Número	
MDVP:APQ	Número	
Shimmer:DDA	Número	
NHR	Número	Dos medidas de la proporción de ruido en las componentes tonales de la voz
HNR	Número	
status	Booleano	El estado de salud del sujeto. Parkinson (1), Sin Parkinson (0)
RPDE	Número	Dos medidas de complejidad dinámicos no lineales
D2	Número	
DFA	Número	Exponente de escalamiento fractal de la señal
spread1	Número	Tres medidas no-lineales de la variación de la frecuencia fundamental
spread2	Número	
PPE	Número	

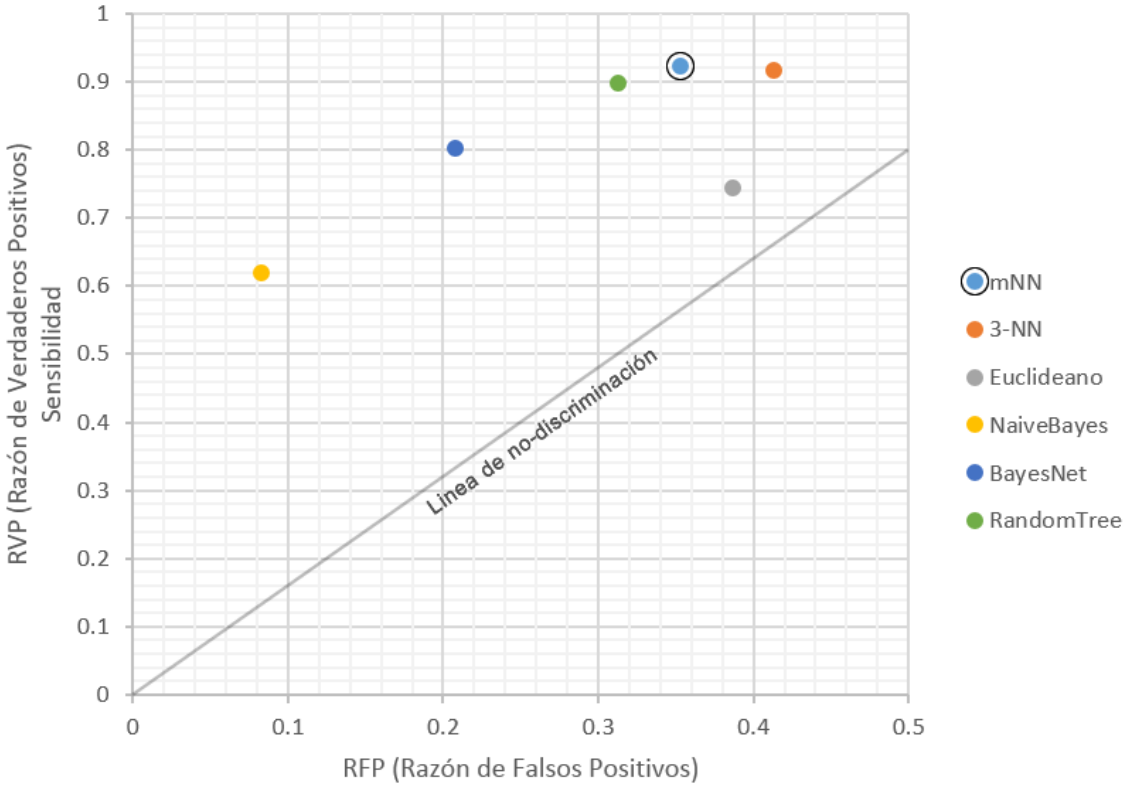
En la tabla 5.4 se presentan los resultados de la clasificación realizada para el banco de datos Parkinsons. Los resultados corresponden al rendimiento y análisis ROC del algoritmo propuesto, junto con otros algoritmos ejecutados en WEKA. Para el análisis ROC se eligió como instancia positiva aquella que tiene un *status* de 1, es decir, que tiene la enfermedad de Parkinson.

Tabla 5.4: Rendimientos de Clasificación y Análisis ROC el banco de datos Parkinsons

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Rendimiento (%)	85.5000	83.5000	71.3333	69.2308	80.0000	84.6154
RVP	0.9233	0.9167	0.7433	0.6190	0.8030	0.8980
RFP	0.3533	0.4133	0.3867	0.0830	0.2080	0.3130

La figura 5.1 muestra el punto en el espacio ROC de los clasificadores mostrados en la tabla anterior.

Figura 5.1: Espacio ROC para el banco de datos Parkinsons



Tomando como base el punto de clasificación correcta (0, 1), la tabla 5.5 muestra las distancias euclidianas obtenidas a partir de los diferentes clasificadores; entre menor sea la distancia representa una mejor prueba diagnóstica:

Tabla 5.5: Distancias hacia la clasificación correcta del banco Parkinsons

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Distancia	0.3615	0.4216	0.4641	0.3899	0.2865	0.3292

5.5. Vertebral Column

El conjunto de datos fue construido por el Dr. Henrique da Mota durante una estancia médica en el Grupo de Investigación Aplicada en Ortopedia del Centro Médico-Quirúrgico de Rehabilitación en Francia.

Para la tarea de clasificación, los datos están organizados en 2 diferentes formas:

1. La primer tarea consiste en clasificar pacientes que pertenezcan a una de las 3 categorías: Normal (100 pacientes), Hernia de disco (60 pacientes) o Espondilolistesis (150 pacientes)

2. Para la segunda tarea, las últimas categorías mencionadas son mezcladas en una sola categoría llamada Anormal (210 pacientes), esto con el fin de separar a los pacientes que están sanos de los que tienen algún problema.

Cada instancia del banco de datos representa un paciente, tomando 6 atributos biomecánicos derivados de la forma y orientación de la pelvis y la espina lumbar, como se muestra en la tabla 5.6.

Tabla 5.6: Información de rasgos del banco de datos Vertebral Column

Rasgo	Unidad	Descripción
1	pelvic incidence	Incidencia pélvica
2	pelvic tilt	Inclinación pélvica
3	lumbar lordosis angle	Ángulo de lordosis lumbar
4	sacral 2ope	Inclinación sacral
4	pelvic radius	Radio pélvico
5	degree spondylolisthesis	Grado de espondilolistesis

Con la finalidad de realizar un análisis de curvas ROC en el banco de datos propuesto, se utilizo el que tiene solo dos categorías, donde Hernia de disco y Espondilolistesis se unen en una sola categoría llamada Anormal.

En la tabla 5.7 se presentan los resultados de la clasificación de patrones realizada para Columna Vertebral. Los resultados corresponden al rendimiento y análisis ROC del algoritmo propuesto, junto con otros algoritmos ejecutados en WEKA. Para el análisis ROC se eligió como instancia positiva aquella que pertenece a la clase Anormal, es decir, que tiene algún problema en la columna.

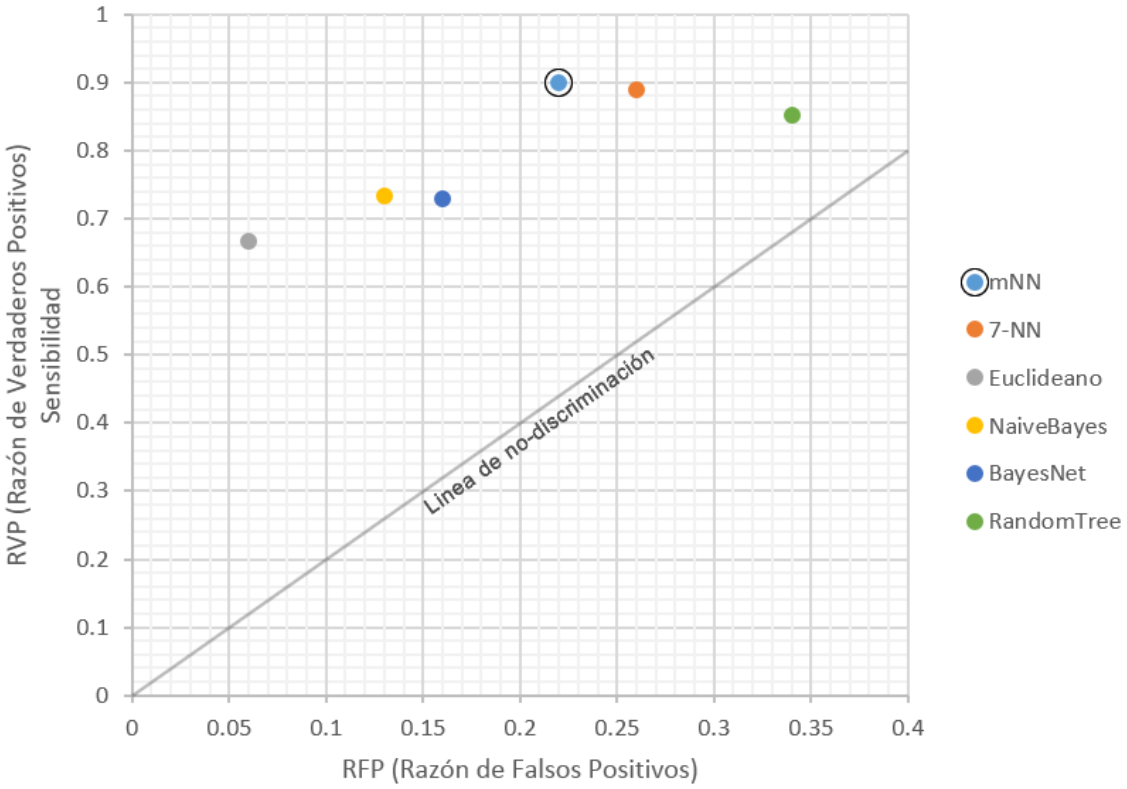
Tabla 5.7: Rendimientos de Clasificación y Análisis ROC para Columna Vertebral

	mNN(m=7)	7-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Rendimiento (%)	85.9032	84.8710	75.4839	77.7419	76.4516	79.0323
RVP	0.9000	0.8900	0.6667	0.7330	0.7290	0.8520
RFP	0.2200	0.2400	0.0600	0.1300	0.1600	0.3400

En este caso, se aprecia una ligera pero importante mejora en el rendimiento del clasificador, usando el 3-NN como base para la comparación. Por otro lado, la clasificación basada en arboles se aproxima mucho al rendimiento obtenido, pero su alta tasa de falsos positivos hace que se descarte como opción en la verificación de algún tipo de síndrome.

Para el análisis ROC, la figura 5.2 muestra el punto en el espacio de los clasificadores mostrados en la tabla anterior.

Figura 5.2: Espacio ROC para el banco de datos Vertebral Column



Tomando como base el punto de clasificación correcta (0, 1), la tabla 5.8 muestra las distancias euclidianas obtenidas a partir de los diferentes clasificadores; entre menor sea la distancia representa una mejor prueba diagnóstica:

Tabla 5.8: Distancias hacia la clasificación correcta del banco Vertebral Column

	mNN(m=7)	7-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Distancia	0.2417	0.2400	0.3387	0.2970	0.3147	0.3708

5.6. Mammographic Masses

La mamografía es el método más efectivo hoy en día para la detección de cáncer de mama, sin embargo, la baja tasa de predicción de valores positivos resulta en aproximadamente un 70 % de biopsias innecesarias con resultados benignos.

Para reducir el número de biopsias de mama, se han desarrollado en los últimos años una serie de sistemas asistidos por computadora para el diagnóstico y toma de decisión ante una lesión sospechosa que requiera de un mamograma o investigación más profunda.

El conjunto de datos de tumores mamográficos puede ser usado para predecir la severidad (benigno o maligno) de un tumor mamográfico con base en atributos llamados BI-RADS y la edad del paciente. Con un total de 961 pacientes identificados mediante mamogramas de campo completo y recolectados en el Instituto de Radiológica de la Universidad de Erlangen-Nuremberg, el banco de datos se distribuye de la siguiente manera: 516 con una detección benigna y 445 con un tumor maligno.

BI-RADS son las siglas del inglés Breast Imaging Report and Database System, una herramienta radiográfica para garantía de calidad durante los reportes e interpretaciones de mamografías

Cada instancia esta asociada a un rango de evaluación BI-RAD que va desde 1 (definitivamente benigno) hasta 5 (alta sugerencia de malignidad). La tabla 5.9, muestra los atributos para las instancias del banco de datos.

Tabla 5.9: Información de rasgos del banco de datos Mammographic Masses

Rasgo	Unidad	Descripción
BI-RADS assessment	Ordinal	1 al 5, no predictivo
Age	Años	La edad del paciente
Shape	Nominal	La forma del tumor, redondo=1, ovalado=2, lobular=3, irregular=4
Margin	Nominal	El margen del tumor, circunscrito=1, microlobulado=2, oscurecido=3, mal-definido=4, espiculado=5
Density	Ordinal	Densidad del tumor, alto=1, iso=2, bajo=3, con grasa=4
Severity	Booleano	Severidad, benigno=0, maligno=1

Debido a que este banco de datos tiene valores perdidos en las instancias, se realizó un pre-procesamiento para completar dichos valores utilizando la media de todos los datos que pertenecen al atributo. Es decir, se obtuvo la media de cada rasgo y se asignó a todos los valores perdidos del mismo atributo en todo el banco de datos. En la tabla 5.10 se muestran los valores obtenidos de la media para cada atributo:

Tabla 5.10: Constantes asignadas (por rasgo) a los valores perdidos del banco de datos Mammographic Masses

Rasgo	Mediana
BI-RADS assessment	4
Age	57
Shape	3
Margin	3
Density	3

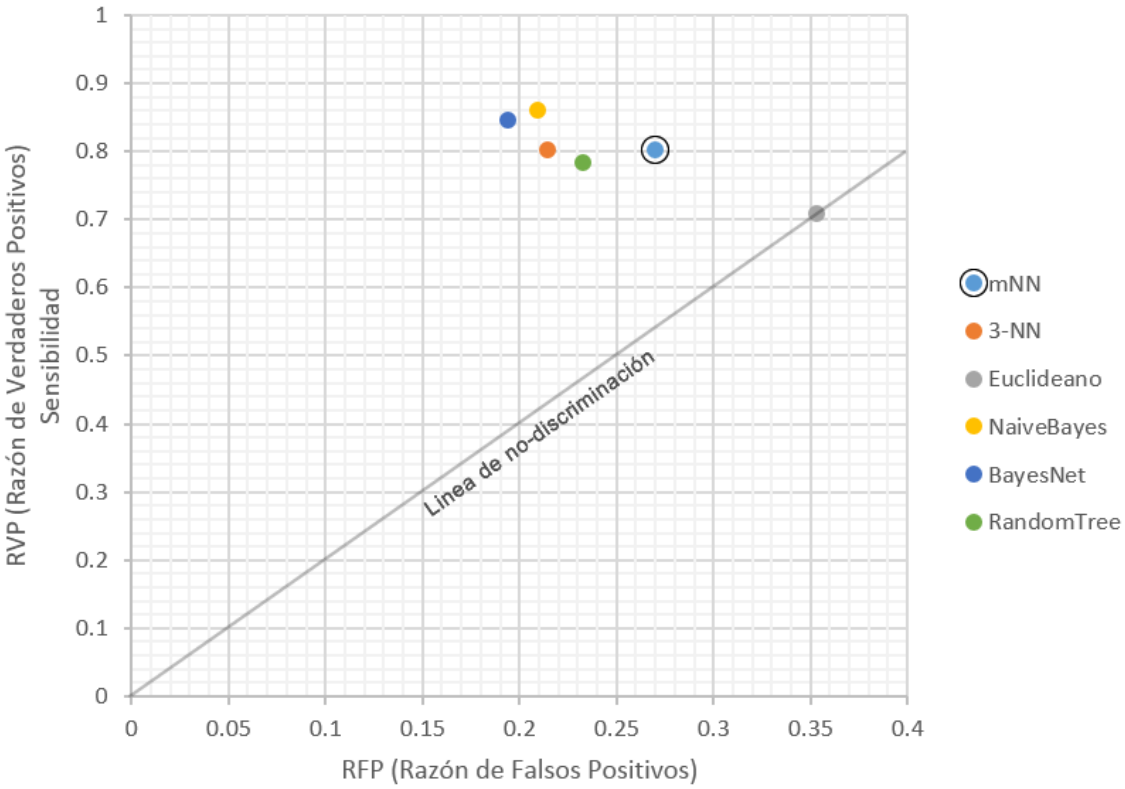
En la tabla 5.11 se presentan los resultados de la clasificación de patrones realizada para tumores mamográficos. Los resultados corresponden al rendimiento y análisis ROC del algoritmo propuesto, junto con otros algoritmos ejecutados en WEKA. Para el análisis ROC se eligió como instancia positiva aquella que representa un tumor maligno, es decir, cuando el paciente requiere de una biopsia mamográfica.

Tabla 5.11: Rendimientos de Clasificación y Análisis ROC para el banco de datos Mammographic Masses

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Rendimiento (%)	76.3871	79.3896	67.5691	82.3101	82.5182	77.5234
RVP	0.8028	0.8031	0.7094	0.8610	0.8470	0.7840
RFP	0.2696	0.2141	0.3534	0.209	0.1940	0.2330

La figura 5.3 muestra el punto en el espacio ROC de los clasificadores mostrados en la tabla anterior.

Figura 5.3: Espacio ROC para el banco de datos Mammographic Masses



Tomando como base el punto de clasificación correcta (0, 1), la tabla 5.12 muestra las distancias euclidianas obtenidas a partir de los diferentes clasificadores; entre menor sea la distancia representa una mejor prueba diagnóstica:

Tabla 5.12: Distancias hacia la clasificación correcta del banco Mammographic Masses

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Distancia	0.3340	0.2909	0.4575	0.2510	0.2471	0.3177

5.7. Wisconsin Breast Cancer

Este banco de datos fué obtenido de los Hospitales de la Universidad de Wisconsin por el Dr. William H. Wolberg. Los datos reflejan en orden cronológico los casos reportados por el Dr. William, con un total de 699 instancias.

Cada instancia esta asociada a 9 rasgos, más la clase; cada uno de ellos con un rango de evaluación que va del 1 al 10. La tabla 5.13, muestra los atributos para las instancias del banco de datos.

Tabla 5.13: Información de rasgos del banco de datos Winsonsin Breast Cancer

Rasgo	Unidad	Descripción
Clump Thickness	Numérico	Grosor de la masa (1 al 10)
Uniformity of Cell Size	Numérico	Uniformidad del tamaño de célula (1 al 10)
Uniformity of Cell Shape	Numérico	Uniformidad de la forma de la célula (1 al 10)
Marginal Adhesion	Numérico	Adhesión Marginal(1 al 10)
Single Epithelial Cell Size	Numérico	Tamaño de la célula epitelial(1 al 10)
Bare Nuclei	Numérico	Núcleolo desnudo(1 al 10)
Bland Chromatin	Numérico	Cromatina blanda(1 al 10)
Normal Nucleoli	Numérico	Núcleolo normal(1 al 10)
Mitoses	Numérico	Mitosis(1 al 10)
Class	Numérico	Clase (4=maligno, 2=benigno)

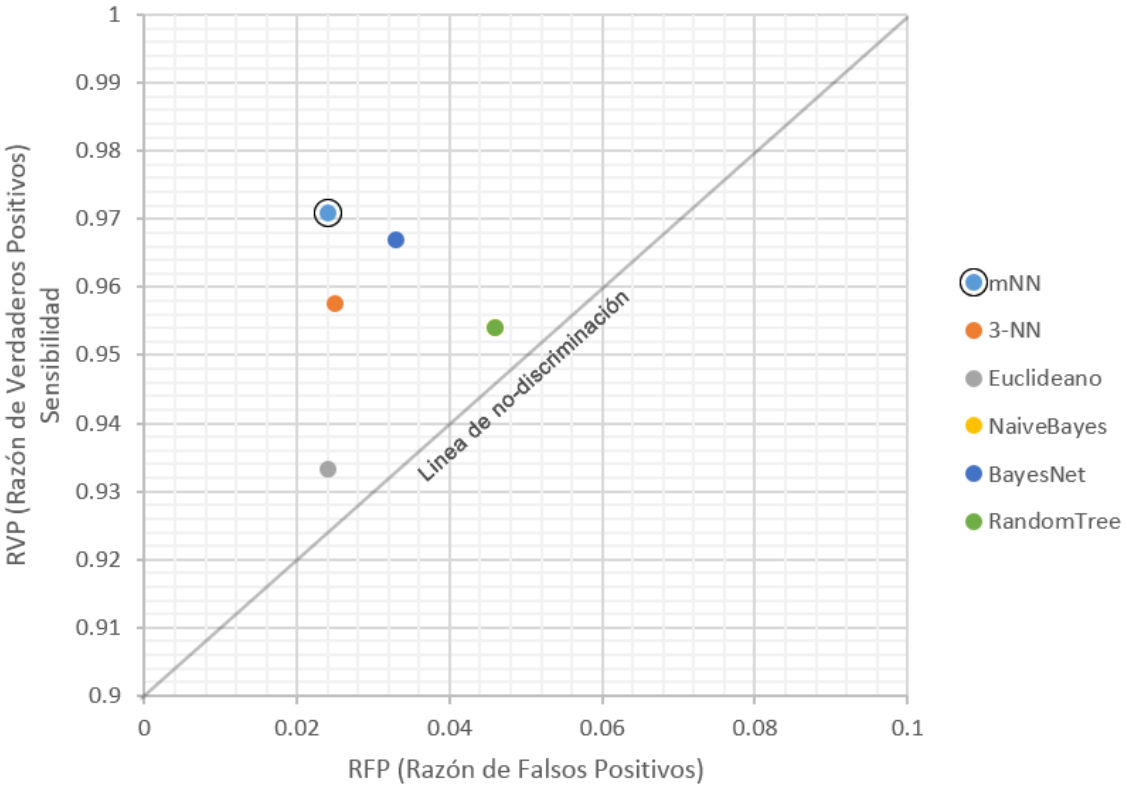
En la tabla 5.14 se presentan los resultados de la clasificación realizada para el banco de datos. Los resultados corresponden al rendimiento y análisis ROC del algoritmo propuesto, junto con otros algoritmos ejecutados en WEKA. Para el análisis ROC se eligió como instancia positiva aquella que corresponde a un tumor maligno, es decir, que tiene clase 4.

Tabla 5.14: Rendimientos de Clasificación y Análisis ROC el banco de datos Wisconsin Breast Cancer

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Rendimiento (%)	97.4202	96.8773	96.1286	96.1373	97.1388	92.4177
RVP	0.9708	0.9576	0.9333	0.9540	0.9670	0.954
RFP	0.0241	0.0249	0.0240	0.0460	0.0330	0.046

La figura 5.4 muestra el punto en el espacio ROC de los clasificadores mostrados en la tabla anterior.

Figura 5.4: Espacio ROC para el banco de datos Wisconsin Breast Cancer



Finalmente, tomando como base el punto de clasificación correcta (0, 1), la tabla 5.15 muestra las distancias euclidianas obtenidas a partir de los diferentes clasificadores; entre menor sea la distancia representa una mejor prueba diagnóstica:

Tabla 5.15: Distancias hacia la clasificación correcta del banco Wisconsin Breast Cancer

	mNN(m=3)	3-NN	Euclidiano	NaiveBayes	BayesNet	RandomTree
Distancia	0.0379	0.0492	0.0709	0.0651	0.0467	0.0651

Capítulo 6

6. Conclusiones y Trabajo Futuro

En este capítulo se presentan las conclusiones a partir de las pruebas con los diferentes bancos de datos analizados en el capítulo anterior. Así mismo, se proponen ideas para trabajo a futuro que podrían considerarse nuevos trabajos de investigación.

6.1. Conclusiones

Las conclusiones a las que se llegó son las siguientes:

1. Se realizó una investigación del estado del arte sobre los algoritmos actuales para el diagnóstico de síndromes, concluyendo en que las SVMs son los modelos de clasifiación más usados junto con las RNAs, y son pocos los trabajos que hacen uso del enfoque asociativo para este trabajo.
2. Se desarrolló un sistema web para la verificación de síndromes basado en las Memorias Morfológicas Autoasociativas *max*, obteniendo resultados competitivos con algunos de los mejores clasificadores del estado del arte.
3. Aunque el algoritmo utilizado es un recuperador, se desarrolló un método nuevo para utilizarlo en la clasificación de patrones.
4. Se llego a la conclusión de que la escalabilidad es una de las ventajas más significativas del uso de una plataforma de cómputo en la nube.
5. Se mostró un caso práctico y necesario para el uso de las bases de datos NoSQL.
6. Se implementó el sistema web en la nube de Amazon sobre los servicios EC2 y S3; brindando una arquitectura para aprovechar las ventajas de la misma.
7. Los resultados obtenidos reflejan un sistema con alto grado de confiabilidad para pruebas diagnósticas.
8. El algoritmo propuesto es eficaz para tratar patrones con ruido aditivo gracias a las Memorias Morfológicas Autoasociativas *max*.

6.2. Trabajo Futuro

1. Probar el algoritmo para bancos de datos más grandes, que contengan al menos 1 millón de instancias.

2. Modificar el algoritmo para implementarlo sobre cómputo paralelo (ej. CUDA).

3. Utilizar otros algoritmos de clasificación para la fase de recuperación del sistema propuesto.

4. Implementar el sistema sobre otras plataformas de cómputo en la nube.

5. Permitir aplicar filtros de pre procesamiento a los bancos de datos de entrenamiento y prueba.

6. Usar la técnica de *Map-Reduce* en la nube.

Referencias

[1] Bamberg, F., Dierks, A., Nikolaou, K., Reiser, M., Becker, C., and Johnson, T. (2011) Metal artifact reduction by dual energy computed tomography using monoenergetic extrapolation. *European Radiology 21*, 1424–1429.

[2] Apostolova, L. G., Klein, E., Cummings, J., Thompson, P., Hwang, K., Kohannim, O., Coppola, G., and Gao, F. (2011) Automated diagnostic classifiers for cognitively normal and mild cognitive impairment subjects using imaging, genotyping, and gene expression. *Alzheimer's & Dementia 7*, S128–S129.

[3] Deng, Y., Wang, Y., and Shen, Y. (2011) An automated diagnostic system of polycystic ovary syndrome based on object growing. *Artificial Intelligence in Medicine 51*, 199–209.

[4] Aussem, A., de Morais, S. R., and Corbex, M. (2012) Analysis of nasopharyngeal carcinoma risk factors with Bayesian networks. *Artificial Intelligence in Medicine 54*, 53 – 62.

[5] Park, H.-S., and Cho, S.-B. (2012) Evolutionary attribute ordering in Bayesian networks for predicting the metabolic syndrome. *Expert Systems with Applications 39*, 4240 – 4249.

[6] Huang, L.-C., Hsu, S.-Y., and Lin, E. (2009) A comparison of classification methods for predicting Chronic Fatigue Syndrome based on genetic data. *Journal of Translational Medicine 7*, 1–8.

[7] Ocampo, E., Maceiras, M., Herrera, S., Maurente, C., Rodríguez, D., and Sicilia, M. A. (2011) Comparing Bayesian inference and case-based reasoning as support techniques in the diagnosis of Acute Bacterial Meningitis. *Expert Systems with Applications 38*, 10343 – 10354.

[8] Tagluk, M. E., and Sezgin, N. (2011) A new approach for estimation of obstructive sleep apnea syndrome. *Expert Systems with Applications 38*, 5346 – 5351.

[9] Yang, M., Zheng, H., Wang, H., McClean, S., Hall, J., and Harris, N. (2012) A machine learning approach to assessing gait patterns for Complex Regional Pain Syndrome. *Medical Engineering & Physics 34*, 740 – 746.

[10] Hirose, H., Takayama, T., Hozawa, S., Hibi, T., and Saito, I. (2011) Prediction of metabolic syndrome using artificial neural network system based on clinical data including insulin resistance index and serum adiponectin. *Computers in Biology and Medicine 41*, 1051 – 1056.

[11] Tagluk, M. E., Akin, M., and Sezgin, N. (2010) Classification of sleep apnea by using wavelet transform and artificial neural networks. *Expert Systems with Applications 37*, 1600 – 1607.

[12] Güneş, S., Polat, K., and Şebnem Yosunkaya, (2010) Multi-class f-score feature selection approach to classification of obstructive sleep apnea syndrome. *Expert Systems with Applications 37*, 998 – 1004.

[13] Álvarez Estévez, D., and Moret-Bonillo, V. (2009) Fuzzy reasoning used to detect apneic events in the sleep apnea-hypopnea syndrome. *Expert Systems with Applications 36*, 7778 – 7785.

[14] Yáñez, I. L., Carapia, R. F., Márquez, C. Y., and Nieto, O. C. (2012) Automatic detection of cranial fractures in radiological images using a pattern classifier. *Revista Facultad de Ingeniería 0*.

[15] Worachartcheewan, A., Nantasenamat, C., Isarankura-Na-Ayudhya, C., Pidetcha, P., and Prachayasittikul, V. (2010) Identification of metabolic syndrome using decision tree analysis. *Diabetes Research and Clinical Practice 90*, e15 – e18.

[16] Son, C.-S., Kim, Y.-N., Kim, H.-S., Park, H.-S., and Kim, M.-S. (2012) Decision-making model for early diagnosis of congestive heart failure using rough set and decision tree approaches. *Journal of Biomedical Informatics 45*, 999 – 1008.

[17] Scalzo, F., Hamilton, R., Asgari, S., Kim, S., and Hu, X. (2012) Intracranial hypertension prediction using extremely randomized decision trees. *Medical Engineering & Physics 34*, 1058 – 1065.

[18] Dobrowolski, A. P., Wierzbowski, M., and Tomczykiewicz, K. (2012) Multiresolution MUAPs decomposition and SVM-based analysis in the classification of neuromuscular disorders. *Computer Methods and Programs in Biomedicine 107*, 393 – 403.

[19] Subasi, A. (2013) Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Computers in Biology and Medicine –*.

[20] Dukart, J., Mueller, K., Barthel, H., Villringer, A., Sabri, O., and Schroeter, M. L. (2012) Meta-analysis based SVM classification enables accurate detection of Alzheimer's disease across different clinical centers using FDG-PET and MRI. *Psychiatry Research: Neuroimaging* –.

[21] Shen, C.-P., Kao, W.-C., Yang, Y.-Y., Hsu, M.-C., Wu, Y.-T., and Lai, F. (2012) Detection of cardiac arrhythmia in electrocardiograms using adaptive feature extraction and modified support vector machines. *Expert Systems with Applications 39*, 7845 – 7852.

[22] Übeyli, E. D. (2008) Support vector machines for detection of electrocardiographic changes in partial epileptic patients. *Engineering Applications of Artificial Intelligence 21*, 1196 – 1203.

[23] Orrù, G., Pettersson-Yeo, W., Marquand, A. F., Sartori, G., and Mechelli, A. (2012) Using Support Vector Machine to identify imaging biomarkers of neurological and psychiatric disease: A critical review. *Neuroscience & Biobehavioral Reviews 36*, 1140 – 1152.

[24] Khandoker, A., Palaniswami, M., and Karmakar, C. (2009) Support Vector Machines for Automated Recognition of Obstructive Sleep Apnea Syndrome From ECG Recordings. *Information Technology in Biomedicine, IEEE Transactions on 13*, 37 –48.

[25] Übeyli, E. D., Cvetkovic, D., Holland, G., and Cosic, I. (2010) Adaptive neuro-fuzzy inference system employing wavelet coefficients for detection of alterations in sleep EEG activity during hypopnoea episodes. *Digital Signal Processing 20*, 678 – 691.

[26] Ushida, Y., Kato, R., Niwa, K., Tanimura, D., Izawa, H., Yasui, K., Takase, T., Yoshida, Y., Kawase, M., Yoshida, T., Murohara, T., and Honda, H. (2012) Combinational risk factors of metabolic syndrome identified by fuzzy neural network analysis of health-check data. *BMC Medical Informatics And Decision Making 12*, 80.

[27] Nakayama, N., Oketani, M., Kawamura, Y., Inao, M., Nagoshi, S., Fujiwara, K., Tsubouchi, H., and Mochida, S. (2011) Novel classification of acute liver failure through clustering using a self-organizing map: usefulness for prediction of the outcome. *Journal of Gastroenterology 46*, 1127–1135.

[28] Lee, H., Malaspina, D., Ahn, H., Perrin, M., Opler, M. G., Kleinhaus, K., Harlap, S., Goetz, R., and Antonius, D. (2011) Paternal age related schizophrenia (PARS): Latent subgroups detected by k-means clustering analysis. *Schizophrenia Research 128*, 143 – 149.

[29] Elmoataz, A., Schüpp, S., Clouard, R., Herlin, P., and Bloyet, D. (1998) Using active contours and mathematical morphology tools for quantification of immunohistochemical images. *Signal Processing 71*, 215 – 226.

[30] Naegel, B. (2007) Using mathematical morphology for the anatomical labeling of vertebrae from 3D CT-scan images. *Computerized Medical Imaging and Graphics 31*, 141 – 156.

[31] Michaelson, J., Chen, L., Bush, D., Fong, A., Smith, B., and Younger, J. (2011) Improved web-based calculators for predicting breast carcinoma outcomes. *Breast Cancer Research and Treatment 128*, 827–835.

[32] Mori, I., Ozaki, T., Muragaki, Y., Ibata, T., Ueda, H., Shinagawa, T., and Osamura, Y. (2013) Construction of web-based remote diagnosis system using virtual slide for routine pathology slides of the rural hospital in Japan. *Diagnostic Pathology 8*, 1–4.

[33] Lin, H.-C., Wu, H.-C., Chang, C.-H., Li, T.-C., Liang, W.-M., and Wang, J.-Y. (2011) Development of a real-time clinical decision support system upon the web mvc-based architecture for prostate cancer treatment. *BMC Medical Informatics and Decision Making 11*, 1–11.

[34] Díaz-de León, J., and Yáñez-Márquez, *Memorias Morfológicas Autoasociativas*; Serie Verde 58, 2001.

[35] Díaz-de León, J., and Yáñez-Márquez, *Memorias Morfológicas Heteroasociativas*; Serie Verde 57, 2001.

[36] Berners-Lee, T., Fielding, R., and Frystyk, H. (1995) Hypertext Transfer Protocol - http/1.0. *Internet draft, IETF*

[37] Fielding, R., Frystyk, H., and Berners-Lee, T. (1995) Hypertext Transfer Protocol - http/1.1. *Internet draft, IETF*

[38] Berners-Lee, T. (1994) The World-Wide Web. *Commun. ACM 37*, 76–82.

[39] Schulzrinne, H. (1996) World Wide Web: whence, whither, what next? *Network, IEEE 10*, 10–17.

[40] Ku, H.-H., and Huang, C.-M. (2010) Web2OHS: A Web2.0-Based Omnibearing Homecare System. *Information Technology in Biomedicine, IEEE Transactions on 14*, 224–233.

[41] Wei, C., Khoury, R., and Fong, S. (2012) Web 2.0 Recommendation service by multi-collaborative filtering trust network algorithm. *Information Systems Frontiers* 1–19.

[42] Díaz, O., Puente, G., Cánovas Izquierdo, J., and García Molina, J. (2011) Harvesting models from web 2.0 databases. *Software & Systems Modeling* 1–20.

[43] Hickson, I. *The WebSocket API*; Candidate Recommendation, 2012; http://www.w3.org/TR/2012/CR-websockets-20120920/.

[44] Hoetzlein, R. (2012) Graphics Performance in Rich Internet Applications. *Computer Graphics and Applications, IEEE 32*, 98–104.

[45] Jacobs, J. H. P. L., Ian; Jaffe (2012) How the Open Web Platform Is Transforming Industry. *Internet Computing, IEEE 16*, 82–86.

[46] Kim, E., Schissel, D., Abla, G., Flanagan, S., and Lee, X. (2012) Web-based (HTML5) interactive graphics for fusion research and collaboration. *Fusion Engineering and Design –*.

[47] Ahmad, F., Mat Isa, N., Hussain, Z., and Sulaiman, S. (2013) A genetic algorithm-based multi-objective optimization of an artificial neural network classifier for breast cancer diagnosis. *Neural Computing and Applications 23*, 1427–1435.

[48] Rogers, R. L., Feller, E. D., and Gottlieb, S. S. (2006) Acute Congestive Heart Failure in the Emergency Department. *Cardiology Clinics 24*, 115 – 123.

[49] Duda R.O., S. D., Hart P.E. In *Pattern Classification*, 2nd ed.; USA,, Ed.; McGraw-Hill., 2001.

[50] Azar, A., and El-Said, S. (2013) Performance analysis of support vector machines classifiers in breast cancer mammography recognition. *Neural Computing and Applications* 1–15.

[51] MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*; Cambridge University Press: New York, NY, USA, 2002.

[52] Coates, A., and Ng, A. In *Neural Networks: Tricks of the Trade*; G. Montavon, G. B. O., and Müller, K.-R., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg, 2012; Vol. 7700; pp 561–580.

[53] Robles, S., Fernández, J. L., Fortier, A., Rossi, G., and Gordillo, S. E. (2012) Improving the model view controller paradigm in the web. *Int. J. Web Eng. Technol. 7*, 22–44.

[54] Brambilla, M., Ceri, S., Fraternali, P., and Manolescu, I. (2006) Process modeling in Web applications. *ACM Trans. Softw. Eng. Methodol. 15*, 360–409.

[55] Flores Carapia, R., and Yáñez Márquez, C. (2005) Minkowski's Metrics-Based k-NN Clasifier Algorithm: A Comparative Study. *Research on Computing Science 14*, 191–202.

[56] Zhu, P., Hu, Q., and Yang, Y. In *Rough Sets and Current Trends in Computing*; Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., and Hu, Q., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg, 2010; Vol. 6086; pp 347–355.

[57] Dadhania, S. S., and Dhobi, J. S. (2012) Improved kNN Algorithm by Optimizing Cross-validation. *International Journal of Engineering Research & Technology 1*.

[58] Spackman, K. A. Signal detection theory: valuable tools for evaluating inductive learning. 1989.

[59] Zou, K., O'Malley, A., and Mauri, L. (2007) Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models. *Circulation 115*, 654.

[60] Bradley, A. P. (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition 30*, 1145–1159.

[61] Winkler, V. J. *Securing the Cloud*; Syngress: Boston, 2011; pp 1 – 27.

[62] Lee, B. S., Yan, S., Ma, D., and Zhao, G. Aggregating IaaS Service. 2011.

[63] Fan, P., Chen, Z., Wang, J., and Zheng, Z. Online Optimization of VM Deployment in IaaS Cloud. 2012.

[64] Mauch, V., Kunze, M., and Hillenbrand, M. (2013) High performance cloud computing. *Future Generation Computer Systems 29*, 1408 – 1416.

[65] Bojanova, I., and Samba, A. Analysis of Cloud Computing Delivery Architecture Models. 2011.

[66] Lourenço, A., Silva, H. P., Carreiras, C., Alves, A. P., and Fred, A. L. N. (2013) A web-based platform for biosignal visualization and annotation. *Multimedia Tools and Applications* 1–28.

[67] MongoDB. `http://www.mongodb.org/`.

[68] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009) The WEKA data mining software: an update. *SIGKDD Explor. Newsl. 11*, 10–18.