

**MÉTODOS DE APRENDIZAJE DE MÁQUINA PARA LA
DETECCIÓN DE ARRITMIAS CARDIACAS EN TIEMPO REAL**

**JORGE HUMBERTO GALLEGO ORREGO
CARLOS ANDRÉS RODRIGUEZ LONDOÑO**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
INGENIERÍA ELECTRÓNICA
MEDELLÍN
2013**

**MÉTODOS DE APRENDIZAJE DE MÁQUINA PARA LA
DETECCIÓN DE ARRITMIAS CARDIACAS EN TIEMPO REAL**

**JORGE HUMBERTO GALLEGO ORREGO
CARLOS ANDRÉS RODRIGUEZ LONDOÑO**

Trabajo de grado para optar al título de Ingeniero Electrónico

**Director
Iván Darío Mora Osorio
Magister en Ingeniería**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
INGENIERÍA ELECTRÓNICA
MEDELLÍN
2013**

Nota de aceptación

Firma
Nombre:
Presidente del jurado

Firma
Nombre:
Jurado

Firma
Nombre:
Jurado

Medellín, Julio 18 de 2013

Para cada una de las personas que un día confiaron en nosotros. A nuestras familias por su apoyo y acompañamiento en el proceso. A Dios por brindarnos la sabiduría para el desarrollo de este proyecto y los ánimos para seguir adelante en los momentos difíciles de nuestra carrera.

AGRADECIMIENTOS

Al especialista en Ingeniería Biomédica Andrés Orozco, por el tiempo invertido en el proyecto en cada una de sus etapas, el interés y dedicación en el proyecto, asesoría brindada en cuanto a investigación e información, solución de dudas sobre los algoritmos y corrección de errores.

Al Centro de Bioingeniería de la UPB por facilitar cada uno de los algoritmos pertinentes la etapa de pre-procesamiento de las señales extraídas de las bases de datos.

CONTENIDO

	Pág.
INTRODUCCIÓN	14
1. OBJETIVOS	15
1.1. OBJETIVO GENERAL	15
1.2. OBJETIVOS ESPECÍFICOS	15
2. MARCO TEÓRICO	16
2.1. SEÑALES BIOELÉCTRICAS	16
2.2. ELECTROCARDIOGRAFÍA (ECG)	16
2.2.1. Ondas ECG y sus intervalos de tiempo	17
2.2.2. Ritmos Cardíacos	18
2.2.3. Latidos Prematuros	19
2.2.4. Arritmias auriculares	19
2.2.5. Arritmias ventriculares	20
3. ESTADO DEL ARTE	21
4. MÉTODOS	24
4.1. Transformada Wavelet Discreta (DWT)	24
4.1.1. Wavelets Daubechies (DbN)	25

4.1.2.	Wavelets Symlets (SymN)	25
4.2.	Análisis de Componentes Principales (PCA)	26
4.3.	Base de datos	27
4.4.	Pre-procesamiento	28
4.5.	Segmentación	29
4.6.	Extracción de características	29
4.6.1.	Espacio de características híbrido	32
4.7.	Parámetros de desempeño	33
5.	ALGORITMOS	36
5.1.	Aprendizaje de Máquina	36
5.2.	Métodos de Aprendizaje Supervisado	37
5.2.1.	Máquinas de Soporte Vectorial (SVM)	37
5.2.2.	K-Nearest Neighbor (KNN)	41
5.3.	Métodos de Aprendizaje No Supervisado	44
5.3.1.	KMEANS	44
5.3.2.	Mapas Auto-Organizados (SOM)	47
6.	RESULTADOS	52
7.	ANÁLISIS	61
7.1.	Métodos Supervisados	61
7.2.	Métodos No Supervisados	62
8.	ALGORITMO PARA LA DETECCIÓN DE LATIDOS PVC EN C++	65

CONCLUSIONES 67

BIBLIOGRAFÍA 70

LISTA DE FIGURAS

	Pág.
1 Señal ECG	17
2 Esquema de descomposición <i>wavelet</i> de un nivel	25
3 Reducción de 2 dimensiones a 1 con PCA	27
4 Latido normal	28
5 Latido PVC	28
6 Función <i>wavelet</i> ψ Db9	30
7 Función <i>wavelet</i> ψ Sym9	30
8 Espacio de las energías de las sub-bandas <i>wavelet</i> con 3 componentes principales	31
9 Clasificación de clases por medio de soportes vectoriales	38
10 Funciones $cost_0(z)$ (línea azul) y $cost_1(z)$ (línea roja)	39
11 Diagrama de flujo del algoritmo SVM para la clasificación de arritmias ventriculares	40
12 Decisión basada en KNN	42
13 Diagrama de flujo del algoritmo KNN para la clasificación de arritmias ventriculares	43
14 Formación de grupos o clusters con <i>Kmeans</i>	45

15	Diagrama de flujo del algoritmo <i>Kmeans</i> para la clasificación de arritmias ventriculares	47
16	Formación de grupos a partir de SOM en un mapa bidimensional	49
17	Diagrama de flujo del algoritmo SOM para la clasificación de arritmias ventriculares	51
18	Separación de latidos normales (marcas rojas) y PVC (marcas verdes) en el espacio híbrido con algoritmo <i>Kmeans</i>	58
19	Separación de latidos normales (marcas rojas), PVC (marcas amarillas), APB (marcas verdes) y SVPB (marcas azules) en el espacio híbrido con algoritmo <i>Kmeans</i>	59
20	Separación de latidos normales (celdas cían) y PVC (celdas verdes) del espacio híbrido en el mapa creado por SOM	59
21	Separación de latidos normales (celdas rojas), PVC (celdas verdes), APB (marcas azules) y SVPB (marcas violetas) del espacio de las sub-bandas de las energías <i>wavelet</i> (Sym9) en el mapa creado por SOM	60
22	Diagrama de flujo del código en C++ del algoritmo de predicción SOM	66

LISTA DE TABLAS

	Pág.
1 Matriz de confusión para 2 clases	33
2 Medidas de desempeño	34
3 Resultados del algoritmo SVM con 8 espacios de características diferentes .	53
4 Resultados del algoritmo KNN con 8 espacios de características diferentes .	54
5 Resultados del algoritmo <i>Kmeans</i> con 8 espacios de características diferentes	55
6 Resultados del algoritmo SOM con 8 espacios de características diferentes .	56
7 Resultados SVM multiclase con el espacio temporal	57
8 Resultados KNN multiclase con el espacio temporal	57
9 Resultados <i>Kmeans</i> multiclase con el espacio híbrido	57
10 Resultados SOM multiclase con el espacio de las energías de las sub-bandas <i>wavelet</i> Sym9	58

GLOSARIO

ALGORITMO: conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

APRENDIZAJE DE MÁQUINA: métodos de programación de computadores para optimizar un criterio de desempeño usando datos de ejemplo o experiencia del pasado [1].

APREDIZAJE NO SUPERVISADO: algoritmos en los cuales el entrenamiento se realiza sin ayuda de ningún agente externo. Este tipo de algoritmo organiza sus pesos a partir de la variación de las entradas del sistema e intenta crear grupos de separación adecuados [2].

APRENDIZAJE SUPERVISADO: es el tipo de algoritmos en que interviene un agente externo en la etapa de entrenamiento, asignando los valores correctos a los datos [2].

ARRITMIA: ritmo anormal del corazón.

BRADICARDIA: ritmo cardíaco lento y regular, normalmente de menos de 60 latidos por minuto. A esta frecuencia el corazón no puede bombear suficiente sangre con altos niveles de oxígeno al cuerpo [3].

ELECTROCARDIOGRAMA (ECG): gráfica de trazado que representa la actividad eléctrica del corazón en función del tiempo.

ESPECIFICIDAD: habilidad de una prueba para detectar correctamente muestras negativas [2].

F1 SCORE: es la media armónica de la predicción positiva y la sensibilidad [4].

FALSO NEGATIVO (FN): latido clasificado como negativo, siendo positivo.

FALSO POSITIVO (FP): latido clasificado como positivo, siendo negativo.

FIBRILACIÓN VENTRICULAR (FV): es un tipo de arritmia caracterizada por una frecuencia cardíaca muy rápida y la ausencia total de contracciones eficaces de los ventrículos, lo que provoca que no llegue sangre a los órganos vitales [3].

FOCO ECTÓPICO: refiere a un grupo de células, que durante ciertas condiciones pueden tomar precedencia sobre el nodo SA. Puede estar localizado en las aurículas o en los ventrículos. Determinan el ritmo del corazón cuando la descarga del ritmo del nodo SA cae por debajo de cierto nivel [3].

KMEANS: es un algoritmo de agrupamiento, el cual divide el conjunto de entrada en k grupos, cada grupo con patrones similares [5].

K NEAREST NEIGHBOR (KNN): es un algoritmo de clasificación estadístico usado para clasificar objetos basados en la distancia más cercana de los datos de entrenamiento del espacio de características [6].

NODO SINOATRIAL (SA): es un conjunto de células marca-pasos situadas en la parte superior de la aurícula derecha, que se encargan de disparar un impulso eléctrico, iniciando el ciclo cardíaco [3].

PREDICCIÓN POSITIVA: número de registros recuperados y relevantes divididos por el número de registros recuperados [2].

PREMATURE BEAT (PB): latido prematuro en español, es un latido que ocurre antes del siguiente latido SA [3].

RITMO SINOATRIAL: es un ritmo originado por el nodo sinusal que tiene un promedio de 50 a 100 latidos por minuto en reposo [3].

SELF-ORGANIZED MAP (SOM): mapas auto-organizados en español, es un método que distribuye los datos de un espacio de entrada en una rejilla, usualmente bidimensional, de tal forma que los datos con patrones similares se agrupen en regiones del mapa [7].

SENSIBILIDAD: habilidad de una prueba para detectar correctamente muestras positivas [2].

SUPPORT VECTOR MACHINES (SVM): máquinas de soporte vectorial en español, este método se encarga de tomar los datos como un conjunto de vectores creando un hiperplano que los separe, maximizando el margen entre el conjunto de datos [8].

SUPRAVENTRICULAR PREMATURE BEAT (SVPB): latido supra-ventricular prematuro en español, es un latido prematuro que se origina en la parte superior de los ventrículos [3].

TAQUICARDIA: ritmo cardíaco acelerado o irregular normalmente de más de 100 latidos por minuto y hasta 400 latidos por minuto [3].

VENTRICULAR PREMATURE BEAT (VPB): latido ventricular prematuro en español, es un latido prematuro que tiene origen en los ventrículos [3].

VERDADERO NEGATIVO (VN): latido clasificado como negativo, siendo negativo.

VERDADERO POSITIVO (VP) : latido clasificado como positivo, siendo positivo.

RESUMEN

En la actualidad la búsqueda de nuevos métodos de clasificación de arritmias cardíacas ha sido una de las mayores preocupaciones en el sector de la cardiología; por esta razón y buscando suplir parte de esta necesidad, este trabajo propone 4 algoritmos para la detección de arritmias cardíacas basados en aprendizaje de máquina supervisado y no supervisado. Para cada uno de los algoritmos, se emplearon como base para su desarrollo y comprobación algunas bases de datos del MIT-BIH, extrayendo características como la señal ECG temporal, los coeficientes de la transformada *wavelet* discreta y el análisis de componentes principales.

Las bases de datos del MIT-BIH fueron pre procesadas y filtradas con anterioridad por el Grupo de Dinámica Cardiovascular de la UPB; cuentan con las etiquetas de cada latido según el tipo de afección, lo cual da soporte para el entrenamiento y la obtención del desempeño de cada método. En los algoritmos se implementaron dos métodos de aprendizaje supervisado, uno basado en el método estadístico KNN y el otro basado en máquinas de soporte vectorial; también se implementaron dos métodos de aprendizaje no supervisado, uno de *clustering* tipo *Kmeans* y el otro basado en aprendizaje competitivo tipo SOM. Dichos algoritmos fueron desarrollados con el software MATLAB y las siguientes Toolbox: *bioinformatics*, *Statistics* y SOM Toolbox 2.0.

Se evaluaron otros parámetros adicionales a la precisión, debido a que no entregaron información suficiente para mostrar la efectividad de los algoritmos en casos biomédicos. Los parámetros de desempeño que se evaluaron fueron: la especificidad, la sensibilidad, *F1 score*, predicción positiva, y tiempo de ejecución, el cual sirvió como indicador del costo computacional.

Adicionalmente, se desarrollo el algoritmo de clasificación de SOM en lenguaje C como apoyo a posibles implementaciones en sistemas embebidos por parte del Centro de Bioingeniería de la UPB.

PALABRAS CLAVE:

Aprendizaje de Maquina, *Clustering*, Clasificación de Arritmias, Algoritmo, KNN, SVM, Mapas auto-organizados, *Kmeans*.

INTRODUCCIÓN

Las enfermedades cardiovasculares son la causa de mayor mortalidad del planeta. En 2008, la Organización Mundial de la Salud (OMS) registró 17,3 millones de muertes por estas causas, lo que es más del 30 % de las muertes totales de ese año [9]. Las arritmias cardíacas son una consecuencia de las enfermedades cardiovasculares; algunas arritmias conllevan a paros cardíacos o muertes inmediatas, como la taquiarritmia ventricular; por lo tanto, se debe tener un constante monitoreo de la actividad eléctrica del corazón del paciente para detectar arritmias fatales a tiempo y tratarlas adecuadamente.

Una prueba para monitorear este tipo de afecciones es el electrocardiograma (ECG), el cual es un método no invasivo utilizado para el diagnóstico de arritmias cardíacas. Este examen permite adquirir el biopotencial, el cual refleja eventos eléctricos a nivel celular. Éste registro ha llegado a ser una herramienta clínica indispensable en diferentes contextos. Patrones presentes en el ritmo y la morfología de la señal ECG, permiten identificar enfermedades cardíacas. Por esta razón, el desarrollo de algoritmos que identifiquen éstos patrones, ha sido un campo de estudio importante en el área de la cardiología [10].

Para un adecuado diagnóstico de afecciones, se procesa la señal ECG en 3 etapas: pre-procesamiento, extracción de características y clasificación. Las etapas de pre-procesamiento y extracción de características se describen en la sección 4. Para la etapa de clasificación se usan los algoritmos SVM, KNN, *Kmeans* y SOM, explicados detalladamente en la sección 5. Se evalúan éstos 4 algoritmos para 8 espacios de características diferentes y son comparados en términos de especificidad, sensibilidad, costo computacional y otros parámetros, explicados en la sección 4. Los resultados y su respectivo análisis se presentan en las secciones 6. y 7.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Evaluar el desempeño de 4 métodos de aprendizaje de máquina para la detección de arritmias cardíacas.

1.2. OBJETIVOS ESPECÍFICOS

- Implementar 4 algoritmos para la detección de arritmias cardíacas basados en aprendizaje supervisado y no supervisado.
- Evaluar el desempeño de los algoritmos seleccionados en cuanto a especificidad y sensibilidad para la detección de arritmias cardiacas.
- Implementar en C el método con mejor relación entre desempeño y costo computacional para su posterior implementación en sistemas embebidos.

2. MARCO TEÓRICO

2.1. SEÑALES BIOELÉCTRICAS

Las señales bioeléctricas están relacionadas a los procesos iónicos, que surgen como resultado de la actividad electroquímica de un grupo especial de células (no todas las células son excitables). Las concentraciones de los iones de sodio y potasio, en el interior y exterior de la célula, se forma un potencial debido a esta difusión de iones, a lo que se denomina potencial de acción. Algunos de estos potenciales se ven reflejados por la actividad interna de las células o por estímulos externos. Independientemente del tipo de células, ya sean las del cerebro, corazón o músculos, sus mecanismos son similares. Estas señales pueden estar directamente relacionadas a diagnósticos clínicos, mientras que otras requieren de un análisis más detallado antes de dar alguna interpretación significativa [3].

2.2. ELECTROCARDIOGRAFÍA (ECG)

El ECG describe la actividad eléctrica del corazón registrada por electrodos ubicados en la superficie del cuerpo (el pecho, brazos y piernas). En cada latido, un impulso viaja a través del corazón determinando su ritmo y velocidad, causando que el músculo del corazón se contraiga y bombee sangre. En el ECG, un latido se manifiesta en una serie de ondas cuyas morfologías y tiempos, contienen información, la cual es usada para diagnosticar enfermedades reflejadas en los disturbios de la actividad eléctrica del corazón. Otro patrón importante de la señal ECG es el tiempo transcurrido entre dos latidos sucesivos [3].

2.2.1. Ondas ECG y sus intervalos de tiempo

En la morfología de la señal ECG se identifican varias formas de onda que reflejan la despolarización y repolarización de las aurículas y los ventrículos. Estas formas de ondas son:

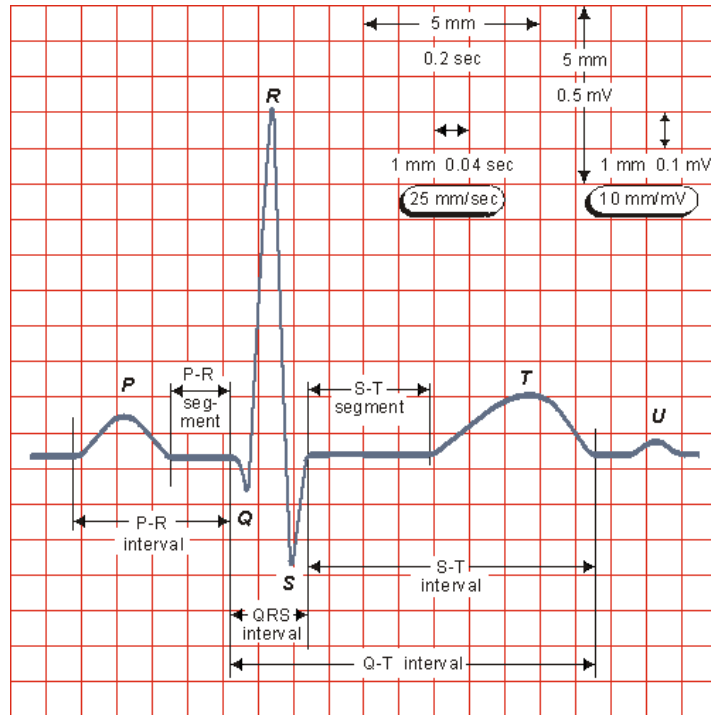


Figura 1. Señal ECG

Fuente: Electrocardiografia.es, Ondas Componentes del ECG, 2013. [En línea]. Available: http://www.electrocardiografia.es/ondas_ecg.html. [Último acceso: 13 05 2013].

- **Onda P:** Refleja la despolarización de la aurícula derecha e izquierda. En muchas de las derivaciones, la onda P tiene una polaridad positiva y una morfología monofásica suave. Normalmente su amplitud es de $300 \mu\text{V}$ y su duración es menor a 120 ms. Una ausencia de la onda P sugiere que el ritmo se origina en los ventrículos.
- **Complejo QRS:** refleja de despolarización de los ventrículos. Su duración es alrededor de 70 a 110 ms. Se compone de 3 ondas, aunque siempre se le refiere como el complejo QRS: la onda Q, que es la primera deflexión negativa, luego una deflexión positiva dada por la onda R, y la siguiente deflexión negativa la denota la onda S (ver Figura 1). Su morfología es muy variable y depende del origen del

latido. Normalmente su amplitud es de 2-3 mV siendo ésta la mayor amplitud entre las formas de la onda ECG.

- **Onda T:** refleja la repolarización ventricular y su duración es de aproximadamente 300 ms. Su posición depende mucho de la velocidad del corazón, causando que en altas velocidades se acerque más al complejo QRS. Su morfología es suave y se asocia a un pico positivo. En altas velocidades, la onda P y la onda T se unen, causando una dificultad al determinar el final de la onda T y el inicio de la onda P.
- **Onda U:** ocurre después de la onda T. Aunque su origen no es claro, se cree que ésta es probablemente sea una post-repolarización.

Conociendo los tiempos y formas de ondas de la señal ECG, hay otros indicadores que aportan información importante sobre la actividad cardíaca, como el intervalo RR, el cual representa la longitud de un ciclo cardíaco ventricular medido entre dos ondas R sucesivas e indica el ritmo ventricular. Este intervalo es la cantidad del ritmo fundamental en cualquier interpretación de la señal ECG y es usado para caracterizar diferentes arritmias [3].

2.2.2. Ritmos Cardíacos

El ritmo normal de un latido es controlado por impulsos eléctricos formados en el nodo SA, y produce un ritmo cardíaco entre 50 y 100 latidos por minuto durante el reposo, aunque este ritmo no es totalmente regular, incluso si hay ausencia de estrés mental o físico, y es debido a una variación continua entre dos componentes del sistema nervioso autónomo, el cual influencia el disparo del nodo SA: un incremento en la actividad parasimpática disminuye el ritmo, mientras que un incremento en la actividad simpática aumenta el ritmo.

Un disturbio en el ritmo sinusal normal es llamado arritmia, la cual puede ocurrir cuando en vez del nodo SA, otro marca-pasos del corazón inicia la despolarización. En consecuencia se altera la formación de los impulsos eléctricos. El fenómeno de re-entrada es otro problema de conducción de impulsos que confirma una arritmia, y se debe a cambios en el periodo refractario y la velocidad de propagación del corazón [3]. La clasificación de arritmias cardíacas se realiza con respecto al lugar de su origen y su velocidad: bradicardia (ritmo del corazón es lento, de menos de 60 latidos por minuto)

y taquicardia (el ritmo del corazón es rápido, de más de 100 latidos por minuto).

2.2.3. *Latidos Prematuros*

El ritmo sinusal normal algunas veces es interrumpido por un latido que ocurre antes del siguiente latido sinusal, y por lo tanto se le llama latido prematuro. El foco ectópico, que produce un latido prematuro, puede estar localizado en cualquier parte del corazón diferente al nodo SA. Dependiendo del lugar de la localización del foco, el latido prematuro puede tener una morfología normal o anormal y también puede estar precedido o no de una onda P.

Un latido prematuro es también llamado latido prematuro supraventricular si su origen se encuentra más arriba de los ventrículos, i.e. en la aurícula o nodo AV; y se le llama latido prematuro ventricular si su origen se encuentra en los ventrículos. Los tipos de SVPB más comunes están asociados a una onda P con morfología anormal y un complejo QRS con una morfología parecida a la de un latido sinusal normal [3].

2.2.4. *Arritmias auriculares*

Varios disturbios en el ritmo se originan de uno o múltiples focos ectópico en las aurículas. Las arritmias resultantes son caracterizadas también por la presencia de ondas P anormales o una completa ausencia de éstas. Las ondas P anormales aparecen cuando el foco ectópico se encuentra muy lejos del nodo SA, e.g. cerca al nodo Auriculoventricular (AV). Las principales arritmias auriculares son las siguientes:

- **Taquicardia auricular:** es una arritmia producida por un automatismo incrementado en las células marca-pasos de uno o múltiples focos entre las aurículas. La mayoría de los impulsos eléctricos son conducidos hacia los ventrículos, elevando el ritmo cardíaco de 140 a 220 latidos por minuto. Debido a esto, las ondas P son enmascaradas por las ondas T o incluso por el complejo QRS del anterior ciclo del corazón.
- **Aleteo auricular y fibrilación auricular:** ambas son taquiarritmias en las cuales las aurículas están desincronizadas con los ventrículos. Ambas son causadas por la continua re-entrada de un impulso eléctrico en las aurículas, causando que se

pierda la línea isoelectrica y la onda P. La rapidez e irregularidad del ritmo causa que la sangre fluya más lentamente en las aurículas, incrementando la probabilidad de que se produzca un coágulo de sangre.

La diferencia entre ambas radica en la velocidad y la forma de onda. El aleteo auricular tiene una morfología que se asemeja a un diente de sierra y su ritmo usualmente esta alrededor de 300 latidos por minuto; mientras que la fibrilación auricular es demasiado rápida, con un ritmo caótico (400-700 latidos por minuto), el cual vuelve las aurículas parpadeantes y los latidos en los ventrículos, irregulares [3].

2.2.5. *Arritmias ventriculares*

Las taquiarritmias ventriculares más comunes resultan del mecanismo de re-entrada e incluye la taquicardia ventricular, aleteo ventricular y fibrilación ventricular, los cuales tienen un mecanismo similar a las tres mencionadas taquiarritmias auriculares, i.e., son taquiarritmias ventriculares entrantes, sin embargo sus manifestaciones en el ECG son completamente diferentes.

- **Taquicardia ventricular:** ocurre a un ritmo alrededor de 120 latidos por minuto y consisten en latidos con una morfología similar a la de los latidos prematuros. Las ondas P algunas veces se pierden debido al ancho de los complejos QRS ó a una onda T repetitiva.
- **Aleteo ventricular:** es similar al aleteo auricular, no tiene complejos QRS ni ondas T discernibles, pero tiene una amplitud mucho mayor que fluctúa considerablemente en el tiempo. El aleteo auricular puede conllevar a una **fibrilación ventricular**, que debido a su ritmo caótico, no puede enviar sangre oxigenada al cerebro, lo que conlleva a un paro cardíaco, un cese de la respiración, la pérdida de conciencia y si no es tratado inmediatamente resulta fatal.

3. ESTADO DEL ARTE

El desarrollo de monitores cardíacos portátiles se ha convertido en un campo de investigación importante, debido a la tasa de mortalidad que presentan las enfermedades cardiovasculares. Por esta razón, son necesarios algoritmos para la detección de arritmias en tiempo real para mejorar este tipo de dispositivos. Nuevos métodos para la clasificación de arritmias cardíacas han sido propuestos a lo largo de los últimos años. En esta sección se realiza un breve repaso sobre algunos de esos trabajos, haciendo énfasis en los métodos implementados en este trabajo.

En 2006, Ramirez L. [11] usa métodos de aprendizaje supervisado para clasificar arritmias cardíacas, diseñando una interfaz de usuario para que el especialista registre los datos del ECG y visualice los resultados obtenidos en la clasificación. Se implementó métodos como KNN, *Naive Bayes*, *Bayes Net*, *MultiLayer Perceptron* (MLP) y SVM. En este artículo se clasificaron arritmias tales como los latidos de Contraccion Ventricular Prematura (en inglés PVC, Premature Ventricular Contraction), Contracciones Auriculares Prematuras (en inglés APB, Atrial Premature Beat), asistolia y taquicardia ventricular.

Karimifard, Ahmadian, Khoshnevisan y Nambakhsh en [12] muestran que la señal ECG puede ser expresada como una combinación lineal de las funciones básicas de *Hermite*, donde el error es menor al 10%; usan KNN para clasificar algunos tipos de arritmias como: *Normal Sinus Rhythm* (NSR), latidos prematuros ventriculares y auriculares, entre otros.

Kinoshita y colaboradores [13] desarrollaron un algoritmo para la detección de arritmias cardíacas fatales para un desfibrilador automático implantable, extrayendo las características de la señal ECG y realizando pruebas en un perro. Las neuronas las etiquetaron con el método KNN, lo que conllevó a que la precisión de este método fuera mejor que otros métodos competitivos.

En 2007, Rodríguez, Cuesta y Castellanos en [14] desarrollan un algoritmo de agrupamiento, donde usaron la transformada *wavelet* discreta (en inglés DWT, Discrete Wavelet Transform) de cuarto orden para la extracción de características de la señal ECG. Con estas características se agrupan los datos en 2 etapas, con el fin de reducir el costo computacional: primero, como etapa de *pre-clustering*, se implementaron algoritmos como *Dynamic Time Warping* (DTW) y un *Kmeans* modificado denominado *Kmedians*.

En 2008 Mohammadzadeh, Kamaledin y Mohebbi [15] se basaron en el algoritmo SVM para clasificar 6 tipos de arritmias realizando la siguiente secuencia: se remueve el ruido en la etapa de pre-procesamiento, se detectan los complejos QRS y se extrae la señal de variabilidad del ritmo cardíaco usando métodos lineales en el dominio del tiempo y la frecuencia y métodos no lineales; la reducción de dimensiones se realiza con un análisis discriminante generalizado; y por último se aplica un algoritmo SVM, realizando el *test* con dos métodos: uno contra todos (en inglés OAA, One Against Another) y uno contra uno (en inglés OAO, One Against One) y comparan los resultados obtenidos.

Mehta y Lingayat en [16] desarrollaron un algoritmo para la detección de complejos QRS en base al algoritmo SVM, y en [17] usaron un algoritmo SVM para detectar ondas P y ondas T en un electrocardiograma de 12 derivaciones. Para la detección de éstas ondas, Mehta y Lingayat realizaron la siguiente secuencia: pre-procesamiento de la señal ECG removiendo los complejos QRS; el entrenamiento de SVM se realizó con una matriz de m instancias, y n características que corresponden a cada derivación; luego realizaron un *test* para la detección de onda T, siendo esta clase etiquetada con 1's; luego se remueven las ondas T obtenidas de la señal ECG y se aplica el mismo procedimiento. En ambos artículos, usaron un algoritmo de optimización mínima secuencial (en inglés SMO, Sequential Minimal Optimization) modificado para realizar el entrenamiento del SVM para reducir el costo computacional.

Wen y colaboradores en 2008 [18], combinan dos métodos de redes neuronales, SOM y el controlador de la articulación del modelo cerebelosa, con el fin de obtener un algoritmo más robusto. Realizan la clasificación en base a las características del complejo QRS.

En 2009, Korürek y Nizam en [19] desarrollaron un algoritmo semi-supervisado, el cual luego del pre-procesamiento y extracción de características con métodos como Análisis de Componentes Principales (en inglés PCA, Principal Component Analysis)

y DWT, propusieron un algoritmo de *clustering*, basado en la forma como las hormigas crean sus colonias (algoritmo ACO). Luego del agrupamiento, implementaron KNN para asignar las clases a cada grupo. Por otra parte, implementaron un algoritmo con redes neuronales y se compararon los métodos, obteniendo mejores resultados con el algoritmo propuesto ACO.

Talbi y Charef [20] implementaron SOM para la detección de latidos PVC a partir del espectro del complejo QRS, el cual se interpola en dos intervalos de diferentes frecuencias, obteniendo de cada uno una pendiente distinta; cada pendiente es una característica del espacio de entrada del SOM.

En 2010, Chitta y Murty en [21] desarrollaron un algoritmo que consta de 2 etapas, cada una con el algoritmo *Kmeans*, donde el radio de los grupos los define el usuario. Se desarrolló este algoritmo con el fin de reducir costo el computacional, ya que agrupar todos los datos en una sola etapa es más demorado, así que ésta tarea se divide en dos etapas.

Mehta, Shete, Lingayat y Chouhanc en [22] implementan el algoritmo *Kmeans* para la detección de complejos QRS a partir de la señal ECG, formando dos grupos: un grupo con los latidos que tienen complejos QRS y otro grupo con los latidos que no tienen. La principal característica utilizada es la pendiente de la señal ECG.

En 2011, Homaeinezhada, Tavakkoli, Atyabi, Ghaffari y Ebrahimpour [23] se enfocaron en formar un algoritmo más robusto, combinando diferentes clasificadores: SVM, red neuronal probabilística (en inglés PNN, Probabilistic Neural Network) y 2 perceptrones multicapa con retropropagación (MLP-BP), y a partir de esta fusión clasificar 7 tipos de arritmias. Al ser el algoritmo más robusto, se obtiene un mejor desempeño pero el costo computacional aumenta. El procesamiento se realiza usando métodos de análisis en el tiempo y DWT. Cada clasificador se entrena por aparte y se fusionan los resultados comparandos con las etiquetas para realizar la predicción.

También se ha implementado SOM como método de pre-procesamiento de la señal ECG, como lo hizo Gacek [24] el cual usa este algoritmo para realizar una interfaz de usuario que visualiza los patrones en las regiones del mapa, a partir de los cuales ayuda a tomar decisiones sobre la selección de los algoritmos de clasificación a implementar, entre ellos usan otros métodos no supervisados como *Cmeans* difuso y *Kmeans*.

4. MÉTODOS

4.1. TRANSFORMADA WAVELET DISCRETA (DWT)

Una *wavelet* es una función oscilatoria, donde su energía esta concentrada en el tiempo para representar mejor señales transitorias y señales no estacionarias. Tiene características de un filtro pasa banda. Un aspecto importante del análisis *wavelet* es que captura información de localización en el tiempo y la frecuencia. La representación de *wavelet* en el dominio del tiempo es la función $\psi(t)$, también llamada *wavelet* madre o *kernel wavelet* con la cual se define su transformada [3].

La transformada *wavelet* representa una señal en una secuencia de coeficientes basados en bases ortogonales de ondas finitas (wavelet madre, trasladada y escalada en el tiempo). La función ψ le permite a la transformada ser integrable e invertible. A partir de esto, se define la transformada *wavelet* para señales continuas como la correlación entre la señal y la función ψ escalada y trasladada en el tiempo. Para aplicaciones prácticas se definió la DWT, la cual es un muestreo discreto de la transformada *wavelet*. El muestreo se realiza con el fin de reducir la redundancia [25]. La ecuación que define la DWT es

$$w_{j,k} = \int_{-\infty}^{\infty} x(t)2^{j/2}\psi(2^j t - k)dt \quad j,k \in \mathbb{Z},$$

donde $\psi(t)$ la define una familia de *wavelets*. Se han definido diferentes tipos de familias, tales como la familia Haar, Daubechies, Biortogonales, Coiflets, Symlets, Morlet, Meyer, entre otras [26]. A continuación se definen las familias Daubechies y *Symlets*, ya que en este trabajo de grado se hace mayor énfasis en estos tipos de familias.

4.1.1. *Wavelets Daubechies (DbN)*

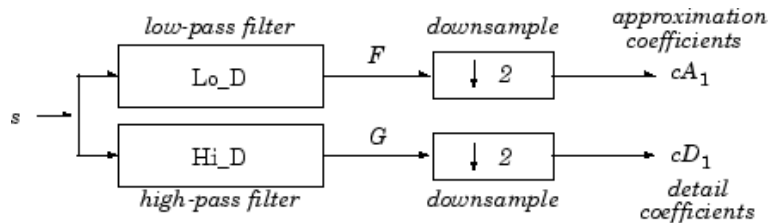
Es una familia de *wavelets* ortogonales que definen una DWT, inventadas por la matemática Ingrid Daubechies. Se caracterizan por tener un alto número de momentos de desvanecimiento para un ancho de apoyo dado. A cada tipo de *wavelet* de esta clase, también se le asigna una función de escala $\varphi(t)$, llamado *wavelet* padre; donde N es el orden (otros autores usan $2N$), el cual se refiere al número de momentos de desvanecimiento. Entre mayor sea el orden, más suaves son las gráficas de las funciones *wavelet* y las funciones escala. Una desventaja de esta familia es que tienen una morfología altamente asimétrica [3].

4.1.2. *Wavelets Symlets (SymN)*

Es una familia de *wavelets* casi simétricas, ortogonales y biortogonales propuestas por Ingrid Daubechies como modificaciones de la familia Db, por lo tanto sus propiedades son similares [26].

Las transformadas *wavelet* han llegado a ser una de las herramientas más útiles en aplicaciones para el procesamiento de señales. La DWT ha sido empleada en aplicaciones de: codificación, compresión de imágenes e interfaz de usuario [27]. Para la DWT, el algoritmo Mallat consiste en $\log_2 N$ etapas, donde el primer paso genera 2 conjuntos de coeficientes a partir de la señal de entrada, estos son: los coeficientes de aproximación cA_1 y los coeficientes de detalle cD_1 , los cuales resultan de convolucionar la señal de entrada con un filtro pasa-bajos y con un filtro pasa-altos respectivamente [28]. Éste esquema se muestra en la Figura 2.

Figura 2. Esquema de descomposición *wavelet* de un nivel



Fuente: The MathWorks, Introduction to Wavelet Families, 2013. [En línea]. Available: <http://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html#3->

1008627. [Último acceso: 20 04 2013].

4.2. ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

También conocido como un método de proyección, es un método de aprendizaje no supervisado desarrollado por Karl Pearson en el año de 1901, siendo este uno de los métodos pioneros para el Análisis Multivariable [29]. En este tipo de métodos, el objetivo es reducir un conjunto de datos n-dimensional en un espacio k-dimensional, siendo $k < n$, con la mínima pérdida de información y maximizando la varianza. La proyección de una observación x en dirección de un vector de proyección o componente principal W , es

$$z = W^T x, \quad (1)$$

siendo la dirección un factor importante, por lo que se requiere que la norma de la componente principal sea igual a 1. El error de proyección se define como la distancia perpendicular entre la observación y W [2]. De la ecuación 1 se tiene que la i -ésima proyección se define como

$$z_i = W_i^T x,$$

con la covarianza $\text{Cov}(x) = \Sigma$, que se calcula así

$$\Sigma = \frac{1}{m} \sum_{i=1}^n x_i x_i^T,$$

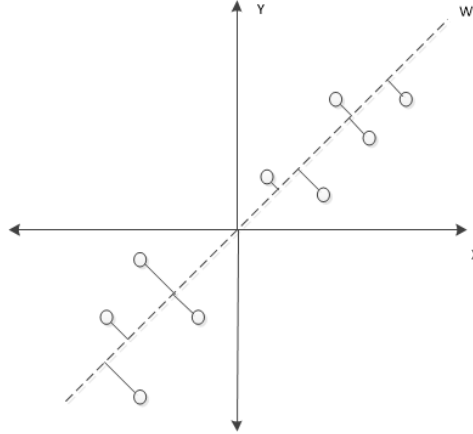
y se define la varianza como

$$\text{Var}(z_i) = W_i^T \Sigma W_i.$$

Este método se ha implementado en aplicaciones como: reducción de dimensiones, extracción de características, compresión de imágenes, entre otras. En la Figura 3

se ilustra el esquema general de PCA, donde se proyectan los datos de un espacio bidimensional en el vector W (unidimensional).

Figura 3. Reducción de 2 dimensiones a 1 con PCA



4.3. BASE DE DATOS

Se usaron las bases de datos del *MIT-BIH Arrhythmia (mitdb)* [30]. Sólo se consideraron latidos normales y PVC, los cuales son el principal enfoque de este trabajo. Se extrajeron, en total, 4234 latidos, de los cuales se tomaron 2538 latidos (60%) para el conjunto de entrenamiento, 850 latidos para el conjunto del *test* (20%) y 846 latidos para el conjunto de la validación cruzada (20%). Cada conjunto se construye con igual número de latidos normales (0's) y PVC (1's) seleccionados aleatoriamente para evitar un desbalance en las clases. En las Figuras 4 y 5 se muestra un ejemplo de un latido normal y de una PVC.

También se realiza la etapa de multi-clasificación con el fin de comparar el desempeño de cada algoritmo cuando se tienen más de dos clases de latidos. Para esta etapa, se clasifican 4 clases: latidos normales, PVC, APB y SVPB; los cuales se extraen de otras bases de datos adicionales, tales como: *Long Term Atrial Fibrillation (ltafdb)*, *MIT-BIH Long term ECG (ltdb)*, *MIT-BIH Normal Sinus Rhythm (nsrdb)* y *MIT-BIH Supraventricular Arrhythmia (svdb)* [30]. De estas bases se extraen 2524 latidos para el conjunto de entrenamiento, 840 latidos para el conjunto de validación cruzada y 840

Figura 4. Latido normal

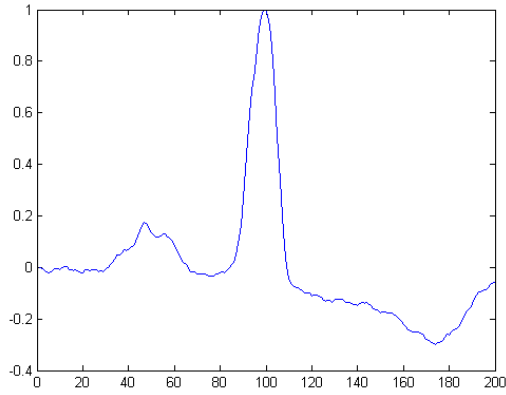
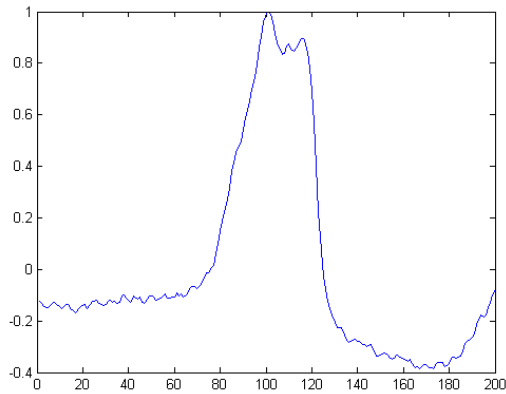


Figura 5. Latido PVC



latidos para el conjunto del *test*. Cada conjunto se construye con igual número de latidos de cada clase, seleccionados aleatoriamente.

4.4. PRE-PROCESAMIENTO

El pre-procesamiento de las señales extraídas de las bases de datos fue desarrollado por Bustamante C., Duque S., Orozco A. y Bustamante J. en [31], quienes usan DWT para la reducción de ruido, remover el ruido de la línea base y reducir la interferencia de línea de potencia.

En el algoritmo de reducción de ruido, descomponen la señal en escalas de frecuencia usando la Transformada *wavelet* Discreta Rápida y el umbral se toma con respecto al nivel de la relación señal a ruido. El umbral se aplica en los coeficientes de detalle cD1-cD8. Se remueven los coeficientes cA8 por lo que corresponde a las bajas frecuencias y contiene el ruido de la línea base.

Se realizó un análisis comparativo entre el método DWT con madres *wavelet* Db4, Db9, Sym4 y Sym9, y los métodos de mediana simple y mediana doble. Para éste análisis se usaron medidas de distorsión como la correlación, la diferencia de raíz media cuadrada (PRD) y la relación señal a ruido. Se obtuvo que la DWT tiene mejor desempeño y menor costo computacional.

Para la reducción de interferencia de línea de potencia se implementaron varios métodos como el filtro Notch, el filtro Ziriani, filtro Kalman y DWT, obteniendo que el filtro Kalman y DWT tienen un desempeño similar, pero DWT tiene menor costo computacional.

4.5. SEGMENTACIÓN

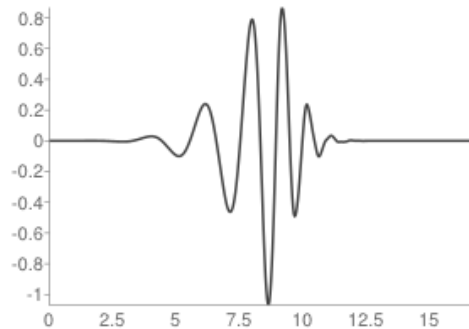
Esta etapa también fue desarrollada en [31], en el cual se implementa un algoritmo basado en transformada *wavelet* para detectar el pico R. La suma de los coeficientes cD2, cD3, cD4 y cD5 son usados para detectar el complejo QRS. Se calcula el valor absoluto de la señal reconstruida y los picos son detectados usando un umbral adaptativo. Luego de identificar los picos R, se extraen 200 puntos alrededor de cada pico R.

4.6. EXTRACCIÓN DE CARACTERÍSTICAS

De la etapa de segmentación, se conforma cada latido con los 200 puntos alrededor del pico R y amplitud normalizada. A partir de los latidos se crea un espacio de características, donde cada latido es una instancia y cada punto del latido representa una característica del espacio. Este espacio de características se le denomina espacio

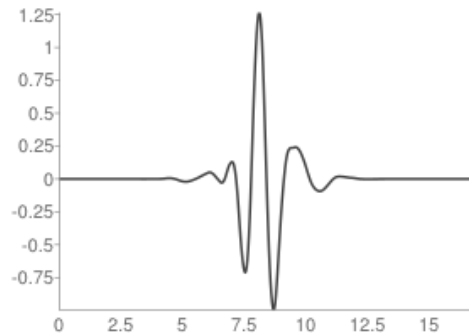
temporal. A partir de este espacio, se crean otros espacios de características basados en los métodos DWT y PCA, con el fin de evaluar los algoritmos y comparar sus resultados. En la etapa de pre-procesamiento, Db9 y Sym9 se obtuvo el mejor desempeño y un bajo costo computacional. Por esta razón, se seleccionaron éstos 2 métodos para esta etapa. Las Figuras 6 y 7 ilustran las morfologías de las funciones *wavelet* de la familia Daubechies y *Symlets* de orden 9 respectivamente.

Figura 6. Función *wavelet* ψ Db9



Fuente: F. Wasilewski, Wavelet Daubechies 9 (db9), Free CSS Templates, 2013. [En línea]. Available: <http://wavelets.pybytes.com/wavelet/db9/>. [Último acceso: 20 04 2013].

Figura 7. Función *wavelet* ψ Sym9



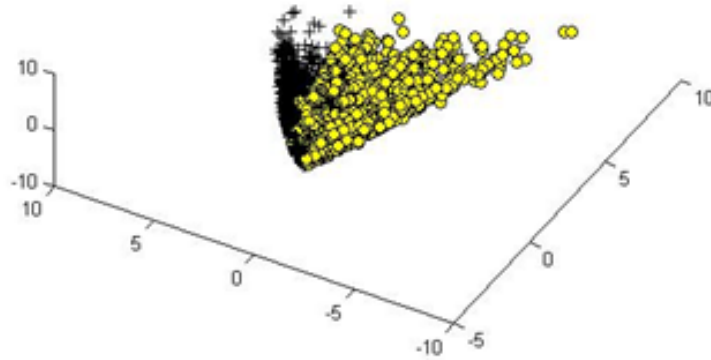
Fuente: F. Wasilewski, Wavelet Symlets 9 (sym9), Free CSS Templates, 2013. [En línea]. Available: <http://wavelets.pybytes.com/wavelet/sym9/>. [Último acceso: 20 04 2013].

Se crea un espacio de características con el método Db9 y otro con el método Sym9, a

partir de los coeficientes *wavelet* resultantes de la descomposición en 8 niveles de cada latido del espacio temporal, los cuales pertenecen a las sub-bandas $cA_8, cD_8, cD_7 \dots cD_1$; aunque los coeficientes cD_1 y cD_2 se descartan debido a que corresponden a las altas frecuencias de la señal y solo representan el 2% de la energía total. Como resultado se genera un espacio de 160 dimensiones por cada método.

A partir de los coeficientes seleccionados, se extraen los porcentajes de la energía contribuida por cada sub-banda y se almacenan como características para la creación de un nuevo espacio. Como resultado se genera un espacio de 7 dimensiones por cada método.

Figura 8. Espacio de las energías de las sub-bandas *wavelet* con 3 componentes principales



Se aplica PCA a los coeficientes *wavelet* del método Sym9 (ya que éste método obtuvo mayor desempeño que Db9 en la etapa de pre-procesamiento [31]) y se crea un espacio de características donde se reducen 160 dimensiones a 10 dimensiones, preservando así el 81,55% de la varianza. Además se crea otro espacio de características con las componentes principales de las energías de las sub-bandas. En la Figura 8 se muestra un ejemplo ilustrativo de PCA, donde se reduce este último espacio a 3 dimensiones para visualizar las 3 componentes con la máxima variabilidad, representando el 94,61% de ésta.

En resumen se crearon 7 espacios de características, tanto para los conjuntos con dos comportamientos como para los conjuntos con cuatro:

- El espacio temporal.
- El espacio de los coeficientes *wavelet* con el método Db9.
- El espacio de los coeficientes *wavelet* con el método Sym9.
- El espacio de las energías de las sub-bandas *wavelet* con el método Db9.
- El espacio de las energías de las sub-bandas *wavelet* con el método Sym9.
- El espacio con las componentes principales de los coeficientes *wavelet* con el método Sym9.
- El espacio con las componentes principales de las energías de las sub-bandas *wavelet* con el método Sym9.

4.6.1. *Espacio de características híbrido*

Se creó un espacio de características adicional con el objetivo de mejorar el desempeño de los métodos no supervisados. En este tipo de métodos, donde no se tienen las etiquetas de los datos, la eficiencia radica en las características del espacio de entrada. Estas características deben permitir al algoritmo identificar patrones únicos de cada clase.

Para la creación de este espacio, se observaron las morfologías de los latidos de la señal ECG temporal, como las que se ilustran en las figuras 4 y 5. De estas señales se identificaron los siguientes patrones:

- El ancho del complejo QRS es mayor en los latidos ventriculares.
- La onda S es más negativa en los latidos ventriculares.
- Se distingue claramente la onda P, la onda Q y algunas veces la onda T en los latidos normales.

Luego de varias pruebas se concluyó que las características más dominantes son: el ancho del pulso QRS y la amplitud de la onda S. Estas características se calculan como el área bajo la curva del valor absoluto del complejo QRS, y el área bajo la curva del valor absoluto de la onda S medido desde el nivel de polarización del latido.

También se evaluaron las características de los anteriores espacios definidos y se encontraron 3 características, de las componentes principales de los coeficientes *wavelet*, que ayudan a mejorar la caracterización de la señal ECG. De esta forma se crea el espacio

con las 5 mejores características, al que se le denomina espacio híbrido.

4.7. PARÁMETROS DE DESEMPEÑO

Para la clasificación de 2 clases, una gran variedad de medidas han sido propuestas. Se tienen cuatro posibles casos, como se muestra en la Tabla 1. Para un ejemplo positivo, si la predicción es positiva, entonces se dice que es un Verdadero Positivo (VP), pero si la predicción es negativa entonces se dice que es un Falso Negativo (FN); de igual forma en un ejemplo negativo, si la predicción también es negativa, se dice que es un Verdadero Negativo (VN), pero si la predicción es positiva, se dice que es un Falso Positivo (FP) [2]. Para la clasificación de dos clases, los resultados positivos hacen referencia a los pacientes enfermos (PVC) y los resultados negativos como pacientes sanos (latidos normales).

Tabla 1. Matriz de confusión para 2 clases

Resultado de la prueba	Predicción	
	Positivo	Negativo
Etiquetas		
Positivo	VP	FN
Negativo	FP	VN

Para nuestro trabajo de grado se usan las medidas que se muestran en la Tabla 2.

Comúnmente la precisión, también llamada predicción positiva, y el *recall* o *sensitivity* (traducido al español como sensibilidad), son medidas que se usan para evaluar el desempeño en el campo de recuperación de información (del inglés *Information Retrieval*). Consiste en que a partir de una base de datos de registros (en nuestro caso las bases de datos del MIT-BIH) se realice una petición, y un sistema (clasificador de dos clases) retorna un número de registros. En las bases de datos, hay registros que son relevantes y para una petición, el sistema puede recuperar algunas de ellas (VP) pero probablemente no todas (FN); también ocurre que recupera registros que no son relevantes (FP). Con esta información se calcula la predicción positiva como la fracción de registros recuperados que son relevantes, i.e. el número de registros recuperados y

relevantes dividido el número total de registros recuperados. Si la precisión es 1, todos los registros recuperados son relevantes, pero pueden quedar datos relevantes sin ser recuperados.

La sensibilidad se calcula como la fracción de registros relevantes que son recuperados, i.e. el número de registros recuperados y relevantes dividido por el número total de registros relevantes. Análogamente a la precisión, si la sensibilidad es 1, da a entender que todos los registros relevantes fueron recuperados pero puede que hayan registros recuperados que no sean relevantes. La especificidad da información de que tan bien se detectaron los ejemplos negativos [2].

El *F1 score* es una medida que considera la precisión y la sensibilidad. Puede ser interpretado como un promedio ponderado entre éstas; también se define como el promedio armónico de éstas. El *F1 score* es conocido en estadística como la proporción de un convenio específico aplicado a una clase específica, el cual se calcula como el número de verdaderos negativos dividido el número total de negativos [4].

Tabla 2. Medidas de desempeño

Nombre	Formula
Precisión	$\frac{VP}{VP + FP}$
Especificidad	$\frac{VN}{VN + FP}$
Sensibilidad	$\frac{VP}{VP + FN}$
F1 Score	$\frac{2 * precision * recall}{Precision + recall}$

Para una clasificación entre más de 2 clases, una forma de evaluar el desempeño es usando la matriz de confusión de clases, la cual es una matriz $K \times K$ (siendo K el número de clases) donde su entrada (i,j) contiene el número de instancias que pertenecen a C_i pero son asignadas a C_j . La otra forma es definir K problemas de dos clases, i.e. con el método OAO [2]. Éste último método es el que se emplea para este trabajo.

Un indicador del costo computacional del algoritmo es el tiempo de ejecución, el cual se mide con los comando tic-toc de MATLAB, usando una CPU con un procesador de 1.65 GHz AMD E-450 APU con 3GB de RAM y sistema operativo Windows 8 de 32 bits. Principalmente se mide el tiempo de ejecución de la predicción del algoritmo, ya que el tiempo de ejecución del entrenamiento no es relevante, debido a que en una aplicación en tiempo real se usa la predicción del algoritmo. En el caso de KNN, la Toolbox de MATLAB realiza el entrenamiento y la predicción en un solo comando, por lo cual se mide el tiempo de ejecución de ese comando.

5. ALGORITMOS

Se desarrollaron cuatro algoritmos para clasificar y detectar arritmias cardíacas según lo planteado en los objetivos del trabajo, dos de ellos basados en aprendizaje supervisado y dos en aprendizaje no supervisado, con el fin de observar que tan útiles son estos tipos de algoritmos a la hora de resolver problemas de clasificación en el área de procesamiento de señales biomédicas.

5.1. APRENDIZAJE DE MÁQUINA

El aprendizaje de datos ha sido investigado por filósofos a través de la historia, bajo el nombre de inferencia inductiva. Hasta el siglo XX se consideró imposible una inducción pura sin tener algún conocimiento a priori. Este problema se ha estudiado largamente desde el marco estadístico, e.g. Gauss propuso la idea de regresión de mínimos cuadrados en el siglo XVIII, mientras que la aproximación para la clasificación de Fisher en la década de los 30, todavía proporciona la base para el desarrollo de métodos y análisis [1].

Alan Turing propuso la idea de máquinas de aprendizaje en 1950. La idea de modelar problemas de aprendizaje como problemas de búsqueda en un espacio de hipótesis adecuado, es característico de un enfoque de la inteligencia artificial, por lo que el desarrollo de algoritmos de aprendizaje llegaron a ser un sub-campo importante de este tipo de inteligencia; eventualmente formando el área separada de aprendizaje de máquina. Existen varios tipos de aprendizaje de máquina: aprendizaje supervisado, no supervisado, aprendizaje reforzado, aprendizaje semi-supervisado, aprendizaje inductivo [1], aprendizaje activo [2], entre otros.

5.2. MÉTODOS DE APRENDIZAJE SUPERVISADO

En este tipo de aprendizaje, se conocen (algunas veces de forma aproximada) los valores de y para cada instancia del conjunto de entrenamiento Θ . Se asume que si puede encontrar una hipótesis, h , que se aproxime a y para los miembros de Θ , entonces se tiene que h es un buen supuesto para y , especialmente si Θ es grande [32].

Existen varios modelos que han sido implementados en el campo biomédico, entre las cuales se encuentran: árboles de decisión, máquinas de soporte vectorial, *Naive Bayes*, vecinos más cercanos y redes neuronales. Éstas últimas se han trabajado ampliamente desde 1943 cuando McCulloch y Pitts modelaron el procesamiento de información que tiene una neurona real, donde más tarde se empiezan a trabajar redes más complejas, a desarrollar nuevos modelos como el perceptrón de Rosenblatt, modelos multicapa [33]. Métodos como la dimensión de Vapnik-Chervonenkis (VC), propuesta en la década de los 70 y el modelo *Probably Approximately Correct* (PAC) propuesto por Valiant en 1984[2] también se reportan en la literatura.

A continuación se explican ampliamente los métodos supervisados implementados.

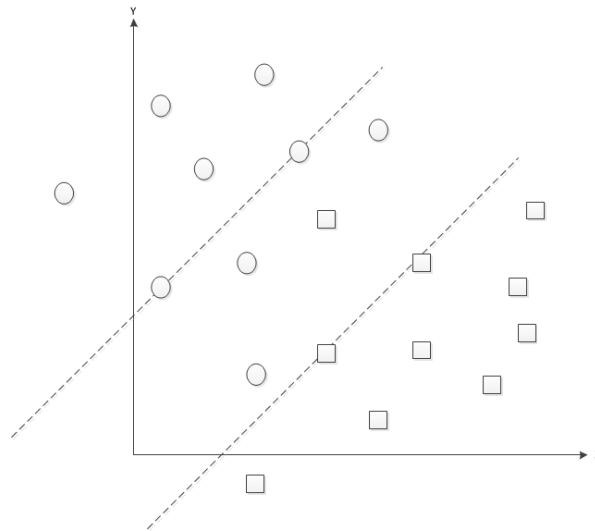
5.2.1. Máquinas de Soporte Vectorial (SVM)

Las máquinas de soporte vectorial son un tipo de algoritmos caracterizados por el uso de *kernels*, la ausencia de mínimos locales, la dispersión de la solución y la capacidad de control obtenidas al actuar sobre la margen u otras cantidades de dimensiones independientes como el número de soportes vectoriales. Las máquinas de soporte vectorial fueron inventados por Vladimir Vapnik y colaboradores, introduciendo el tema en 1992 en la conferencia COLT; aunque características como los hiperplanos de margen grande en espacios de entrada, *kernels* y métodos estadísticos ya se habían trabajado en aprendizaje de máquina desde los años 60; y fue hasta 1992 que estos elementos fueron reunidos para formar las primeras máquinas de soporte vectorial. En 1964 se introdujo la interpretación geométrica de los *kernels* como productos internos en un espacio de características. En 1995 se presentó la versión de margen suave por Cortes y Vapnik [34].

Este método selecciona el hiperplano que maximiza la distancia y la margen entre las clases [8]. Dado un espacio de entrada de dimensión m , se calcula el vector parámetro $f \in R^{(m-1)}$, que depende de la proximidad de los vectores en el conjunto de entrenamiento. Para calcular la proximidad se usa una función *kernel*, que para este trabajo es el *kernel* gaussiano. La predicción se calcula con el producto interno entre el vector f y los pesos del modelo θ^T , donde la predicción es 1 si el producto es mayor a 0.

En la Figura 9 se observa la línea de separación entre el grupo de cuadrados y círculos. Los que están fuera de la margen de error son los que fueron clasificados correctamente, los que están dentro son los que fueron clasificados erróneamente y los que se encuentran en las fronteras son los soportes vectoriales.

Figura 9. Clasificación de clases por medio de soportes vectoriales



El *kernel* Guassiano, definido por la ecuación

$$GaussianKernel(x^{(i)}, x^{(j)}) = e^{-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}},$$

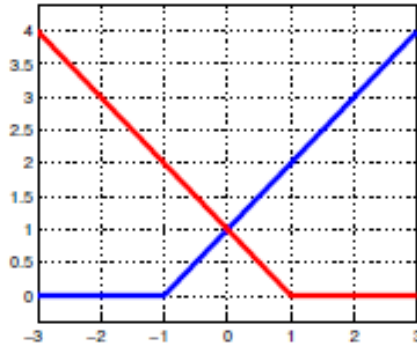
se puede entender como una función de similitud que mide la distancia entre un par de ejemplos del espacio de entrada. Este kernel también está parametrizado por un ancho de banda σ , el cual determina que tan rápido decrece la similitud métrica al estar los ejemplos más separados. Los resultados de este *kernel* se guardan en el vector f .

El algoritmo SVM minimiza la función costo

$$\min_{\theta} C \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^m \theta_j^2$$

donde las funciones $\text{cost}_0(z)$ y $\text{cost}_1(z)$ representan las funciones de pérdida *hinge* simétricas que se muestran en la Figura 10, donde el eje horizontal representa el producto $\theta^T f^{(i)}$. Si $y=1$ se quiere que $z \geq 1$ y si $y=0$ entonces que $z \leq -1$.

Figura 10. Funciones $\text{cost}_0(z)$ (línea azul) y $\text{cost}_1(z)$ (línea roja)



Fuente: N. Vasconcelos y H. Masnadi-Shirazi, Risk minimization, probability elicitation, and cost-sensitive SVMs, 2010. [En línea]. Available: http://www.svcl.ucsd.edu/publications/conference/2010/icml2010/FinalAccepted_ICML2010_CostSVM.pdf. [Último acceso: 15 04 2013].

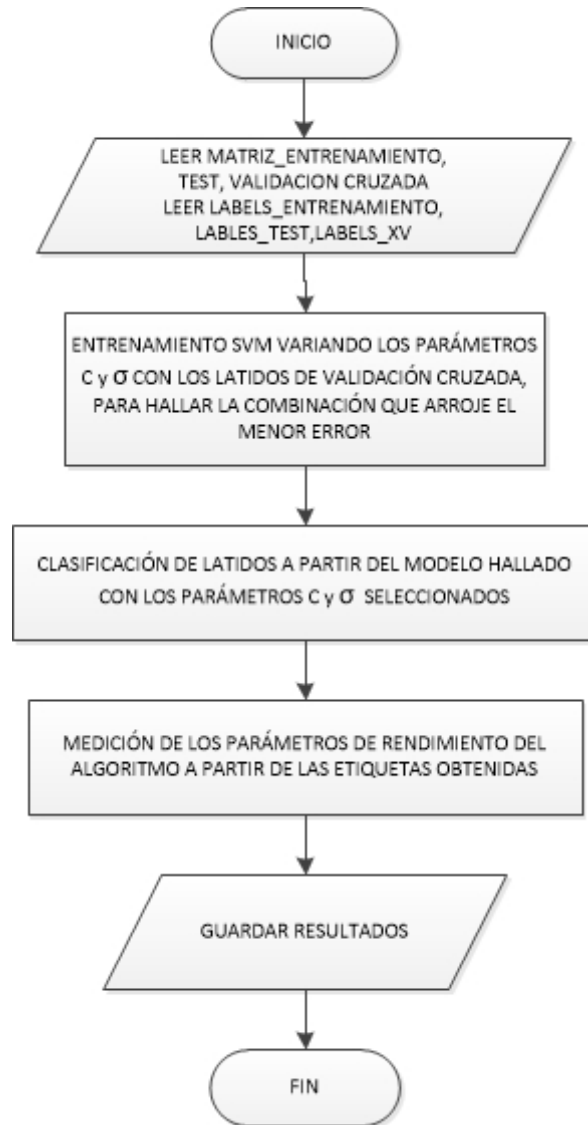
El vector y es el vector de salida (el vector de las predicciones). C es el parámetro de regularización, donde su función es controlar el sobreajuste, el cual indica que si al evaluar un conjunto de datos el desempeño es muy bueno, entonces existe la posibilidad de que para otros conjuntos de datos éste no sea tan bueno.

La varianza indica que tanto varían las predicciones para un punto dado entre varias repeticiones del modelo; y el *Bias* es la diferencia entre el valor esperado de la predicción y el valor correcto. Se requieren valores no muy pequeños ni muy grandes de C y σ para que el *Bias* y la varianza sean mínimos. Si los datos del entrenamiento dan un sobreajuste entonces se busca disminuir C y aumentar σ .

Algoritmo SVM: es implementado en MATLAB usando una versión simplificada del

algoritmo SMO del curso *online Machine Learning* de Andrew NG de la Universidad de Stanford [35]. En la Figura 11 se ilustra el diagrama de flujo general del algoritmo.

Figura 11. Diagrama de flujo del algoritmo SVM para la clasificación de arritmias ventriculares



Para la etapa de entrenamiento del SVM, se emplea un algoritmo SMO simplificado, con el fin de resolver el problema de programación cuadrática (QP) del SVM sin almacenamiento extra de matrices ni usar la optimización numérica del problema QP. El algoritmo SMO descompone el problema QP en sub-problemas QP usando el teorema de Osuna para asegurar la convergencia [36].

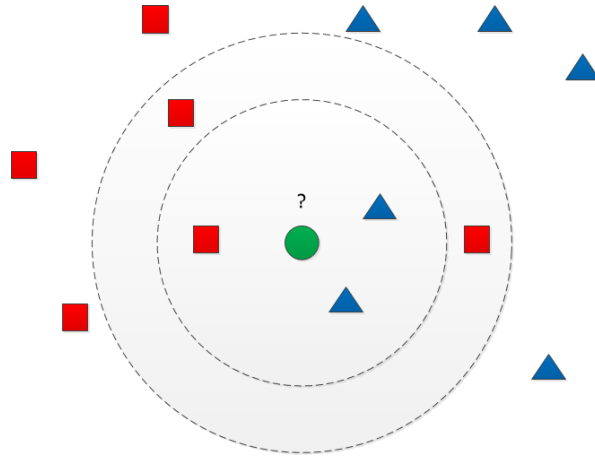
5.2.2. *K-Nearest Neighbor (KNN)*

El algoritmo KNN es uno de los métodos de clasificación más simples, fue desarrollado de la necesidad de realizar un análisis discriminante cuando las estimaciones paramétricas confiables de densidades de probabilidad son desconocidos o difíciles de determinar [6]. Este método fue propuesto en 1951 por Fix y Hodges en un reporte de medicina no publicado de la escuela de aviación de las fuerzas aéreas de US. Este reporte contenía trabajo de investigación sobre el análisis discriminante no paramétrico, estimación de la densidad de probabilidad y otros conceptos que fueron introducidos por primera vez [37]. luego en 1967 se trabajaron algunas de las propiedades del KNN, e.g. si $k=1$ y $n \rightarrow \infty$ el error de clasificación tiene su cota superior al doble de la tasa de error de *Bayes*. Y en la década de los 70 hubo una larga línea de investigación sobre las propiedades de este método, como refinamientos con respecto a tasa de error de *Bayes* y aproximaciones de distancias ponderadas. Pero como se requería de altos registros de memoria y computación, el método no fue tan conocido hasta 1991, que gracias al avance del procesamiento en paralelo, y los bajos costos de memoria y computación, éste método empezó a ser ampliamente usado [2].

KNN es un método de clasificación estadístico usado para clasificar objetos, basado en los ejemplos del entrenamiento más cercanos en el espacio de características. Se le asigna al objeto la clase que es más común dentro de sus vecinos más cercanos [38]. E.g., en la Figura 12 se quiere clasificar un objeto de clase desconocida, en este caso el círculo verde; entonces si tomamos los 3 vecinos más cercanos ($k=3$), la clase a la que pertenece el círculo es a la de los triángulos, mientras que si $k=5$, ya pertenece a la clase de los cuadrados, y así sucesivamente si se quieren tener en cuenta más vecinos. Si k es muy grande, decrece el riesgo de que la predicción se vea influenciado por un patrón de entrenamiento ruidoso cercano a la observación, pero también reduce la agudeza del método, aunque la estimación es mejor si el número de ejemplos alrededor de la observación es mayor [32].

Uno de los parámetros de este método es la Función que determina la distancia métrica entre dos instancias, debido a la necesidad de conocer los vecinos más cercanos. Se requieren de varias funciones ya que la distancia euclidiana no siempre es la más efectiva en un espacio de características de más de 2 dimensiones. El algoritmo toma cada instancia del conjunto de entrenamiento y mide la distancia entre ésta y todas las

Figura 12. Decisión basada en KNN



instancias del conjunto muestra, y luego registra las k menores distancias para realizar la clasificación. Las distancias métricas que se usan para este algoritmo son las siguientes:

Distancia Euclidiana Cuadrada (*SqEuclidean*): La distancia entre 2 puntos x y y del espacio euclidiano m -dimensional se define como la suma de las diferencias al cuadrado

$$\sum_{i=1}^m (x_i - y_i)^2, \quad (2)$$

es una función simple y es una de las más usadas.

Distancia *Cityblock*:

Ésta distancia se define como la suma del valor absoluto de las diferencias entre dos puntos del espacio.

$$\sum_{i=1}^m |x_i - y_i| \quad (3)$$

Distancia Correlación:

Ésta distancia se define como

$$d_{\text{correlacion}} = 1 - \frac{S_{xy}}{S_x S_y} \quad (4)$$

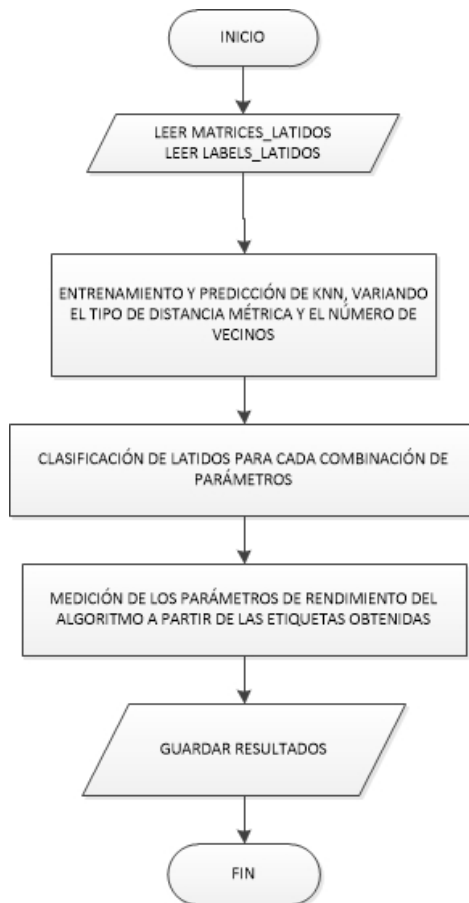
$$S_{xy} = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x}_i)(y_i - \bar{y}_i) \quad (5)$$

$$S_x = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x}_i)^2} \quad (6)$$

donde la ecuación 5 representa la muestra de covarianza, y la ecuación 6 representa las muestras de las desviaciones estándar de x y y [39].

Algoritmo KNN: es implementado en MATLAB usando las *bioinformatics Toolbox* para obtener el resultado de cada latido. Al igual que SVM, se ilustra el diagrama de flujo del algoritmo en la Figura 13.

Figura 13. Diagrama de flujo del algoritmo KNN para la clasificación de arritmias ventriculares



5.3. MÉTODOS DE APRENDIZAJE NO SUPERVISADO

En muchas aplicaciones no se cuentan con las etiquetas del conjunto de datos, ya que puede ser costoso, propenso a errores y algunas veces imposible. En estos casos se recurre al aprendizaje no supervisado, donde solo se tienen los datos de entrada. El objetivo del aprendizaje no supervisado es encontrar los patrones que ocurren con más frecuencia dentro del espacio de entrada y ver como se comportan. En estadística es llamado estimación de densidad [5].

Uno de los métodos más comunes para la estimación de densidad es el *clustering* o agrupamiento, donde el objetivo es agrupar los patrones similares.

A lo largo de los años se ha trabajado ampliamente en métodos de aprendizaje no supervisado, como la regla Hebb, en 1949, el cual enlaza métodos estadísticos para experimentos neurofisiológicos en plasticidad; en 1970, David Marr realizó un postulado de aprendizaje no supervisado sobre el objetivo del aprendizaje en su modelo de neocortex. Otros trabajos como el de Hinton y Sejnowski, que en 1986 inventaron un modelo de aprendizaje llamado la máquina de Boltzmann, la cual importó muchos de los conceptos estadísticos que hoy se trabajan en aprendizaje no supervisado [40].

En cuanto a métodos de clustering [2], Titterington, Smith y Makov en 1985 dedicaron textos a los modelos de mezcla, que son usados frecuentemente en estadística; en 1988 [41], Jain y Bubes recopilan trabajos realizados sobre clustering en su libro *Algorithms for Clustering Data*; en 1997 se discutieron los desarrollos recientes en el algoritmo de *Expectation-Maximization* (EM) por McLachlan y Krishnan.

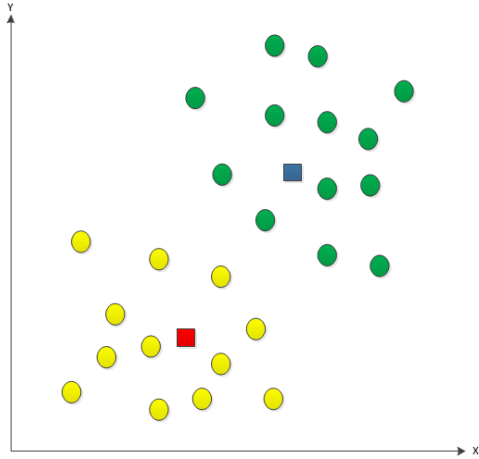
A continuación, se explican ampliamente los métodos de aprendizaje no supervisados implementados.

5.3.1. KMEANS

Kmeans es uno de los métodos de *clustering* más conocidos. Éste término fue utilizado por primera vez por MacQueen en 1967 [42] aunque Steinhaus, en 1956, trabajó en un algoritmo que realiza la división de un sólido n-dimensional en k partes tales que la suma de las clases específicas de las inercias fueran minimizadas; en 1957, Lloyd propuso

el algoritmo estándar en su trabajo en modulación por codificación de pulsos, que es una transmisión de información cuantizada, y planteó transmitir una señal aleatoria X n -dimensional con densidad $f(x)$; Forgy en 1965 propone un algoritmo discreto [43]. Jain y Dubes en [41] tienen una historia más detallada de *Kmeans* sobre sus diversas variaciones y modificaciones.

Figura 14. Formación de grupos o clusters con *Kmeans*



Dado un espacio de entrada m -dimensional, el objetivo de *Kmeans* es particionarlo en k grupos de tal manera que los puntos de un grupo sean similares entre sí, e.g. en la Figura 14, donde los centroides son representados por los cuadrados y los datos por los círculos, el algoritmo asigna el grupo verde a los datos más cercanos al centroide azul, y el grupo amarillo a los datos más cercanos al centroide rojo. Los centroides o vectores prototipo se definen como $\mu_1, \mu_2, \dots, \mu_k$, y el vector indicador como r_{ij} , el cual es 1 si el x_i es asignado al grupo j . Por lo tanto, para agrupar los datos se requiere minimizar la función costo dada por la ecuación

$$J(r, \mu) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^k r_{ij} \|x_i - \mu_j\|^2, \quad (7)$$

que a su vez minimiza la distancia de cada punto a partir del centroide. $\|\cdot\|^2$ denota la norma euclidiana cuadrada. El objetivo es encontrar r y μ , pero al ser la ecuación 7 de orden 2, i.e. de dos variables, entonces se minimiza la función en 2 etapas:

Primera etapa (asignación de grupos): se mantiene fijo μ y se determina r .

Por consiguiente, la minimización se descompone en m-problemas independientes. La solución para el i-ésimo punto x_i puede ser encontrado resolviendo la siguiente ecuación

$$r_{ij} = 1 \text{ si } j = \operatorname{argmin}_{j'} \|x_i - \mu_{j'}\|^2, \quad (8)$$

y de lo contrario $r_{ij} = 0$. La ecuación 8 encuentra el índice donde la norma euclidiana cuadrada es mínima.

Segunda etapa (actualizar los centros): se mantiene fija la r y se determina μ . Por consiguiente, la función J es cuadrática de μ . Para la minimización se iguala a 0 la derivada de la función J con respecto a μ_j :

$$\sum_{i=1}^m r_{ij} \|x_i - \mu_j\| = 0 \text{ para cada grupo,}$$

simplificando se tiene que

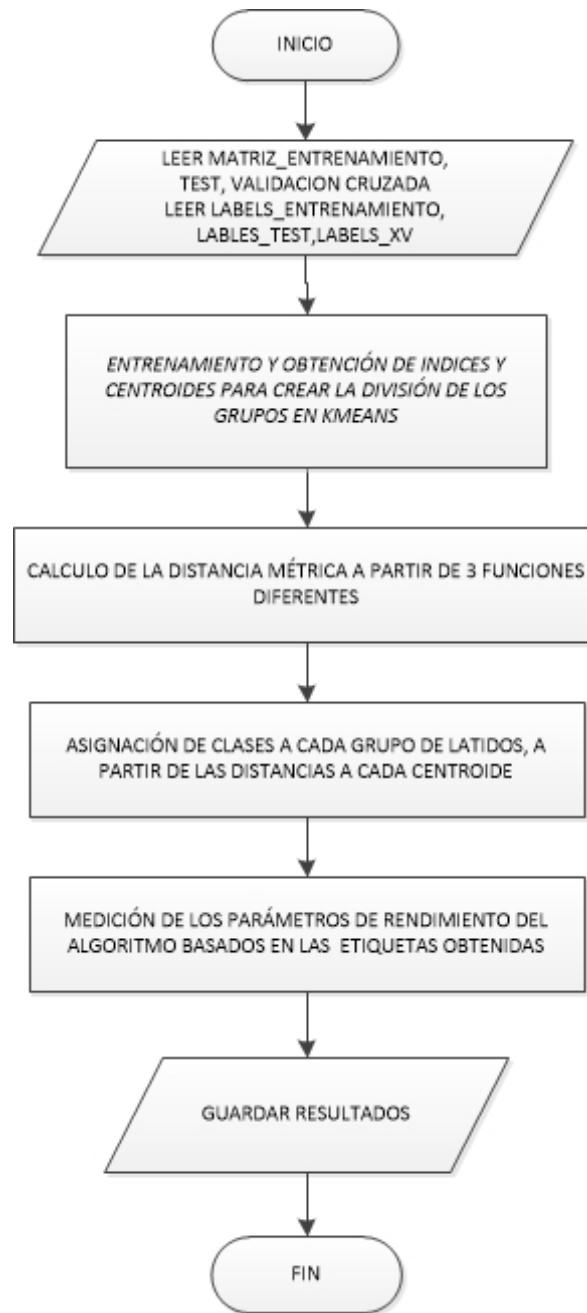
$$\mu_j = \frac{\sum_{i=1} r_{ij} x_i}{\sum_{i=1} r_{ij}} \quad (9)$$

De la ecuación 9 se dice que μ_j es el promedio de los datos asignados al grupo j. El algoritmo se detiene cuando la asignación de grupos no cambie significativamente.

Este método presenta dos inconvenientes. Primero, es muy sensible a la elección de los centroides iniciales, para ello se han desarrollado un número de heurísticas prácticas, aunque usualmente se recurren a otras modalidades. Una de ellas es elegir aleatoriamente k puntos del conjunto de datos y asignarlos como centroides, o escoger los k puntos más alejados del conjunto de datos. Segundo, se hace difícil la asignación de cada punto a un centroide, por esta razón existen variantes donde se reemplaza $r_{ij} \in [0, 1]$ por la probabilidad de que un x_i dado pertenezca al grupo j [5]. Para el cálculo de las distancias entre el punto y el centroide, se usan las ecuaciones 2,3 y 4 de la página 42.

Algoritmo Kmeans: es implementado en MATLAB usando las *Statistics Toolbox* para obtener el resultado de cada latido. El diagrama de flujo general del algoritmo se muestra en la Figura 15.

Figura 15. Diagrama de flujo del algoritmo *Kmeans* para la clasificación de arritmias ventriculares



5.3.2. Mapas Auto-Organizados (SOM)

En 1982, Kohonen, basado en evidencias a nivel cerebral presentó un modelo de red denominado mapas auto-organizados o red Kohonen. Este método surge de los primeros

modelos de redes neuronales, destacándose los modelos de memoria asociativa y aprendizaje adaptativo, con el fin de explicar la organización espacial de las funciones del cerebro, como se observaba principalmente en la corteza cerebral. Sin embargo, ya habían trabajos anteriores en esa área [7].

SOM es un método que distribuye los datos de un espacio de entrada en una rejilla, usualmente bidimensional, de tal forma que los datos con patrones similares se agrupan en regiones. SOM también se conoce como un aprendizaje competitivo debido a que las neuronas compiten unas contra otras con el fin de llevar a cabo una función; solo una de las neuronas sale vencedora, activándose y anulando las demás, las cuales son forzadas a sus respuestas mínimas.

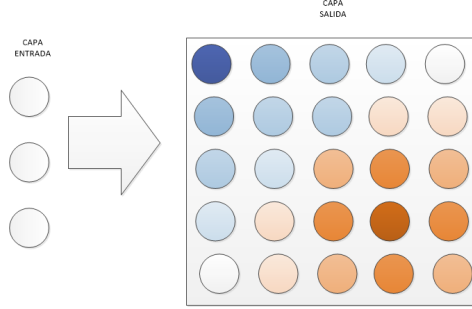
El modelo SOM se compone de 2 capas de neuronas: la capa de entrada, la cual recibe los datos del espacio de entrada y los transmite a la capa de salida, esta se encarga procesar la información y formar las regiones en el mapa. Las neuronas de la capa de salida se organizan en un mapa bidimensional como se dijo antes. Cada neurona de entrada i esta conectada con cada neurona de salida j mediante un peso w_{ij} , el cual se almacena en un vector de pesos W_j llamado vector de referencia o *codebook*. Mediante este proceso, se realiza una reducción de dimensiones debido a que se proyecta un espacio m -dimensional en un mapa bidimensional.

Cada neurona de la salida tiene influencia sobre las neuronas vecinas o las neuronas adyacentes N_j de la neurona j . E.g. si se activa una neurona, las neuronas vecinas también se activan pero con un peso menor, donde su nivel de activación depende de la distancia que las separan, lo cual indica que se requiere de una función distancia. Por lo tanto, la topología del mapa también influye en la activación de neuronas, producto de la competencia; las que se usan con más frecuencias son la rectangular y hexagonal. El algoritmo de SOM al igual que *Kmeans*, se itera varias veces hasta que la variación del *codebook* no sea tan significativa. Éstos vectores de referencia tienden a acercarse a las zonas con mayor densidad de datos [44].

En la Figura 16 se visualiza un ejemplo básico de SOM, donde se tienen las neuronas de la capa de entrada y se tienen los pesos en el mapa bidimensional formando 2 grupos: un grupo de neuronas en tonos de azul y el otro grupo en tonos de café. Las neuronas con los tonos más fuertes son las que tienen mayor peso dentro del grupo (centroides) y las neuronas vecinas tienen un color más claro que los centroides y por ende menor

peso.

Figura 16. Formación de grupos a partir de SOM en un mapa bidimensional



En su proceso de aprendizaje, SOM toma un vector aleatorio del conjunto de datos y mide su similitud con los vectores del *codebook*, con el fin de encontrar el vector más próximo o también llamado *Best Matching Unit* (BMU), los otros vectores se actualizan; para este fin se usa comúnmente la distancia euclidiana (ver ecuación 2). El BMU y sus vecinos se mueven cerca del vector escogido del conjunto de datos a razón de una tasa de aprendizaje α . Esta tasa de aprendizaje decrece gradualmente a cero (depende de su magnitud la rapidez con la que decrece) mientras se actualizan las neuronas del mapa. También reduce el radio de vecindad [44]. El siguiente valor del *codebook*, $m_i(t + 1)$ se actualiza de forma iterativa, tomando en cuenta el valor del *codebook* anterior $m_i(t)$ y un nuevo vector del conjunto de datos $x(t)$,

$$m_i(t + 1) = m_i(t) + \alpha(t)h_{ci}(t)(x(t) - m_i(t)),$$

siendo $\alpha(t)$ la tasa de aprendizaje o tamaño de la corrección; su valor decrece con el índice de paso t . El índice i se refiere al *codebook* en proceso, y c al índice del *codebook* más próximo a $x(t)$. $h_{ci}(t)$ es el *kernel* usado como factor de escala, también llamado función de vecindad, el cual es igual a 1 si y solo si $i = c$ y su valor decrece cuando la distancia entre los *codebooks* m_i y m_c aumenta en el mapa. El ancho espacial del *kernel* en el mapa debe disminuir con el índice de paso t . Este proceso se repite hasta que la distribución en el mapa no cambie de forma significativa. Los *kernels* y la inicialización de los *codebooks* son factores que determinan la convergencia del algoritmo, por lo que han sido temas muy discutidos en este método. Aunque el algoritmo descrito anteriormente funciona y ha sido implementado numerosas veces, se encontró que el esquema del mapa *Batch* produce resultados similares y con menor costo computacional.

Se tiene que para cada cada nodo j en el mapa, el promedio \bar{x}_j del vector de entrada $x(t)$ se forma de tener m_j como el *codebook* más cercano. De esta forma los nuevos *codebooks* son calculados así,

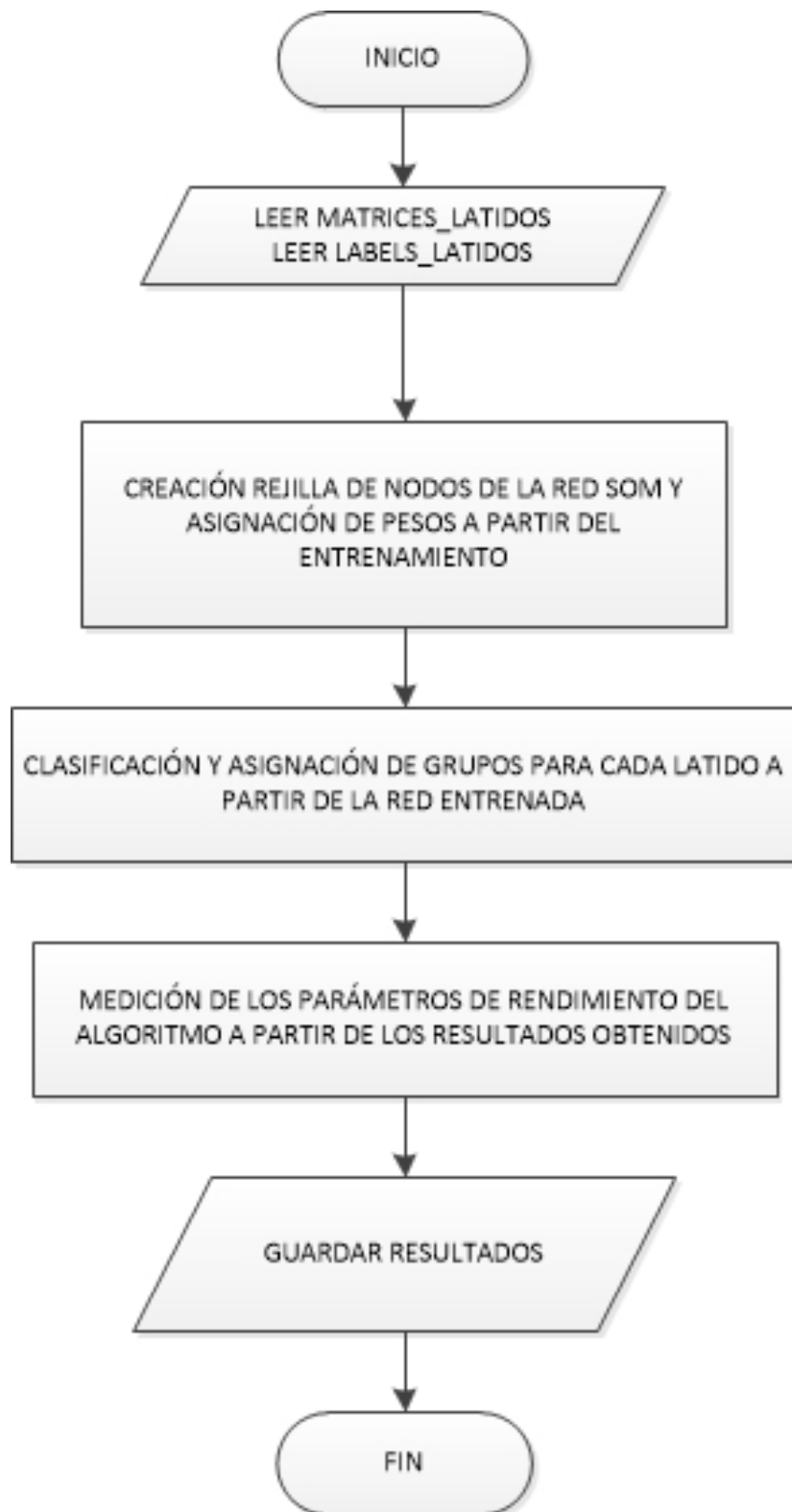
$$m_i = \frac{\sum_j n_j h_{ij} \bar{x}_j}{\sum_j n_j h_{ij}},$$

donde n_j es el número de entradas asignadas al nodo j , y el índice j hace parte de los nodos que se encuentran en la vecindad del nodo i [7].

Algoritmo SOM: es implementado en MATLAB usando la SOM Toolbox 2.0 [45], con el fin de obtener el resultado de cada latido y visualizar su distribución en el mapa de pesos. El esquema general del algoritmo se muestra en la Figura 17.

A continuación se ilustra un diagrama de flujos simplificado del algoritmo SOM en la Figura 17. Se realiza el entrenamiento a partir de la estructura creada, donde los parámetros que buscan optimizar el algoritmo son los que están directamente relacionados con la construcción del mapa; entre los parámetros más destacados están: el tipo del algoritmo (*batch* o secuencial), tamaño del mapa, duración del entrenamiento (número de etapas). Con el cálculo de las BMU, se conocen los latidos que compiten en cada celda del mapa; puede ocurrir que en una sola celda compitan muchos latidos y que otras queden vacías.

Figura 17. Diagrama de flujo del algoritmo SOM para la clasificación de arritmias ventriculares



6. RESULTADOS

A continuación, se muestran los resultados obtenidos de los algoritmos SVM, KNN, *Kmeans* y SOM. Para cada uno de éstos algoritmos se comparan, en cada tabla, las medidas de desempeño de 8 espacios de características diferentes para la etapa de clasificación binaria. Para la etapa de multi-clasificación, se consignan los resultados del espacio con el mejor desempeño de cada algoritmo, teniendo en cuenta las siguientes connotaciones: N-V (Normales contra PVC), N-A (Normales contra APB), N-S (Normales contra SVPB), V-A (PVC contra APB), V-S (PVC contra SVPB) y A-S (APB contra SVPB). En la Figura 18 se ilustra cómo el algoritmo *Kmeans*, con las 2 primeras dimensiones del espacio híbrido correspondientes al área bajo la curva del valor absoluto de la onda S y del complejo QRS, separa los latidos normales de los PVC. De igual forma se ilustra en la Figura 19 como *Kmeans* separa 4 clases de latidos. En la Figura 20 se ilustra, para el espacio híbrido, la agrupación de los latidos normales y PVC en un mapa creado por SOM. En la Figura 21 se ilustra la agrupación de 4 clases de latidos en el mapa. Los mapas creados por SOM muestran la distribución de pesos (*codebooks*) en el mapa con una escala de grises, donde el negro indica el mayor peso. Los valores de dichos pesos están ilustrados en las Figuras 20 y 21.

Tabla 3. Resultados del algoritmo SVM con 8 espacios de características diferentes

Espacio de Características	Parámetros	Precisión del Test (%)	Predicción Positiva (%)	Especificidad (%)	Sensibilidad (%)	<i>F1 Score</i>	Tiempo de ejecución (ms)
1. Temporal	C=10 y $\sigma=0,3$	98,94	99,29	99,29	98,59	0,9894	0,78
2. Coeficientes <i>wavelet</i> (Db9)	C=1 y $\sigma=0,5$	97,8824	99,04	99,06	99,04	0,9786	1,35
3. Coeficientes <i>wavelet</i> (Sym9)	C=10 y $\sigma=0,7$	97,2941	99,02	99,06	95,53	0,9725	0,94
4. Energías de las sub-bandas <i>wavelet</i> (Db9)	C=0,1 y $\sigma=0,5$	95,0588	95,43	95,53	94,59	0,9504	0,8
5. Energías de las sub-bandas <i>wavelet</i> (Sym9)	C=10 y $\sigma=1$	92,9411	93,35	93,41	92,47	0,9291	0,47
6. Componentes Principales de las Energías de las sub-bandas <i>wavelet</i> (Sym9)	C=1 y $\sigma=0,5$	95,647	93,69	93,41	97,88	0,9574	0,26
7. Componentes Principales de los Coeficientes <i>wavelet</i> (sym9)	C=1 y $\sigma=0,7$	96,82	97,16	97,18	96,47	0,9681	0,56
8. Híbrido	C=0,3 y $\sigma=1$	96,1176	98,76	98,82	93,41	0,9601	0,626

Tabla 4. Resultados del algoritmo KNN con 8 espacios de características diferentes

Espacio de Características	Parámetros	Precisión del Test (%)	Predicción Positiva (%)	Especificidad (%)	Sensibilidad (%)	$F1$ Score	Tiempo de ejecución (ms)
1. Temporal	K=1 y Función Correlación	98,588	98,135	98,118	99,059	0,986	14,55
2. Coeficientes <i>wavelet</i> (Db9)	K=3 y Función <i>sqEuclidean</i>	97,0588	97,39	97,41	96,71	0,9705	11,23
3. Coeficientes <i>wavelet</i> (Sym9)	K=1 y Función Correlación	96,9412	96,5	96,47	97,41	0,9696	1,63
4. Energías de las sub-bandas <i>wavelet</i> (Db9)	K=3 y Función <i>sqEuclidean</i>	95,6471	96,17	96,24	95,06	0,9562	1,26
5. Energías de las sub-bandas <i>wavelet</i> (Sym9)	K=3 y Función <i>Cityblock</i>	94,1176	94,75	94,82	93,41	0,9408	1,23
6. Componentes Principales de las Energías de las sub-bandas <i>wavelet</i> (Sym9)	K=3 y Función <i>sqEuclidean</i>	95,8824	95,99	96	95,76	0,9588	1,29
7. Componentes Principales de los Coeficientes <i>wavelet</i> (sym9)	K=3 y Función <i>sqEuclidean</i>	96,7059	96,29	96,24	97,18	0,9672	1,28
8. Híbrido	K=3 y Función <i>Cityblock</i>	96,7059	97,60	97,65	95,77	0,9668	0,903

Tabla 5. Resultados del algoritmo *Kmeans* con 8 espacios de características diferentes

Espacio de Características	Parámetros	Precisión del Test (%)	Predicción Positiva (%)	Especificidad (%)	Sensibilidad (%)	<i>F1 Score</i>	Tiempo de ejecución (μs)
1. Temporal	Función <i>City-block</i>	74,235	99,05	99,52	48,94	0,6551	22
2. Coeficientes <i>wavelet</i> (Db9)	Función <i>City-block</i>	54,353	54,59	56,94	51,77	0,5314	25,88
3. Coeficientes <i>wavelet</i> (Sym9)	Función Coseno	58,82	58,43	56,47	61,18	0,5977	20
4. Energías de las sub-bandas <i>wavelet</i> (Db9)	Función relación	61,765	58,01	38,35	85,18	0,6902	17,88
5. Energías de las sub-bandas <i>wavelet</i> (Sym9)	Función relación	53,176	51,81	15,53	90,82	0,6598	20,82
6. Componentes Principales de las Energías de las sub-bandas <i>wavelet</i> (sym9)	Función relación	61,059	59,48	52,71	69,41	0,6406	22
7. Componentes Principales de los Coeficientes <i>wavelet</i> (sym9)	Función relación	54,471	54,55	55,29	53,65	0,5409	19,29
8. Híbrido	Función <i>City-block</i>	95,176	97,29	97,41	92,94	0,9507	18,71

Tabla 6. Resultados del algoritmo SOM con 8 espacios de características diferentes

Espacio de Características	Parámetros	Precisión del Test (%)	Predicción Positiva (%)	Especificidad (%)	Sensibilidad (%)	F1 Score	Tiempo de ejecución (μs)
1. Temporal	Entrenamiento largo BMUS=189	76,4706	94,3	95,6	70,25	0,8052	349,06
2. Coeficientes <i>wavelet</i> (Db9)	Celdas 20x20 y Entrenamiento largo BMUS=245	63,0588	65,29	62,53	67,94	0,6659	428,71
3. Coeficientes <i>wavelet</i> (Sym9)	<i>Default</i> BMUS=158	63,4118	62,92	57,11	72,6	0,6747	277,06
4. Energías de las sub-bandas <i>wavelet</i> (Db9)	<i>Default</i> BMUS=1	91,4118	94,24	94,58	89,31	0,9171	70
5. Energías de las sub-bandas <i>wavelet</i> (Sym9)	<i>Default</i> BMUS=1	91,0588	94,19	94,58	88,39	0,912	69,18
6. Componentes Principales de las Energías de las sub-bandas <i>wavelet</i> (sym9)	Celdas 20x20 y Entrenamiento largo BMUS=1	91,5294	94,31	94,52	90,71	0,9248	83,29
7. Componentes Principales de los Coeficientes <i>wavelet</i> (sym9)	Entrenamiento largo BMUS=188	58,8235	64,15	67,92	57,54	0,6066	183,06
8. Híbrido	Celdas 20x20 y Entrenamiento largo BMUS=1	95,1765	96,17	96,22	95,04	0,9560	79,41

Tabla 7. Resultados SVM multiclase con el espacio temporal

<i>Parámetros\clases</i>	N-V	N-A	N-S	V-A	V-S	A-S
1. Precisión del Test	94,58	94,58	96,46	99,06	98,58	99,76
2. Predicción Positiva	92	91,63	95,39	98,6	97,25	99,53
3. Especificidad	91,51	91,04	95,28	98,58	97,17	99,53
4. Sensibilidad	97,64	98,11	97,64	99,53	100	100
5. <i>F1 score</i>	94,74	94,76	96,5	99,06	98,6	99,76
6. Parámetros SVM	C=0,3 y $\sigma=0,3$	C=10 y $\sigma=1$	C=0,3 y $\sigma=0,3$	C=0,1 y $\sigma=0,3$	C=10 y $\sigma=0,7$	C=0,1 y $\sigma=0,3$
7. Tiempo de ejecución (ms)	0,29	0,13	0,83	0,39	0,29	0,51

Tabla 8. Resultados KNN multiclase con el espacio temporal

Parámetros KNN	K=3 y Función <i>Cityblock</i>					
Precisión del Test	93,0425					
Tiempo de ejecución (ms)	10,70					
<i>Parámetros\clases</i>	N-V	N-A	N-S	V-A	V-S	A-S
1. Predicción Positiva	93,52	92,92	97,1	99,53	97,57	100
2. Especificidad	92,63	91,67	96,7	99,51	97,58	100
3. Sensibilidad	98,06	99,53	94,81	99,53	100	100
4. <i>F1 score</i>	95,73	96,11	95,94	99,53	98,77	100

Tabla 9. Resultados *Kmeans* multiclase con el espacio híbrido

Función	<i>Cityblock</i>					
Precisión del Test	51,42					
Tiempo de ejecución (μs)	26.35					
<i>Parámetros\clases</i>	N-V	N-A	N-S	V-A	V-S	A-S
1. Predicción Positiva	97,55	55,56	82,35	74,47	78,65	83,33
2. Especificidad	97,16	70,98	90,13	86,89	89,33	83,33
3. Sensibilidad	94,08	36,08	47,62	94,59	85,37	56,91
4. <i>F1 score</i>	95,78	43,75	60,34	83,33	81,87	67,63

Tabla 10. Resultados SOM multiclase con el espacio de las energías de las sub-bandas *wavelet* Sym9

Parámetros SOM	<i>Default</i> y BMUS=236					
Precisión del Test	43,2786					
Tiempo de ejecución (μs)	184,47					
<i>Parámetros</i> \ <i>clases</i>	N-V	N-A	N-S	V-A	V-S	A-S
Predicción Positiva	78,77	70,99	43,37	93	45	65,85
Especificidad	73,61	73,61	77,94	95,27	81,03	86,92
Sensibilidad	81,98	56,71	19,71	73,23	37,5	47,37
<i>F1 score</i>	80,34	63,05	27,84	81,94	40,91	55,1

Figura 18. Separación de latidos normales (marcas rojas) y PVC (marcas verdes) en el espacio híbrido con algoritmo *Kmeans*

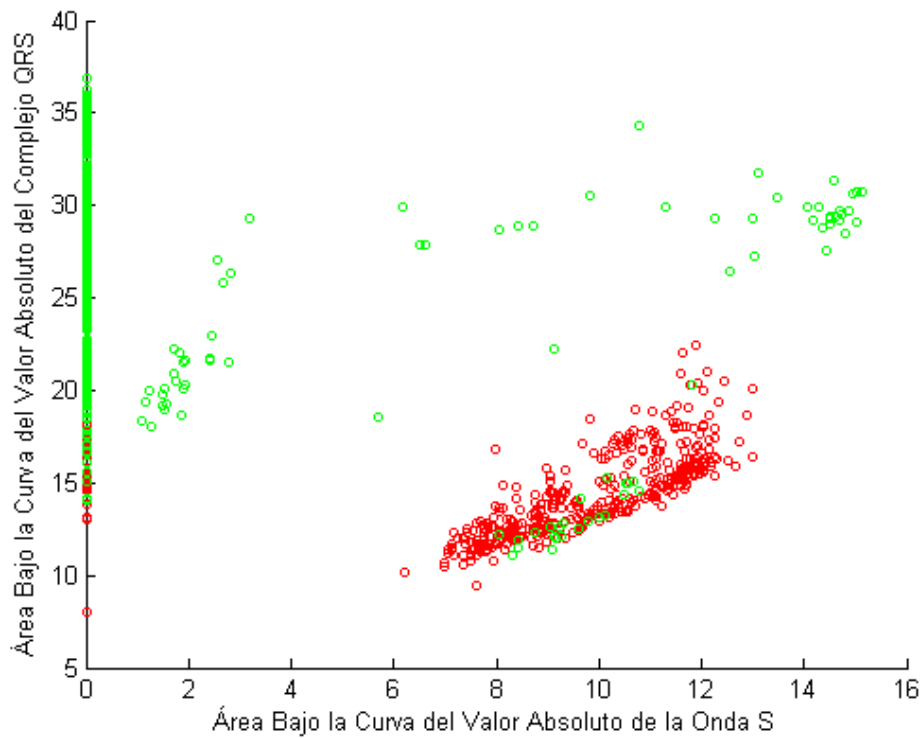


Figura 19. Separación de latidos normales (marcas rojas), PVC (marcas amarillas), APB (marcas verdes) y SVPB (marcas azules) en el espacio híbrido con algoritmo *Kmeans*

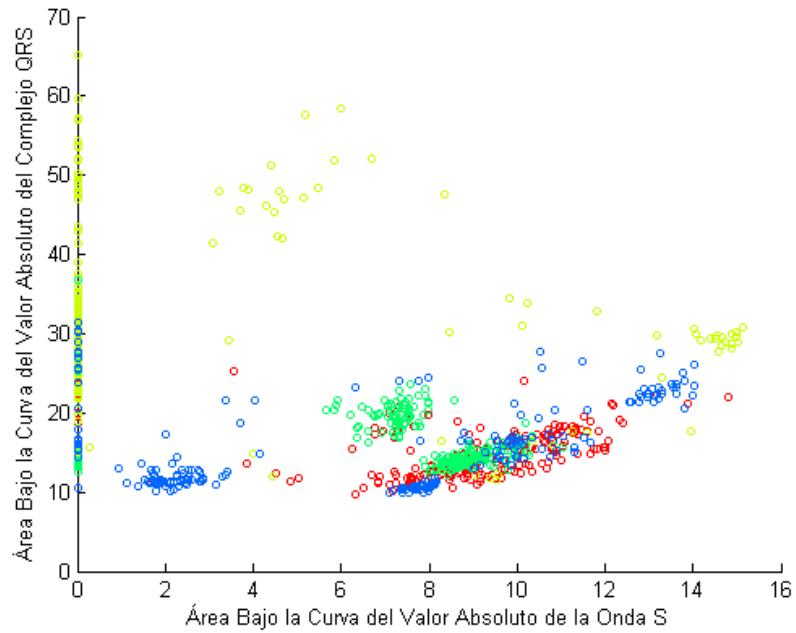


Figura 20. Separación de latidos normales (celdas cían) y PVC (celdas verdes) del espacio híbrido en el mapa creado por SOM

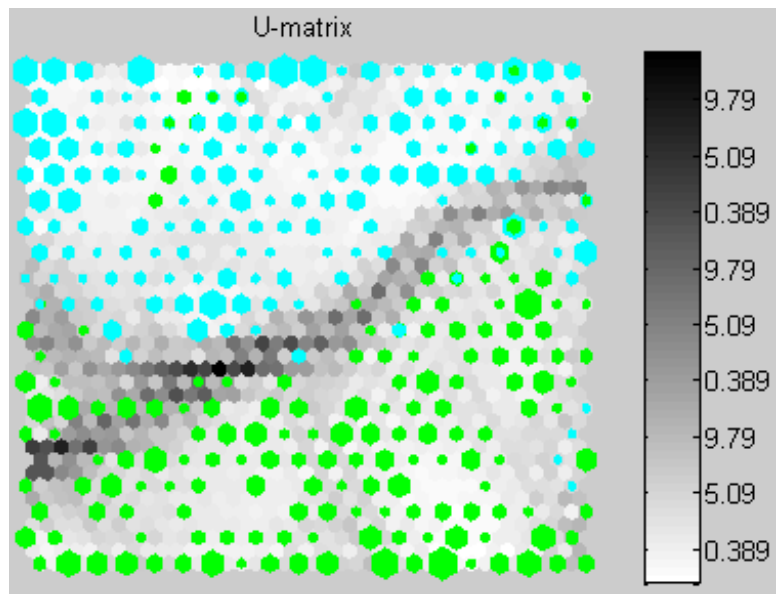
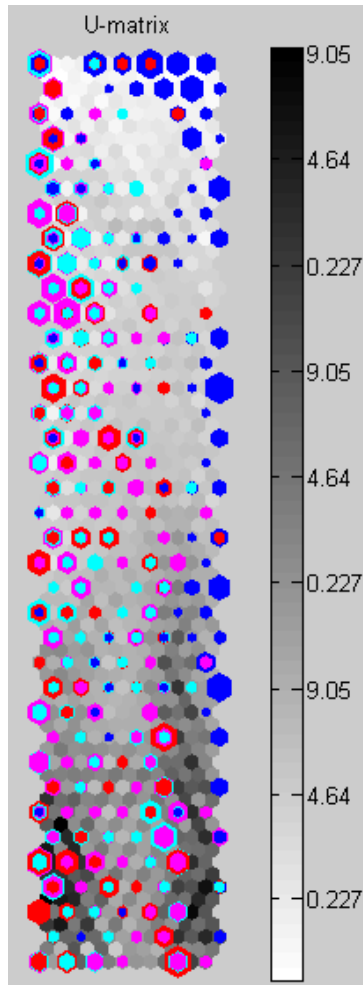


Figura 21. Separación de latidos normales (celdas rojas), PVC (celdas verdes), APB (marcas azules) y SVPB (marcas violetas) del espacio de las sub-bandas de las energías *wavelet* (Sym9) en el mapa creado por SOM



7. ANÁLISIS

En esta sección se realiza el análisis correspondiente a los resultados obtenidos en la sección 6.

7.1. MÉTODOS SUPERVISADOS

En los resultados del algoritmo SVM, se observa en la Tabla 3, que entre más características tenga el espacio mayor es el desempeño, ya que tiene más información para discernir entre los latidos aunque su costo computacional aumenta debido a que es mayor el número de cálculos que tiene que realizar. De los dos métodos basados en *wavelet* implementados, la mayor sensibilidad se obtuvo con Db9, aunque Sym9 presenta un menor costo computacional. El espacio temporal obtuvo la mejor precisión con 98,94% y la mejor especificidad con 99,29%, con un tiempo de ejecución de 0,78 ms por latido. El espacio de las componentes principales de las sub-bandas *wavelet* Sym9 tiene el menor costo computacional con un tiempo de ejecución de 0,26 ms por latido, y su *F1 score* es de 0,9574.

En la Tabla 7, se muestra que el espacio temporal obtuvo los mejores resultados para SVM multiclase. Gracias a que este espacio tiene un gran número de características y a la provisión de etiquetas, el algoritmo discierne bien entre las diferentes clases. La sensibilidad es el parámetro con mayor porcentaje, donde se observa que el algoritmo discierne completamente los latidos SVPB de los latidos APB y PVC.

De la Tabla 4 se observa que para KNN, el espacio temporal obtuvo los mejores resultados con una precisión del 98,58%, una especificidad del 98,12% y una sensibilidad del 99,06%, pero tiene el costo computacional más alto (14,55 ms por latido) ya que requiere más tiempo para realizar todos los cálculos. Para los métodos basados en *wavelet*, Db9

obtiene mejores resultados, aunque Sym9 presenta menor costo computacional y mayor sensibilidad. El espacio híbrido tiene el menor costo computacional, con un tiempo de ejecución de 0,903 ms por latido y un *F1 score* de 0,9668.

Para el KNN multiclase, el espacio temporal también presentó el mejor desempeño. Los resultados se muestran en la Tabla 8. Se observa que los parámetros presentan mayores porcentajes en las clasificaciones que comparan dos tipos de afecciones. Su costo computacional es menor que el obtenido en la clasificación binaria, debido a que al haber más centroides, el cálculo de las distancias converge en menor tiempo.

De los 2 métodos supervisados desarrollados, el algoritmo de predicción de SVM tiene un menor costo computacional que KNN en los 8 espacios de características evaluados; esto se debe a que el algoritmo KNN, para calcular las distancias más cercanas, requiere de muchas iteraciones, mientras que la predicción de SVM solo necesita calcular el producto punto entre el nuevo vector parámetro y el vector de pesos ya entrenado. SVM obtuvo la mejor especificidad (99,29%), sin embargo KNN obtuvo la mejor sensibilidad (99,059%).

7.2. MÉTODOS NO SUPERVISADOS

Al ser un método no supervisado, los resultados disminuyen su rendimiento, como se observa en la Tabla 5, ya que no tienen la información de las etiquetas. Para el algoritmo *Kmeans*, el espacio híbrido obtuvo el mayor desempeño con un *F1 score* de 0,9507 y un costo computacional de 18,71 μ s; esto ocurre debido a que este espacio contiene las características que le permiten al algoritmo discernir de manera eficiente entre los latidos normales y ventriculares sin la ayuda de las etiquetas; situación que no ocurre con los demás espacios, ya que a pesar de tener muchas más características, éstas pueden ser redundantes para el algoritmo, influyendo negativamente en los resultados.

A partir de los resultados consignados en la Tabla 5 se realiza un análisis cuantitativo del algoritmo, pero en las aplicaciones de este tipo de métodos se busca es una visualización gráfica que permita realizar un análisis cualitativo de la clasificación del algoritmo. De la Figura 18 se observa que la agrupación muestra unos resultados acordes a los resultados cuantitativos; el porcentaje de error de la clasificación se aprecia en las marcas verdes

ubicadas en la zona roja y viceversa.

Para *Kmeans* multiclase, el rendimiento disminuye mucho más, incluso el mejor resultado, consignado en la Tabla 9, se logró con el espacio híbrido con una precisión apenas del 51,42%, donde solo obtienen resultados por encima del 90% entre latidos normales y latidos PVC. Su tiempo de ejecución es mayor debido a que la agrupación se realiza para 4 clases. Incluso en la Figura 19 se observa que los grupos están distribuidos de forma desigual en el espacio, i.e., no se disciernen fronteras o grupos conformados entre las diferentes clases.

Los resultados del algoritmo SOM se consignan en la Tabla 6. El espacio híbrido obtuvo el mejor desempeño, con una especificidad del 96,22%, una sensibilidad del 95,04% y un tiempo de ejecución de 79,41 μ s. El espacio de las energías de las sub-bandas *wavelet* Db9 tiene el menor costo computacional con un tiempo de ejecución de 70 μ s; de los resultados obtenidos se puede deducir que entre más características tenga el espacio, menor es el desempeño; esto ocurre debido a que en el algoritmo, cada característica aporta una distribución de pesos en el mapa, dando indicio de la celda en la que debe competir el latido, por ende si se tienen demasiadas características, cada latido tiene más celdas en las que compite y por lo tanto el vector más cercano (BMU) no consigue el mejor resultado; mientras que si se tiene un menor número de características, la variabilidad de los pesos en el mapa disminuye y es mayor el desempeño.

De la Figura 20, se observa que los resultados gráficos son acordes a los resultados analíticos; el porcentaje de error de la clasificación se aprecia en las celdas verdes ubicadas en la zona cian y viceversa.

Para SOM multiclase, el espacio con las energías de las sub-bandas *wavelet* Sym9 obtuvo el mayor desempeño entre los espacios evaluados, con apenas 43,27% en la precisión. Los resultados de este espacio de características se consignaron en la Tabla 10. De la Figura 21 no se observa una formación de grupos discernible.

De los 2 métodos no supervisados, el algoritmo de predicción de *Kmeans* tiene el costo computacional más bajo (18,71 μ s por latido) y la mejor especificidad (97,41%), mientras que SOM tiene la mejor sensibilidad (95,04%) y *F1 score* (0,956). SOM obtuvo resultados con más del 90% de precisión en 4 espacios de características diferentes, a diferencia de *Kmeans*. Ésto indica que el algoritmo SOM es más efectivo en encontrar

patrones en un espacio de características, lo que hace que el algoritmo sea más confiable.

Ambos algoritmos presentaron un desempeño por debajo del 52% en la etapa de multi-clasificación, indicando que se requiere una mejor caracterización de la señal ECG con varios tipos de afecciones.

8. ALGORITMO PARA LA DETECCIÓN DE LATIDOS PVC EN C++

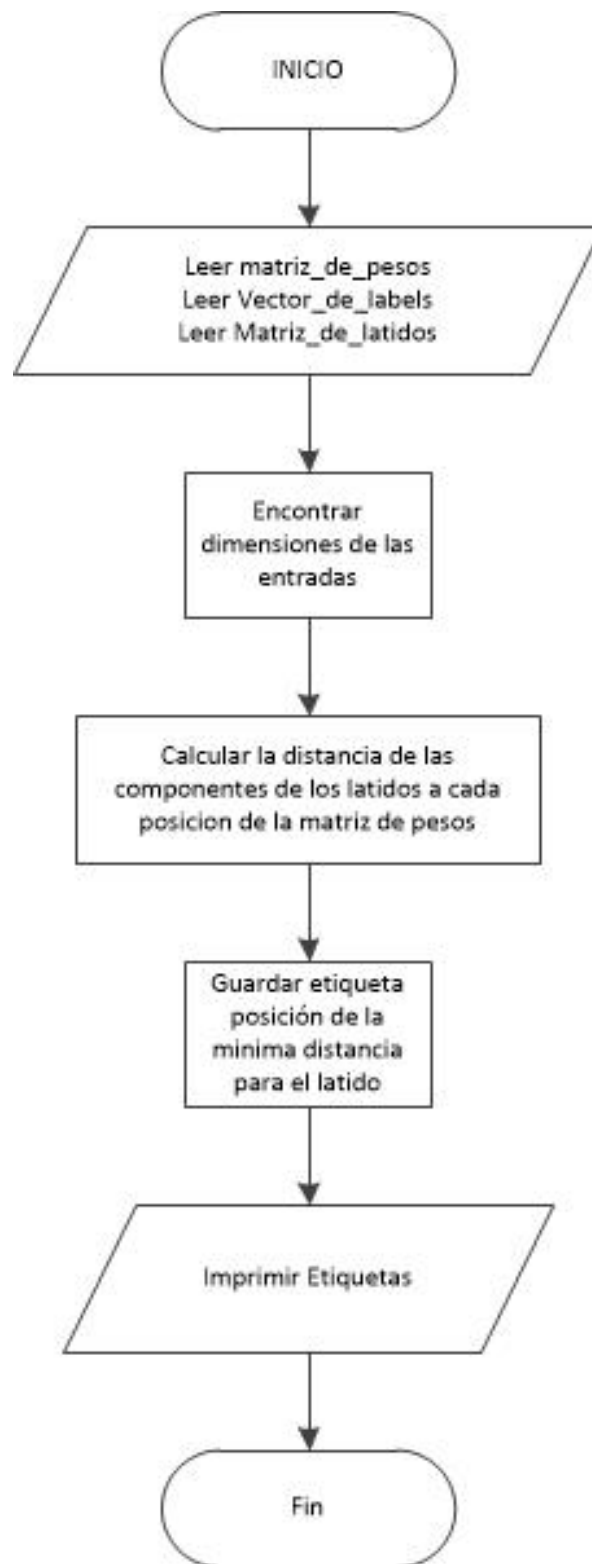
Ante la necesidad creciente de obtener nuevas herramientas que faciliten el monitoreo y la detección de arritmias en pacientes, se desarrolló un algoritmo de clasificación de PVC basado en SOM empleando el lenguaje de programación C++, el cual permite la integración con sistemas embebidos de alto nivel como microcontroladores y FPGAs de manera directa, lo que facilita su implementación a futuro; para el desarrollo del algoritmo se empleó la plataforma dev C++ de la empresa *bloodshed*. La ventaja de implementar éste algoritmo es que posee una precisión de aproximadamente 95% y un costo computacional reducido, permitiendo una clasificación más rápida y efectiva. Para garantizar ésta precisión, las matrices de entrada deben pertenecer al espacio híbrido; aunque otra ventaja de SOM es que también se podrían realizar pruebas ingresando espacios de características como las energías de las sub-bandas *wavelet*, las cuales arrojan resultados alrededor del 91% de precisión.

El algoritmo recibe como datos de entrada:

- El mapa entrenado previamente desde MATLAB con un conjunto representativo de latidos y/o bases de datos de arritmias. Dicha matriz debe ser ubicada en la misma carpeta del algoritmo con el nombre
- El vector de etiquetas de los nodos obtenidos a partir del entrenamiento del mapa auto-organizado en MATLAB, lo que permitirá calcular cada uno de los parámetros de desempeño del algoritmo en caso de ser necesario; dicho vector debe ser guardado en la misma ubicación del algoritmo con el nombre de "*Labels.txt*".
- Los latidos que se desean clasificar por el algoritmo, los cuales deben contener el mismo número de características (columnas) que los latidos empleados para entrenar el mapa y deben ser guardados en la misma carpeta del algoritmo con el nombre "*latidos.txt*"

En la Figura 22 se ilustra el diagrama de flujo simplificado del algoritmo de predicción de SOM desarrollado en C++.

Figura 22. Diagrama de flujo del código en C++ del algoritmo de predicción SOM



CONCLUSIONES

Se implementaron 4 algoritmos para la detección de arritmias cardíacas basados en métodos de aprendizaje supervisado y no supervisado. Se comparan los métodos supervisados y no supervisados de forma separada, ya que no es relevante comparar dos métodos que se usan para aplicaciones distintas. Cada método tiene sus ventajas y desventajas, e.g. los métodos supervisados suelen tener mucho sobreajuste, i.e. suelen dar buenos resultados para un grupo determinado de pacientes, pero para otros, estos resultados disminuyen. Por lo regular, tienen fronteras de decisión más complejas, permitiendo una separación de clases más efectiva; mientras que los métodos no supervisados se adaptan más fácil a cualquier tipo de paciente, aunque sus fronteras de decisión suelen ser más simples.

Para la detección de PVC, de los métodos supervisados, SVM obtuvo el mejor desempeño con una especificidad del 99,29% y sensibilidad del 98,59% y el costo computacional más bajo, con un tiempo de ejecución de 0,78 ms por latido; esto se debe a que el uso de *kernels* le permite a SVM tener fronteras de decisión más complejas, mientras que las distancias usadas en KNN generan fronteras de decisión circulares, las cuales limitan la clasificación. El costo computacional de SVM es menor gracias a que su etapa de predicción solo necesita realizar el producto punto entre el vector de pesos entrenados y el vector parámetro f , mientras que KNN necesita calcular la distancia más cercana para cada instancia, lo cual requiere de muchas iteraciones. En pocas palabras, SVM es un método más complejo y más efectivo.

Una posible limitación de los resultados de este algoritmo, es que solo se consideran latidos normales y latidos PVC de una sola base de datos, por lo que estos resultados podrían estar mostrando un sobreajuste. A este problema se recomienda incluir otras bases de datos y otras clases de arritmias.

De los métodos no supervisados, SOM presenta el mejor resultado en cuanto a su desempeño, con *F1 score* de 0,956 y un tiempo de ejecución de 79,41 μ s, aunque el

algoritmo *Kmeans* presenta un costo computacional más bajo (18,71 μs); sin embargo ésta diferencia de tiempos no es relevante para un sistema embebido, el cual gracias a su nivel de procesamiento, permite realizar la clasificación del latido en tiempo real, ya que la duración de la predicción es mucho menor a la duración de cada latido (750 ms o menos si el ritmo cardíaco es muy rápido); además *Kmeans* requiere una etapa de caracterización mucho más riguroso. Por otra parte SOM es un algoritmo más complejo, ya que usa el concepto de redes neuronales para generar competencia entre las neuronas, usando funciones *kernel* para la activación de las neuronas vecinas; mientras que *Kmeans* genera grupos con fronteras de separación más simples, motivo que limita el desempeño del algoritmo.

Debido a la necesidad de desarrollar dispositivos electrónicos portátiles que registren constantemente la señal cardíaca en pacientes con enfermedades cardiovasculares severas y le permitan al médico realizar un diagnóstico de forma más rápida y sencilla, se implementó el algoritmo SOM en código C++ como base de apoyo para una futura implementación en sistemas embebidos, siendo éste un lenguaje común en muchas clases de microcontroladores. Éste código le permite al usuario, en la interfaz del DevC++, variar el número de latidos a clasificar, obteniendo el resultado en tiempo real gracias a su costo computacional reducido y parámetros de medición eficientes. Sólo se tiene en cuenta la predicción del algoritmo, ya que esta etapa es la que se implementa en este tipo aplicaciones.

En la etapa de multi-clasificación se incluyeron otras bases de datos y otras clases de arritmias para la reducción del sobreajuste en los métodos supervisados. En consecuencia se obtuvo menor desempeño, principalmente en los métodos no supervisados, motivo por el cuál se propone en trabajos futuros, el estudio de otros métodos y técnicas para la etapa de pre-procesamiento de diferentes bases de datos.

Los métodos no supervisados se caracterizan por identificar los patrones de cada clase, y en base a esto realizar la separación de clases. Por lo tanto la eficiencia de este tipo de métodos radica en las características del espacio de entrada. Por esta razón y por los resultados obtenidos en este trabajo de grado, se propone el estudio de nuevos métodos para la extracción de características de la señal ECG; se recomiendan pocas características para un desempeño más eficiente y una reducción del costo computacional en los algoritmos.

También se propone el estudio de algoritmos que combinen diferentes métodos no supervisados que permitan mejorar el desempeño de la clasificación; también se puede pensar en la implementación de algoritmos que combinen diferentes métodos supervisados, aunque se debe tener en cuenta que el costo computacional aumenta.

BIBLIOGRAFÍA

- [1] N. Shawe-Taylor, J. and Cristianini, “Further Reading: Chapter 1,” 2007. [Online]. Available: http://www.support-vector.net/chapter_1.html
- [2] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Massachusetts Institute of Technology, 2010.
- [3] L. Sormo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Academic Press, 2005.
- [4] D. Powers, “EVALUATION : FROM PRECISION , RECALL AND F-MEASURE TO ROC , INFORMEDNESS , MARKEDNESS & CORRELATION,” vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: http://www.bioinfo.in/uploadfiles/13031311552_1_1_JMLT.pdf
- [5] A. Smola and S. Vishwanathan, “Introduction to machine learning,” 2008. [Online]. Available: <http://alex.smola.org/drafts/thebook.pdf>
- [6] Scholarpedia, “K-nearest neighbor,” 2009. [Online]. Available: http://www.scholarpedia.org/article/K-nearest_neighbor
- [7] —, “Kohonen network,” 2007. [Online]. Available: http://www.scholarpedia.org/article/Kohonen_network
- [8] I. Steinwart and A. Christmann, *Support Vector Machines*, 1st ed., 2008.
- [9] O. M. de la Salud, “Enfermedades cardiovasculares,” 2013. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs317/es/X>
- [10] W. Uribe, M. Duque, and E. Medina, *Electrocardiografía y arritmias*. P.L.A. Export Editores Ltda, 2005.

- [11] L. López, “MEDIANTE APRENDIZAJE DE MAQUINA CON INTERFAZ A USUARIO MODEL OF DYNAMIC CLASSIFICATION OF ARRHYTHMIAS CARDIAC BY MEANS OF LEARNING OF MACHINE WITH INTERFACE TO USER,” vol. 16, no. 2, pp. 86–95, 2006. [Online]. Available: http://www.umng.edu.co/documents/63968/74789/16-2_8.pdf
- [12] S. Karimifard, a. Ahmadian, M. Khoshnevisan, and M. S. Nambakhsh, “Morphological heart arrhythmia detection using Hermitian basis functions and kNN classifier.” *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 1, no. 4, pp. 1367–70, Jan. 2006. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/17946458>
- [13] H. Kinoshita, M. Yoshizawa, M. Inagaki, K. Uemura, M. Sugimachi, and K. Sunagawa, “Development of an algorithm for detection of fatal cardiac arrhythmia for implantable cardioverter-defibrillator using a self-organizing map,” pp. 5–7, 2006. [Online]. Available: http://www.nanobme.org/en/annals/articles07/2007_en_045-054.pdf
- [14] J. L. Rodríguez, D. Cuesta, G. Castellanos, and Valencia, “Dynamic Time Warping y k-means modificado,” pp. 1173–1177, 2007.
- [15] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, “Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal.” *Artificial intelligence in medicine*, vol. 44, no. 1, pp. 51–64, Sep. 2008. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/18585905>
- [16] S. Mehta and N. Lingayat, “SVM-based algorithm for recognition of QRS complexes in electrocardiogram,” *Irbm*, vol. 29, no. 5, pp. 310–317, Nov. 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1297956208000764>
- [17] S. S. Mehta and N. S. Lingayat, “Application of support vector machine for the detection of P- and T-waves in 12-lead electrocardiogram.” *Computer methods and programs in biomedicine*, vol. 93, no. 1, pp. 46–60, Jan. 2009. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/18835057>

- [18] C. Wen, T.-C. Lin, K.-C. Chang, and C.-H. Huang, "Classification of ECG complexes using self-organizing CMAC," *Measurement*, vol. 42, no. 3, pp. 399–407, Apr. 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0263224108001292>
- [19] M. Korürek and A. Nizam, "Clustering MIT-BIH arrhythmias with Ant Colony Optimization using time domain and PCA compressed wavelet coefficients," *Digital Signal Processing*, vol. 20, no. 4, pp. 1050–1060, Jul. 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1051200409001973>
- [20] M. L. Talbi and a. Charef, "PVC discrimination using the QRS power spectrum and self-organizing maps." *Computer methods and programs in biomedicine*, vol. 94, no. 3, pp. 223–31, Jun. 2009. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19215994>
- [21] R. Chitta and M. Narasimha Murty, "Two-level k-means clustering algorithm for kâ relationship establishment and linear-time classification," *Pattern Recognition*, vol. 43, no. 3, pp. 796–804, Mar. 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S003132030900363X>
- [22] S. Mehta, D. Shete, N. Lingayat, and V. Chouhan, "K-means algorithm for the detection and delineation of QRS-complexes in Electrocardiogram," *Irbm*, vol. 31, no. 1, pp. 48–54, Feb. 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1959031809001079>
- [23] M. Homaeinezhad, E. Tavakkoli, S. Atyabi, a. Ghaffari, and R. Ebrahimpour, "Synthesis of multiple-type classification algorithms for robust heart rhythm type recognition: Neuro-svm-pnn learning machine with virtual QRS image-based geometrical features," *Scientia Iranica*, vol. 18, no. 3, pp. 423–431, Jun. 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1026309811000988>
- [24] A. Gacek, "Preprocessing and analysis of ECG signals â A self-organizing maps approach," *Expert Systems with Applications*, vol. 38, no. 7, pp. 9008–9013, Jul. 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417411001394>

- [25] N. Nieto and M. Orozco, “SEÑALES SENOSOIDALES,” no. 38, pp. 381–386, 2008. [Online]. Available: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/3809/1975>
- [26] Mathworks, “Introduction to wavelet families,” 2013. [Online]. Available: <http://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html#f3-1008627>
- [27] O. Rioul and P. Duhamel, “Fast algorithms for discrete and continuous wavelet transforms,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 569–586, Mar. 1992. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=119724>
- [28] Mathworks, “Fast wavelet transform (fwt) algorithm,” 2013. [Online]. Available: <http://www.mathworks.com/help/wavelet/ug/fast-wavelet-transform-fwt-algorithm.html>
- [29] R. S. Gallego, *Introducción al Análisis de Datos Experimentales: Tratamiento de Datos en Bioensayos*. Universitat Jaume I, June 2002.
- [30] G. AL, A. LAN, G. L, H. JM, I. PCh, M. RG, M. JE, M. GB, P. C-K, and S. HE, “Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals,” 2013. [Online]. Available: <http://circ.ahajournals.org/content/101/23/e215.full>
- [31] C. Bustamante Arcila, S. Duque Vallejo, A. Orozco-Duque, and J. Bustamante Osorno, “Development of a segmentation algorithm for ecg signals, simultaneously applying continuous and discrete wavelet transform,” pp. 44,49, Sep. 2012. [Online]. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6340555&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6340555
- [32] N. J. Nilsson, “INTRODUCTION TO MACHINE LEARNING AN EARLY DRAFT OF A PROPOSED TEXTBOOK Department of Computer Science,” 2005.
- [33] D. D. Ingenier, “Estudio y aplicación de técnicas de aprendizaje automático orientadas al ámbito médico : estimación y explicación de predicciones

- individuales,” 2012. [Online]. Available: http://www.eps.uam.es/esp/alumnos/trabajos_fin_master/Di_Deco_Sampedro_Javier.pdf
- [34] J. Shawe-Taylor and N. Cristianini, “Further reading: Chapter 6,” 2007. [Online]. Available: http://www.support-vector.net/chapter_6.html
- [35] A. NG, “Online Course Machine Learning,” 2013. [Online]. Available: <https://www.coursera.org/course/ml>
- [36] J. C. Platt, “Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machines,” pp. 1–21, 1998. [Online]. Available: http://www.bradsblock.com/Sequential_Minimal_Optimization_A_Fast_Algorithm_for_Training_Support_Vector_Machine.pdf
- [37] E. Fix and J. Hodges, “Discriminatory analysis, nonparametric discrimination: Consistency properties. technical report 4, usaf school of aviation medicine, randolph field, texas,” 1951.
- [38] A. K. I. Saini, D. Singh, “Qrs detection using k-nearest neighbor algorithm (knn) and evaluation on standard ecg databases,” May 2012.
- [39] V. Laboratories, “Sample correlation and regression i,” 2013. [Online]. Available: <http://www.math.uah.edu/stat/sample/Covariance.html>
- [40] P. Dayan, “Unsupervised Learning,” pp. 1–7. [Online]. Available: <http://www.gatsby.ucl.ac.uk/~dayan/papers/dun99b.pdf>
- [41] A. Jain and R. Dubes, *Algorithms for Clustering Data*. New York: Prentice Hall, 1998.
- [42] J. Macqueen, “SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS,” vol. 233, no. 233, pp. 281–297.
- [43] H.-h. Bock and R. Aachen, “A history of the k-means algorithm 1 . Clustering with SSQ and the k-means algorithm,” pp. 161–172, 2007. [Online]. Available: <http://www.modulad.fr/dac/Slides/Bock/Ahistoryofthek-means.pdf>
- [44] J. Marin, “Kohonen (SOM),” pp. 1–13. [Online]. Available:

<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>

- [45] CIS, “Som toolbox 2.0 downloadables,” 2013. [Online]. Available: <http://www.cis.hut.fi/projects/somtoolbox/download/>