

Universitat de Valencia
Escuela Técnica Superior de Ingeniería
Programa Doctorado en Ingeniería Electronica



TESIS DOCTORAL

**Determinación mediante minería de datos del núcleo
subtalámico utilizando registros MER de cirugías
para la implantación de neuroestimuladores en
pacientes con Parkinson**

Autor: Luciano Schiaffino
Director: Dr. Alfredo Rosado Muñoz
Co-Director: Dr. Juan Guerrero Martínez

Julio, 2018

DECLARACIÓN DEL AUTOR

Este documento de Tesis ha sido presentado y aprobado como parte de los requisitos para obtener el grado académico de Doctor ante la Universidad de Valencia.

A MI HIJO LUCCA POR SER LA LUZ DE MI VIDA

AGRADECIMIENTOS

- A la Universidad Nacional de Entre Ríos, la Fundación Carolina y la Universidad de Valencia, por haber posibilitado la realización del Doctorado en Ingeniería Electrónica y el desarrollo de la presente tesis.

- A mi director, Dr. Alfredo Rosado Muñoz y co director Dr. Juan Guerrero Martínez, por sus enseñanzas, asesoramiento y constante disposición durante el desarrollo de todo mi Doctorado.

- A mis compañeros y docentes de la cátedra de Control de Procesos por la colaboración recibida que posibilitó mi labor.

- Al Dr. Med. Antonio J. Gutierrez Martín y demás médicos que integran la Unidad de Trastornos del Movimiento del Hospital Universitario y Politécnico La Fe de Valencia por su asesoramiento y aportes para el desarrollo experimental de la tesis.

- A los investigadores y docentes del ETSE por sus aportes a las investigaciones y publicaciones realizadas.

RESUMEN

La enfermedad de Parkinson (EP) es una degeneración del sistema nervioso central (SNC) caracterizada por un deterioro progresivo de funciones motoras como lentitud de movimientos, temblor, rigidez e inestabilidad postural.

Un estimulador cerebral profundo (DBS por sus siglas en inglés) es una tecnología novedosa para el tratamiento de los trastornos motores de la EP en los casos donde la medicación no es efectiva o cuando dichos fármacos generan efectos secundarios muy discapacitantes como las discenesias. Básicamente el DBS consiste en un neuroestimulador implantado o marcapasos del cerebro que mediante 2 microelectrodos estimula con pulsos de alta frecuencia al núcleo subtalámico (STN por sus siglas en inglés) de ambos hemisferios cerebrales para reducir la hiperactividad crónica de las neuronas involucradas.

Para la implantación del DBS se emplean procedimientos estereotáctico con el objetivo de realizar una cirugía lo menos invasiva posible para el paciente. Utilizando un instrumento fijo al cráneo del paciente, es posible localizar tridimensionalmente una estructura cerebral de referencia. Para localizar la zona STN y dependiendo de la tecnología disponible en cada institución se utilizan: imágenes de resonancia previas a la cirugía, atlas genéricos que en algunos casos se superponen a las imágenes obtenidas, imágenes de tomografía realizadas durante la cirugía, reconstrucción 3D de la fusión de imágenes de resonancia y tomografía (co-registro), en el análisis de ritmo beta del STN y en el análisis visual y acústico de las señales obtenidas mediante microelectrodos de registro (MER por sus siglas en inglés). Los MER tienen menos de 100 micrómetros de diámetro y mediante el sistema estereotáctico mecanizado ingresan al cerebro atravesando distintas estructuras funcionales del mismo como son el tálamo anterior (TH), la zona incerta (ZI), el núcleo subtalámico (STN) y la sustancia negra pars reticulata (SNr). Cada una de estas zonas presenta registros eléctricos específicos. El análisis de los registros MER para determinar si los electrodos están en la zona de implantación es uno de los procedimientos más utilizado en las instituciones de salud.

La detección del área objetivo (STN) es una tarea compleja y la exactitud con que se cumpla depende del éxito del tratamiento con un DBS. Recientemente se han publicado los primeros trabajos que utilizando la minería de datos mediante algoritmos individuales de clasificación supervisados determinan si los registros obtenidos mediante los MER corresponden al STN.

En la presente tesis se plantea como hipótesis que es posible obtener modelos de clasificación supervisados con un buen rendimiento, haciendo uso de la combinación de clasificadores y una adecuada selección de características, con el fin de que puedan ser utilizados como herramienta de soporte para detectar las señales MER correspondientes al STN durante una cirugía para implantar un DBS.

Se adquirieron señales MER de 22 pacientes con Parkinson a los cuales se les realizó una cirugía bilateral para la implantación de un dispositivo DBS en STN. Las cirugías fueron desarrolladas en su totalidad en el Hospital Universitario y Politécnico La Fe de Valencia, España. Dos especialistas experimentados analizaron durante la intervención quirúrgica y para cada nivel de profundidad, si las señales MER visualizadas en el monitor se correspondían al STN u a otra zona del cerebro. Con la idea de minimizar posibles errores humanos en el proceso de etiquetación de los registros se analizaron posteriormente a cada cirugía las imágenes de corregramas realizadas durante la cirugía (al inicio y al final de la misma) que permiten determinar mediante imágenes la profundidad a la cual se ingresó al STN.

Cada registro se dividió en ventanas de 1 s solapadas al 50% a las cuales se calcularon y estandarizaron *off line* 16 características temporales utilizando el software Matlab[®]. Las características utilizadas permiten describir las principales variables temporales de las señales obtenidas mediante los MER asociadas a la actividad de fondo sin considerar las espigas y otras que sólo caracterizan a las espigas. La eliminación automática de registros asociados a ruidos de movimiento, mecánicos o eléctricos fue el único pre procesamiento que se aplicó a los registros MER utilizados, asegurado que la base de datos represente situaciones semejantes de la vida real.

Con la base de datos obtenida se procedió a entrenar los clasificadores con los datos de 21 pacientes (datos de entrenamiento) dejando los datos del paciente excluido para validación (datos de validación). Este proceso se repitió 22 veces dejando un paciente distinto por vez en cada conjunto de entrenamiento-validación realizando una generación de conjuntos mediante validación cruzada “*leave one out*” por paciente.

Se configuraron y entrenaron algoritmos de clasificación individuales: K vecinos más cercanos (KNN) y árboles de decisión tipo CART y CHAID. También se trabajó con las siguientes estrategias combinadas: *bagging*, *boosting*, *random forest* y *stacking*. Para todos los

casos se seleccionaron los parámetros que permitieron configurar cada algoritmo para minimizar el error de clasificación mediante optimización Bayesiana.

Cuando se utilizaron las 16 características temporales para entrenamiento y validación, *stacking* obtuvo los mejores resultados utilizando 3 clasificadores diferentes como base (nivel 0): KNN, árboles de decisión tipo CART y *bagging* y como meta clasificador (nivel 1) a *random forest*. Este algoritmo combinado se denominó Stack2 y presentó una exactitud promedio porcentual (ACC) del 94,6%, una especificidad (ESP) del 95,6% y una sensibilidad (SEN) del 95,6%.

Se ensayaron algoritmos de selección de características tipo filtro, de envoltura e integrados. *Branch and bound* resultó el mejor para los datos de la tesis y permitió seleccionar 6 características de las 16 totales (4 vinculadas a la actividad de fondo y 2 a las espigas) que mejoraron el rendimiento de Stack2 en lo que se denominó Stack2_BBoud, pasando ACC al 95%, ESP al 96% y SEN al 94%. Este clasificador tuvo el mejor desempeño de todos los analizados en la presente tesis, indicando que dicho clasificador presenta mayor poder de discriminación de clases mediante las características vinculadas a la actividad de fondo.

Con el desarrollo de las investigaciones realizadas en la presente tesis doctoral se pudo obtener un modelo combinado de clasificación supervisada mediante la metodología *stacking*, cuyos indicadores de desempeño y tiempos de validación indican que puede ser utilizado, con excelente rendimiento, en un proceso de clasificación automática para detectar señales del STN a partir de procesar señales eléctricas cerebrales provenientes de micro electrodos de registros durante cirugías para implantar un estimulador cerebral profundo en paciente con Parkinson. Los indicadores de desempeño obtenidos son superiores a los reportados por otros trabajos de investigación para este mismo tipo de aplicación.

Los resultados de la presente tesis contribuyen con un modelo novedoso (Stack2_BBoud) que constituye el primer paso para un sistema de clasificación automático que trabaje en el quirófano como herramienta de soporte a los neurofisiólogos y neurocirujanos al momento de definir la localización óptima del electrodo de estimulación de un sistema DBS en pacientes con Parkinson. Un sistema de éstas características permitirá reducir los tiempos de una cirugía de esta naturaleza además de brindar un resultado de clasificación objetivo.

ABREVIATURAS Y ACRÓNIMOS

ACC	Exactitud
A/D	analógico digital
AD	árbol de decisión
AdaBoost	<i>adaptive boosting</i>
ALIC	extremidad anterior de la cápsula interna
ANOVA	análisis de varianza
ATN	núcleo anterior del tálamo
AUC	área bajo la curva ROC
bagging	<i>bootstrap aggregating</i>
Basal	valor de la amplitud basal
BUA	<i>background unit activity</i> o actividad de fondo
CART	<i>classification and regression tree</i> o árbol de clasificación y regresión
CCA	análisis de correlación canónica
Cg25	área subcallosa del cíngulo
CHAID	<i>chi-square automatic interaction and detection</i> o detección e interacción automática chi-cuadrado
CI	núcleo cerebral claustrum
CL	longitud de la curva
CMt	núcleo centro medial del tálamo
DBS	estimulador cerebral profundo o estimulación cerebral profunda
DDR3	<i>double data rate type three synchronous dynamic random-access memory</i>
DLOR	<i>dorsolateral oscillatory region</i> o región dorsolateral

DOR	capacidad de diagnóstico
ECoG	Electrocorticograma
EDT	distancia estimada a la zona de implantación del electrodo
EEG	electroencefalograma
EM	algoritmo <i>estimated means</i>
EP	enfermedad de Parkinson
EPA	enfermedad de Parkinson avanzada
ESP	Especificidad
FDA	Administración de Alimentos y Medicamentos de Estados Unidos
GB	ganglios basales
GentleBoost	<i>gentle adaptive boosting</i>
GI	ganancia de información para árboles de decisión
GPe	globo pálido externo
GPi	globo pálido interno
H2	campos de Forel
HFD-P	descargas neuronales de alta frecuencia con pausas
HIFU	ultrasonido focalizado de alta intensidad
IG	algoritmo de ganancia de información para selección de características
IPG	generador de pulsos implantable
ITP	pedúnculo inferior talámico
K	Curtosis
KNN	clasificador K-vecinos más cercanos

LASSO	<i>least absolute shrinkage and selection operator</i>
LDA	análisis discriminante lineal
LFD-B	descargas neuronales de baja frecuencia con ráfagas
LFP	<i>local field potential</i> o potenciales de campo local
LH	hipotálamo lateral
LogitBoost	<i>adaptive logistic regression</i> o regresión logística adaptativa
LR	tasa de pérdida
MD	minería de datos
MER	microelectrodos de registro
MPTP	1-metil-4-fenil-1,2,3,6-tetrahidropiridina
MUA	<i>multi unit activity</i> o actividad multi neuronal
MUST	<i>multi unit spike train</i> o tren de espigas de varias neuronas
NAc	núcleo accumbens
NBM	núcleo basal de Meynert
NE	energía no lineal promedio
NRMS	media cuadrática normalizada o valor cuadrático medio normalizado
ODA	análisis discriminante cuadrático
PAG	sustancia gris periacueductal
PCA	análisis de componentes principales
PK	cantidad de picos
QUEST	<i>quick, unbiased, efficient, statistical tree</i>
RA	media cuadrática normalizada o valor cuadrático medio normalizado

RAM	memoria de acceso aleatorio
RForest	<i>random forest</i> o árboles aleatorios
RG	ratio de ganancia
RMI	Imágenes de resonancia magnética
RMS	valor cuadrático medio
ROC	<i>receiver operating characteristic</i> o característica operativa del receptor
RPM	revoluciones por minuto
Rt	núcleo reticular
RUS	muestreo aleatorio
RUSBoost	<i>random undersampling boosting</i>
SATA	<i>serial advanced technology attachment</i>
SBI	índice de ráfaga de espigas
SC	frecuencia de disparos
SCR	ratio del conteo de espigas
SEN	Sensibilidad
SF	mediana de la frecuencia de disparo de las espigas en Hz
Sign	función signo
SMAD	amplitud diferencial media de espigas
SNC	sistema nervioso central
SNr	sustancia nigra pars reticulata
SNr	sustancia nigra pars compacta
SPI	índice de pausa de espigas

SPR	ratio de pausa de espigas
SSD	desviación estándar de los intervalos entre espigas
SSF	mediana de la frecuencia de disparo de las espigas en Hz
STN	núcleo subtalámico
SUA	<i>single unit activity</i> o actividad de una neurona
SUST	<i>single unit spike train</i> o tren de espigas de una neurona
SVM	máquinas de soporte vectorial
TC	tomografía computada
TH	tálamo anterior
TH	Umbral
tiempo Entre.	tiempo de entrenamiento
tiempo Val.	tiempo de validación
UPDRS	<i>unified Parkinson's disease rating scale</i> o escala unificada de la enfermedad del Parkinson
VC/VS	cápsula interna/estriado ventral
Vim	núcleo ventral intermedio del tálamo
VMH	hipotálamo ventromedial
VPM/VPL	tálamo ventral posteromedial/posterolateral
VTA	área tegmental ventral
ZC	cruces por cero
ZI	zona incerta

TABLA DE CONTENIDOS

<i>Capítulo 1 Introducción</i>	1
1.1 Descripción del problema a resolver	1
1.2 Objetivos de la tesis	4
1.3 Organización del documento de tesis	5
1.4 Publicaciones y actividades vinculadas a la investigación de la tesis	5
<i>Capítulo 2 Enfermedad de Parkinson</i>	7
2.1 Introducción	7
2.2 Ganglios basales del cerebro involucrados en el Parkinson	8
2.3 Etiología de la Enfermedad de Parkinson	9
2.4 Fisiopatología de la Enfermedad de Parkinson	11
2.5 Síntomas de la Enfermedad de Parkinson	13
2.5.1 Síntomas motores	13
2.5.2 Síntomas No motores	15
2.6 Evolución y pronosis	15
2.7 Tratamientos	17
2.7.1 Tratamientos farmacológicos	18
2.7.2 Tratamientos quirúrgicos	19
<i>Capítulo 3 Estimulación cerebral profunda</i>	21
3.1 Definición y características generales	21
3.1.1 Detalles técnicos de la DBS	22
3.1.2 Pacientes con EP candidatos para la DBS.....	24
3.2 Acción fisiológica de la DBS	25
3.2.1 La DBS inhibe elementos neuronales locales	25
3.2.2 La DBS excita elementos neuronales locales.....	26
3.2.3 La DBS interrumpe o modifica el flujo patológico de información	26
3.3 Aspectos quirúrgicos	27
3.3.1 Marco estereotáctico	28
3.3.2 Neuroimágenes para la identificación del target	28
3.3.3 Registros de microelectrodos (MER) en la identificación del target	29

Capítulo 4 Registro de las señales eléctricas neuronales	31
4.1 Técnicas de registro	31
4.2 Factores que afectan al registro de actividad eléctrica neuronal	33
4.2.1 Sincronización temporal.....	33
4.2.2 Factores geométricos.....	34
4.2.3 Filtrado y atenuación de las señales	35
4.3 registros de microEEG	35
4.4 Señales de microEEG en pacientes con enfermedad de Parkinson.....	37
4.4.1 Características temporales de los registros MER del STN.....	39
4.4.2 Características frecuenciales de los registros MER del STN	43
Capítulo 5 Reconocimiento de patrones y minería de datos.....	47
5.1 Reconocimiento de Patrones	47
5.2 Fase de Minería de datos.....	50
5.2.1 Taxonomía de la Minería de datos	50
5.2.2 Técnicas de aprendizaje en la minería de datos.....	51
5.2.3 Algoritmos para aprendizaje supervisado y no supervisado.....	52
5.3 Estado del arte de clasificadores en detección de STN en base a registros MER	53
Capítulo 6 Materiales y metodología general	57
6.1 Materiales	57
6.1.1 Registro de Datos	57
6.1.2 Software	59
6.1.3 Hardware.....	60
6.2 Metodología general	60
6.2.1 Índices de desempeño de los clasificadores	61
6.2.2 Tiempos de entrenamiento y validación.....	63
6.2.3 Análisis Estadístico	64
Capítulo 7 Experiencia 1: Clasificación de registros MER mediante algoritmos individuales para la detección del STN.....	67
7.1 Algoritmo de K Vecinos más cercanos (KNN)	67
7.2 Algoritmo de Árboles de Decisión	71
7.2.1 Algoritmos para la Construcción de Árboles de Decisión	73

7.2.2 Criterios de división univariante	75
7.2.3 Criterios de división multivariante.....	78
7.2.4 Criterios de detención o stop.....	78
7.2.5 Métodos de poda.....	79
7.2.6 Algoritmos que implementan árboles de decisión (inductores)	80
7.2.7 Ventajas y desventajas de los árboles de decisión	82
7.3 Resultados con KNN	84
7.4 Resultados con Árboles de Decisión.....	86
7.5 Análisis estadístico y de resultados de los clasificadores del presente capítulo	95
<i>Capítulo 8 Experiencia 2: Clasificación de registros MER mediante combinación de clasificadores para la detección del STN.....</i>	<i>103</i>
8.1 Metodología secuencial o serie	105
8.1.1 Modelos guiados por la Selección de instancias	105
8.2 Métodos concurrentes o paralelos.....	111
8.2.1 Bagging.....	111
8.2.2 Árboles aleatorios (<i>random forest</i> por su denominación en inglés).....	115
8.3 Combinación de clasificadores de diferente tipo	116
8.3.1 Métodos de combinación no generadores	116
8.3.2 Métodos de Meta Combinación	119
8.4 Tamaño de clasificador combinado	120
8.5 Resultados obtenidos con <i>Boosting</i>	121
8.5.1 Resultados obtenidos con árboles de decisión podados	124
8.6 Resultados obtenidos con <i>bagging</i>.....	125
8.6.1 Resultados obtenidos con árboles de decisión podados	126
8.7 Resultados para Árboles Aleatorios.....	126
8.8 Resultado de la combinación de clasificadores diferentes mediante métodos no generadores.....	130
8.9 Resultado de la combinación de clasificadores mediante <i>Stacking</i> como Meta Combinador.....	132
8.10 Análisis estadístico y de resultados de los clasificadores del presente capítulo	135
<i>Capítulo 9 Experiencia 3: Selección de características</i>	<i>141</i>

9.1 Reducción de la dimensión del espacio de las características	141
9.2 Selección supervisada de características.....	142
9.2.1 Modelos de filtro	143
9.2.2 Modelos de envoltura o envolventes	144
9.2.3 Modelos Integrados	146
9.3 Resultados obtenidos con modelos de filtro para la selección de características	148
9.4 Resultados obtenidos con modelos de envoltura para la selección de características	150
9.5 Resultados obtenidos con modelos integrados para la selección de características....	151
9.5.1 Auto selección de características de los clasificadores entrenados en los capítulos 7 y 8.....	152
9.5.2 Resultados de la selección de características con LASSO	153
9.5.3 Resultados de la selección de características con ELASTIC NET	155
9.6 Análisis estadístico y de resultados de la selección de características del presente capítulo	157
.....	
9.6.1 Estudio estadístico y análisis de resultados de los métodos de autoselección de características	158
9.6.2 Estudio estadístico y análisis de resultados de los métodos ReliefF, branch and bound, LASSO y Elastic Net	163
9.6.3 Comparación de los clasificadores con mejor rendimiento	171
Capítulo 10 Discusiones y conclusiones.....	177
10.1 Clasificadores individuales	177
10.2 Clasificadores combinados.....	180
10.3 Base de datos	182
10.4 Selección de características	183
10.5 Limitaciones y perspectivas futuras de la investigación.....	185
10.6 Conclusión general.....	186
Referencias	189
ANEXO 1: Informe de actividades y publicaciones vinculadas a la tesis Doctoral	195

Capítulo 1 Introducción

1.1 DESCRIPCIÓN DEL PROBLEMA A RESOLVER

La enfermedad de Parkinson (EP) es una degeneración del sistema nervioso central (SNC) caracterizada por un deterioro progresivo de funciones motoras como lentitud de movimientos, temblor, rigidez, inestabilidad postural y no motoras tales como trastornos del sueño, estreñimiento, problemas visuales, respiratorios, urinarios, cognitivos o problemas psicológicos [1], [2]. Constituye un problema importante de salud que afecta a nivel mundial estimativamente a 2 personas de cada 100 mayores de 64 años [3]. Las causas de esta patología aún no son conocidas completamente, pero se sabe que existe un deterioro significativo de las células del núcleo basal denominado sustancia negra pars reticulata, lo que conlleva a la disminución de los niveles de dopamina, produciendo un efecto en otras estructuras como el estriado, el núcleo subtalámico y el globo pálido; las cuales juegan un papel relevante en la inhibición y control de los movimientos del ser humano [1], [2].

Aunque el tratamiento inicial de la EP es farmacológico, en algunos casos no se consigue un control adecuado de los síntomas, a pesar de las diversas combinaciones de los medicamentos actualmente disponibles. En otros casos, la medicación provoca efectos secundarios discapacitantes, como la aparición de movimientos anormales (discinesias) o intolerancia. Asimismo se ha observado que luego 5 o 6 años de tratamiento farmacológico el mismo deja de tener efecto y re aparecen los síntomas iniciales de la EP [1], [2]. En las tres situaciones antes descritas la implantación de un estimulador cerebral profundo (DBS por sus siglas en inglés) mediante una cirugía es un tratamiento novedoso utilizado en la actualidad [4]. Dicho neuroestimulador, con pulsos de alta frecuencia, actúa como un marcapasos del cerebro y es particularmente exitoso en la EP. La zona de estimulación más habitual es el núcleo subtalámico (STN por sus siglas en inglés) de ambos hemisferios cerebrales para reducir la hiperactividad crónica de las neuronas involucradas [5], [6]. El STN es una estructura discoidea de aproximadamente 5.9 mm×3.7 mm×5 mm situada en la transición entre el diencefalo y el mesencefalo rodeada de estructuras funcionalmente denominadas sustancia negra, núcleo rojo y zona incierta [1].

Para la implantación del neuroestimulador se emplean procedimientos estereotácticos con el objetivo de realizar una cirugía lo menos invasiva posible para el paciente. Utilizando un

instrumento fijo al cráneo del paciente, es posible localizar tridimensionalmente una estructura cerebral de referencia [5], [7]. Para localizar la zona STN y dependiendo de la tecnología disponible en cada institución se utilizan: imágenes de resonancia previas a la cirugía, atlas genéricos que en algunos casos se superponen a las imágenes obtenidas, imágenes de tomografía realizadas durante la cirugía, reconstrucción 3D de la fusión de imágenes de resonancia y tomografía (coregistro), en el análisis de ritmo beta del STN y en el análisis visual y acústico de las señales obtenidas mediante microelectrodos de registro (MER por sus siglas en inglés) [5], [7]. Los MER tienen menos de 100 micrómetros de diámetro y mediante el sistema estereotáctico mecanizado van ingresando al cerebro atravesando distintas estructuras funcionales del mismo como son el tálamo anterior (TH), la zona incerta (ZI), el núcleo subtalámico (STN) y la sustancia negra pars reticulata (SNr) [4]. Cada una de estas zonas presenta registros eléctricos específicos [4], [5]. El análisis de los registros MER para determinar si los electrodos están en la zona de implantación es uno de los procedimientos más utilizado en las instituciones de salud [4].

Además de los registros MER las otras técnicas (imágenes de resonancia, coregistros, ritmo beta, etc.), cuando están disponibles en la institución, le ayudan al neurocirujano en la tarea crucial de localizar el área objetivo para ubicar los electrodos del neuroestimulador en el cerebro del paciente. Debido a las diferencias entre la localización del área objetivo planificada antes de la cirugía y la posición real del STN, el neurocirujano define la posición final de los electrodos del neuroestimulador principalmente en base a la forma de onda de las señales obtenidas mediante los MER [4], [5].

Los registros MER son señales no estacionarias, conformadas por secuencias de disparos provenientes de: i) la actividad neuronal denominadas spikes o espigas, ii) por una actividad neuronal de fondo y iii) por artefactos [8].

La detección del área objetivo es una tarea compleja y la exactitud con que se cumpla depende del éxito del tratamiento. Recientemente se han publicado los primeros trabajos [4], [9], [10] que utilizando el procesamiento de señales y la minería de datos para localizar regiones específicas del cerebro basados únicamente en los MER. Básicamente se intenta clasificar mediante distintos algoritmos si las señales obtenidas mediante los MER corresponden a TH, ZI, STN o la SNr.

En la minería de datos se han desarrollado un gran número de clasificadores con el objetivo de encontrar uno generalizable que pueda utilizarse con cualquier conjunto de datos. Cada método de clasificación presenta sus ventajas y desventajas y es importante tener en cuenta que ningún algoritmo siempre resulta mejor que otro para todos los procesos de clasificación [11]. En la búsqueda por mejorar el porcentaje de aciertos por parte de cada clasificador, el problema encontrado se centra en predecir a qué clase pertenecen las muestras evaluadas, así como en reducir el tiempo de ejecución ante cualquier proceso relacionado con la minería de datos [12].

Específicamente para determinar mediante clasificadores zonas cerebrales en base a los registros MER se han reportado las siguientes publicaciones científicas relevantes:

Art Chaovallitwongse y col. [4] trabajaron con 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) y señales MER provenientes de 17 pacientes. Reportaron resultados de clasificadores supervisados con algoritmos de K-vecinos más cercanos (KNN), árboles de decisión binarios y Naive Bayes. Los mejores resultados fueron obtenidos con los árboles de decisión que utilizaron como criterio de división de ganancia de información logrando una exactitud promedio del 89,61%. También utilizaron para entrenar KNN y árboles de decisión un indicador frecuencial-temporal, pero los resultados obtenidos no fueron alentadores (máxima exactitud promedio de 50,2%). Para estimar los indicadores de desempeño utilizaron validación cruzada dejando siempre un paciente fuera de los datos de entrenamiento y utilizando ese paciente para validación (denominada *leave one patient out*). En el trabajo no se realiza un estudio estadístico comparativo de los desempeños de los clasificadores.

En Cagnan y col. [9] proponen un método de clasificación propio con una estructura similar a un árbol de decisión binario, trabajan con 2 características temporales y 2 frecuenciales vinculadas a la potencia en el rango de frecuencias beta y gamma, con el objetivo de detectar la entrada y salida del electrodo en STN. Logran un porcentaje de acuerdo promedio con las clasificaciones que realizan los neurofisiólogos del 87-88. Realizaron un estudio estadístico comparativo.

Rajpurohit y col. [10] reportaron resultados de clasificadores supervisados con algoritmos regresión logística, KNN, Naive Bayes y máquinas de soporte vectorial (SVM). Utilizaron 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) con datos provenientes de 26 pacientes. Los mejores resultados

se obtuvieron con KNN. Cuando entrenaron al clasificador con las 13 características estandarizadas con los valores de todos los pacientes KNN presentó un error promedio de 0,2. El trabajo también reportó la selección de características por 2 métodos de envoltura: *backward* seleccionando 7 características y *forward* seleccionando 6 características en ambos casos 5 de ellas vinculadas a la actividad de fondo. El error promedio de clasificación en ambos casos disminuye a 0,1. El trabajo realiza un estudio estadístico comparativo y utiliza la validación de los clasificadores tipo *leave one patient out*.

Los trabajos científicos analizados al presente, vinculados a la clasificación de señales MER para detección de STN, reportan algoritmos supervisados que realizan el proceso de clasificación en forma individual (KNN, árboles de decisión, máquinas de soporte vectorial, Bayes, etc.). Asimismo la selección de características para la eliminación de aquellas redundantes o ruidosas se ha explorado limitadamente en las publicaciones científicas antes reportadas. La hipótesis propuesta plantea que es posible obtener modelos de clasificación supervisados con un buen rendimiento, haciendo uso de la combinación de clasificadores y una adecuada selección de características, con el fin de que puedan ser utilizados como herramienta de soporte para detectar las señales MER correspondientes al STN durante una cirugía para implantar un DBS.

1.2 OBJETIVOS DE LA TESIS

Objetivo General:

Obtener un algoritmo de clasificación supervisado que, a partir de las señales obtenidas con microelectrodos de registros (MER) en cirugías realizadas para implantar un neuroestimulador profundo, permita determinar en forma automática y con un desempeño aceptable las señales que se corresponden con el núcleo subtalámico (STN).

Objetivos específicos:

- Utilizar la combinación de clasificadores con el fin de obtener modelos de clasificación supervisados que a partir de características temporales de los registros MER pueda detectar la zona STN con un desempeño superior al reportado actualmente.

- Utilizar métodos de selección de características para mejorar el rendimiento de los clasificadores obtenidos.
- Analizar los aspectos de configuración, entrenamiento y validación de los clasificadores propuestos que permitirían su utilización real durante una cirugía para implantar un DBS.

1.3 ORGANIZACIÓN DEL DOCUMENTO DE TESIS

La presente tesis se encuentra organizada de la siguiente forma: de los capítulos 2 al 5 se desarrolla el marco teórico general. En particular en los capítulos 2 y 3 se aborda la enfermedad del Parkinson y su tratamiento mediante estimulación cerebral profunda, en el capítulo 4 se enmarcan las señales eléctricas cerebrales con foco a las obtenidas mediante micro electrodos de registro y el capítulo 5 presenta el proceso de reconocimiento de patrones y desarrolla los aspectos generales de la minería de datos.

Los materiales y la descripción de la metodología para realizar la tesis se detallan en el capítulo 6. Los capítulos 7, 8 y 9 desarrollan las bases teóricas específicas de los algoritmos utilizados en cada uno de esos capítulos y sus posibilidades de configuraciones, junto con los resultados obtenidos en cada experimento y un análisis de los resultados del propio capítulo.

El capítulo 10 presenta las discusiones y conclusiones de la tesis además de proponer las perspectivas futuras de investigación.

1.4 PUBLICACIONES Y ACTIVIDADES VINCULADAS A LA INVESTIGACIÓN DE LA TESIS

En el Anexo I se detallan las publicaciones y actividades resultado de las investigaciones realizadas durante la presente tesis.

Capítulo 2 Enfermedad de Parkinson

En el presente capítulo se desarrollan los conceptos fundamentales de la enfermedad del Parkinson y sus tratamientos actuales.

2.1 INTRODUCCIÓN

James Parkinson fue el primero en describir la enfermedad que lleva su nombre en 1817, denominándola ‘parálisis agigante’. En 1960 se identificaron los cambios bioquímicos asociados a la enfermedad, y sólo en la década de los ochenta se identificó la causa subyacente de los síntomas del párkinson. Actualmente, se entiende a la Enfermedad de Parkinson (EP) como un desorden neurodegenerativo crónico y progresivo provocado por la pérdida de células productoras de dopamina en la sustancia negra. La dopamina es un neurotransmisor implicado en la transmisión de señales para la coordinación de los movimientos. La falta de dopamina altera la respuesta motora en los enfermos de EP [1]

El porcentaje de personas que pueden desarrollar la enfermedad a nivel mundial se estimaba en el 0,3%, aunque a partir de los 60 años esta prevalencia aumentaría entre el 1 y 2%. En el ámbito internacional, según un informe de la ‘*Global Declaration for Parkinson’s Disease*’ elaborado en 2004, se estima que el Parkinson podría estar afectando a 6,3 millones de personas en todo el mundo. Aunque la mayor cantidad de éstos pacientes tiene más de 60 años de edad, un 10 % de los pacientes tiene menos de 40 años [2], [13].

En cuanto a Europa, se estima que la EP afecta a más de 1,2 millones de personas, pero esta CIFRA IRÁ CRECIENDO A MEDIDA PASEN LOS AÑOS. EN 2007, SOBRE LA BASE DE ESTUDIOS DE PREVALENCIA publicados, se proyectó el número de individuos con EP en las cinco naciones más pobladas de Europa Occidental (Alemania, Francia, Reino Unido, Italia y España) y en las diez naciones más pobladas del mundo (China, India, Estados Unidos, Indonesia, Brasil, Pakistán, Bangladesh, Rusia, Nigeria y Japón). El número de personas con EP, mayores de 50 años, en estos países fue de entre 4,1 y 4,6 millones en 2005 y se duplicarían a entre 8,7 y 9,3 millones en 2030. Como puede observarse en la figura 2.1, en algunas naciones, el crecimiento superaría el 100 % [3], [14].

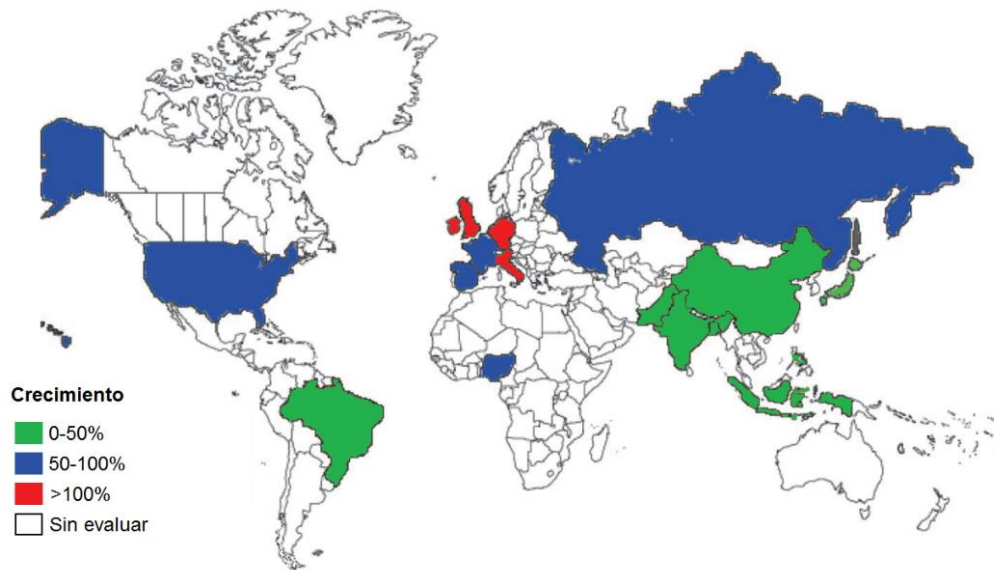


Figura 2.1: Tasas de crecimiento proyectadas al 2030 en cantidad de individuos mayores de 50 años con Enfermedad de Parkinson en los países más poblados de Europa occidental y del mundo. Adaptado de [3].

2.2 GANGLIOS BASALES DEL CEREBRO INVOLUCRADOS EN EL PARKINSON

Los ganglios basales (GB) son un grupo de núcleos subcorticales, que controlan la actividad cortical referente a los movimientos voluntarios y otras funciones de alto nivel. Se encuentran involucrados en el control motor, en el aprendizaje motor, en las funciones ejecutivas y del comportamiento, y en las emociones.

En la parte superior de la figura 2.2 se muestra un corte frontal de un cerebro humano, en el cual puede apreciarse la ubicación anatómica de los GB y de otras estructuras relacionadas.

En su sentido más estricto, el término ganglios basales refiere a los núcleos embebidos en lo profundo de los hemisferios cerebrales, es decir: el estriado y el globo pálido. Según esta acepción, el núcleo subtalámico, la sustancia negra y el núcleo pedunculopontino, que se encuentran en el diencefalo, son llamados núcleos relacionados [15]. Sin embargo, en la práctica de la clínica moderna, se utiliza el término ganglios basales para referirse a todas estas estructuras, ya que se encuentran fuertemente conectadas tanto anatómica como fisiológicamente [16]. En la presente tesis se hará uso de esta última interpretación, por lo que al hablar de GB se hará referencia a:

- el estriado, que incluye al núcleo caudado y al putamen.
- el globo pálido, que consiste en los segmentos interno (GPi) y el externo (GPe).
- el núcleo subtalámico (STN)
- la sustancia nigra, que se divide en: pars reticulata (SNr) y pars compacta (SNc).

Los GB reciben estímulos de varias áreas de la corteza cerebral; la información que se procesa en los GB vuelve principalmente a la corteza (algunas salidas también descienden hacia el tronco encefálico), especialmente al lóbulo frontal, pasando a través del tálamo como se observa en la parte inferior de la figura 2.2. Así, los GB y la corteza cerebral forman circuitos cerrados, que permiten la modulación de la actividad cortical. Los GB pueden dividirse en cuatro categorías según las funciones que cumplen en esos circuitos [5]:

1. Núcleos de entrada: el estriado y el STN, que reciben estímulos corticales.
2. Núcleos de salida: el GPi y la SNr, que proyectan fuera de los ganglios basales, hacia el tálamo y el tronco cerebral.
3. Núcleos conectores: el GPe, que conecta a los núcleos de entrada con los de salida.
4. Núcleos moduladores: SNc, que modula la actividad de los ganglios basales.

El recorrido general de los circuitos corteza - GB puede verse en la figura 2.2 inferior mediante el cual los GB están directamente interconectados con la corteza cerebral y el tálamo a través de diversos lazos independientes y paralelos. Se observan tres lazos los que constituyen tres dominios fisiológicos diferentes: el motor, que trata con el control del movimiento; el asociativo, que se relaciona con el comportamiento y la cognición; y el límbico, relacionado a los estados emocionales [16]. Cualquier desorden en el funcionamiento de estos lazos, se reflejará en desórdenes del movimiento, del comportamiento o la cognición, y/ o en cambios emocionales [16].

2.3 ETIOLOGÍA DE LA ENFERMEDAD DE PARKINSON

A lo largo de los años, el conocimiento sobre los factores etiológicos de la EP ha cambiado notablemente. En la actualidad, se reconoce la influencia de factores tanto ambientales como genéticos, pero se le otorga mayor relevancia a la predisposición genética del individuo [1].

La variable que se relaciona más fuertemente con el inicio de la enfermedad es el envejecimiento. A pesar de este reconocimiento, pareciera ser poco lo que se ha hecho por intentar comprender cómo está involucrado el proceso de envejecimiento con el inicio de la EP.

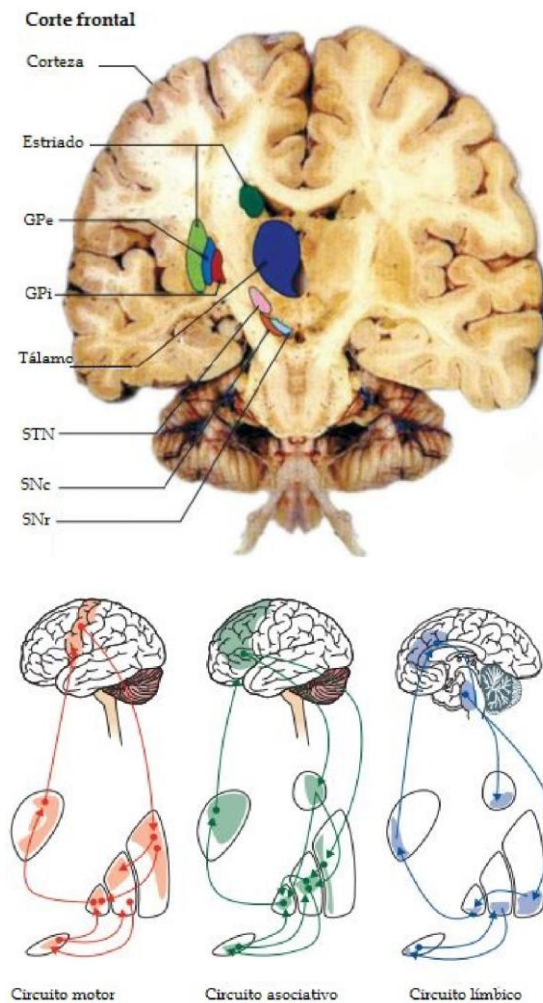


Figura 2.2: La figura superior representa un corte frontal o coronal de un cerebro humano donde se observan los núcleos de los ganglios basales y otras estructuras relacionadas. En la gráfica inferior se observan los diagramas esquemáticos de los circuitos principales entre la corteza y los ganglios basales. Adaptado de [16].

De hecho, la mayoría de los estudios que se reportan ignoran este factor, dado que utilizan individuos jóvenes de modelos animales. La explicación que se puede inferir del conocimiento del proceso de envejecimiento es que a medida que el individuo envejece, los procesos celulares

pierden eficiencia y eficacia, lo que hace a las células más vulnerables a determinadas agresiones tóxicas [1], [14].

Otro factor que no se ha estudiado suficientemente tiene que ver con la predominancia de hombres sobre mujeres cuando se analiza la prevalencia de la EP. Estudiar esto podría arrojar luz a la etiología de la enfermedad, más aun teniendo en cuenta la menor expectativa de vida de la población masculina [1].

La influencia del ambiente en la ocurrencia de la EP está determinada por factores que van desde los más generales -como el rol potencial de la industrialización, vida rural, calidad del agua, infecciones virales o bacterianas-, a los más específicos -como la exposición a solventes orgánicos o a monóxido de carbono-. Recientemente ha habido un interés creciente con respecto a la exposición a agentes agro tóxicos [1].

El hecho que proporcionó el mayor estímulo al interés en el componente etiológico ambiental de la enfermedad, y que dio lugar a una nueva era en la investigación, fue el descubrimiento de los efectos neurotóxicos del 1-metil-4-fenil-1,2,3,6-tetrahidropiridina (MPTP). Este acontecimiento proporcionó el primer modelo viable de los déficits motores de la EP en animales, y generó la posibilidad de predecir la acción de fármacos en humanos. Sin embargo, el parkinsonismo inducido por MPTP difiere de la EP que se produce en los seres humanos, debido a que no es progresiva, no hay formación de cuerpos de Lewy y no hay ninguna patología en otras zonas del cerebro, que sí se ven afectadas en la enfermedad humana. Por lo tanto, aunque muy importante para el avance en el conocimiento, el modelo MPTP podría no transmitir los mensajes críticos necesarios para resolver el misterio de la etiología de la EP [1], [14].

2.4 FISIOPATOLOGÍA DE LA ENFERMEDAD DE PARKINSON

A los fines de la presente tesis, la descripción de la fisiopatología de la EP se abordará desde el modelo clásico del circuito motor, si bien es importante aclarar que existen numerosos hallazgos clínicos y experimentales que no pueden ser explicados por este modelo [14], [17].

De acuerdo con el modelo clásico de la EP, que puede observarse en la figura 2.3, la característica fisiopatológica esencial del estado parkinsoniano es el incremento de la actividad

neuronal en los núcleos de salida GPi/SNr de los ganglios basales, lo que conduce a la inhibición excesiva del sistema motor tálamo-cortical.

En condiciones fisiológicas, la dopamina tiene un rol modulador del circuito motor. Así, su acción consiste en favorecer la vía directa (excitándola) y desfavorecer la vía indirecta (inhibiéndola). Entonces, según las predicciones del modelo clásico, el déficit de dopamina resulta en el cese de esa modulación, por lo que la vía directa deja de ser favorecida y la vía indirecta deja de ser inhibida [18].

La reducción en la inhibición de la vía indirecta conduce a la sobre inhibición del GPe, la desinhibición del STN y el aumento de la excitación de las neuronas de GPi/SNr. Por su parte, la actividad disminuida de la vía directa causa una reducción en su influencia inhibitoria sobre el GPi/SNr (ver figura 2.3).

El resultado neto del déficit de dopamina es una activación excesiva de las neuronas de salida de los GB, lo que resulta en la inhibición excesiva de los sistemas motores, dando lugar a los síntomas característicos de la EP [14].

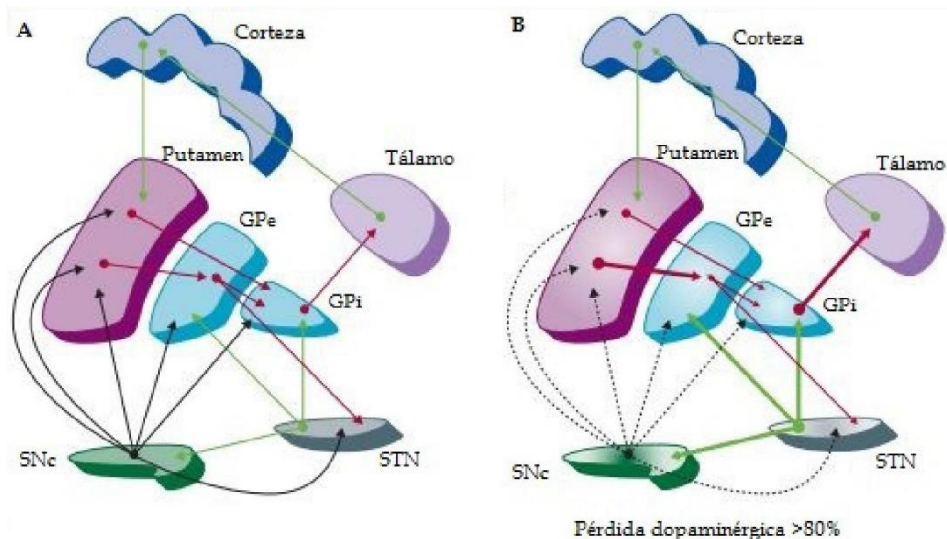


Figura 2.3: A) Esquema del modelo clásico del circuito motor en condiciones fisiológicas normales. B) El mismo circuito pero afectado por una pérdida de células dopaminérgicas mayor al 80 %. Las flechas rojas indican proyecciones inhibitorias GABA-érgicas; las flechas verdes representan proyecciones excitatorias glutamatérgicas; y las flechas negras indican inervaciones dopaminérgicas. Adaptado de [17].

2.5 SÍNTOMAS DE LA ENFERMEDAD DE PARKINSON

Aunque los principales síntomas son motores, porque responden al síndrome dopaminérgico, los circuitos de los GB que utilizan otros neurotransmisores además de la dopamina también se ven afectados. Debido a esto, los pacientes pueden desarrollar síntomas no motores con el progreso de la enfermedad [1].

En la tabla 2.1, se listan varias características clínicas asociadas a la EP. La sintomatología de esta enfermedad es muy compleja y su expresión varía de un sujeto a otro. Sin embargo, existen ciertos síntomas que se aparecen con mayor frecuencia en los pacientes, y son esos los que se detallarán a continuación [18].

SÍNTOMAS MOTORES	SÍNTOMAS NO MOTORES
Bradicinesia, temblor en reposo, rigidez, inestabilidad postural	Desórdenes cognitivos, bradipsiquia
Hipomimia, disartria, disfagia	Depresión, apatía, fatiga
Balanceo de brazos disminuido, marcha arrastrada	Anosmia, ageusia, parestesias
Micrografía, lentitud en la ejecución de actividades diarias	Disfunciones autonómicas
Distonía, blefarospasmo	Desórdenes del sueño

Tabla 2.1: Listado parcial de signos y síntomas de la EP.

2.5.1 SÍNTOMAS MOTORES

Dentro de los síntomas motores que se listaron en la tabla 2.1, existen cuatro que son los más importantes y que se los considera los cuatro síntomas cardinales de la EP. Estos son: la bradicinesia, el temblor en reposo, la rigidez y la inestabilidad postural. A continuación se dan algunos detalles de los mismos:

Bradicinesia: Significa ‘lentitud para realizar un movimiento’ y en EP se define específicamente como la dificultad para planificar, iniciar y ejecutar un movimiento, en el desempeño secuencial y en realizar tareas simultáneas. Es el factor clínico más característico de la EP y es el que pareciera correlacionarse mejor con el grado de deficiencia dopaminérgica [2], [19].

Como en otros de los síntomas de la EP, la bradicinesia es afectada por el estado emocional del paciente. Por ejemplo, pacientes que están inmóviles y que son estimulados por algún factor externo, podrían realizar movimientos rápidos, como el de agarrar un objeto en el aire o escapar corriendo ante una señal de alerta. Este fenómeno, que en principio pareciera paradójico, sugiere que los pacientes con EP preservan intactos los programas motores pero tienen dificultades para iniciarlos sin un estímulo externo

Algunas manifestaciones asociadas a este síntoma son las expresiones faciales disminuidas, la voz suave, el babeo y la marcha arrastrada.

Temblores en reposo: Se trata, tal vez, del síntoma más socialmente relacionado con la EP. Si bien este síntoma se considera distintivo de la patología, suele no ser el más discapacitante para los pacientes, dado que el temblor suele disminuir durante la ejecución de un movimiento y durante el sueño.

La frecuencia característica del temblor en la EP es de 3-5 ciclos/s. Los síntomas usualmente comienzan en un lado del cuerpo y lentamente progresan, en años a décadas, para incluir al otro lado [14]. El temblor de las manos es de tipo supinación-pronación. Este síntoma puede extenderse a los labios, mentón, mandíbula y piernas.

Rigidez: es un síntoma muy común en los pacientes con EP, y está caracterizado por un aumento en la resistencia al movimiento pasivo de los miembros, tanto en extensión, flexión o rotación sobre una articulación [14], [19]. Los músculos permanecen constantemente tensos y contraídos y la persona tiene dolor o se siente tiesa. La rigidez se vuelve obvia cuando alguien más trata de mover o extender un miembro de la persona.

Trastornos posturales: Es el signo típico de la EP y uno de los síntomas más incómodos de la enfermedad, se manifiesta como inestabilidad postural o alteración del equilibrio. A medida que la enfermedad avanza, las personas con la EP tienden a adoptar una postura encorvada, con las piernas un poco dobladas, el tronco echado hacia delante y los brazos flexionados [2]. La alteración del equilibrio puede ocasionar inestabilidad y riesgo de caídas en situaciones de la vida cotidiana, como levantarse de una silla o realizar un movimiento brusco [19].

2.5.2 SÍNTOMAS NO MOTORES

Estos síntomas pueden tener un gran impacto en la calidad de vida de los pacientes, tanto o más que los síntomas motores mencionados anteriormente. Los principales aspectos no motores de la sintomatología de la EP se han listado en la tabla 2.1. A continuación se dan algunos detalles de los que se presentan con mayor frecuencia [2], [14]:

Desórdenes cognitivos: El deterioro cognitivo asociado a la EP está caracterizado por una deficiencia leve en la capacidad intelectual del paciente, pudiéndose afectar también la memoria. Este síntoma ocurre frecuentemente y puede progresar a demencia con el avance de la patología [13]. Varios estudios han reportado una importante incidencia de depresión, apatía, trastornos obsesivos compulsivos, ansiedad y alucinaciones [19].

Disfunciones autonómicas: Las disfunciones autonómicas pueden incluir hipotensión ortostática, estreñimiento, disfunción de la sudoración, disfunción urinaria y sexual [19].

2.6 EVOLUCIÓN Y PROGNOSIS

La EP comienza años antes de que los pacientes y los médicos noten los primeros síntomas motores [1].

Una fase asintomática ha sido identificada en autopsias cerebrales de sujetos clínicamente normales (enfermedad de cuerpos de Lewy incidental) [20]. Esta fase asintomática, sumada a la falta de un test diagnóstico específico, dificulta mucho la detección y el tratamiento temprano de la enfermedad.

Se desconoce la duración de la primera fase asintomática, pero su existencia podría reflejar el hecho de que el cerebro puede compensar con éxito la disfunción dopaminérgica hasta que se alcanza un umbral (aproximadamente 50 % de pérdida neuronal) [20].

Considerando las manifestaciones motoras pueden evidenciarse tres fases en la EP [2], [14]:

Fase inicial: En el primer estadio de la EP, la mayoría de personas pueden realizar sin ayuda muchas de las actividades que conforman la vida cotidiana. Por lo general, en esta primera etapa se pueden seguir manteniendo las obligaciones laborales y las relaciones sociales. Aunque los síntomas son leves y unilaterales (se manifiestan en un lado del cuerpo), es una etapa decisiva en el desarrollo de la enfermedad de acuerdo al tratamiento que pueda iniciarse.

Fase intermedia: En esta etapa, la rigidez y la bradicinesia son marcadas, el temblor ya es manifiesto, al caminar se arrastra una pierna y el braceo desaparece. Sin embargo, los reflejos posturales todavía se conservan y no suelen producirse aún episodios de congelación de la marcha.

Fase avanzada: Esta fase viene acompañada de una importante disminución en las capacidades de las personas con la EP. Aunque durante algunos momentos del día se encuentran menos limitados (lo que se denomina ‘periodos *on*’), predominan aquellos en los que se tiene un difícil control de los síntomas motores (los llamados ‘periodos *off*’). Cuando la pérdida de independencia alcanza un punto en el que, para llevar a cabo acciones de la vida cotidiana - como vestirse o asearse- la persona necesita de la ayuda de un cuidador, es cuando la persona con EP se encuentra en la fase de enfermedad de Parkinson avanzada (EPA). Aparte de estos casos, existen otros síntomas que son definitivos para diagnosticar EPA, como son la disfagia severa, la demencia o las caídas recurrentes.

En la figura 2.4, se presenta un diagrama esquemático de la relación entre la función neuronal, los síntomas clínicos y el paso del tiempo. Como puede observarse en esas curvas, la progresión de la enfermedad no es lineal y la velocidad es mayor en la fase avanzada [14], [19].

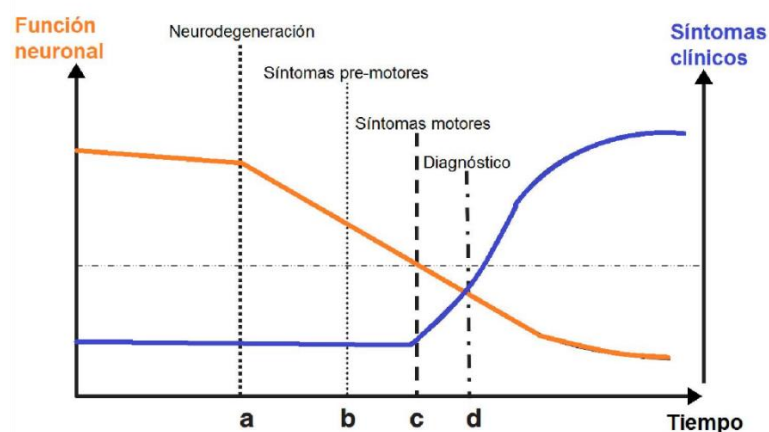


Figura 2.4: Esquema de la evolución de los síntomas clínicos y la degeneración neuronal, en función del tiempo. En **a**, si bien la degeneración neuronal comienza, el cuerpo puede compensar este factor, y los síntomas no se expresan. En **b** la neurodegeneración ha alcanzado un 50 % aproximadamente, y se pueden comenzar a percibir los primeros síntomas no motores: pérdida del sentido del olfato, afectación de sueño, etc. En **c** han comenzado los síntomas motores, pero el paciente aún no ha sido diagnosticado, en este punto un 80 % de las células dopaminérgicas podrían haber degenerado. En **d** ya se ha logrado el diagnóstico. Adaptado de [20].

Para evaluar el avance de la progresión de la enfermedad y de la discapacidad del paciente con EP existen diversas escalas. La escala más ampliamente utilizada en la clínica es la escala de evaluación unificada de la enfermedad de Parkinson (UPDRS por sus siglas en inglés de *Unified Parkinson's Disease Rating Scale*). La misma consiste en la evaluación de 46 ítems divididos en las seis categorías que se listan a continuación [21]:

- Memoria, comportamiento y humor
- Actividades de la vida diaria
- Estado motor
- Complicaciones de la terapia
- Fluctuación clínica
- Otras complicaciones

Con respecto al pronóstico de la enfermedad es importante remarcar que varía de un sujeto a otro. Si bien la EP no es considerada una patología mortal, es crónica y progresiva, acompañada del empeoramiento en los síntomas a lo largo de los años. Dependiendo de las actividades que acostumbre a realizar el paciente, la EP afectará en mayor o menor medida su vida diaria [1].

2.7 TRATAMIENTOS

Aunque aún no se ha descubierto la cura de la EP, existen distintas líneas de tratamientos que permiten mejorar la sintomatología de esta enfermedad, mejorando la calidad de vida de los pacientes. Se pueden clasificar los tratamientos para EP en dos grupos: los farmacológicos, que incluyen a la Levodopa como principal componente; y los quirúrgicos, que pueden ser ablativos o estimulativos [14].

Como no existe actualmente un tratamiento que detenga o modifique el curso de la EP, todas las opciones terapéuticas, ya sean farmacológicas o quirúrgicas, están orientadas a tratar los síntomas que experimenta el paciente. Dado que la expresión sintomática varía entre un paciente y otro, los tratamientos se adaptan a los síntomas particulares de cada paciente. En la mayoría de los casos, las terapias comienzan cuando la sintomatología interfiere con la

capacidad del paciente de realizar sus actividades diarias. Con el progreso de la enfermedad, los tratamientos deben ser revisados y ajustados [13].

A continuación se describen con más detalles los tratamientos utilizados con mayor frecuencia.

2.7.1 TRATAMIENTOS FARMACOLÓGICOS

Los tratamientos farmacológicos para mejorar los síntomas motores se pueden clasificar según sus mecanismos de acción [13], [19]:

Terapia de reemplazo de dopamina: La Levodopa es un precursor de la dopamina que atraviesa la barrera hematoencefálica. Es la droga más utilizada en el tratamiento de la EP.

Inhibidor de la decarboxilasa: La Carbidopa evita que la Levodopa se convierta en dopamina antes de llegar al encéfalo. Además, ayuda a prevenir los efectos secundarios de la Levodopa, tales como las discinesias.

Agonistas dopaminérgicos: Estimulan los receptores dopaminérgicos de las neuronas indistintamente de la disponibilidad de dopamina presináptica. Se utiliza en estadios iniciales para retrasar el inicio del tratamiento con Levodopa.

Inhibidores de la Monoamina oxidasa: Estos fármacos disminuyen la actividad de la monoamina oxidasa, enzima que destruye a la dopamina luego de haber actuado en el encéfalo.

Inhibidores de la Catecol O-metiltransferasa: Estos fármacos evitan la degradación de la Levodopa y están indicados en EP avanzados combinado con Levodopa.

Medicación anticolinérgica: Este fármaco disminuye la concentración de acetilcolina en el cerebro, mejorando el balance entre ese neurotransmisor y la dopamina.

Amantadina: Este medicamento aumenta la actividad de la dopamina en el cerebro y afecta el sistema glutamatérgico. Se utiliza para tratar las discinesias causadas por Levodopa.

La terapia de reemplazo de Dopamina es la más ampliamente utilizada y la más eficaz cuando se quieren controlar los síntomas motores de la EP. Una terapia prolongada con Levodopa puede producir discinesias. La discinesia es la dificultad o la distorsión en la

ejecución de movimientos voluntarios, y aparece usualmente como efecto secundario del tratamiento crónico con Levodopa [13].

Con respecto al tratamiento de los síntomas no motores, las opciones son muy variadas y responden exclusivamente a la sintomatología particular del paciente, por lo que no se tratarán en este texto.

2.7.2 TRATAMIENTOS QUIRÚRGICOS

En pacientes en los que la medicación ya no está controlando los síntomas, en los que los efectos secundarios son muy discapacitantes, algunas opciones quirúrgicas pueden ser consideradas como parte del tratamiento [10]. Dentro de las terapias quirúrgicas disponibles, se encuentran las ablativas y las estimulativas [14].

Los tratamientos ablativos producen muerte neuronal por radiofrecuencia de una estructura determinada del cerebro. Dependiendo de la estructura donde se realiza la ablación reciben distintos nombres [10]:

Palidotomía: cuando el núcleo basal globo pálido es la estructura objetivo. Normalmente son ablaciones unilaterales ya que son bien toleradas por los pacientes (las bilaterales han presentado efectos adversos severos). Los pacientes continúan con la mismas dosis de Levodopa previas a la palidotomía pero se suprimen la discinesia producida como efecto colateral del medicamento. Estudios a corto plazo han demostrado que la palidotomía unilateral mejora entre un 25% a un 30% la escala motora UPDRS.

Talamotomía: cuando el núcleo de Vim es la estructura objetivo. Es recomendada en pacientes con la EP si existe mucho temblor. El beneficio es comparable con la terapia estimulativa Vim-DBS con mayor tasa de riesgos durante las cirugías lo que puede producir efectos adversos posteriores y permanentes tales como trastorno de la programación motora del habla (disartria) y/o falta de equilibrio.

Subtalamotomía: cuando el STN o una porción del mismo es la estructura objetivo. Actualmente se prefiere aplicar la ablación sobre la porción sensorio-motora del STN. En general el resultado ha sido positivo en el caso de la acinesia, rigidez, temblor y discinesia inducida por Levodopa. Permite reducir la dosis de dopamina en forma similar a los tratamientos estimulativos como STN-DBS. En general se realiza unilateral dado que hay

mayor riesgo de afectar el habla o producir ataxia en los casos bilaterales. La subtalantomía presenta como efectos adversos el hemibalismo o hemicorea que suelen desaparecer a corto plazo, quedando presente a largo plazo en el 15% de los casos.

Los tratamientos estimulativos o DBS consisten en estimular eléctricamente una estructura cerebral profunda y ofrece muy buenos resultados para mejorar los síntomas motores de los EP y otros trastornos del movimiento. Actúan básicamente como un marcapasos del cerebro abordándose los detalles de la terapia DBS en el capítulo 3.

Existen otros tratamientos quirúrgicos que realizan el transplante de células adrenales autólogas o células fetales en la sustancia negra con el objetivo de que produzcan dopamina, pero los trabajos más recientes ponen de manifiesto su baja eficacia y seguridad [10].

2.7.2.1 Consideraciones sobre la elección de un tratamiento ablativo o DBS

La DBS tiene como ventaja que es parcialmente reversible con solo apagar la estimulación, adaptable a las necesidades del paciente a lo largo de los años, es bilateral y en general presenta menos efectos adversos. La ubicación adecuada de los electrodos de estimulación en la DBS es un aspecto crítico en esta técnica, ya que una incorrecta implantación implica beneficios nulos y la aparición de efectos adversos [22].

La cirugía ablativa exitosa presenta sus ventajas. En primer lugar, no requiere un tratamiento postoperatorio. En segundo lugar, no lleva ningún riesgo de hardware, cambio de baterías y complicaciones tales como erosiones de la piel e infecciones.

En general los tratamientos ablativos son más utilizados en los países donde la terapia DBS es muy costosa y poco accesible.

Recientemente, las lesiones mediante ultrasonido focalizado de alta intensidad (HIFU) y los bisturís de radiación gamma, los cuales no requieren una cirugía incisiva del cráneo y cerebro para producir la ablación, han comenzado a explorarse abriendo un panorama nuevo e incierto en este tipo de terapias [10].

Si bien estos procedimientos ablativos unilaterales se realizan en determinadas circunstancias como se describió anteriormente, los tratamientos con DBS, en la actualidad han ganado terreno progresivamente y son los más utilizados dentro de los tratamientos quirúrgicos [22].

Capítulo 3 Estimulación cerebral profunda

Inicialmente, los tratamientos quirúrgicos de los desórdenes del movimiento eran exclusivamente ablativos y la estimulación eléctrica de los ganglios basales se realizaba únicamente para asegurar que la lesión de una estructura cerebral determinada causara los efectos deseados. En la década de 1960, la Levodopa para el tratamiento de la EP reemplazó casi por completo al tratamiento quirúrgico de la enfermedad [23].

El primer indicio del desarrollo de estimuladores implantables de características aptas para el uso crónico en humanos fue en el área de la cardiología, específicamente con el marcapasos cardiaco. En 1960, se logró tratar un paciente con el primer marcapasos recargable a través de la piel, totalmente implantado. La disponibilidad de esos dispositivos allanó el camino para la utilización de la estimulación eléctrica en trastornos funcionales del sistema nervioso, como desordenes del movimiento, dolor y epilepsia, entre otros [23].

La idea de utilizar la estimulación subcortical crónica como una terapia en sí misma para los trastornos del movimiento fue presentada por primera vez en la década de 1970 [14]. Recién en 1997 la Administración de Alimentos y Medicamentos de Estados Unidos (FDA) aprobó el uso de la estimulación cerebral profunda (DBS) para el tratamiento de EP. En la actualidad, la DBS ya ha sido utilizada rutinariamente en la clínica por casi dos décadas y es una alternativa terapéutica de alto valor para pacientes con EP [6].

3.1 DEFINICIÓN Y CARACTERÍSTICAS GENERALES

La DBS comprende un conjunto de técnicas quirúrgicas cuyo objetivo terapéutico es modificar la transmisión de las señales nerviosas, a fin de producir la inhibición, excitación o interrupción de la actividad neuronal patológica mediante la utilización de un equipo implantable que descarga impulsos eléctricos en una estructura profunda del cerebro [24]. Esta terapia es lo más avanzado del tratamiento neuroquirúrgico para los síntomas motores de la EP.

La terapia consiste en la liberación de una corriente eléctrica de frecuencia, amplitud y ancho de pulso controlados, sobre una estructura del encéfalo determinada. Una característica que diferencia este tratamiento de sus precursores ablativos, es que no es destructivo, es reversible y es adaptable a la expresión sintomática del paciente [6]

3.1.1 DETALLES TÉCNICOS DE LA DBS

Un estimulador cerebral profundo consta de tres partes: (1) el electrodo de estimulación, (2) la guía o alambre de conexión, y (3) el generador de pulsos implantable (ver figura 3.1). Estas partes se detallan a continuación [5], [14]:

1. El **electrodo de estimulación** consta de cuatro contactos en su extremo. Durante la cirugía, se inserta profundamente en el cerebro de tal manera que, al menos uno de los contactos del electrodo, se ubique en el interior de la estructura objetivo. El otro extremo de la guía es anclado al cráneo del paciente.

2. La **guía de conexión** va por debajo de la piel desde el sitio donde fue anclado el electrodo de estimulación hasta el pecho o abdomen donde se conecta al generador de pulsos implantable.

3. El **generador de pulsos implantable (IPG)** se implanta en el pecho (bajo la clavícula) o en el abdomen. El IPG es un dispositivo similar a un marcapasos cardíaco que envía impulsos eléctricos a través del cable de conexión al electrodo de estimulación implantado en el cerebro.

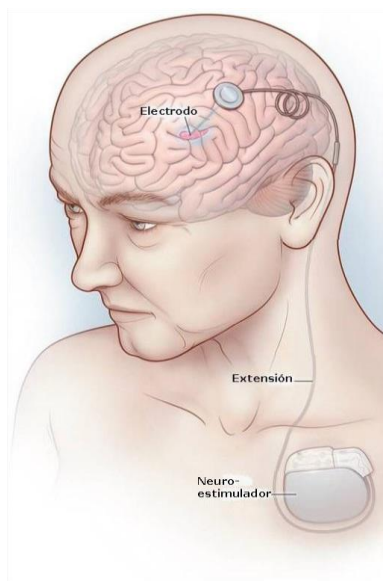


Figura 3.1: Esquema de un estimulador cerebral profundo implantado. Pueden observarse las partes componentes. Adaptado de [25] .

Dentro de las modalidades de estimulación, puede optarse entre unipolar o bipolar, continua o cíclica, con un voltaje de entre 0 y 10.5 V, una frecuencia entre 2 y 185 Hz y una amplitud de pulso de 60 a 450 μ s [24]. Los parámetros estándares de estimulación para EP son [14], [26]:

- Frecuencia de estimulación: 130 Hz
- Ancho de pulso: 60 μ s
- Amplitud del pulso: 1-4 V

La DBS fue aprobada por la FDA para el tratamiento de las siguientes afecciones [23]:

- Temblor esencial
- Enfermedad de Parkinson
- Epilepsia
- Trastorno Obsesivo Compulsivo

No obstante, la DBS está siendo utilizada -de manera experimental- en otras indicaciones, tales como síndrome de Gilles de la Tourette, depresión, obesidad, dolor crónico, esquizofrenia y otros.

Dependiendo de la patología que se quiera tratar, la estimulación eléctrica se realiza sobre diferentes estructuras cerebrales [23].

Particularmente, para la EP, existen dos estructuras cerebrales profundas, cuya estimulación ha demostrado tener buenos efectos sobre los síntomas motores de la enfermedad. Estas estructuras son el STN y GPi. La estimulación de cualquiera de estas estructuras arroja resultados similares y, en los ensayos clínicos, ambas aparecen como igualmente efectivas y seguras [6]. Sin embargo, la estimulación del STN tiene la ventaja de permitir una mayor reducción de la dosis del tratamiento farmacológico dopaminérgico del paciente. Además, la estimulación del STN requiere menor energía, lo que resulta en mayor vida útil del IPG [6].

En la tabla 3.1 pueden observarse algunas estructuras objetivos para el tratamiento de otras patologías. A excepción de los cuatro primeros, los demás procedimientos se encuentran en etapas experimentales.

PATOLOGÍA	ESTRUCTURA OBJETIVO
Enfermedad de Parkinson	STN, GPi
Temblor Esencial	Vim, STN
Epilepsia	ATN, foco epileptogénico (STN, hipocampo, cerebelo, te)
TOC	VC/VS, (ALIC, NAc, STN, ITP)
Depresión	Cg25, ALIC, NAc
Obesidad	VMH, LH
Dolor crónico	PAG, VPM/VPL
Síndrome de Tourette	CMt, GPi, ALIC, NAc
Alzheimer	NBM, fornix
Esquizofrenia	NAc/VS, VTA

Tabla 3.1: Patologías que podrían ser tratadas con DBS y su estructuras objetivo. STN: núcleo subtalámico, GPi: globo pálido interno, Vim: núcleo ventral intermedio del tálamo, ATN: núcleo anterior del tálamo, VC/VS: cápsula interna/estriado ventral, ALIC: extremidad anterior de la cápsula interna, NAc: núcleo accumbens, ITP: pedúnculo inferior talámico, Cg25: área subcallosa del cíngulo, VMH: hipotálamo ventromedial, LH: hipotálamo lateral, PAG: sustancia gris periacueductal, VPM/VPL: tálamo ventral posteromedial/posterolateral, CMt: núcleo centromedial del tálamo, NBM: núcleo basal de Meynert, VTA: área tegmental ventral. Adaptado de [23].

3.1.2 PACIENTES CON EP CANDIDATOS PARA LA DBS

El criterio de selección de pacientes para ser sometidos a la terapia DBS es variable y depende tanto de la institución en la que se llevará adelante el procedimiento quirúrgico como del criterio del equipo médico a cargo de la cirugía. Si bien es un tratamiento sintomático y potencialmente reversible, los costos de la intervención son muy altos y se debe ser estricto a la hora de seleccionar a un paciente como posible candidato a ser sometido a esta terapia.

A continuación se exponen algunos criterios de consenso para seleccionar pacientes para la DBS [2], [14].

Diagnóstico claro: Debe estar claramente definido el diagnóstico de EP idiopática. Los pacientes con síndromes relacionados, no responden a la terapia DBS, por lo que deben excluirse.

Sensibilidad a la medicación oral: Debe existir evidencia clara de que los síntomas motores del paciente disminuyen considerablemente en respuesta a la medicación oral dopaminérgica.

Funciones cognitivas intactas: Los pacientes con disfunción cognitiva tienen dificultades para tolerar la cirugía sin anestesia total; sus condiciones pueden empeorar luego de la

operación incluso si el rendimiento motor mejora. En pacientes con disfunciones cognitivas está contraindicada la terapia DBS.

Grado de avance de la enfermedad: Pacientes en estadios muy avanzados de EP, que suelen ser los que más requieran recibir un tratamiento eficaz, en general no responden de la mejor manera al tratamiento. Por otro lado, pacientes en estadios muy tempranos, cuyos síntomas motores están bien controlados con la medicación farmacológica, tampoco son buenos candidatos para este tratamiento.

Ausencia de otras patologías crónicas: La presencia de otras patologías como hipertensión arterial crónica no controlada, o cualquier otra disfunción sistémica mayor, aumenta el riesgo y relativiza el beneficio del tratamiento.

Edad del paciente: Los beneficios de la terapia DBS se ven reducidos en pacientes de edad avanzada. Pacientes de más de 85 años son raramente sometidos a esta terapia.

3.2 ACCIÓN FISIOLÓGICA DE LA DBS

El mecanismo a través del cual la DBS logra controlar los síntomas motores del paciente con EP aún no está totalmente aclarado. A continuación se dan detalles de las tres teorías que han sido argumentadas por diferentes autores y que podrían explicar el efecto de la DBS en el control de la sintomatología motora del paciente con EP. Estos autores han reportado que la DBS podría inhibir elementos neuronales locales, excitarlos, o bien producir una respuesta multifásica cuyo resultado final sería la interrupción del flujo patológico de información. En las secciones siguientes, se explican brevemente estas tres teorías [14], [23].

3.2.1 LA DBS INHIBE ELEMENTOS NEURONALES LOCALES

Como los efectos derivados de la terapia DBS son muy similares a los resultados del alivio de los síntomas motores alcanzados con terapias ablativas, la suposición más directa es que el mecanismo que subyace a la estimulación es el de la inhibición local de la actividad neuronal. De hecho, varios estudios han encontrado que la DBS en el STN y en el GPi produce inhibición en neuronas vecinas al foco de estimulación [5].

Uno de los efectos más comunes de la DBS en STN y GPi es la reducción de la tasa de disparo de neuronas cercanas al sitio de estimulación. En pacientes con EP, durante la cirugía estereotáctica, se ha registrado supresión de actividad neuronal selectiva en los alrededores de los sitios de estimulación [5].

3.2.2 LA DBS EXCITA ELEMENTOS NEURONALES LOCALES

Varios autores han reportado resultados que sugieren que la DBS produce una excitación en elementos neuronales locales.

Se ha reportado que la DBS en STN produce un aumento en la actividad de neuronas del GPi a través de la vía excitatoria STN-GPi. Asimismo, la DBS en GPi produce inhibición de la actividad de neuronas talámicas, dado que estimula la vía inhibitoria GPi-tálamo [5].

3.2.3 LA DBS INTERRUMPE O MODIFICA EL FLUJO PATOLÓGICO DE INFORMACIÓN

Algunos autores han examinado la respuesta de neuronas del GPi ante una estimulación en la región motora de la corteza cerebral. Como resultado encontraron que la estimulación cortical induce una respuesta trifásica compuesta de una excitación temprana, una inhibición y, por último, una excitación tardía en el GPi; mediadas por las vías hiperdirecta córtico-STN-GPi, directa córtico-estriado-GPi, y la indirecta córtico-estriado-GPe-STN-GPi, respectivamente [5]

En los mismos estudios, se analizó el impacto de la estimulación eléctrica en GPi sobre el flujo de información descrito. Como resultado, se encontró que, tanto las respuestas corticales evocadas como las descargas espontáneas, fueron completamente inhibidas, sugiriendo que la DBS en GPi bloquearía el flujo de información a través de la estructura estimulada (Ver figura 3.2) [5].

Se han obtenido resultados similares en estudios realizados en STN. Tanto la DBS en STN como la terapia lesionante en STN bloquearían el flujo de información a través del mismo y normalizarían el balance entre las entradas inhibitorias de la vía directa y las entradas excitatorias de las otras dos vías hacia el GPi, llevando a un alivio efectivo de la sintomatología motora del paciente con EP [14].

Estudios más recientes [27] también han demostrado en monos que la DBS en STN modifica la actividad neuronal aberrante en un patrón neuronal más fisiológico.

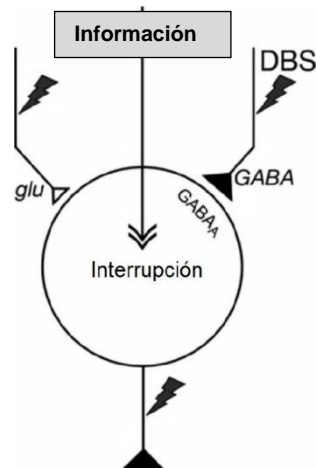


Figura 3.2: Mecanismo de interrupción. La DBS disocia las señales de entrada y salida en el núcleo estimulado, resultando en la interrupción del flujo patológico de información. GABA: ácido gamma-aminobutírico; GABA_A: receptor GABA_A; glu: glutamato. Adaptado de [5].

3.3 ASPECTOS QUIRÚRGICOS

La mayoría de personas con EP que se someten a la terapia DBS requieren implantación en ambos lados del cerebro para controlar totalmente los síntomas. Para personas con EP que tienen temblor u otros síntomas unilateralmente, se puede considerar la estimulación en un solo hemisferio. La DBS también puede mejorar los síntomas parkinsonianos en personas con EP que se han sometido a cirugías cerebrales previas, tales como los antiguos procedimientos como la palidotomía o la talamotomía [13].

El tiempo total del procedimiento suele insumir unas 7-10 horas y en los controles posteriores a la implantación, son necesarias numerosas pruebas para determinar cuáles son los parámetros de los impulsos eléctricos que mejor respuesta clínica producen; este procedimiento suele durar varios días, aunque a veces puede prolongarse hasta meses [24].

El objetivo de la cirugía de DBS es implantar los electrodos en una estructura determinada del cerebro. Para ello, el cirujano utiliza diversas herramientas que lo auxilian en el procedimiento.

Las principales herramientas de las que se vale el equipo médico durante la cirugía para lograr la localización de la estructura objetivo lo más rápido posible y generando el mínimo daño son [5], [7], [14]:

- Marco estereotáctico
- Estudios de imágenes médicas
- Registros de microelectrodos

Estas herramientas se describirán en las secciones siguientes. Una vez que se cree haber alcanzado el sitio correcto de estimulación, se realiza una descarga de corriente exploratoria para evaluar la respuesta clínica del paciente y descartar la aparición de efectos secundarios.

3.3.1 MARCO ESTEREOTÁCTICO

El marco estereotáctico cumple la función de proveer de un marco de referencia para el abordaje preciso y mínimamente invasivo de una estructura cerebral profunda. Este sistema posee tornillos y barras para sostener la cabeza y colocarla en una posición fija con referencia a un sistema de coordenadas.

El marco estereotáctico puede utilizar las tres coordenadas en un esquema ortogonal de referencia (coordenadas cartesianas), o bien un sistema de coordenadas polares, también con tres coordenadas: ángulo, profundidad y posición antero-posterior. En la figura 3.3 puede observarse un paciente con el marco estereotáctico ya colocado.

3.3.2 NEUROIMÁGENES PARA LA IDENTIFICACIÓN DEL TARGET

Además del uso del marco estereotáctico, las herramientas más importantes de las utilizadas para la determinación del lugar de estimulación son las neuroimágenes. En este sentido, se utilizan tanto imágenes de resonancia magnética (RMI) como imágenes de tomografía computada (TC), que suelen combinarse para aprovechar las ventajas que cada técnica presenta.



Figura 3.3: Foto de paciente con el marco estereotáctico colocado. En este caso se utiliza el modelo G del Sistema Esterotáctico Leksell. Extraída de [5].

En particular, la resonancia magnética tiene mejor resolución espacial en los tejidos blandos, por lo que permite distinguir entre diferentes estructuras cerebrales máxime con equipos RMI de 1,5 T o superiores; por su parte, la tomografía computada tiene la ventaja de que, si se cuenta con la tecnología apropiada, puede ser realizada en el quirófano, durante la cirugía de implantación, otorgando una vía de control intraoperatorio.

Las trayectorias seguidas en cada cirugía se planifican en base a estudios de RMI y de TC. En la planificación se busca evitar que las cánulas atraviesen los surcos principales, los ventrículos y los vasos sanguíneos de mayor calibre, para evitar posibles hemorragias u otras complicaciones quirúrgicas.

Las neuroimágenes pueden utilizarse también como una vía para el control intraoperatorio de la implantación de los electrodos. Si el quirófano se encuentra equipado con un equipo para realizar estudio de TC, es posible tomar estas imágenes durante la cirugía.

Estas imágenes, se toman con el marco estereotáctico ya fijado, por lo que son utilizadas como imágenes de referencia. Luego, las imágenes de TC se fusionan con RMI obtenidas previas a la cirugía.

3.3.3 REGISTROS DE MICROELECTRODOS (MER) EN LA IDENTIFICACIÓN DEL TARGET

En la implantación de electrodos de estimulación para terapia DBS, se utilizan microelectrodos que registran la actividad neuronal local a medida que se penetra hacia el

núcleo basal objetivo o target. Además de la forma de la señal y de los patrones que se evidencian en ella, y que pueden ser visualizados en un monitor, los médicos disponen de la señal sonora correspondiente a estas señales. El objetivo es localizar el target, y en éste, el lugar óptimo de implantación de los electrodos.

A medida que se avanza con los microelectrodos, se registran las descargas neuronales locales. Estas descargas son características de la estructura cerebral que se está atravesando. Es decir, los patrones de disparo que observa el equipo médico varían de una estructura a otra, e incluso la variación puede ser grande en diferentes dominios de una misma estructura cerebral. En la figura 3.4 se puede observar ejemplos de las diferentes señales obtenidas con los MER en distintas estructuras cerebrales hasta llegar al núcleo objetivo. En el capítulo 4 se desarrollaran con mayor detalle las características temporales y frecuenciales de los registros eléctricos neuronales, incluyendo las señales de los MER de la presente sección. Cuando se ha seleccionado el lugar en el que se hará la implantación definitiva, se retiran estos microelectrodos y se colocan los electrodos de DBS.

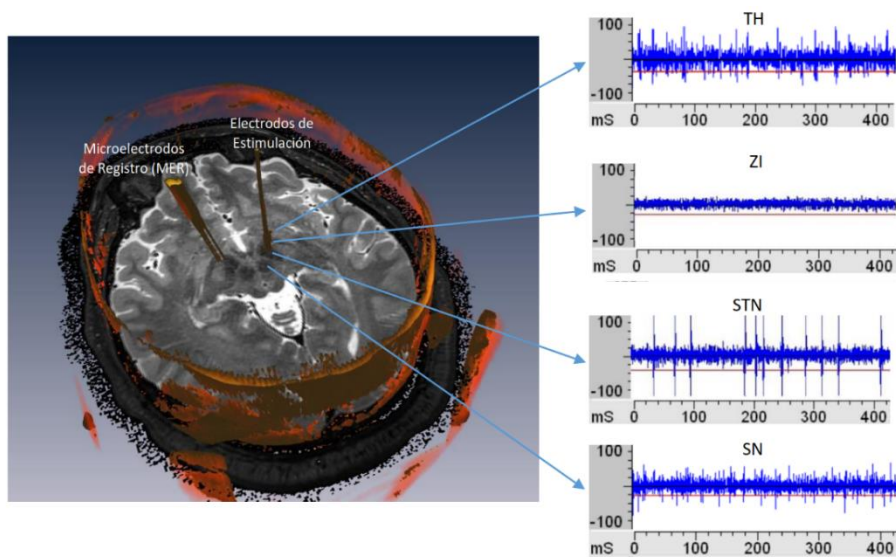


Figura 3.4: Neuroimagen de un paciente con electrodo de estimulación implantado (hemisferio izquierdo), microelectrodos de registro (MER) en el hemisferio derecho. Se observa también registros eléctricos característicos del tálamo (TH), zona incerta (ZI), núcleo subtalámico (STN) y sustancia negra pars reticulata (SNr).

Capítulo 4 Registro de las señales eléctricas neuronales

Las neuronas producen dos tipos principales de actividad eléctrica: potenciales de acción y potenciales postsinápticos. Los potenciales de acción son espigas de voltaje discretas que viajan desde el inicio del axón, en el cuerpo celular, hasta los terminales del mismo, donde los neurotransmisores se liberan. Por su parte, los potenciales postsinápticos son los voltajes que emergen cuando los neurotransmisores se unen a receptores en la membrana de la célula postsináptica, provocando que se abran o cierren canales iónicos, y llevando a un cambio gradual en el voltaje a través de la membrana celular [18].

Aunque estos dos fenómenos son los principales responsables de la actividad eléctrica cerebral, existen muchos otros eventos eléctricos que suceden entre las células del tejido nervioso y que aportan componentes a las señales cerebrales. Cualquier tipo de corriente transmembrana contribuye al campo eléctrico extracelular, que se constituye por la superposición de todos los procesos iónicos, desde los potenciales de acción hasta las fluctuaciones más lentas en las células gliales. Todas las corrientes en el cerebro se superponen en cualquier punto del espacio para generar una diferencia de potencial en ese lugar.

En las siguientes secciones se hará referencia a los eventos eléctricos neuronales y su registro abordando los vinculados con la presente tesis, sin dar un detalle pormenorizado de los procesos involucrados, dado que excede los propósitos del trabajo.

4.1 TÉCNICAS DE REGISTRO

Los diferentes procesos que se suceden en el tejido nervioso involucran flujos de cargas que atraviesan las membranas celulares excitables de las neuronas, generando diferencias de potencial. Existen diversas técnicas que permiten medir la actividad eléctrica del cerebro y, según estas, las señales resultantes suelen identificarse con diferentes nombres. Así, la diferencia de potencial registrada ha sido referida como:

- EEG (Electroencefalograma) cuando se la registra desde el cuero cabelludo,
- ECoG (Electrocorticograma) cuando se la registra con una grilla de electrodos subdurales en la superficie de la corteza

- microEEG (También conocido como EEG profundo, EEG intracraneal cuando se lo registra con un electrodo pequeño ubicado dentro del cerebro.

En la figura 4.1 se puede ver la ubicación de los electrodos de registro utilizados en cada una de las técnicas mencionadas. En esa imagen, los electrodos para registros de microEEG se muestran en la corteza cerebral, pero estos pueden ubicarse también en estructuras más profundas.

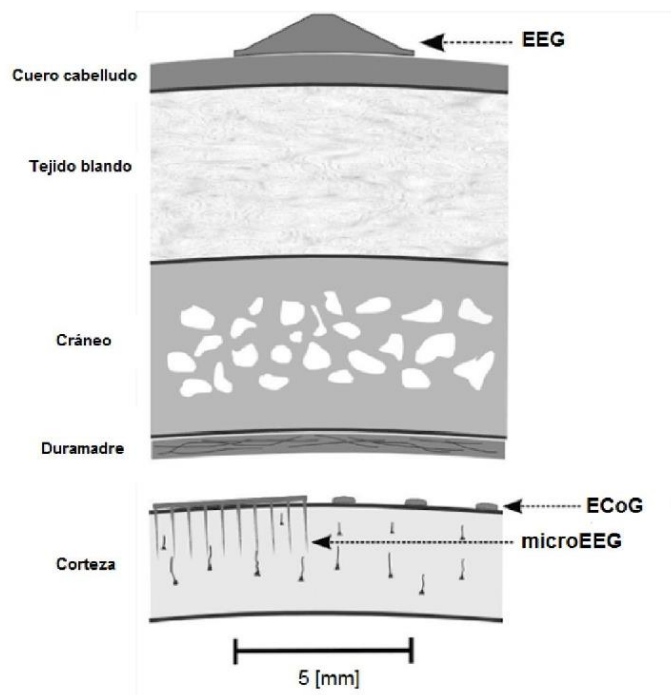


Figura 4.1: Técnicas de registro de actividad eléctrica cerebral. Adaptado de [28].

Aunque todas estas técnicas miden la actividad eléctrica neuronal, las señales obtenidas no son iguales unas a otras. Esto se debe, en parte, a que la diferente ubicación de los electrodos provoca que las señales sean filtradas por diferentes tejidos que se interponen entre los eventos generadores y el lugar de registro.

Más allá de cuál sea la técnica utilizada para el registro, las señales obtenidas dan información acerca del comportamiento colectivo de grupos de neuronas. Descomponer esas señales de manera que se pueda interpretar cuáles son los eventos individuales que la componen es un problema complejo que se conoce como "el problema inverso" [29]. Parte de la complejidad de este problema radica en que diversos eventos celulares combinados pueden

generar una misma señal macroscópica. Además, la complejidad es mayor porque deben tenerse en cuenta otros factores que afectan de diversas maneras a la señal extracelular que se registra, como el filtrado generado por los diversos tejidos.

4.2 FACTORES QUE AFECTAN AL REGISTRO DE ACTIVIDAD ELÉCTRICA NEURONAL

Los mayores determinantes de la fuerza del campo extracelular son la alineación espacial de las neuronas, la sincronía temporal y las propiedades de filtrado de los tejidos que se interponen entre la fuente de señal y el electrodo. En esta sección se describirá cómo estos sucesos se reflejan en las señales de EEG y microEEG.

4.2.1 SINCRONIZACIÓN TEMPORAL

Las corrientes extracelulares de varios sucesos individuales deben solaparse en el tiempo para inducir una señal medible, lo cual es claramente registrable en los eventos lentos, tales como las corrientes sinápticas [29].

Cuando se genera un potencial de acción, la corriente fluye rápidamente hacia adentro y hacia afuera en un punto del axón, y esto se repite en puntos sucesivos del mismo, hasta que se alcanza un terminal sináptico. Si dos neuronas con sus axones paralelos transmiten potenciales de acción exactamente al mismo tiempo, los voltajes de las dos neuronas se suman. Sin embargo, si una neurona dispara apenas después de la otra, la corriente va a fluir hacia adentro de un axón al mismo tiempo que está fluyendo hacia afuera del otro, y esas señales se van a cancelar. Dado que grandes grupos de neuronas no suelen disparar precisamente al mismo tiempo (esto es, dentro de micro segundos de diferencia), los potenciales de acción no suelen ser registrados desde el cuero cabelludo.

Aunque los potenciales de acción generan deflexiones de voltaje de gran amplitud cerca del axón, son de corta duración (menor a 2 ms) y neuronas vecinas raramente disparan de manera sincrónica dentro de ventanas temporales tan pequeñas, en condiciones fisiológicas [29].

Por su parte, los potenciales postsinápticos típicamente duran decenas o centenas de milisegundos. Si bien solamente es posible registrarlos desde el cuero cabelludo cuando varios

ocurren al mismo tiempo, el solapamiento temporal es más fácil de alcanzar para estos eventos lentos.

En el caso de las señales de microEEG, los potenciales de acción pueden contribuir sustancialmente a los componentes de alta frecuencia de la señal. La amplitud de la señal extracelular registrada se ve influenciada por la sincronía en el disparo de las neuronas [29]. Los potenciales postsinápticos de las neuronas cercanas al electrodo también son registrados por los microEEG.

4.2.2 FACTORES GEOMÉTRICOS

La orientación espacial de las neuronas influye en la sumación de sus eventos eléctricos. Para que ocurra sumación espacial de los potenciales de acción, debe haber grandes grupos de neuronas orientadas de manera similar. En el caso de los potenciales postsinápticos, la sumación espacial ocurre más fácilmente, dado que éstos están confinados en las dendritas y los cuerpos celulares neuronales.

Las neuronas piramidales de la corteza cerebral están orientadas perpendicularmente a la superficie cortical y sus dipolos se suman entre sí. Consecuentemente, las señales registradas desde el cuero cabelludo, casi siempre reflejan neurotransmisiones que ocurren en las células piramidales de la corteza [29]. Estructuras no laminares, como los ganglios basales, no generan señales que puedan ser registradas desde el cuero cabelludo. Así, solamente una fracción de la actividad cerebral conduce a señales en el EEG.

Como resultado de lo hasta aquí expresado, las señales de EEG casi siempre reflejan potenciales postsinápticos más que potenciales de acción, y proveen una medida directa, con una resolución de milisegundos, de la actividad neuronal mediada por neurotransmisores [29].

Por su parte, en los microEEG, se captan las señales de potenciales de acción de las neuronas cercanas. Cuanto mayor sea la distancia entre el electrodo de registro y la fuente de corriente, menos informativa será la medida respecto de los eventos que ocurren en la ubicación de esa fuente, dado que la amplitud del potencial registrado decrece con la distancia entre la fuente y el sitio de registro. A medida que aumenta la distancia hasta el electrodo, también se produce la inclusión de otras señales interferentes, que llevan a una promediación espacial [29]. Además,

el campo extracelular se ve influenciado no solo por la magnitud y el signo de las corrientes individuales, sino también por la densidad espacial de fuentes de corriente [29].

4.2.3 FILTRADO Y ATENUACIÓN DE LAS SEÑALES

Además de los factores geométricos y de sincronización temporal, un factor importante que impacta en las señales registradas es el filtrado que realizan los diferentes tejidos que se interponen entre las fuentes generadoras de señal y el electrodo de registro.

Las señales de EEG deben propagarse a través de varios medios, tales como el líquido cefalorraquídeo, la duramadre, el cráneo, los músculos y la piel, por lo que están mucho más sujetas a fenómenos de filtrado y difusión en esos medios. Las señales de microEEG se registran muy cerca de las fuentes neuronales pero también son filtradas debido a que los electrodos de registro están separados de las fuentes por pequeñas porciones de tejido [29].

Por las propiedades de filtrado del tejido nervioso, los potenciales de acción tienen una participación limitada en las actividades registradas en los EEG y los microEEG. Las altas frecuencias (mayores a 100 Hz), tales como las producidas por los potenciales de acción, están sujetas a una gran atenuación, mientras que los eventos de bajas frecuencias, tales como los potenciales postsinápticos, se atenúan menos con la distancia. Consecuentemente, el reflejo extracelular de potenciales de acción solamente es visible para electrodos inmediatamente adyacentes a la célula registrada, mientras que los eventos sinápticos pueden propagarse largas distancias en el espacio extracelular y ser registrados tan lejos como en la superficie del cuero cabelludo, donde participan de la génesis del EEG [29].

Esto contribuye a que los registros de EEG tengan una contribución de los potenciales de acción mucho más acotada que la que puede obtenerse en los microEEG.

4.3 REGISTROS DE MICROEEG

Los registros de microelectrodos reflejan actividad extracelular que puede ser dividida en tres categorías [30]:

- LFP (*Local Field Potential* por sus siglas en inglés) que involucra frecuencias hasta 300Hz (algunos autores proponen hasta 500 Hz)

- MUA (*Multi-Unit Activity* por sus siglas en inglés) que involucra señales con componentes frecuenciales superiores a 300 Hz
- SUA o actividad de única neurona (*Single-Unit Activity* por sus siglas en inglés)

Como ya se ha expresado la señal registrada por el electrodo en el medio extracelular es la suma de una variedad de señales generadas por muchos procesos y elementos neurales.

Los cambios de baja frecuencia (típicamente $<300\text{Hz}$) en la señal extra celular registrada por el microelectrodo es lo que se conoce como LFP. Esta señal se deriva de los procesos más lentos y es menos atenuada en grandes distancias porque es de baja frecuencia, como se explicó en la sección 4.2.3. Así, puede reflejar procesos remotos en el rango de distancias entre 0.5 y 3.0 mm desde el extremo del microelectrodo [8].

Por su parte, la actividad de potenciales de acción se refleja principalmente en los cambios de alta frecuencia de la señal (típicamente $>300\text{ Hz}$), por lo que puede ser extraída desde el registro con un filtro pasa altos. Esa señal llamada MUA es la suma de potenciales de acción de varias neuronas que están muy cerca del electrodo de registro. Como se explicó en la sección 4.2.3, las altas frecuencias de la actividad de los potenciales de acción decaen rápidamente con la distancia; la distancia a la cual la actividad de potenciales de acción puede ser distinguida de la actividad de fondo (que es la suma de señales producidas por neuronas más distantes) depende del tamaño y la forma de las neuronas y de las propiedades del electrodo de registro, pero es generalmente del orden de $100\text{-}300\ \mu\text{m}$ [8].

A su vez, la señal de MUA puede descomponerse en otras señales. Puntualmente, la señal de MUA puede ser dividida en trenes de potenciales de acción y actividad de fondo, BUA (*Background Unit Activity* por sus siglas en inglés). La BUA representa potenciales de acción más pequeños, con niveles del orden del ruido, que son generadas por poblaciones neuronales de los alrededores [8].

Los trenes de potenciales de acción que pueden obtenerse a partir de la señal de MUA pueden ser de dos tipos: los correspondientes a la actividad de neuronas individuales, donde cada señal representa un tren de espigas de una única neurona (SU-ST, *single-unit spike-train* por sus siglas en inglés) o puede extraerse un tren de espigas de varias neuronas no separables (MU-ST, *multi-unit spike-train* por sus siglas en inglés). La diferencia entre la señal MUA y la MU-ST es que, mientras que la señal de MUA es una señal filtrada y muestreada que representa

la actividad promedio registrada extracelularmente de neuronas vecinas, la MU-ST es un conjunto de puntos que representan momentos de disparo de varias neuronas [8].

En la figura 4.2 pueden apreciarse las señales que se han mencionado en esta sección y que pueden obtenerse a partir de un único registro extracelular.

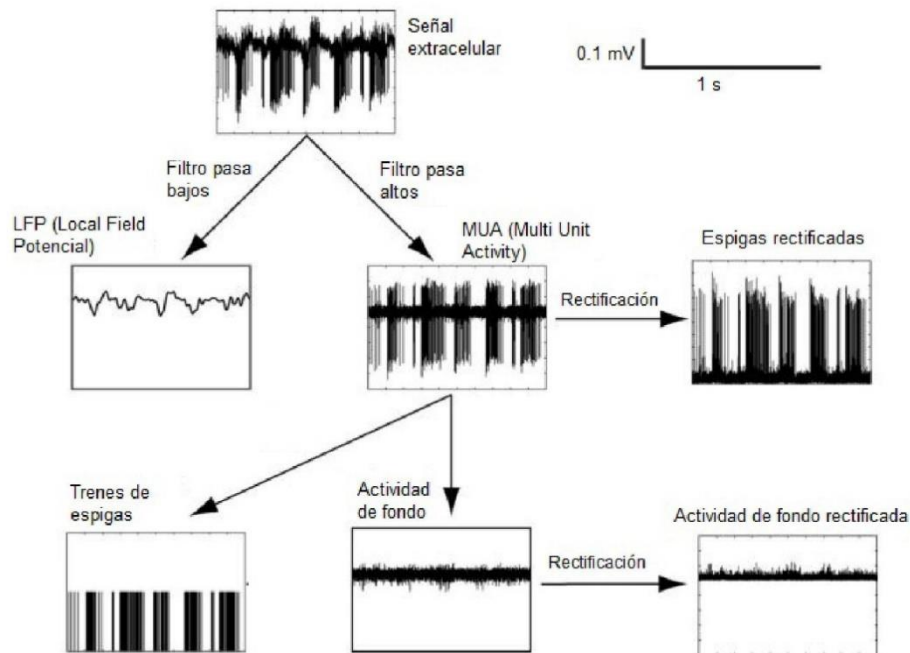


Figura 4.2: Señales derivadas de un único registro extracelular. Adaptado de [8].

4.4 SEÑALES DE MICROEEG EN PACIENTES CON ENFERMEDAD DE PARKINSON

Como se explicó en el capítulo 3, durante la cirugía de implantación de electrodos para DBS se registran señales con microelectrodos de registro (MER) para definir la localización del núcleo elegido como objetivo (*target* por su denominación en inglés). También se ha mencionado que en los tratamientos de pacientes con EP se suele elegir como sitio de implantación al STN.

La trayectoria que sigue cada MER es planificada previo a la cirugía para cada paciente para minimizar riesgos de lesiones vasculares y neuronales. Normalmente el registro comienza unos 10 mm antes del *target* y a medida que los MER avanzan hacia la zona de implantación van recorriendo distintas estructuras funcionales como TH, ZI, STN y SNr como puede observarse en la figura 3.5.

Las señales obtenidas por los MER en un núcleo basal del cerebro presentan características temporales (ver figura 4.3) y frecuenciales particulares que permiten distinguirlo de las otras estructuras que la rodean. A continuación se detallan las propiedades de disparo de cada uno de los núcleos de los GB [5]:

Estriado: sus neuronas disparan a frecuencias entre 2 y 10 Hz.

GPe: está compuesto por dos tipos de neuronas: HFD-P (descargas de alta frecuencia con pausas) que disparan a frecuencias entre 10 y 100 Hz y componen el 85 % del GPe; y LFD-B (descargas de baja frecuencia con ráfagas) que disparan a frecuencias entre 1 y 30 Hz y conforman el 15 % del GPe;

GPI: neuronas HFD (descargas de alta frecuencia), que disparan a frecuencias entre 10 y 110 Hz sin pausas.

Existen **neuronas "de borde"** que disparan regularmente a 20-50Hz y se encuentran en el borde entre el putamen y el GPe y entre el GPe y el GPI.

STN: las neuronas glutamatérgicas excitatorias que componen al STN disparan irregularmente y a intervalos medios de frecuencia, entre 25 y 45 Hz.

SNr: se compone principalmente de neuronas GABA enérgicas inhibitorias que disparan espontáneamente, a intervalos irregulares de alta frecuencia, como las neuronas HFD del GPI.

SNe: se compone de neuronas dopaminérgicas que se caracterizan por espigas amplias y bajas tasas de disparo (menores a 6 Hz).

Cuando se registran estas señales con los MER además se presentan regiones de materia blanca por ser zonas de menor actividad. En particular los registros del STN se distinguen de la SNr porque en el primero la actividad es irregular, en tanto que en la SNr aparece una actividad regular de alta frecuencia.

Dado que el STN es el núcleo basal más utilizado como *target* se procederá a analizar las características temporales y frecuenciales de las señales obtenidas con los MER en dicho núcleo basal.

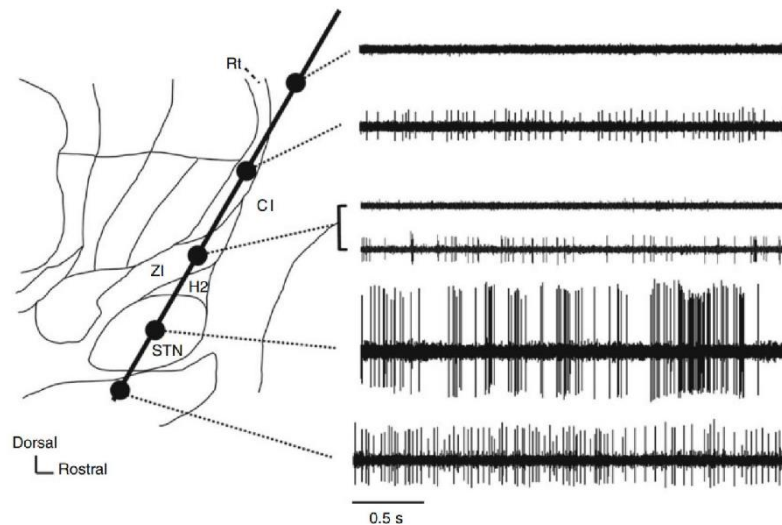


Figura 4.3: Esquema de algunos núcleos basales del cerebro y sus registros eléctricos característicos: núcleo subtalámico (STN), zona incerta (ZI), campos de Forel (H2), *Claustrum* (CI), núcleo reticular (Rt). Adaptado de [5].

4.4.1 CARACTERÍSTICAS TEMPORALES DE LOS REGISTROS MER DEL STN

En la verificación intraoperatoria de la localización del STN, el equipo médico analiza las características temporales de las señales captadas por los MER.

Como se ha explicado en la sección 4.3, la señal de MUA se compone por un agregado de actividad de potenciales de acción de la vecindad del electrodo de registro; esta señal de fondo es de mayor amplitud en el STN que en estructuras vecinas [30].

Las características o parámetros temporales más reportados por diversos autores [4], [31]–[33] de los registros temporales adquiridos con los MER se clasifican en aquellas independientes de la detección de las espigas neuronales (*spikes* por su denominación en inglés) y que reflejan la actividad de fondo de los registros y las características que son dependientes de la detección de espigas y reflejan principalmente la presencia de las mismas. El análisis conjunto cuantitativo de las características más relevantes permite a un neurofisiólogo entrenado determinar cuando los MER ingresan o egresan del STN

4.4.1.1 Características temporales independientes de la detección de spikes

La media cuadrática normalizada o valor cuadrático medio normalizado (NRMS o RA): Chaovalitwongse y col. [4] dividieron los registros MER en segmentos de 4 s con solapamiento

del 50% y obtuvieron el valor cuadrático medio (RMS) utilizando la ecuación 4.1, donde X es el vector discreto de la señal temporal de n componentes y X_i cada elemento del vector.

$$RMS = \sqrt{\frac{\sum_{i=1}^n (X_i)^2}{n}} \quad (4.1)$$

La detección de los bordes del STN usando el RMS del registro de MUA puede ser buena, existiendo estudios en los que las guías de DBS fueron implantadas exitosamente usando este criterio [30]. En general, la medida del RMS requiere ser normalizada, dado que sus valores son susceptibles a las propiedades de los electrodos de registro tales como su impedancia. En el estudio realizado por Zaidel y col. [34] se propone normalizar al RMS dividiéndolo por el RMS de la sustancia blanca que precede al STN en la trayectoria del electrodo. De esa manera, los autores crean lo que denominan RMS normalizado (NRMS). En ese estudio, los autores detectan la entrada y salida del electrodo en el STN por un incremento o descenso dramático del NRMS, respectivamente [34]. En la parte A de la figura 4.4, se ve el NRMS de una trayectoria de un electrodo como función de la distancia estimada hasta el target (EDT), que en el referido estudio era el centro de la región dorsolateral del STN. En la parte B de la figura 4.4 puede observarse también un incremento de la banda beta en las señales del STN tal como se desarrollará en la sección 4.4.2.

Valor de la amplitud basal (Basal): Dolan y col. [31] propusieron un método robusto utilizando la transformada de Hilbert para estimar la envolvente de un registro temporal MER mediante la ecuación 4.2, donde $E(t)$ es la envolvente, $HT\{X(t)\}$ es la transformada de Hilbert de la señal temporal $X(t)$.

$$E(t) = \sqrt{X(t)^2 + (HT\{X(t)\})^2} \quad (4.2)$$

Curtosis (K): Guerrero y col. [33] propusieron la curtosis de segmentos de 1 s de la señal MER solapados al 50% como una característica temporal. Estadísticamente representa la proporción de la varianza que se explica por la combinación de datos extremos respecto a la media de la señal en contraposición con datos poco alejados de la misma. Cuando no se conocen las distribuciones de las variables pero se dispone de datos discretos K puede estimarse según la ecuación 4.3, donde μ_4 es el cuarto momento respecto a la media y σ la desviación estándar.

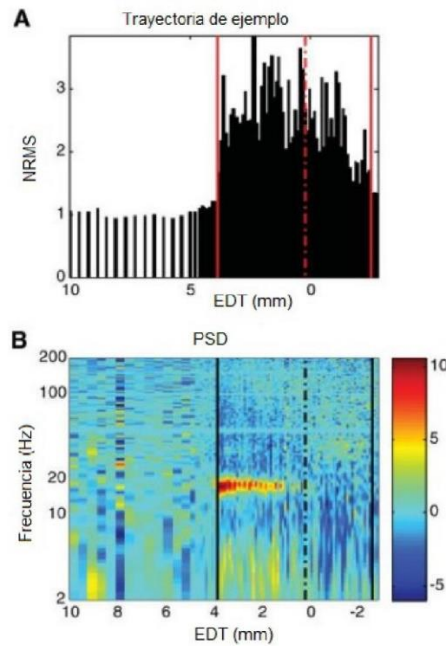


Figura 4.4: A) Incremento del NRMS cuando los electrodos MER ingresan al STN. B) Incremento de la banda beta de frecuencia para la señales del STN. Adaptado de [34].

$$K = \frac{\mu_4}{\sigma^2} \quad (4.3)$$

Los valores de K obtenidos se comparan con el valor de la curtosis de la distribución de Normal, que da un valor de 3. Un valor mayor implica, generalmente, la presencia de picos o artefactos concentrados en una zona de la señal.

Longitud de la curva (CL): Wongs y col. [32] dividieron los registros MER en segmentos de 4 s con solapamiento del 50% y obtuvieron la suma del valor absoluto de la señal derivada según la ecuación 4.4, donde X_i cada elemento del vector de longitud n . Valores elevados de CL representan variaciones rápidas de la señal.

$$CL = \sum_{i=1}^{n-1} |X_{i+1} - X_i| \quad (4.4)$$

Umbral (TH): Chaovalitwongse y col. [4] dividieron los registros MER en segmentos de 4 s con solapamiento del 50% y obtuvieron el umbral según la ecuación 4.5, donde X_i cada elemento del vector de longitud n y media μ .

$$TH = \frac{3}{n-1} \sqrt{\sum_{i=1}^n (X_i - \mu)^2} \quad (4.5)$$

Cantidad de Picos (PK): Chaovalitwongse y col. [4] estimaron la cantidad de picos presentes en segmentos de 4 s de la señal MER solapados al 50% según la ecuación 4.6, donde X_i cada elemento del vector de longitud n .

$$PK = \frac{1}{2} \sum_{i=1}^{n-2} \max\{0, \text{sgn}[X_{i+2} - X_{i+1}] - \text{sgn}[X_{i+1} - X_i]\} \quad (4.6)$$

Energía no lineal promedio (NE): Wongs y col. [32] dividieron los registros MER en segmentos de 4 s con solapamiento del 50% y obtuvieron la energía de la señal según la ecuación 4.7, donde X_i cada elemento del vector de longitud n . Valores elevados de NE manifiestan la presencia en los segmentos estudiados de señales de alta frecuencia.

$$NE = \frac{1}{n-2} \sum_{i=2}^{n-1} X_i^2 - X_{i-1}X_{i+1} \quad (4.7)$$

Cruces por cero (ZC): Chaovalitwongse y col. [4] dividieron los registros MER en segmentos de 4 s con solapamiento del 50% y obtuvieron la cantidad de cruces por cero según la ecuación 4.8, donde X_i cada elemento del vector de longitud n .

$$ZC = \frac{1}{2} \sum_{i=1}^{n-1} |\text{sgn}(X_{i+1}) - \text{sgn}(X_i)| \quad (4.8)$$

4.4.1.2 Características temporales dependientes de la detección de spikes

Son características relacionadas directamente a las espigas presentes en los registros temporales adquiridos con los MER. Se requiere para el cálculo de las mismas primero detectar las espigas en segmentos del registro MER, en la presente tesis se realizó según lo propuesto por [4]. En el estado del arte [4], [31]–[33] se proponen las siguientes características las cuales por su propia definición pueden calcularse y por lo tanto no se detalla una fórmula de cálculo:

Índice de ráfaga de espigas (SBI): Mide la relación de la cantidad intervalos de tiempo entre espigas menores a 10 ms versus la cantidad de intervalos mayores a 10 ms.

Índice de pausa de espigas (SPI): Mide la relación de la cantidad intervalos de tiempo entre espigas mayores a 50 ms versus la cantidad de intervalos menores a 50 ms.

Ratio de pausa (SPR): relación entre el tiempo acumulado entre espigas en un tiempo mayor y menor a 50 ms.

Frecuencia de disparos (SC) de las espigas calculadas en espigas por segundo.

Amplitud diferencial media de espigas (SMAD): calculado como el valor medio de la amplitud de 2 espigas consecutivas.

Desviación estándar de los intervalos entre espigas (SSD).

Ratio del conteo de espigas (SCR)

Mediana de la frecuencia de disparo de las espigas en Hz (SF o SSF).

4.4.2 CARACTERÍSTICAS FRECUENCIALES DE LOS REGISTROS MER DEL STN

En la comunidad científica, es bien aceptado que los registros de LFP del STN en humanos con EP revelan una potencia elevada en la banda beta que se corresponde al rango de frecuencias entre 13 y 35 Hz cuando el paciente está en reposo y sin medicación. Varias investigaciones han confirmado esta relación tal como lo revela la recopilación realizada en 2012 por Little y Brown [35]. Bronte-Stewart y col. [36] registraron LFP de STN de pacientes con EP y demostraron que la actividad oscilatoria en la banda beta de frecuencias es un fenómeno consistente a lo largo de varios minutos de registro con el paciente en reposo.

El STN posee tres porciones somatotópicas o territorios funcionales: sensorimotor, límbico y cognitivo/asociativo. Cada uno involucrado con su respectivo circuito tálamo-corteza-GE. El dominio sensorimotor del STN se localiza en la porción dorsolateral.

Particularmente, es en la porción dorsolateral del STN donde se suele presentar mayor energía en la banda beta [34], [37]. A la porción dorsolateral que presenta oscilaciones en la banda beta se la llama DLOR (*Dorsolateral Oscillatory Region* por sus siglas en inglés).

Por otra parte, según estudios de inactivación de primates tratados con MPTP, la región dorso lateral del STN es el target óptimo de las cirugías de implantación de electrodos para DBS [38]. Esta región coincide con la ubicación que parece otorgar beneficios terapéuticos óptimos para los pacientes con EP que se someten a DBS en el STN [34].

En [37] se buscó localizar al STN de acuerdo a la energía en diferentes bandas de frecuencia, reportando que ni la banda 4-10 Hz ni la banda 65-85 Hz ayudaron a localizar al STN.

En el estudio realizado por Zaidel y col. [34] se utilizó la densidad espectral de potencia en la banda de frecuencias beta como un marcador para la DLOR del STN, basándose en que la actividad oscilatoria en beta podría predecir el sitio de estimulación más efectivo para la terapia DBS en este núcleo. En la parte B de la figura 4.4, se ve la densidad espectral de potencia de una trayectoria de un electrodo como función de la distancia estimada hasta el EDT [34].

También se ha demostrado que el tratamiento con DBS suprime la actividad en beta y que las mejoras en rigidez y bradiquinesia (pero no en temblor) mediadas por DBS en el STN se correlacionan con la supresión de la sincronización [35]. En la figura 4.5 puede verse el efecto de la estimulación eléctrica sobre la densidad espectral de potencia de las señales registradas en el STN.

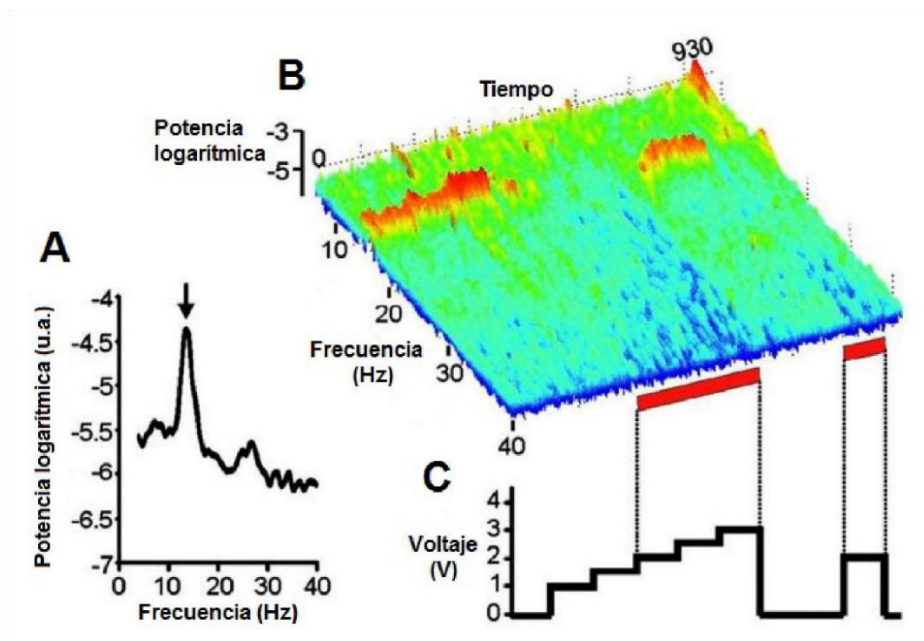


Figura 4.6: Señales derivadas de un único registro extracelular. Adaptado de [35].

Mientras que la Levodopa reduce casi completamente las oscilaciones en beta, la DBS solamente lo hace parcialmente. Cuando se combinan ambas terapias, se reduce en mayor medida la potencia en la banda beta [39].

Más allá de la discusión respecto de si la potencia elevada en la banda beta es fisiológica, en parte fisiológica o completamente patológica, existen estudios que muestran que implantar el electrodo de DBS en la región del STN con mayor potencia en esa banda de frecuencias tiene resultados óptimos en la terapia.

El estudio realizado por Zaidel y col. [38] sugieren localizar el electrodo de estimulación DBS en la región dorsolateral del STN para obtener mejores resultados clínicos del tratamiento.

Las características frecuenciales mas reportadas por distintos trabajos científicos [34]–[36], [38], son:

- Energía normalizada en banda beta (13-35Hz), respecto de la energía en banda de interés (1-45Hz).
- Energía media en banda beta (13-35Hz), respecto de energía en banda de interés ampliada (2-200Hz) [6].
- Energía máxima en banda beta (13-35Hz), respecto de energía en banda de interés ampliada (2-200Hz) [6].
- Frecuencia dominante en banda beta (13-35Hz).

Capítulo 5 Reconocimiento de patrones y minería de datos

5.1 RECONOCIMIENTO DE PATRONES

El reconocimiento de patrones representa un proceso organizado para identificar patrones válidos, novedosos, útiles y comprensibles a partir de una gran cantidad de datos frecuentemente complejos. La minería de datos (DM) es el núcleo del proceso del reconocimiento de patrones, que implica utilizar algoritmos que exploran los datos, desarrollan el modelo y descubren patrones previamente desconocidos. El modelo se utiliza para comprender fenómenos a partir de los datos, el análisis y la predicción [40].

En general, como se observa en la figura 5.1, el reconocimiento de patrones presenta las siguientes fases:

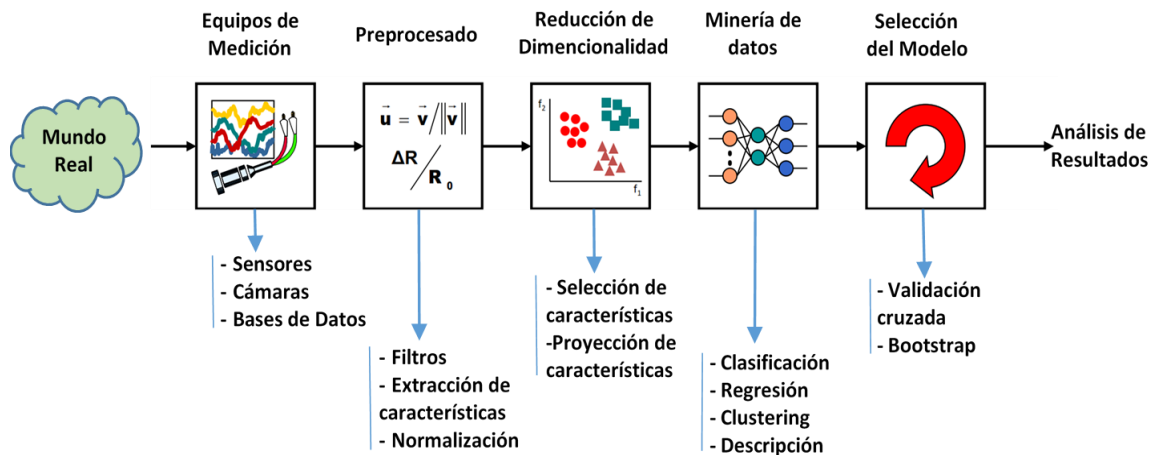


Figura 5.1: Fases y componentes de un sistema de reconocimiento de patrones. Adaptado de [40]

La primera de ellas es la **fase de obtención de los datos** mediante sensores, cámaras o bases de datos, entre otros dispositivos o medios, se obtienen datos del mundo real. La identificación de los datos relevantes es una tarea que no puede ser automatizada y que debe ser realizada por un analista [12]. Consiste en crear un conjunto de datos objetivo, seleccionando un conjunto de variables o muestras relevantes. Implica considerar la homogeneidad, la variación temporal y la disponibilidad de los mismos.

La segunda **fase es de pre procesado** e involucra operaciones de filtrado, transformación y reducción de datos con el fin de favorecer el éxito de las etapas posteriores [12]. En esta fase se eliminan aquellos datos innecesarios para el proceso de extracción de conocimiento como

así también pueden crearse datos derivados de la combinación o transformación de los datos iniciales. En algunas ocasiones es necesario realizar transformaciones en la distribución de los datos, como puede ser la normalización.

Básicamente la fase de preprocesado tiene como fin [40]:

- Detectar valores anómalos como los *outliers*.
- Detección de valores faltantes y de registros de baja calidad y su eliminación o reemplazo.
- Eliminar el ruido.
- Preparar los datos para optimizar el proceso de extracción de conocimiento de fases posteriores

Aún después de haber sido pre procesados los datos, en la mayoría de los casos se tiene una cantidad muy grande de información que no siempre son necesarios para encontrar un buen modelo, por lo que la siguiente **fase es la de reducción de la dimensionalidad**. En la misma se aborda la selección de características para reducir el tamaño de los datos eligiendo las variables más influyentes en el problema [12]. Esta fase de acuerdo a la problemática que se quiere abordar y/o los algoritmos utilizados puede estar integrada a la fase siguiente de minería de datos o en algunas ocasiones realizarse después de esa fase [12].

Los métodos para la selección de características son básicamente dos: a) aquellos basados en la elección de los mejores atributos del problema, b) aquellos que buscan variables independientes mediante tests de sensibilidad, algoritmos de distancia o heurísticos [41].

Una vez obtenidos los datos significativos, se debe decidir qué tipo de patrón o modelo se quiere descubrir y para eso se trabaja en la **fase de minería de datos** la que se desarrollará en la siguiente sección. El tipo de conocimiento que se desea extraer determinará la técnica de minería de datos a utilizar [40]. También, en muchas ocasiones, la elección de la técnica depende de los tipos de datos que se dispone y si los mismos contienen suficientes ejemplos de cada clase como para obtener un modelo fiable. Para llevar adelante esta fase es necesario disponer de datos que ya han sido asignados a una clase y que servirán para el algoritmo de aprendizaje o datos sin clase asignada en donde el resultado del algoritmo es la asignación de clase [41]. No todas las técnicas trabajan bien con datos faltantes, con datos escasos o con desbalances importantes entre las clases. De acuerdo al objetivo del proceso de reconocimiento

de patrones se utilizarán técnicas de minería de datos de clasificación, regresión, *clustering* o agrupamiento de conceptos o descripción [40]. La taxonomía de la minería de datos se abordará en detalle en la sección 5.2.

La fase anterior produce una o más hipótesis de modelos. Para seleccionar y validar estos modelos es necesario el uso de criterios de evaluación de hipótesis y esto se corresponde con la **fase de selección del modelo**. En esta fase se realiza la comprobación de la exactitud del modelo utilizando un conjunto de ejemplos independiente de los utilizados para entrenar el mismo. Se pueden utilizar estrategia de validación cruzada de *n*-pliegues, *bootstrapping*, *leave one out* u otras que resulten apropiadas según el caso de estudio [42].

La medida tradicional para evaluar un modelo, por ejemplo en el caso de un clasificador, es el error que representa el porcentaje de instancias mal clasificadas respecto al conjunto de test. Existen otras valoraciones para la selección de un modelo desde el punto de vista de quien evalúa los mismos, alguna de ellas son [40], [42]:

A) **Comprensibilidad**: se trata de una medida completamente subjetiva y depende del evaluador, pero en el caso de reglas de asociación se pueden seguir algunas medidas simples, tales como no contemplar reglas con muchos antecedentes (o consecuentes). Desde el punto de vista del evaluador la obtención de reglas que expliquen al modelo y resulten de fácil comprensión puede significar una opción viable de selección.

B) **Interés**: existen dos tipos de medidas de interés, las objetivas y subjetivas. Las subjetivas se basan en la novedad que aportan los resultados y su relevancia para el sujeto evaluador. Las objetivas son medidas que se obtienen de los propios resultados y tratan de hallar aquellos que presentan una tendencia estadísticamente significativa.

C) **Aplicabilidad**: un modelo debe ser aplicable a la predicción o la descripción de los datos para que su evaluación subjetiva sea positiva. Por ejemplo, es poco aplicable una regla cuyo antecedente, en la práctica, no se conozca o no sea utilizado en el campo de aplicación. Dentro de esta evaluación se incluye la realización de experiencias pilotos con los modelos obtenidos utilizándolos en el mundo real de donde se obtuvieron los datos iniciales.

5.2 FASE DE MINERÍA DE DATOS

La minería de datos (o *data mining* por su denominación en inglés) se define como una disciplina que basada en el aprendizaje computacional inductivo estudia métodos y algoritmos que permiten la extracción automática de información para caracterizar o descubrir relaciones, tendencias, desviaciones comportamientos atípicos, patrones y trayectorias ocultas en una gran cantidad de datos [40]. Se pretende también que la información obtenida posea capacidad predictiva, facilitando el análisis de los datos de forma eficiente. La minería de datos utiliza diversas técnicas estadísticas, de inteligencia artificial y del aprendizaje automático (*machine learning* por su denominación en inglés) enfocadas, principalmente, a la visualización, análisis, y modelización de información de bases de datos con gran tamaño [43].

5.2.1 TAXONOMÍA DE LA MINERÍA DE DATOS

Las tareas que aborda la minería de datos (MD) son fundamentalmente de dos tipos:

- Predictivas: donde se pretende predecir el valor que tendrá en un futuro uno o más valores en función de los datos que se disponen hasta el momento.
- Descriptivas: donde se realiza una labor de análisis acerca de los datos que se disponen, intentando describirlos para obtener información de ellos que mediante técnicas estadísticas normales sería dificultoso.

La clasificación de la MD difiere entre los autores, para la presente tesis se ha adoptado la de Maimon y col. [40] que describe las siguientes funciones:

- *Clustering*: donde se analizan los datos y se generan conjuntos de reglas que agrupan y clasifican los datos futuros.
- Reglas de asociación: reglas que presentan ciertas relaciones entre un grupo de objetos de una base de datos.
- Análisis de secuencias: trata de encontrar patrones que ocurren con una secuencia determinada.
- Reconocimiento de patrones: analiza la asociación de una señal de información de entrada con aquella o aquellas con las que guarda mayor similitud, de entre las catalogadas por el

sistema. Se usan para identificar causas de problemas o incidencias y buscar posibles soluciones, siempre y cuando se disponga de la base de información necesaria.

- **Predicción:** se busca establecer el comportamiento futuro más probable de una variable o un conjunto de variables a partir de la evolución pasada y presente de las mismas o de otras de las que dependen.

- **Simulación:** comparan la situación actual de una variable y su posible evolución futura.

- **Optimización:** resuelven problemas de la minimización o maximización de una función que depende de una serie de variables.

- **Clasificación:** permiten asignar a un elemento la pertenencia a un determinado grupo o clase. Se define un perfil característico de cada clase y su expresión en términos de un algoritmo o reglas como función de distintas variables. También se establece el grado de discriminación o influencia de estas últimas. Con ello es posible clasificar un nuevo elemento una vez conocidos los valores de las variables presentes en él.

5.2.2 TÉCNICAS DE APRENDIZAJE EN LA MINERÍA DE DATOS

Se pueden clasificar las técnicas de aprendizaje en dos grande grupos: técnicas inductivas y abductivas [40], [42], [44]. Dentro del primer grupo (inductivas) existe una sub clasificación detallada a continuación:

Técnicas predictivas:

- **Interpolación:** genera una función continua sobre varias dimensiones
- **Predicción secuencial:** las observaciones están ordenadas secuencialmente y se predice el siguiente valor de la secuencia.
- **Aprendizaje supervisado:** en éstas técnicas cada observación, compuesta por muchos valores de atributos, incluye un valor de la clase a la que corresponde. Se genera o aprende un clasificador a partir de clases que se proporcionan. Es un caso particular de interpolación en el que la función genera un valor discreto en lugar de continuo. Dentro de las técnicas de aprendizaje supervisados encontramos aquellas utilizadas para clasificación y otras destinadas a la regresión.

Técnicas descriptivas:

- Aprendizaje no supervisado: en estas técnicas el conjunto de observaciones no tienen clases asociadas. El objetivo es detectar regularidades en los datos de cualquier tipo: agrupaciones de datos parecidos o próximos, contornos de delimitación de grupos, asociaciones o valores anómalos. El *clustering* es la técnica de aprendizaje no supervisado más común.
- Análisis exploratorios: tales como estudios correlacionales, asociaciones, estudios de dispersión y valores anómalos.

Dentro de las técnicas abductivas, cuyo fin es principalmente exploratorio, existe la siguiente sub clasificación:

Técnicas de Aprendizaje Analítico:

- En estas técnicas se pretende, partiendo de los valores generados y de las reglas, obtener los datos de origen. El objetivo es explicar la evidencia respecto a los hechos que se han producido.

5.2.3 ALGORITMOS PARA APRENDIZAJE SUPERVISADO Y NO SUPERVISADO

Se presentan a continuación los algoritmos que implementan las técnicas de aprendizaje supervisado y no supervisado dado que son las de más amplia utilización dentro de la MD para la solución de problemas de la vida real [40], [42], [44] . Existen numerosos algoritmos para implementar ambas técnicas, siendo a su vez una rama del conocimiento que está en constante investigación y desarrollo. Los algoritmos nombrados en la presente sección son aquellos más ampliamente utilizados y que representan sólo una muestra de la gran diversidad que existe.

Como se abordó en la sección anterior el **aprendizaje no supervisado** halla patrones ocultos o estructuras intrínsecas en los datos. Se emplea para inferir información a partir de conjuntos de datos de entrada que no presentan sus correspondientes respuestas, clase o etiquetas. En el aprendizaje no supervisado el algoritmo debe agrupar los conceptos, ya que dispone de ejemplos sin clasificar, mediante la observación de las características de los datos de entrada y en base a criterios propios del algoritmo.

Algunos algoritmos habituales para realizar el *clustering* son: *k-means* y *k-medoids*, *clustering* jerárquico, modelos de mezclas gaussianas, modelos ocultos de Markov, mapas auto organizados, *clustering* difuso *c-means*, *clustering* sustractivo, redes neuronales de Kohonen, *Estimated Means* (EM), *Cobweb*, *Autoclass*, etc [40], [42], [44].

El **aprendizaje supervisado** supone la participación de un experto que define clases y provee ejemplos de cada una de esas clases. Un algoritmo de aprendizaje supervisado toma un conjunto de datos de entrada y respuestas conocidas para estos datos (salidas o clases) y entrena un modelo con el objeto de generar predicciones razonables como respuesta a datos nuevos.

El aprendizaje supervisado emplea técnicas de regresión y clasificación para desarrollar modelos predictivos. Las técnicas de regresión predicen respuestas cuando los datos de entrada son continuos. Algunos algoritmos habituales de regresión son [40], [42], [44]: modelo lineal, modelo no lineal, regularización, regresión por pasos, árboles de decisión, redes neuronales, lógica difusa y aprendizaje neurodifuso adaptativo.

Las técnicas de clasificación predicen respuestas discretas clasificando los datos de entrada en categorías. Algunos algoritmos habituales para realizar la clasificación son [40], [42], [44]: máquina de soporte vectorial, árboles de decisión, k-vecino más cercanos (KNN), clasificadores bayesianos (Naive Bayes), análisis discriminante, lógica difusa, regresión logística y redes neuronales.

5.3 ESTADO DEL ARTE DE CLASIFICADORES EN DETECCIÓN DE STN EN BASE A REGISTROS MER

Dado que los registros MER son analizados por neurofisiólogos durante una cirugía para implantar un DBS, naturalmente resultan ser datos clasificados o etiquetados. Los neurofisiólogos determinan en base a su experticia si dichos registros pertenecen por sus características temporales (amplitud, cantidad de *spikes*, ruido de fondo, etc.) a algún ganglio basal o zona del cerebro.

En los últimos años han surgido investigaciones para utilizar el aprendizaje supervisado mediante clasificadores, con el fin de obtener modelos que en base al aprendizaje de las señales MER de pacientes anteriores pueda clasificar los registros de un nuevo paciente. Si se tienen en cuenta que en tiempo medio de una cirugía para implantar en los dos hemisferios cerebrales un

DBS demandan de 6 a 8 horas es lógico pensar que el cansancio humano puede ser un factor que introduzca errores. Por lo tanto disponer de una herramienta de soporte que realice la clasificación automática de las señales MER puede ser de utilidad a los neurofisiólogos intervinientes durante una cirugía.

Art Chaovalitwongse y col. [4] trabajaron con 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) y señales MER provenientes de 17 pacientes. Reportaron resultados de clasificadores supervisados con algoritmos de K-vecinos más cercanos (KNN), árboles de decisión binarios y Naive Bayes. Los mejores resultados fueron obtenidos con los árboles de decisión que utilizaron como criterio de división de ganancia de información logrando una exactitud promedio del 89,61%. También utilizaron para entrenar KNN y árboles de decisión un indicador frecuencial-temporal, pero los resultados obtenidos no fueron alentadores (máxima exactitud promedio de 50,2%). Para estimar los indicadores de desempeño utilizaron validación cruzada dejando siempre un paciente fuera de los datos de entrenamiento y utilizando ese paciente para validación (denominada *leave one patient out*). En el trabajo no se realiza un estudio estadístico comparativo de los desempeños de los clasificadores.

En Cagnan y col. [9] proponen un método de clasificación propio con una estructura similar a un árbol de decisión binario, trabajan con 2 características temporales y 2 frecuenciales vinculadas a la potencia en el rango de frecuencias beta y gamma, con el objetivo de detectar la entrada y salida del electrodo en STN. Logran un porcentaje de acuerdo promedio con las clasificaciones que realizan los neurofisiólogos del 87-88%. Realizaron un estudio estadístico comparativo.

Francés Villora y col. [45] trabajaron sólo con características frecuenciales de la señal MER de 15 pacientes con clasificadores de tipo Bayesianos (cuadrático y lineal), KNN, SVM (con *kernel* exponencial y basado en distancias) y de Parzen. En general no obtuvieron resultados alentadores, siendo el mínimo error de clasificación encontrado del 25,7%.

Rajpurohit y col. [10] reportaron resultados de clasificadores supervisados con algoritmos regresión logística, KNN, Naive Bayes y máquinas de soporte vectorial (SVM). Utilizaron 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) con datos provenientes de 26 pacientes. Los mejores resultados se obtuvieron con KNN. Cuando entrenaron al clasificador con las 13 características

estandarizadas con los valores de todos los pacientes KNN presentó un error promedio de 20%. El trabajo también reportó la selección de características por 2 métodos de envoltura: *backward* seleccionando 7 características y *forward* seleccionando 6 características en ambos casos 5 de ellas vinculadas a la actividad de fondo. El error promedio de clasificación en ambos casos disminuyó al 10%. El trabajo realizó un estudio estadístico comparativo y utilizó la validación de los clasificadores tipo *leave one patient out*.

Todas las publicaciones científicas analizadas al presente, vinculadas a la clasificación de señales MER para detección de STN, reportan algoritmos que realizan el proceso de clasificación en forma individual (KNN o árboles de decisión o Bayes, etc) y no se ha indagado la combinación de dichos algoritmos para lograr un modelo más robusto y con mejor desempeño.

Capítulo 6 Materiales y metodología general

En el presente capítulo se presentan los materiales necesarios para el desarrollo de la tesis así como la descripción de la metodología general, la cual se desarrolla en los capítulos 7, 8, 9 y 10.

6.1 MATERIALES

6.1.1 REGISTRO DE DATOS

Se adquirieron señales MER de 22 pacientes con Parkinson de 57 ± 6 años (12 hombres/10 mujeres) a los cuales se les realizó una cirugía bilateral para la implantación de un dispositivo DBS en STN (88 *tracks* o trayectorias totales). Todos los pacientes fueron seleccionados mediante criterio médico y firmaron un consentimiento informado. Las cirugías fueron desarrolladas en su totalidad en el Hospital Universitario y Politécnico La Fe de Valencia, España.

Según el protocolo clínico estándar, las coordenadas del STN y la trayectoria planificada de inserción de los electrodos de registro se identificaron mediante resonancia magnética (MRI) estereotáxica preoperatoria, que se fusionó con una tomografía computada (TC) estereotáxica (co-registro de neuroimágenes) realizada en el quirófano el día de la cirugía una vez colocado el marco estereotáxico o estereotáctico. La fusión de imágenes se realizó en una plataforma de neuro navegación (StealthStation, Medtronic Corp, Minneapolis, MN, USA) con el fin de ajustar las coordenadas del STN. Los pacientes habían interrumpido la medicación para el Parkinson previo a la cirugía siguiendo el protocolo clínico.

Se siguieron técnicas estereotácticas estándares para la inserción de dos cánulas cerebrales y los microelectrodos MER mediante los cuales se obtuvieron los registros eléctricos neuronales monopares individuales utilizando el sistema "MicroGuide" (AlphaOmega Engineering, Nazareth, Israel). Los electrodos MER de tungsteno recubiertos de poliamida (Alpha Omega) presentan una impedancia típica de $1M\Omega$ a 1000 Hz. La impedancia fue monitorizada por el equipo MicroGuide durante todas las cirugías con el fin de asegurar la adecuada adquisición. Por cada hemisferio cerebral se insertaron 2 MER paralelos separados por una distancia de 2 mm. La señal se amplificó 10000 veces y se aplicó un filtro pasabanda entre 200 y 6000 Hz, de

tipo *Butterworth* de 4° orden para la frecuencia de corte bajo y 2° orden para la frecuencia de corte alto. La frecuencia de muestreo fue de 12 kHz y se utilizó un convertidor A/D de 12 bits. Durante una cirugía de DBS, los electrodos de registro descendieron en la trayectoria planificada hasta localizarse 7 mm antes de STN, de acuerdo con la posición de STN establecida por imágenes (corregistros). A partir de esta posición, se comenzó la adquisición de la señal eléctrica mediante los MER en pasos de 0,2 mm hasta llegar a la sustancia negra luego de atravesar el STN. En cada paso, se registraron como mínimo 30 s de actividad eléctrica cerebral.

Dos especialistas neurofisiólogas experimentadas, pertenecientes al equipo de neurocirugía del Hospital Universitario y Politécnico La Fe, analizaron durante la intervención quirúrgica y para cada nivel de profundidad, si las señales MER visualizadas en el monitor se correspondían al STN u a otra zona del cerebro. Por lo que los registros quedaron etiquetados como “STN” o “no STN”. Con la idea de minimizar posibles errores humanos en el proceso de etiquetación de los registros se analizaron posteriormente a cada cirugía las imágenes de corregistros realizadas durante la cirugía (al inicio y al final de la misma) que permiten determinar mediante imágenes la profundidad a la cual se ingresó al STN. Luego de todo este proceso se contó con una base de datos precisa donde para cada nivel de profundidad del registro MER y por paciente se definió su etiqueta como “no STN” o “STN”.

6.1.1.1 Características obtenidas de los registros MER

Para cada nivel de profundidad se eliminaron en forma automática los registros asociados a ruidos de movimiento, mecánicos o eléctricos siguiendo la metodología propuesta en [9]. Luego se dividió el registro en ventanas de 1 s solapadas al 50%. Se calcularon *off line* las características temporales por cada ventana de 1 s utilizando el software Matlab®. Las dieciséis características temporales fueron definidas en la sección 4.4.1 y permiten describir las principales variables temporales de las señales obtenidas mediante los MER asociadas a la actividad de fondo sin considerar las espigas y otras que sólo caracterizan a las espigas. Las mismas se enumeran continuación con su codificación entre paréntesis mediante el cual se las referenciará en el presente documento:

- Valor de la amplitud basal (1. Basal).
- Curtosis (2. Curt).
- Longitud de la curva (3. CL).

- Umbral (4. TH).
- Cantidad de Picos (5. PK).
- La media cuadrática normalizada o valor cuadrático medio normalizado (6. RA).
- Energía no lineal promedio (7. NE).
- Cruces por cero (8. ZC).
- Índice de ráfaga de espigas (9. SBI)
- Índice de pausa de espigas (10. SPI)
- Ratio de pausa (11.SPR)
- Frecuencia de disparos de las espigas calculadas en espigas por segundo (12. SC).
- Amplitud diferencial media de espigas (13. SMAD)
- Desviación estándar de los intervalos entre espigas (14. SSD).
- Ratio del conteo de espigas (15. SCR).
- Mediana de la frecuencia de disparo de las espigas en Hz (16. SF).

Se estandarizaron por paciente las características calculadas para conformar una base de datos que por cada nivel de profundidad y por paciente contó con 16 características temporales etiquetada como “no STN” y “STN” que luego se utilizó para el entrenamiento y validación de los clasificadores. En total la base de datos contó con una composición del 59% etiquetados como “no STN” y un 41% como “STN”.

6.1.2 SOFTWARE

Para el desarrollo de la tesis se utilizaron los siguientes programas informáticos:

- Matlab[®] versión 2017a (MathWorks, Massachusetts, Estados Unidos), con los siguientes herramientas específicas: Statistics and Machine Learning Toolbox[™] y Signal Processing Toolbox[™]
- PRTools 4.2.0 que se ejecuta como una aplicación en Matlab.
- Software R versión 3.4.5 (R Foundation for Statistical Computing, Vienna, Austria).
- Sistema operativo Windows[®] 10 Home de 64 bits.

6.1.3 HARDWARE

Para el procesamiento de señales, obtención de las características, generación de modelos, cálculos y validación de los clasificadores se utilizó una computadora portátil con procesador Intel® Core™ i7 – 4710HQ 2,5 GHz, con 16 Gb de memoria RAM DDR3, disco duro SATA-III 6.0 Gb/s y 5400 RPM y capacidad de 1 TB.

Para los cálculos de optimización de algoritmos con el fin de definir los parámetros que logren un mejor desempeño de cada tipo de clasificador se utilizó un sistema de *cluster* informático del centro de Computación de Alto Rendimiento de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos, Argentina.

6.2 METODOLOGÍA GENERAL

Se trabajó inicialmente con el grupo de médicos de la unidad de neurocirugía del Hospital Universitario y Politécnico La Fe para definir el protocolo de registro de datos que permitiera luego su utilización en algoritmos de clasificación y una adecuada validación estadística de los resultados obtenidos. El tesista asistió a varias cirugías durante el proceso de desarrollo de la tesis para participar del proceso de adquisición de señales MER. La base de datos de registros MER se fue consolidando a lo largo de los años de desarrollo de la tesis, primero comenzando con 8 pacientes hasta llegar a la actual con el registro de 22 pacientes.

A medida que se conformaba la base de datos se fueron ensayando distintos algoritmos de clasificación para determinar aquellos que mejores resultados presentaban con los datos disponibles. Se hizo énfasis en aquellos algoritmos reportados por otros trabajos científicos en señales de la misma naturaleza los cuales fueron descriptos en la sección 5.3. En particular se ensayaron los siguientes algoritmos de clasificación individuales:

- Clasificador normal lineal de Bayes
- Clasificador normal cuadrático de Bayes
- Clasificador de Parzen
- K-vecinos más cercanos y su versión en lógica difusa
- Árbol de decisión: CART, CHAID, C5.0 y QUEST
- Clasificador de Naive Bayes
- Máquina de soporte vectorial

- Redes Neuronales: clasificador *feedforward* entrenado por *backpropagation* y clasificador de base radial

A medida que la cantidad de pacientes se incrementó, se volvieron a entrenar los clasificadores antes detallados. De las pruebas exploratorias se seleccionaron los de mejor desempeño como clasificadores individuales para la base de datos utilizada en la presente tesis: K-vecinos más cercanos y árboles de decisión. Luego se procedió a trabajar y explorar en profundidad estos dos tipos de clasificadores individuales con la idea de maximizar su rendimiento, seleccionando los parámetros de cada uno de manera de obtener un menor error de clasificación. En el capítulo 7 se desarrolla el marco teórico de dichos clasificadores individuales junto con los resultados obtenidos para su optimización.

En el capítulo 8 se aborda el marco teórico de la combinación de clasificadores individuales (*ensemble methods* por su denominación en inglés) junto con los resultados obtenidos con la base la datos.

En el capítulo 9 se desarrollan los algoritmos de selección de características tanto para los clasificadores del capítulo 7 como para los del capítulo 8 con la idea de lograr desempeños equivalentes o superiores con menor tiempo de cálculo.

La metodología general abordada en los capítulos 7, 8 y 9 de la presente tesis está centrada en métodos empíricos cuyos resultados son validados por casos para luego lograr una generalización estadística.

6.2.1 ÍNDICES DE DESEMPEÑO DE LOS CLASIFICADORES

Dado que se contó con el registro MER de 22 pacientes se procedió a entrenar los clasificadores con los datos de 21 pacientes (datos de entrenamiento) dejando los datos del paciente excluido para validación (datos de validación). Este proceso se repitió 22 veces dejando un paciente distinto por vez en cada conjunto de entrenamiento-validación. Esta generación de conjuntos de datos es un proceso de validación cruzada donde la formación fue “*leave one out*” por paciente. Cada conjunto de datos de validación, que representa a un paciente, se encuentra conformado por los registros MER de 4 trayectorias totales de los 2 hemisferios cerebrales.

Para obtener el valor medio de cada índice o indicador de desempeño se promediaron los resultados de las 22 pruebas. Dado que se utilizaron para todos los clasificadores diseñados en los capítulos 6, 7 y 8 los mismos 22 conjuntos de entrenamiento – validación los resultados de desempeño son comparables.

De esta forma, ningún dato del proceso de validación formó parte de los datos de entrenamiento siendo a su vez una prueba de cómo sería el desempeño medio de los clasificadores obtenidos funcionando realmente durante una cirugía con los datos de un nuevo paciente.

Un índice o indicador de desempeño es una función que provee una medida de calidad de un clasificador y generalmente se calculan a partir de la matriz de confusión.

Dado un problema con c clases $\{1, \dots, c\}$, la correspondiente matriz de confusión C definida según la ecuación 6.1 es una matriz cuadrada de tamaño $c \times c$ cuya ij -ésima entrada C_{ij} es el número de elementos de la clase verdadera i que fueron asignados por el clasificador a la clase j . Donde a_{ii} es un acierto del clasificador para la clase i y e_{ij} es un error donde el clasificador confundió la clase i con la clase j .

$$C = \begin{pmatrix} a_{11} & \cdots & e_{1c} \\ \vdots & \ddots & \vdots \\ e_{c1} & \cdots & a_{cc} \end{pmatrix} \quad (6.1)$$

En el caso particular de una matriz de confusión para un problema de dos clases definido como {no STN, STN} C puede escribirse como se expresa en la ecuación 6.2.

$$C = \begin{pmatrix} V_{STN} & F_{noSTN} \\ F_{STN} & V_{noSTN} \end{pmatrix} \quad (6.2)$$

Los verdaderos STN (V_{STN}) representan los números de aciertos de la clase STN.

Los verdaderos no STN (V_{noSTN}) representan los números de aciertos de la clase no STN

Las falsos no STN (F_{noSTN}) es el número de errores que resulta de asignar a la clase verdadera STN como no STN.

Las falsos STN (F_{STN}) es el número de errores que resulta de asignar a la clase verdadera no STN como STN.

Los indicadores de desempeño más utilizados en clasificadores para dos clases se detallan en la tabla 6.1 [46], [47]. Los mismos se pueden generalizar para multiclase.

6.2.2 TIEMPOS DE ENTRENAMIENTO Y VALIDACIÓN

En la presente tesis se define como tiempo de entrenamiento individual al tiempo en segundos que le demande a un algoritmo de clasificación tener su modelo o estructura a partir de los datos de entrenamiento. Los conjuntos de entrenamiento y validación se detallaron en la sección 6.2.1, a partir de lo cual surge que el **tiempo de entrenamiento** reportado en los próximos capítulos resulte del promedio de los tiempos de entrenamiento individuales de 22 clasificadores, uno por cada conjunto de entrenamiento. Este tiempo a los fines de la utilización de un sistema de clasificación automático, como herramienta de soporte en una cirugía para implantar un DBS, no tiene relevancia ya que es un proceso que se realizaría previo a la cirugía.

Índice	Ecuación	Evaluación	Rango
Exactitud (ACC)	$\frac{V_{STN} + V_{noSTN}}{V_{STN} + V_{noSTN} + F_{STN} + F_{noSTN}}$	Efectividad global de un clasificador. En clases muy desbalanceadas se utiliza el coeficiente de correlación de Matthews.	[0,1]
Precisión	$\frac{V_{STN}}{V_{STN} + F_{STN}}$	Concordancia entre las etiquetas de clase con las etiquetas positivas dadas por el clasificador.	[0,1]
Sensibilidad (SEN) o Razón de verdaderos Positivos	$\frac{V_{STN}}{V_{STN} + F_{noSTN}}$	Efectividad del clasificador para identificar la clase positiva.	[0,1]
Especificidad (ESP)	$\frac{V_{noSTN}}{V_{noSTN} + F_{STN}}$	Efectividad del clasificador para identificar la clase negativa.	[0,1]
Área bajo la curva ROC (AUC)	Cálculo del area bajo la curva para funciones continuas	Probabilidad del clasificador de clasificar correctamente.	[0,1]
Capacidad de diagnóstico (DOR)	$\frac{V_{STN} \cdot V_{noSTN}}{F_{STN} \cdot F_{noSTN}}$	Capacidad de discriminación entre clases.	[0,∞]
Error de clasificación	1-ACC	Efectividad global del clasificador	[0,1]
Razón de Falsos Positivos	$1 - ESP = \frac{F_{STN}}{F_{STN} + V_{noSTN}}$	Relación que evidencia los casos que el clasificador detecta un STN que en realidad es noSTN.	[0,1]

Tabla 6.1: Índices o indicadores de desempeño para clasificadores.

El tiempo de validación individual se define en la presente tesis como el tiempo en segundos que insume un algoritmo ya entrenado en obtener el resultado de la clasificación ante un nuevo conjunto de datos que en la sección 6.2.1 se denominaron datos de validación. Los mismos contienen registros de 4 trayectorias de los 2 hemisferios cerebrales por paciente. El **tiempo de validación** reportado en los próximos capítulos es el promedio de los tiempos de validación individuales de 22 clasificadores, uno por cada conjunto de validación.

En el marco del objetivo de la presente tesis se definen dos escenarios futuros de trabajo para el proceso clasificación automática durante una cirugía DBS:

El primero consiste en adquirir todos los registros MER de un hemisferio cerebral y procesarlos al finalizar la toma de datos por parte del equipo "MicroGuide". Para este caso se exportarían del equipo los archivos a una memoria externa y se procesarían en un portátil destinado a tal fin que ya contenga el modelo entrenado de clasificación. Los resultados indicarían con un grado de probabilidad las profundidades donde el clasificador encuentra que los nuevos datos se corresponden al STN. Para este escenario el tiempo de validación no es crítico siempre que pueda desarrollarse en el orden de los minutos.

El segundo escenario que se define es incluir el algoritmo de clasificación como parte del software que ofrecen los fabricantes de equipos de sistemas de navegación fisiológica para neurocirugía tipo el "MicroGuide". Se propone una actualización de información en un mismo nivel de profundidad cada 5 s. De esta ventana se tomarían 4 s de señal y el proceso de cálculo de características y validación del clasificador debería ocurrir en 1 s. Bajo estas condiciones el MER debería permanecer en cada nivel de profundidad como mínimo 30 s según lo postulado por [4] para tener potencia estadística significativa que permita la discriminación de distintas zonas cerebrales.

En los capítulos 7, 8, 9 y 10 cuando se refiera al **rendimiento** de un clasificador implicará una valoración de su desempeño a través de los indicadores definidos en la sección 6.2.1 como así también de su tiempo de validación.

6.2.3 ANÁLISIS ESTADÍSTICO

Se realizaron estudios estadísticos descriptivos y comparaciones estadísticas con pruebas no paramétricas, después de verificar los supuestos de normalidad (prueba de Kolmogorov-

Smirnov, con rechazo de hipótesis nula con $p < 0,05$). Para las comparaciones pareadas se utilizó el test de Friedman [47] y en caso de significancia global se utilizó Nemenyi como test *post hoc* lo que permitió en los casos posibles la comparación entre pares de clasificadores. El umbral de significancia entre las comparaciones se aceptó al 95% ($p < 0,05$). Todos los resultados se expresan como la media \pm desviación estándar (SD). Los análisis estadísticos parciales se fueron realizando al final de los capítulos 7, 8 y 9 para los clasificadores involucrados en cada caso, para presentar un estudio estadístico final en el capítulo 10.

Capítulo 7 Experiencia 1: Clasificación de registros MER mediante algoritmos individuales para la detección del STN

7.1 ALGORITMO DE K VECINOS MÁS CERCANOS (KNN)

El algoritmo estadístico de K vecinos más cercanos (KNN por su denominación en inglés *K Nearest Neighbours*) es uno de los más simples y atractivos en la clasificación de patrones, demostrando por lo general un elevado rendimiento. La estructura es similar a la utilizada en las redes de Bravais, aunque fue desarrollado por Fix y Hodges en 1951. Realiza una estimación a posteriori de la probabilidad sobre la cercanía de sus vecinos siendo un método no paramétrico dado que no hace ninguna suposición sobre la distribución que siguen los datos [48].

En la fase de entrenamiento se almacenan las clases y las características que se deben tener en cuenta. Posteriormente se pasa a la fase de clasificación, que evalúa las muestras de las que se desconoce la clase a la que pertenece según el patrón de los datos evaluados inicialmente, calculando las distancias y seleccionando los K vecinos más cercanos, que permitirá definir la clase a la que pertenecerá el nuevo elemento.

Sea $R(z) \subset \mathfrak{R}_N$ una hiperesfera con volumen V con centro z , siendo N_k el número de muestras del conjunto de entrenamiento T_k para el clasificador KNN y w_k la clase asignada. La probabilidad de tener exactamente n muestras dentro de $R(z)$ tiene una distribución binomial según la ecuación 7.1 [48].

$$E[n] = N_k \int_{y \in R(z)} p(y | w_k) dy \approx N_k V p(y | w_k) \quad (7.1)$$

Si se selecciona un radio alrededor de z que genere un volumen que contenga exactamente K muestras, entonces ese radio y su volumen dependen de la posición z en el espacio de medición. Por lo tanto podemos escribir $V(z)$ en lugar de V siendo la estimación de la densidad la indicada en 7.2 [48].

En KNN se fija el valor de K para estimar el modelo y se calcula el volumen $V(z)$ mínimo que abarcan esas K muestras [48]. La expresión 7.2 indica que en las regiones donde la estimación de densidad es grande, se espera que el volumen sea chico. Esto es similar a tener una zona reducida de interpolación. Si, por otra parte la estimación es pequeña, la esfera necesita crecer con el fin de recoger las muestras necesarias [48].

$$\hat{p}(z | w_k) dy = \frac{K}{N_k V(z)} \quad (7.2)$$

El parámetro K controla el equilibrio entre el sesgo y la varianza, según lo indicado en 7.3 [48]

$$\begin{aligned} k \rightarrow \infty \quad \text{y} \quad N_k \rightarrow \infty \quad \text{para obtener una variancia baja} \\ k / N_k \rightarrow 0 \quad \text{y} \quad N_k \rightarrow \infty \quad \text{para obtener un sesgo bajo} \end{aligned} \quad (7.3)$$

La técnica KNN tiene un interés práctico ya que trabaja sobre el conjunto de muestras para estimar el modelo sin calcular la densidad de probabilidad. Si K_k es número de muestras vecinas encontradas de la clase w_k entonces un estimador de la densidad condicional se expresa en la ecuación 7.4 [48].

$$\hat{p}(z | w_k) dy = \frac{K_k}{N_k V(z)} \quad (7.4)$$

Combinando 7.4 con el clasificador de Bayes con función de costo uniforme se obtiene la clasificación estimada expresada en 7.5 [48].

$\hat{w}(z) = w_k$ con k definida como

$$k = \arg \max_{i=1, \dots, K} \{ \hat{p}(z | w_k) \cdot \hat{P}(w_i) \} = \arg \max_{i=1, \dots, K} \left\{ \frac{K_i}{N_i V(z)} \frac{N_i}{N_s} \right\} = \arg \max_{i=1, \dots, K} \{ K_i \} \quad (7.5)$$

De la expresión anterior se concluye que la clase asignada al vector z es aquella con mayor número de muestras vecinas de la clase w_k más cercana a z . Para medir la cercanía de los vecinos, dependiendo de la aplicación y las características de los datos, se utilizan distintos tipos de métricas de distancia (D) como se detallan en la figura 7.1 [44], [49].

<p>Minkowsky</p> $D(x, y) = \sqrt[p]{\sum_{i=1}^m x_i - y_i ^p}$	<p>Euclídea</p> $D(x, y) = \sqrt{\sum_{i=1}^m x_i - y_i ^2}$
<p>Manhattan</p> $D(x, y) = \sum_{i=1}^m x_i - y_i $	<p>Camberra</p> $D(x, y) = \sum_{i=1}^m \left \frac{x_i - y_i}{x_i + y_i} \right ^p$
<p>Chebychev o Maximum</p> $D(x, y) = \max_{i=1, \dots, m} x_i - y_i $	<p>Cuadrática</p> $D(x, y) = (x - y)^T \cdot Q \cdot (x - y) = \sum_{j=1}^m \left(\sum_{i=1}^m (x_i - y_i) \cdot q_{ji} \right) (x_j - y_j)$ <p>Q: matriz mxm de pesos definida positiva</p>
<p>Mahalanobis</p> $D(x, y) = \sqrt{\det(V)} \cdot (x - y)^T \cdot V^{-1} \cdot (x - y)$ <p>V: matriz de covarianza de A_1, \dots, A_m donde A_j es el vector de valores para el atributo j del conjunto de entrenamiento</p>	<p>Correlación</p> $D(x, y) = \frac{\sum_{i=1}^m (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^m (x_i - \bar{x}_i)^2 \sum_{i=1}^m (y_i - \bar{y}_i)^2}}$
<p>Chi-cuadrado</p> $D(x, y) = \sum_{i=1}^m \frac{1}{suma_i} \left(\frac{x_i}{tam_x} - \frac{y_i}{tam_y} \right)^2$ <p>suma_i = suma de todos los valores del atributo i. tam_x = suma de todos los valores en el vector x.</p>	<p>Correlación del rango Kendall</p> $D(x, y) = \frac{2}{n(n-1)} \sum_{i=1}^m \sum_{j=1}^{i-1} sign(x_i - x_j) \cdot sign(y_i - y_j)$ <p>Sign(x)=signo de x</p>

Figura 7.1: Métricas de distancia utilizadas para el algoritmo KNN. Adaptado de [50]

En la figura 7.2 se observa una muestra z que se debe clasificar entre 3 clases distintas w_1 , w_2 y w_3 . El ejemplo se describe en 2 dimensiones mediante las características 1 y 2. Si se utiliza la distancia Euclídea y se fija $K=5$ observamos que de las cinco muestras más cercanas a z cuatro están etiquetadas como w_1 y una como w_3 , por lo tanto z será clasificada como clase w_1 .

En función del tipo de datos que queramos evaluar, es conveniente utilizar más vecinos o menos, generalmente incrementar K reduce el ruido de la clasificación, sin embargo hace que haya clases demasiado parecidas, y esto es un inconveniente a la hora de decidir la clase a la que pertenece la muestra evaluada. Ante este problema se aplican reglas que determinan qué hacer en caso de empate entre dos clases respecto a la pertenencia de una muestra.

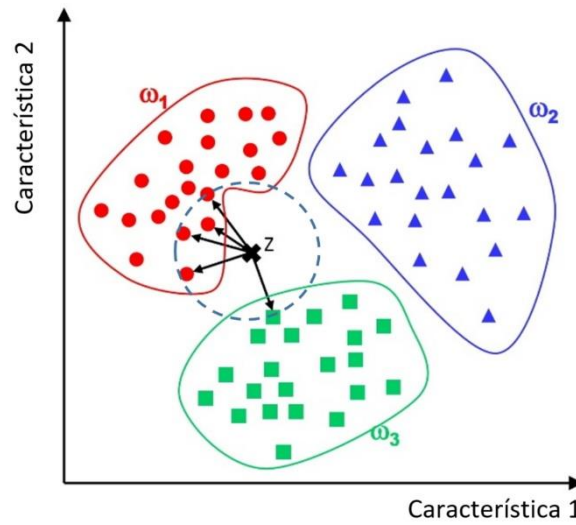


Figura 7.2: Ejemplo de clasificación utilizando con KNN siendo $K=5$ y 3 clases w_1 , w_2 y w_3 .

La figura 7.3 muestra un ejemplo sencillo que intenta clasificar el punto verde. Si con el clasificador KNN se utilizan los 3 vecinos más cercanos ($K=3$), la muestra pertenecerá a la clase formada por triángulos. Sin embargo, si se utiliza el clasificador con 5 vecinos más cercanos ($K=5$), el elemento evaluado pertenecerá a la clase formada por cuadrados ya que hay tres cuadrados y dos triángulos entre las cinco muestras más próximas al dato.

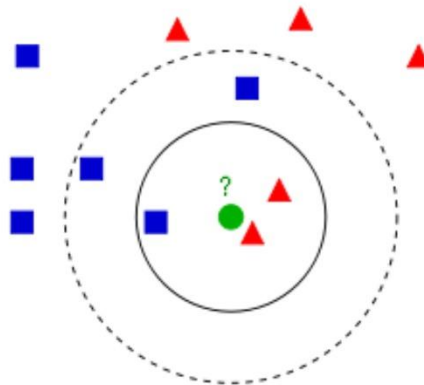


Figura 7.3: Ejemplo de clasificación utilizando KNN con $K=3$ (círculo de línea continua) resultando la muestra a clasificar (punto verde) en un triángulo mientras que si se utiliza $K=5$ (círculo de línea punteada) resulta clasificada como cuadrado. Adaptado de [51].

La mejor forma de seleccionar el número de vecinos K es mediante el entrenamiento de las muestras y/o validación cruzada de los datos de entrenamiento. Si existen solo 2 clases es

recomendable utilizar un K impar para evitar el empate de clases. Asimismo es necesario normalizar las características o atributos para que aquellos con mucho rango no tengan más peso que los de menor rango [52].

En la figura 7.4 puede observarse el desempeño de un clasificador KNN para 2 clases y distancia Euclídea: registros del núcleo subtalámico (etiquetados como “STN”) y aquellos que no pertenecen a ese núcleo basal (etiquetados como “no STN”). Los datos fueron calculados a partir de los registros MER de un paciente con Parkinson al que se le implantó un dispositivo DBS cuando se representó a las muestras mediante dos características SSF (mediana de la frecuencia de disparo de las espigas en Hz) y RA (valor RMS) descritas en la sección 4.4.1. La gráfica superior cuando $K=1$ y la inferior cuando $K=7$: puede observarse que con $K=1$ no se generaliza el modelo tanto como con $K=7$, donde se observan fronteras más generales.

El principal problema que presenta este algoritmo es en datos con ruido o aquellos que contienen características irrelevantes, dado que por ejemplo dos atributos relevantes perderán peso frente a veinte irrelevantes. Una forma de corregir esta desventaja es asignando un peso a las distancias de cada atributo, siendo más importantes unos atributos que otros, otra forma es aplicando algoritmos de selección de atributos para determinar aquellos más relevantes [52].

Es posible seleccionar los datos de entrenamiento más representativos para mejorar la generalidad y velocidad del clasificador KNN como así también eliminar datos con ruido. Las técnicas de edición permiten eliminar datos engañosos o con ruido, permitiendo asimismo suavizar las fronteras (mejora la capacidad de generalización pero no mejora la eficiencia). Los algoritmos de condensación se utilizan para eliminar datos superfluos pero mantiene los datos con ruido, por lo que se sugiere primero editar y luego condensar. Existen algoritmos híbridos que realizan ambos procesos [48].

7.2 ALGORITMO DE ÁRBOLES DE DECISIÓN

Un árbol de decisión (AD) es un clasificador expresado como una partición recursiva del espacio de entrada que consta de nodos que forman un árbol que nace con un nodo raíz. En un AD, cada nodo interno divide el espacio de instancia en dos o más subespacios (ramas) según una determinada función discreta que se evalúa a partir de los valores de los atributos de su entrada [53]. Las ramas del árbol están etiquetadas con los posibles valores de la prueba. A su

vez los nodos terminales se denominan hojas del árbol y especifican el valor de una clase que representa el valor objetivo más apropiado. Alternativamente, la hoja puede contener un vector de probabilidad que indica la probabilidad de que el atributo objetivo tenga un cierto valor. Las instancias se clasifican navegándolas desde la raíz del árbol hasta una hoja, según el resultado de las pruebas a lo largo de las ramas [53].

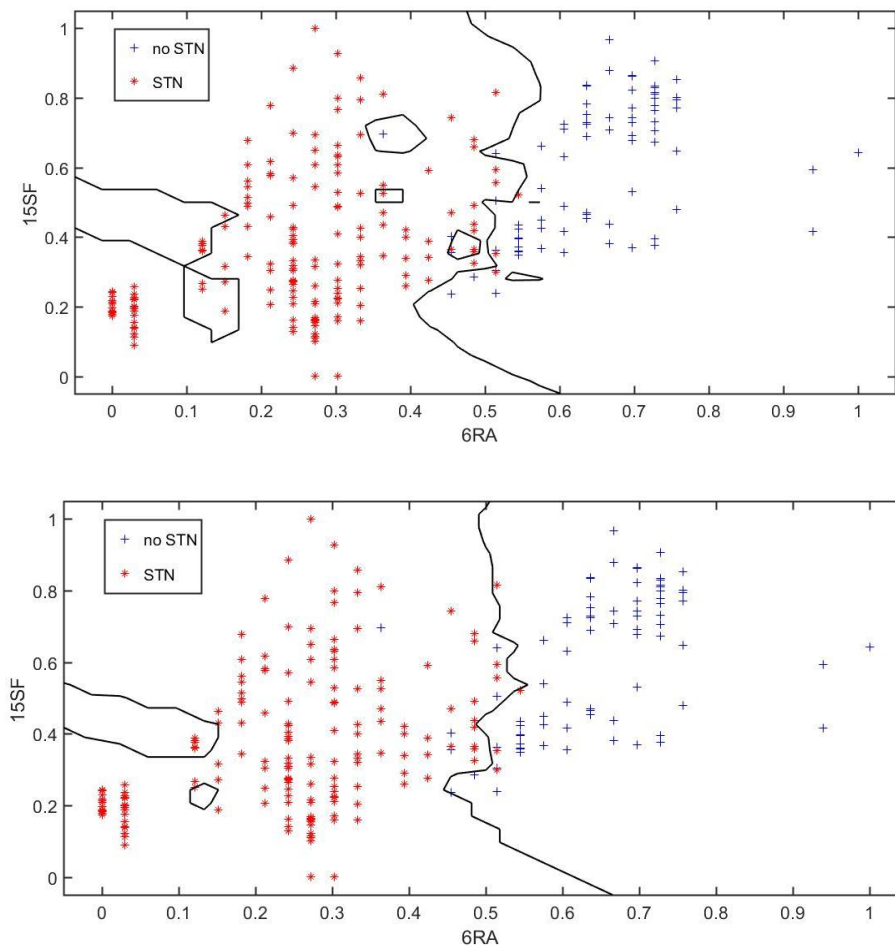


Figura 7.4: Ejemplo de clasificación utilizando KNN con 2 características temporales (SSF y RA) obtenidas de los registros MER de un paciente con Parkinson. Gráfica superior con $K=1$ y gráfica inferior con $K=7$ ambas estimadas con distancia Euclídea.

Los AD fueron desarrollados a principios de los años 60 mediante distintos algoritmos: CLS (Hunt y col., 1966), ID3 (Quinlan, 1979), CART (Breiman y col., 1984), ACLS (Niblett y col., 1982), ASSISTANT (Cestnik y col., 1987), C4.5 (Quinlan, 1993), etc [40], [53].

La figura 7.5 describe un árbol AD que analiza 3 características temporales de los registros MER de 15 pacientes para clasificar la zona STN. Los registros utilizados estaban etiquetados como STN y NO STN por especialistas neurofisiólogos. Los nodos internos se representan como triángulos, mientras que las hojas se indican como círculos. Cada nodo está etiquetado con el atributo o característica que evalúa y sus ramas están etiquetadas con sus valores de decisión correspondientes.

En el caso de los atributos numéricos, los AD se pueden interpretar geoméricamente como una colección de hiperplanos, cada uno ortogonal a uno de los ejes. De acuerdo a Breiman y col. (1984) la complejidad del árbol tiene un efecto crucial en su exactitud. La complejidad está explícitamente controlada por los criterios de stop o finalización utilizados y el método de poda empleado. Por lo general, la complejidad del AD se mide con una de las siguientes métricas: el número total de nodos, el número total de hojas, la profundidad del árbol y la cantidad de atributos utilizados.

Cada ruta desde la raíz de un árbol de decisión hasta una de sus hojas puede transformarse en una regla simplemente combinando las pruebas a lo largo del camino para formar la parte antecedente, y tomando la predicción de clase de la hoja como el valor de clase o consecuente. Por ejemplo, una de las rutas en la figura 7.5 puede transformarse en la regla:

"Si Energía no lineal promedio (NE) es menor a 0,107 y la curtosis (kur) de la señal es menor a 10,6301 y la longitud de la curva (CL) es menor a 1622,35 entonces el registro corresponde a la zona STN".

7.2.1 ALGORITMOS PARA LA CONSTRUCCIÓN DE ÁRBOLES DE DECISIÓN

Los inductores del AD son algoritmos que construyen automáticamente un árbol a partir de un conjunto de datos determinado. Normalmente, el objetivo es encontrar el AD óptimo al minimizar el error de generalización. Sin embargo, también se pueden definir otras funciones objetivos, por ejemplo, minimizar el número de nodos o minimizar la profundidad promedio del árbol.

El uso de algoritmos de AD óptimos solo es factible en pequeños problemas. En consecuencia, se requieren métodos heurísticos para aplicar los AD a problemas de mediana o gran dimensión. En términos generales, estos métodos se pueden dividir en dos grupos: de

7.2.2 CRITERIOS DE DIVISIÓN UNIVARIANTE

En la mayoría de los casos, las funciones de división discretas son univariadas lo que implica que un nodo interno se divide según el valor de un solo atributo. En consecuencia, el algoritmo busca el mejor atributo sobre el cual dividir. Hay varios criterios univariantes que se pueden caracterizar en dos grupos [40]:

- De acuerdo con la estructura de la medida: criterios basados en impurezas o impurezas normalizadas y criterios binarios.
- Según el origen de la medida: teoría de la información, dependencia y distancia.

Se describen a continuación los criterios más utilizados en la literatura:

Criterio basado en impureza

Dada una variable aleatoria x con k valores discretos, distribuidos de acuerdo con $P = (p_1, p_2, \dots, p_k)$, una medida de impureza es una función $\phi: [0, 1]^k \rightarrow R$ que satisface las siguientes condiciones [40]:

- $\phi(P) \geq 0$
- $\phi(P)$ se minimiza si $\exists i$ tal que $p_i = 1$
- $\phi(P)$ se maximiza si $\forall i, 1 \leq i \leq k$ tal que $p_i = 1/k$
- $\phi(P)$ es simétrica con respecto a los componentes de P
- $\phi(P)$ es diferenciable en todo su rango

Si el vector de probabilidad tiene un componente de valor 1, entonces la variable se define como pura. Por otro lado, si todos los componentes son iguales, el nivel de impureza alcanza el máximo.

Dado un conjunto de entrenamiento S , el vector de probabilidad del atributo objetivo y se define según la ecuación 7.6 [40], [53].

$$P_y(S) = \left(\frac{|\sigma_{y=c_1} S|}{|S|}, \dots, \frac{|\sigma_{y=c_{|dom(y)|}} S|}{|S|} \right) \quad (7.6)$$

La bondad del desdoblamiento debido al atributo discreto a_i se define como la reducción en la impureza del atributo objetivo después de la partición S de acuerdo con los valores $v_{ij} \in dom(a_i)$ se define en la ecuación 7.7.

$$\Delta \Phi(a_i, S) = \Phi(P_y(S)) - \sum_{j=1}^{|\text{dom}(a_i)|} \left(\frac{|\sigma_{a_i=v_{i,j}} S|}{|S|} \cdot \Phi(P_y(\sigma_{a_i=v_{i,j}} S)) \right) \quad (7.7)$$

Ganancia de información (GI)

Es un criterio basado en la medida de entropía (cuyo origen proviene de la teoría de la información) como medida de impureza (Quinlan, 1987) [40]. Representa la cantidad de información que se gana al seleccionar un atributo y queda definido según la ecuación 7.8.

$$GI(a_i, S) = Entropia(y, S) - \sum_{v_{i,j} \in \text{dom}(a_i)} \left(\frac{|\sigma_{a_i=v_{i,j}} S|}{|S|} \cdot Entropia(y, \sigma_{a_i=v_{i,j}} S) \right) \quad (7.8)$$

Dónde la entropía se define según se expresa en la ecuación 7.9.

$$Entropia(y, S) = \sum_{c_j \in \text{dom}(y)} \left(- \frac{|\sigma_{y=c_j} S|}{|S|} \cdot \log_2 \frac{|\sigma_{y=c_j} S|}{|S|} \right) \quad (7.9)$$

Básicamente mide que tan bien un atributo separa a los ejemplos de entrenamiento de acuerdo a la clase.

Índice Gini

El índice de Gini es un criterio basado en impurezas que mide las divergencias entre las distribuciones de probabilidad de los valores del atributo objetivo. Queda definido según la ecuación 7.10 [40], [53].

$$Gini(y, S) = 1 - \sum_{c_j \in \text{dom}(y)} \left(\frac{|\sigma_{y=c_j} S|}{|S|} \right)^2 \quad (7.10)$$

En consecuencia, el criterio de evaluación para seleccionar el atributo a_i se define en la ecuación 7.11. En el ejemplo de la Figura 7.5 se utilizó este criterio para establecer las bifurcaciones de cada rama.

$$GananciaGini(a_i, S) = Gini(y, S) - \sum_{v_{i,j} \in \text{dom}(a_i)} \left(\frac{|\sigma_{a_i=v_{i,j}} S|}{|S|} \cdot Gini(y, \sigma_{a_i=v_{i,j}} S) \right) \quad (7.11)$$

Ratio de verosimilitud con el estadístico Chi-Cuadrado

Este criterio está definido por la ecuación 7.12. Su relación es útil para medir la significancia estadística del criterio de ganancia información [40].

$$G^2(a_i, S) = 2 \cdot \ln(2) \cdot |S| \cdot GI(a_i, S) \quad (7.12)$$

La hipótesis nula (H_0) plantea que el atributo de entrada y el atributo objetivo son condicionalmente independientes por lo que puede aplicarse el estadístico Chi-Cuadrado con grado de libertad (gl) según se expresa en la ecuación 7.13 [40].

$$gl = (dom(a_i) - 1) \cdot (dom(y) - 1) \quad (7.13)$$

Criterio basado en la impureza normalizada

El criterio basado en impurezas descrito anteriormente está sesgado hacia atributos con valores grandes (Quinlan, 1986) [40]. Por lo que es necesario normalizar los valores que toman los atributos para mejorar la exactitud de árbol en el proceso de clasificación.

Ratio de Ganancia (RG)

Este criterio normaliza la ganancia de información como se indica en la ecuación 7.15 [40], [53].

$$RG(a_i, S) = \frac{GI(a_i, S)}{Entropia(a_i, S)} \quad \text{con } Entropia(a_i, S) \neq 0 \quad (7.15)$$

Este ratio puede tender a favorecer atributos para los cuales el denominador es muy pequeño. En consecuencia, se sugiere implementarlo en dos etapas: primero, la ganancia de información se calcula para todos los atributos y se seleccionan aquellos que han obtenido al menos una ganancia igual o superior al promedio y luego se selecciona el atributo que ha obtenido la mejor ganancia de relación. Se ha demostrado que el ratio de ganancia tiende a superar los criterios simples de ganancia de información, tanto desde el aspecto de exactitud como desde los aspectos de complejidad del clasificador (Quinlan, 1988).

7.2.2.1 Comparación de los métodos univariados

Estudios comparativos de los criterios de división descritos anteriormente, y otros, han sido realizados por varios investigadores durante los últimos treinta años. La mayoría de estas

comparaciones se basan en resultados empíricos, aunque hay algunas conclusiones teóricas. Muchos de los investigadores señalan que en la mayoría de los casos, la elección de los criterios de división no hará mucha diferencia en el rendimiento del árbol. Cada criterio es superior en algunos casos e inferior en otros, como sugiere el teorema de "*No free lunch*" [11].

7.2.3 CRITERIOS DE DIVISIÓN MULTIVARIANTE

En los criterios de división multivariante, varios atributos pueden participar en una prueba de división de un único nodo. Encontrar el mejor criterio multivariado es más complicado que encontrar la mejor división univariada. Además, aunque este tipo de criterios pueden mejorar drásticamente el rendimiento del árbol, son mucho menos utilizados que los criterios univariados.

La mayoría de los criterios de división multivariante se basan en la combinación lineal de los atributos de entrada. Encontrar la mejor combinación lineal se puede realizar usando una búsqueda codiciosa (Breiman y col., 1984; Murthy, 1998), programación lineal (Duda y Hart, 1973; Bennett y Mangasarian, 1994), análisis discriminante lineal (Duda y Hart, 1973; Friedman, 1977; Sklansky y Wassel, 1981; Lin y Fu, 1983 y otros (Utgoff, 1989a; Lubinsky, 1993; Sethi y Yoo, 1994) [40], [53].

7.2.4 CRITERIOS DE DETENCIÓN O STOP

La fase de crecimiento de un árbol continúa hasta que se desencadena un criterio de detención. Las siguientes condiciones son reglas utilizadas comúnmente como criterio de detención [49]:

1. Todas las instancias en el conjunto de entrenamiento pertenecen a un único valor de y .
2. Se alcanzó la profundidad máxima del árbol.
3. El número de casos en el nodo terminal es menor que el número mínimo de casos para los nodos principales.
4. Si el nodo se dividió, el número de casos en uno o más nodos secundarios sería menor que el número mínimo de casos para los nodos secundarios.
5. El mejor criterio de división no es mayor que un cierto umbral.

7.2.5 MÉTODOS DE PODA

Emplear criterios estrictos de detención tiende a crear AD pequeños y mal adaptados y su extremo opuesto tiende a generar grandes árboles que están sobre ajustados para el conjunto de entrenamiento. Los métodos de poda (*pruning* por su denominación en inglés) originalmente sugeridos en (Breiman y col., 1984) se desarrollaron para resolver este dilema [49]. De acuerdo con esta metodología, se usa un criterio de detención relajado, lo que permite que el árbol de decisión sobreajuste el conjunto de entrenamiento. Luego, el árbol sobre equipado se corta en un árbol más pequeño al eliminar las subdivisiones que no contribuyen a la exactitud de la generalización. Se ha demostrado en varios estudios que el empleo de métodos de poda puede mejorar el rendimiento de generalización de un AD, especialmente en dominios ruidosos.

Hay varias técnicas para podar los AD, la mayoría de ellas realiza un recorrido descendente o ascendente sobre los nodos. Un nodo se elimina si esta operación mejora ciertos criterios. Existen dos tipos de poda:

Pre poda: detener en algún momento el crecimiento del árbol durante la construcción descendente cuando ya no hay suficientes datos para hacer decisiones criteriosas (por ejemplo tener un mínimo número de ejemplo por hoja). Básicamente se modifica el criterio de paro bajo algún criterio.

Post poda: luego de obtener el árbol completo, eliminar las ramas que no mejoran el desempeño del AD.

Los métodos más utilizados para la poda son [49]: a) poda por complejidad de costos (también conocida como poda del enlace más débil o poda por complejidad del error); b) poda por reducción del error; c) poda basada en errores; d) la poda por errores pesimistas; e) poda óptima y e) poda por mínimos errores entre otros.

Varios estudios intentan comparar el rendimiento de diferentes técnicas de poda (Quinlan, 1987; Mingers, 1989; Esposito y col., 1997). Los resultados indican que algunos métodos (como poda de complejidad de costo, reducción de errores de poda) tienden a sobre podar, es decir, crear AD más pequeños pero menos precisos. Otros métodos (como la poda basada en errores, la poda de errores pesimistas y la poda mínima de errores) tienen tendencia a sub podar. La mayoría de las comparaciones concluyeron que el teorema de "*no free lunch*" [11] se aplica

también en este caso, es decir, no existe un método de poda que en cualquier caso supere a otros métodos de poda [40].

7.2.6 ALGORITMOS QUE IMPLEMENTAN ÁRBOLES DE DECISIÓN (INDUCTORES)

7.2.6.1 ID3

El algoritmo ID3 [40] se considera como un algoritmo de árbol de decisión muy simple (Quinlan, 1986). Utiliza la ganancia de información como criterio de división. El crecimiento se detiene cuando todas las instancias pertenecen a un único valor de la característica objetivo o cuando la mejor ganancia de información no es mayor que cero. ID3 no aplica ningún procedimiento de poda ni maneja atributos numéricos o valores perdidos.

7.2.6.2 C4.5

C4.5 es una evolución de ID3, presentada por el mismo autor (Quinlan, 1993) [40]. Utiliza la relación de ganancia como criterio de división. La división cesa cuando el número de instancias a dividir está por debajo de un cierto umbral. La poda basada en errores se realiza después de la fase de crecimiento. C4.5 puede manejar atributos numéricos. Puede inducir a partir de un conjunto de entrenamiento que incorpore valores perdidos mediante el uso de criterios de relación de ganancia corregida [53].

7.2.6.3 C5.0

C5.0 es una evolución de C4.5, presentada por el mismo autor (Quinlan, 2000). Está basado en C4.5 pero realiza un uso más eficiente de la memoria. Es un algoritmo mucho más rápido que C4.5 (como mínimo un orden de magnitud para la construcción de las reglas del AD). Genera AD más chicos para grandes conjuntos de datos con errores de clasificación similares o menores que C4.7. Como clasificador puede anticipar que atributos son relevantes y cuales no [54].

7.2.6.4 CART (*Classification And Regression Tree*, por sus siglas en inglés)

Fue propuesto por (Breiman y col., 1984). Se caracteriza por el hecho de que construye árboles binarios, es decir, cada nodo interno tiene exactamente dos sub nodos. Las divisiones se seleccionan usando los dos criterios y el árbol obtenido se poda mediante el algoritmo de poda por complejidad de costos. Cuando se proporciona, CART puede considerar los costos de

clasificación errónea en la inducción del árbol. También permite a los usuarios proporcionar una distribución de probabilidad previa [40].

Una característica importante de CART [53] es su capacidad para generar árboles de regresión. Los árboles de regresión son árboles donde sus hojas predicen un número real y no una clase. En caso de regresión, CART busca divisiones que minimicen el error cuadrático de predicción al cuadrado. La predicción en cada hoja se basa en la media ponderada para el nodo.

7.2.6.5 CHAID

A principio de los años setenta, los investigadores en estadística aplicada desarrollaron procedimientos para generar AD, tales como: AID (Sonquist y col., 1971), MAID (Gilio, 1972), THAID (Morgan y Messenger, 1973) y CHAID (Kass, 1980). CHAID (*Chi-square Automatic Interaction and Detection* por sus siglas en inglés) fue diseñado originalmente para manejar atributos nominales solamente [53]. Para cada atributo de entrada a_i , CHAID encuentra el par de valores V_i que difiere significativamente menos respecto del atributo objetivo. La diferencia significativa se mide por el valor p obtenido de una prueba estadística. La prueba estadística utilizada depende del tipo de atributo destino: si es continuo se usa una prueba F, si es nominal, se usa una prueba Chi-cuadrado de Pearson y si es ordinal entonces se usa una prueba de probabilidad de ratios.

Para cada par seleccionado, CHAID verifica si el valor de p obtenido es mayor que un determinado umbral de fusión. Si la respuesta es positiva, fusiona los valores y busca un par potencial adicional para fusionar. El proceso se repite hasta que no se encuentran pares significativos [40].

El mejor atributo de entrada que se utilizará para dividir el nodo actual se selecciona a continuación, de modo que cada nodo secundario se compone de un grupo de valores homogéneos del atributo seleccionado. No se realiza ninguna división si el valor p ajustado del mejor atributo de entrada no es inferior a un cierto umbral de división. Este procedimiento también se detiene cuando se cumple una de las siguientes condiciones [40]:

1. Se alcanza la profundidad máxima del árbol.
2. Se alcanza el número mínimo de casos en el nodo para ser uno de los padres, por lo que no se puede dividir más.
3. Se alcanza el número mínimo de casos en el nodo para ser un nodo hijo.

CHAID maneja los valores perdidos al tratarlos a todos como una única categoría válida y el algoritmo no realiza poda del árbol.

7.2.6.6 QUEST

El algoritmo QUEST (*Quick, Unbiased, Efficient, Statistical Tree*, por sus siglas en inglés) admite divisiones univariadas y lineales (Loh y Shih, 1997) [40]. Para cada división, la asociación entre un atributo de entrada y el atributo objetivo se calcula utilizando ANOVA F-test o la prueba de Levene (para atributos ordinales y continuos) o Chi-cuadrado de Pearson (para atributos nominales). El atributo que obtiene la mayor asociación con el atributo objetivo se selecciona para dividir. El análisis discriminante cuadrático (QDA) se aplica para encontrar el punto de división óptimo para el atributo de entrada. QUEST tiene muy poco sesgo y produce árboles de decisión binarios [40]. La validación cruzada con $n=10$ se utiliza para la poda de los árboles generados.

7.2.7 VENTAJAS Y DESVENTAJAS DE LOS ÁRBOLES DE DECISIÓN

Los AD presentan las siguientes desventajas [40], [49], [53]:

1. La mayoría de los algoritmos (como ID3 y C4.5) requieren que el atributo objetivo tenga sólo valores discretos.

2. Como los AD utilizan el método de "divide y vencerás", tienden a formarse bien si existen algunos atributos muy relevantes, pero menos si existen muchas interacciones complejas. Como la mayoría de los árboles de decisión dividen el espacio de la instancia en regiones mutuamente excluyentes para representar un concepto, en algunos casos el árbol debe contener varias duplicaciones del mismo subárbol para representar al clasificador.

3. La característica codiciosa de los AD conduce a otra desventaja: son muy sensibles al conjunto de entrenamiento, a los atributos irrelevantes y al ruido (Quinlan, 1993).

Las ventajas más significativas de los AD como una herramienta de clasificación son las siguientes [40], [49], [53]:

1. Los AD son auto explicativos y cuando son compactados también son fáciles de seguir y analizar. Si el AD tiene un número razonable de hojas, puede ser interpretado por usuarios no

profesionales. Además, los AD se pueden convertir en un conjunto de reglas, por lo tanto, esta representación se considera comprensible.

2. Los árboles de decisión pueden manejar atributos de entrada tanto nominales como numéricos.

3. La representación del árbol de decisión es lo suficientemente rica para representar cualquier clasificador de valores discretos.

4. Los árboles de decisión son capaces de manejar conjuntos de datos que pueden tener errores.

7. Los árboles de decisión son capaces de manejar conjuntos de datos que pueden tener valores faltantes.

6. Los árboles de decisión se consideran un método no paramétrico. Esto significa que los árboles de decisión no tienen suposiciones sobre la distribución del espacio y la estructura del clasificador.

7.3 RESULTADOS CON KNN

Inicialmente se realizó un estudio para determinar los valores óptimos de la cantidad de vecinos (K) y el tipo de medición de distancia que optimizaba el desempeño del algoritmo KNN. Para esto se utilizó la optimización Bayesiana tomando como criterio la minimización de la función error de clasificación cuando los conjuntos de datos de entrenamiento y validación se configuraron según lo detallado en la sección 6.2.1. Los resultados luego de implementar la optimización en Matlab[®] se presentan en la tabla 7.1 y la figura 7.6. El algoritmo de optimización bayesiano indaga sobre valores extremos e intermedios de K que van desde 1 hasta 14049, como así también verifica que tipo de distancia minimiza el error de clasificación.

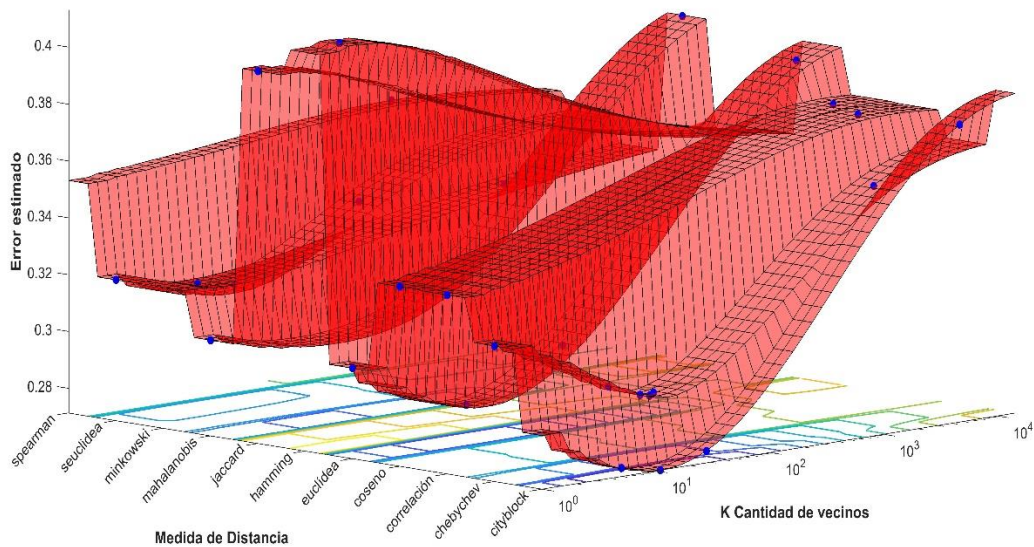


Figura 7.6: Gráfica del estudio de optimización Bayesiana para determinar los valores de K y tipo de distancia óptimos para el algoritmo KNN en base a los datos de la tesis.

Se observa en la tabla 7.1 y la figura 7.6 que existen varias configuraciones posibles para K y tipo de distancia que logran minimizar la función objetivo. En particular se selecciona un valor de $K=11$ y tipo de medición de distancia *Manhattan* (también denominada *Cityblock*) ya que minimizan la función de error siendo a su vez la configuración de menor tiempo de ejecución del optimizador. Asimismo al resultar K impar facilita el proceso de clasificación, ya que evita las situaciones de empate de vecinos.

Iteración	Tiempo ejecución optimización [s]	Error Observado	Cantidad de Vecinos (K)	Tipo de distancia	Valoración resultado
1	69.862	0,31245	11	SEuclídea	Muy buena
2	30,56	0,31245	6447	coseno	Aceptable
3	70.309	0,2936	25	Chebychev	Muy buena
4	18,8	0,2936	688	Spearman	Aceptable
5	70.332	0,2936	25	Chebychev	Aceptable
6	7.064	0,2928	23	Chebychev	Muy buena
7	70.816	0,2928	1	Chebychev	Aceptable
8	7.532	0,2928	10	Chebychev	Aceptable
9	19.727	0,2928	2155	Chebychev	Aceptable
10	71.439	0,2928	137	SEuclídea	Aceptable
11	6.028	0,2928	1	SEuclídea	Aceptable
12	71.604	0,2928	19	Chebychev	Aceptable
13	50.647	0,27434	5	Cityblock	Muy buena
14	50.238	0,27434	28	Cityblock	Aceptable
15	4.816	0,27143	11	Cityblock	La Mejor
16	26.138	0,27143	4710	Cityblock	Aceptable
17	49.504	0,27143	2	Minkowsky	Aceptable
18	11.481	0,27143	992	Minkowsky	Aceptable
19	45.259	0,27143	1	Mahalanobis	Aceptable
20	46.291	0,27143	81	Mahalanobis	Aceptable
21	99,7	0,27143	14049	Mahalanobis	Aceptable
22	93.607	0,27143	1	Jaccard	Aceptable
23	47.261	0,27143	1	Correlación	Aceptable
24	24,3	0,27143	4090	Correlación	Aceptable
25	5,02	0,27143	1	Euclídea	Aceptable
26	55.177	0,27143	71	Euclídea	Aceptable
27	38.905	0,27143	7975	Euclídea	Aceptable
28	67.799	0,27143	10	Euclídea	Aceptable
29	80.904	0,27143	2	Hamming	Aceptable
30	53.787	0,27143	1	coseno	Aceptable

Tabla 7.1: Valores obtenidos de la optimización Bayesina implementada en Matlab de los parámetros cantidad de vecinos (K) y medida de distancia para el clasificador KNN y los datos de la presente tesis.

Con el valor de K y tipo de distancia seleccionados (K=11 y distancia *Cityblock*) se adoptó como peso la inversa de la distancia al cuadrado y se procedió en entrenar los clasificadores con los 22 grupos de entrenamiento según lo descrito en la sección 6.2.1. El tiempo de entrenamiento promedio fue de $0,031 \pm 0,006$ s.

Luego se validó cada clasificador KNN con su conjunto de validación obteniéndose los valores exactitud (ACC), especificidad (ESP), sensibilidad (SEN), área bajo la curva ROC (AUC), índice de capacidad de diagnóstico (DOR) y tiempo de entrenamiento según se detalla en la tabla 7.2. El tiempo promedio de validación por paciente fue de $0,196 \pm 0,018$ s. Tanto para el entrenamiento como para la validación se utilizaron datos estandarizados. En la figura 7.7 se observa la curva ROC.

Se logra con la configuración de KNN propuesta una exactitud promedio con los datos de validación de $0,751 \pm 0,013$. Este desvío estándar indica que no existe una elevada dispersión en los valores individuales de ACC por paciente, lo mismo que ocurre con los otros indicadores de desempeño. Se observa de la tabla 7.2 que el algoritmo KNN presenta una especificidad porcentual promedio (78,6%) superior a la sensibilidad porcentual promedio (70%), lo que representa mayor capacidad de clasificación sobre la clase no STN. De la figura 7.7 se observa que el área bajo la curva ROC es superior a la situación por azar presentando un área de promedio de $0,816 \pm 0,013$. En el algoritmo KNN propuesto el tiempo de validación es aproximadamente 6 veces superior al de entrenamiento.

7.4 RESULTADOS CON ÁRBOLES DE DECISIÓN

De los resultados preliminares detallados en la sección 6.2 los algoritmos que mejor desempeño lograron con los datos de la presente tesis son CART y CHAID con criterios de división univariados. Por lo que los resultados detallados se presentan para esos dos algoritmos. En cada caso se pueden seleccionar parámetros que combinados permiten minimizar una función objetivo como el error de clasificación o la entropía.

En la presente tesis se utilizó la optimización Bayesiana tanto para CART como para CHAID con el objetivo de minimizar el error de clasificación cuando los conjuntos de datos de entrenamiento y validación se configuraron según lo detallado en la sección 6.2.1. Se estudiaron los parámetros: cantidad de hojas mínimas, máxima cantidad de divisiones y criterio de división. Los resultados luego de implementar la optimización en Matlab® se presentan en la figura 7.8 para CART y la figura 7.9 para CHAID. Se puede observar que para CART un valor de mínima cantidad de hojas igual a 2 y máxima cantidad de divisiones en 2221 minimizan la función de error. A su vez para CHAID la optimización arrojó una mínima cantidad de hojas

igual a 3 y máxima cantidad de divisiones en 7091. Para ambos algoritmos se adoptó, aplicando similar estudio de optimización, el criterio de división de Gini.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,758	0,796	0,703	0,823	9,241	0,253
Paciente 2	0,750	0,792	0,690	0,811	8,434	0,186
Paciente 3	0,761	0,781	0,732	0,823	9,770	0,200
Paciente 4	0,754	0,780	0,717	0,815	9,000	0,201
Paciente 5	0,746	0,790	0,682	0,808	8,084	0,229
Paciente 6	0,768	0,788	0,740	0,828	10,544	0,192
Paciente 7	0,770	0,803	0,721	0,834	10,550	0,179
Paciente 8	0,738	0,764	0,699	0,809	7,534	0,172
Paciente 9	0,734	0,773	0,677	0,799	7,144	0,189
Paciente 10	0,754	0,790	0,701	0,822	8,821	0,200
Paciente 11	0,750	0,797	0,682	0,810	8,385	0,190
Paciente 12	0,749	0,790	0,689	0,814	8,346	0,195
Paciente 13	0,768	0,797	0,726	0,829	10,394	0,190
Paciente 14	0,763	0,801	0,709	0,827	9,801	0,190
Paciente 15	0,747	0,779	0,700	0,826	8,215	0,190
Paciente 16	0,763	0,798	0,711	0,830	9,732	0,193
Paciente 17	0,727	0,767	0,669	0,787	6,641	0,170
Paciente 18	0,755	0,779	0,721	0,815	9,075	0,206
Paciente 19	0,750	0,784	0,700	0,815	8,466	0,201
Paciente 20	0,756	0,808	0,680	0,829	8,944	0,203
Paciente 21	0,746	0,792	0,680	0,810	8,056	0,189
Paciente 22	0,724	0,754	0,680	0,789	6,517	0,183
Valor medio	0,751	0,786	0,7	0,816	8,713	0,196
Desvío Estándar	0,013	0,013	0,020	0,013	1,152	0,018

Tabla 7.2: Medidas de desempeño de los 22 clasificadores KNN entrenados utilizando como datos de validación las 16 características calculadas por paciente.

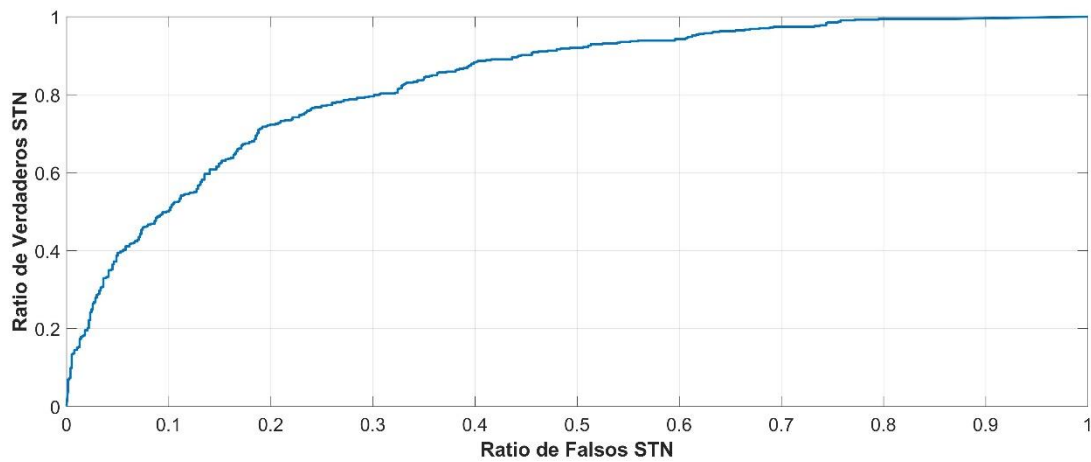


Figura 7.7: Gráfica de la curva ROC para el clasificador KNN.

Optimizados los parámetros de CART se procedió a entrenar los 22 clasificadores según datos de entrenamiento. Los resultados de los indicadores de desempeño con los datos de entrenamiento se detallan en la tabla 7.3. El tiempo de entrenamiento promedio para CART fue de $0,282 \pm 0,027$ s. Las curvas ROC para CART se presentan en la gráfica 7.10.

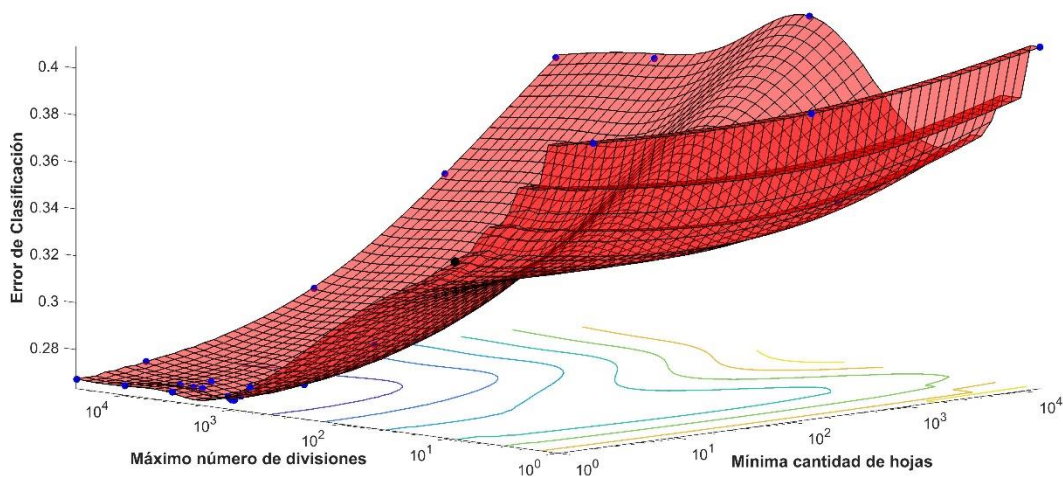


Figura 7.8: Gráfica del estudio de optimización Bayesiana para determinar los valores de máximo número de divisiones y mínima cantidad de hojas el algoritmo CART en base a los datos de la tesis.

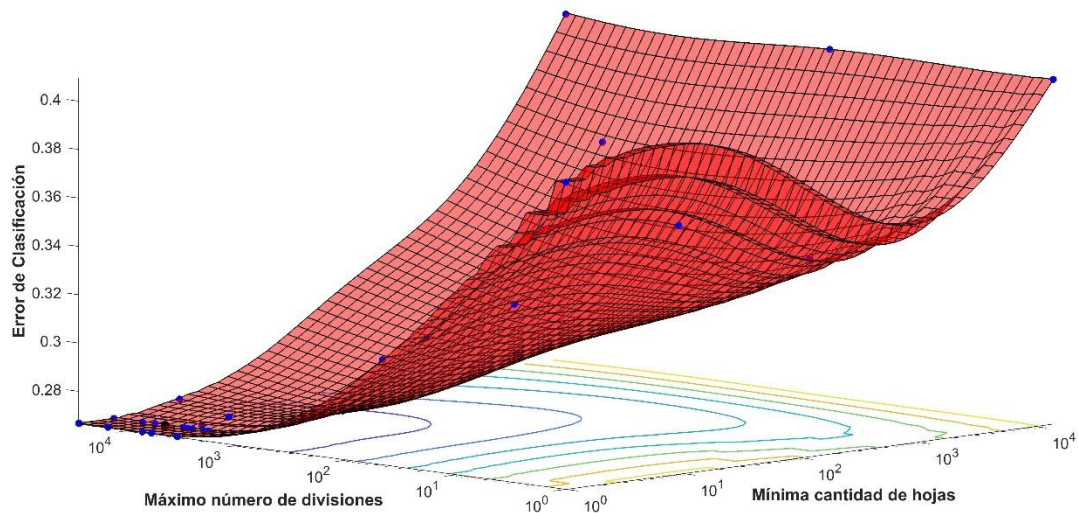


Figura 7.9: Gráfica del estudio de optimización Bayesiana para determinar los valores de máximo número de divisiones y mínima cantidad de hojas el algoritmo CHAID en base a los datos de la tesis.

Con la configuración de CART propuesta se observa en la tabla 7.3 una exactitud promedio con los datos de validación de $0,738 \pm 0,01$. Este desvío estándar indica que no existe una elevada dispersión en los valores individuales de ACC por paciente, lo mismo que ocurre con los otros indicadores de desempeño. Se observa de la tabla 7.3 que CART presenta una especificidad porcentual promedio (77,2%) superior a la sensibilidad porcentual promedio (68,8%), lo que de manera similar a KNN representa mayor capacidad de clasificación CART sobre la clase no STN. De la figura 7.10 (línea azul) se observa que el área bajo la curva ROC mejora la situación por azar presentando un área de promedio de $0,783 \pm 0,01$ inferior a la de KNN. En el algoritmo CART propuesto el tiempo de validación es aproximadamente 140 veces más rápido que el tiempo promedio de entrenamiento, situación extremadamente opuesta a la que presentó el algoritmo KNN. Esto indica que CART es un algoritmo muy veloz para realizar el proceso de clasificación de los datos de validación.

Para los árboles de decisión tipo CHAID se realizó el mismo procedimiento de entrenamiento resultando los indicadores de desempeño con los datos de entrenamiento por paciente que se detallan en la tabla 7.4. El tiempo de entrenamiento promedio fue de $0,342 \pm 0,036$ s y la curva ROC para CHAID se presenta en la gráfica 7.10 (gráfica roja punteada).

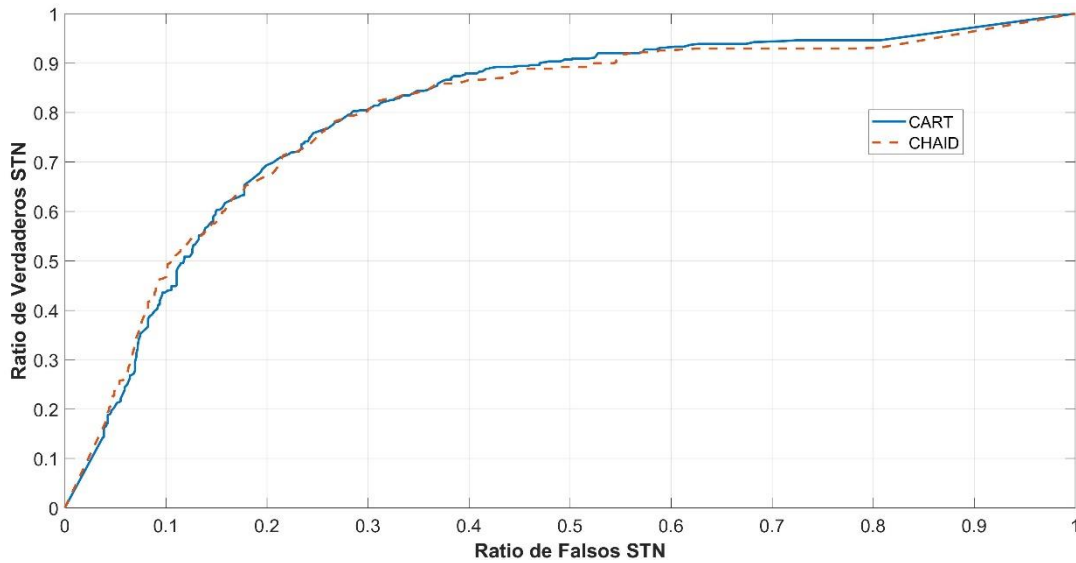


Figura 7.10: Curva ROC del desempeño de CART (gráfica azul) y CHAID (gráfica roja).

Con la configuración de CHAID propuesta se observa en la tabla 7.4 una exactitud promedio con los datos de validación de $0,734 \pm 0,004$. Este desvío estándar de $0,004$ indica que no existe una elevada dispersión en los valores individuales de ACC por paciente, situación que se repite con los otros indicadores de desempeño. Se observa de la tabla 7.4 que CHAID presenta una especificidad porcentual promedio (77,7%) superior a la sensibilidad porcentual promedio (67,1%), lo que de manera similar a KNN y a CART representa mayor capacidad de clasificación CHAID sobre la clase no STN. De la figura 7.10 (línea roja punteada) se observa que el área bajo la curva ROC mejora la situación por azar presentando un área promedio de $0,783 \pm 0,010$ igual a CART. La curva ROC presenta diferencias sutiles entre ambos AD: para valores bajos de ratio de falsos STN CHAID presenta levemente un área mayor y luego esta situación se revierte a partir de un valor de 0,35 y es CART quien mejora su área ROC. Desde el punto de vista del desempeño CART y CHAID no presentan diferencias promedio para los datos de la presente tesis.

El tiempo de entrenamiento de CHAID es aproximadamente un 20% más lento que CART pero ambos presentan el mismo tiempo de validación promedio de 0,002 s. En el algoritmo CHAID propuesto el tiempo de validación es aproximadamente 170 veces más rápido que el tiempo promedio de entrenamiento, situación similar a la de CART y opuesta a KNN. Esto

indica que CHAID al igual que CART son algoritmos muy rápidos para realizar el proceso de clasificación de los datos de validación.

Datos de entrenamiento	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,743	0,774	0,697	0,786	7,901	0,005
Paciente 2	0,736	0,772	0,684	0,780	7,338	0,002
Paciente 3	0,750	0,768	0,723	0,790	8,659	0,002
Paciente 4	0,732	0,759	0,693	0,782	7,132	0,002
Paciente 5	0,739	0,762	0,706	0,788	7,696	0,003
Paciente 6	0,741	0,770	0,701	0,779	7,823	0,002
Paciente 7	0,744	0,776	0,699	0,792	8,043	0,002
Paciente 8	0,741	0,752	0,727	0,768	8,049	0,002
Paciente 9	0,732	0,768	0,680	0,774	7,058	0,002
Paciente 10	0,715	0,763	0,645	0,766	5,855	0,002
Paciente 11	0,737	0,762	0,702	0,780	7,540	0,002
Paciente 12	0,747	0,777	0,702	0,800	8,226	0,002
Paciente 13	0,745	0,786	0,685	0,790	8,015	0,003
Paciente 14	0,739	0,767	0,698	0,784	7,622	0,002
Paciente 15	0,726	0,759	0,678	0,782	6,638	0,002
Paciente 16	0,732	0,786	0,654	0,772	6,946	0,002
Paciente 17	0,734	0,785	0,659	0,784	7,066	0,002
Paciente 18	0,753	0,776	0,719	0,803	8,859	0,002
Paciente 19	0,753	0,798	0,689	0,791	8,749	0,002
Paciente 20	0,731	0,776	0,665	0,775	6,873	0,002
Paciente 21	0,745	0,794	0,674	0,788	7,977	0,002
Paciente 22	0,723	0,763	0,665	0,774	6,399	0,002
Valor medio	0,738	0,772	0,688	0,783	7,528	0,002
Desvío Estándar	0,010	0,012	0,022	0,010	0,771	0,001

Tabla 7.3: Medidas de desempeño de los 22 clasificadores de árboles de decisión tipo CART entrenados utilizando como datos de validación las 16 características calculadas por paciente.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,744	0,781	0,690	0,788	7,919	0,002
Paciente 2	0,737	0,779	0,677	0,781	7,358	0,002
Paciente 3	0,747	0,781	0,697	0,798	8,215	0,002
Paciente 4	0,725	0,773	0,654	0,784	6,462	0,002
Paciente 5	0,739	0,772	0,691	0,790	7,596	0,003
Paciente 6	0,731	0,772	0,671	0,776	6,914	0,002
Paciente 7	0,743	0,768	0,706	0,792	7,977	0,002
Paciente 8	0,729	0,753	0,693	0,756	6,888	0,002
Paciente 9	0,725	0,763	0,669	0,769	6,518	0,002
Paciente 10	0,712	0,762	0,639	0,769	5,674	0,002
Paciente 11	0,725	0,759	0,676	0,775	6,582	0,002
Paciente 12	0,738	0,792	0,661	0,789	7,405	0,002
Paciente 13	0,747	0,792	0,683	0,799	8,196	0,002
Paciente 14	0,732	0,770	0,678	0,785	7,029	0,002
Paciente 15	0,728	0,770	0,669	0,788	6,738	0,002
Paciente 16	0,731	0,798	0,633	0,775	6,816	0,002
Paciente 17	0,736	0,785	0,665	0,776	7,245	0,002
Paciente 18	0,744	0,786	0,682	0,789	7,878	0,002
Paciente 19	0,743	0,808	0,648	0,778	7,761	0,002
Paciente 20	0,733	0,786	0,655	0,777	7,003	0,002
Paciente 21	0,740	0,786	0,674	0,791	7,614	0,002
Paciente 22	0,721	0,765	0,656	0,763	6,228	0,002
Valor medio	0,734	0,777	0,671	0,781	7,151	0,0020
Desvío Estándar	0,009	0,014	0,019	0,011	0,675	0,0002

Tabla 7.4: Medidas de desempeño de los 22 clasificadores de árboles de decisión tipo CHAID entrenados utilizando como datos de validación las 16 características calculadas por paciente.

En particular el entrenamiento de los AD tipo CART y CHAID permite también analizar, en base al criterio de división y a medida que se van construyendo los distintos árboles, la importancia de las características, es decir, aquellas que tienen mayor peso según los datos de entrenamiento. El resultado se presenta en las figuras 7.11 para CART y 7.12 para CHAID, los que serán retomados en el capítulo 9 de selección de características (sección 9.5.1) donde se aborda el tema desde diversas perspectivas.

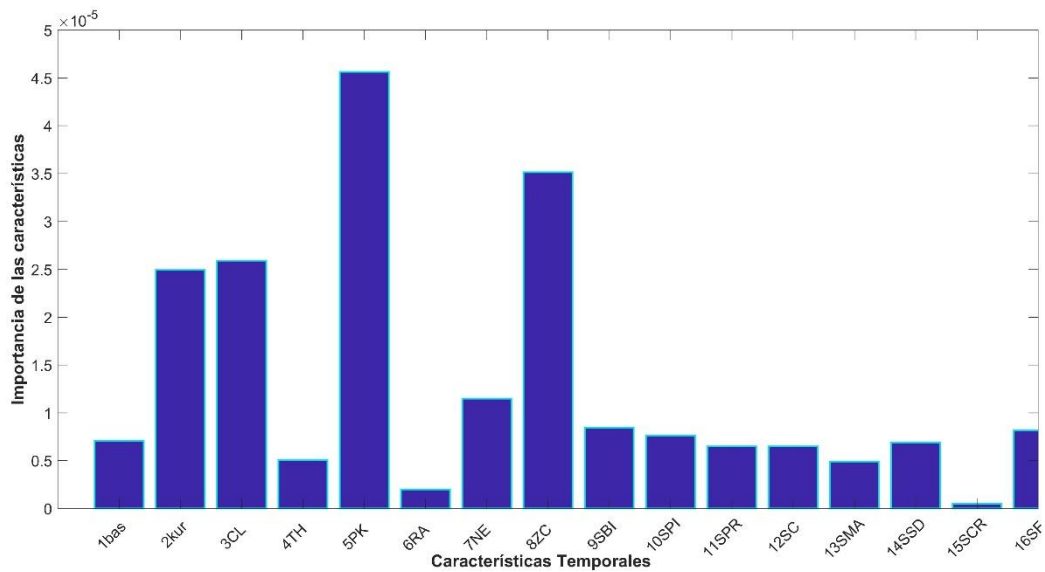


Figura 7.11: Importancia promedio de las características temporales que mayor peso tienen en el algoritmo CART a partir de los datos de entrenamiento.

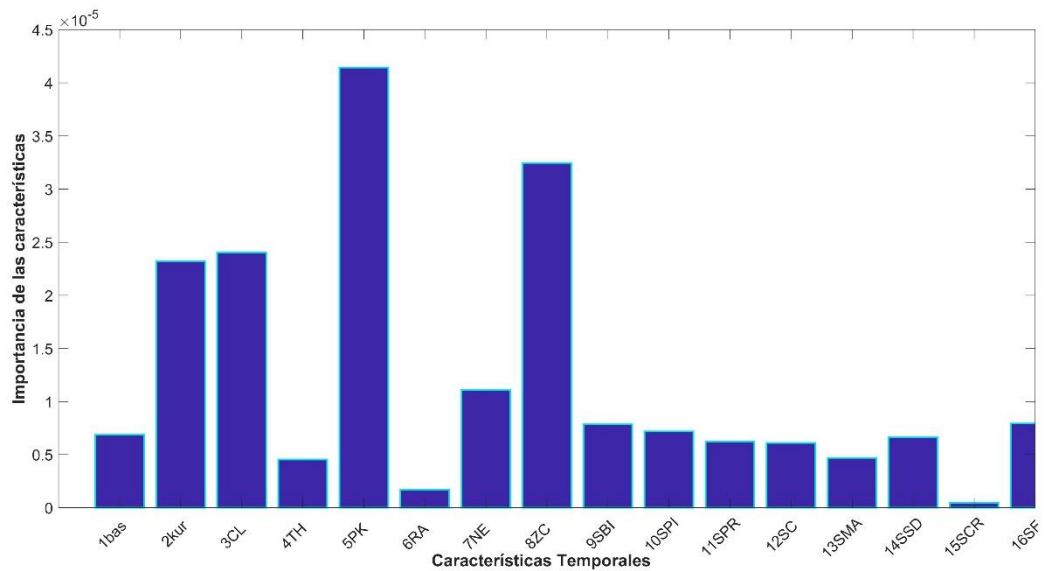


Figura 7.12: Importancia promedio de las características temporales que mayor peso tienen en el algoritmo CHAID a partir de los datos de entrenamiento.

Se aplicó un algoritmo de poda tanto para CART como para CHAID con la idea de indagar si es posible obtener árboles con menor cantidad de ramas con similar exactitud (ACC). Se

aplicó algoritmo de poda por nivel en base a criterio de error, realizando podas de nivel al 25%, 50% y 75%. Los resultados de los índices de desempeño para el AD sin podar y las podas propuestas se presentan en la tabla 7.5 y 7.6 para CART y CHAID respectivamente.

	ACC	ESP	SEN	DOR	tiempo validación [s]
CART sin podar	0,738±0,010	0,772±0,012	0,688±0,022	7,528±0,771	0,0020±0,0002
CART podados al 75%	0,716±0,013	0,772±0,012	0,656±0,027	6,00±0,760	0,0006±0,0001
CART podados al 50%	0,732±0,090	0,774±0,010	0,672±0,016	7,07±0,660	0,0008±0,0002
CART podados al 25%	0,741±0,090	0,781±0,012	0,683±0,019	7,74±0,683	0,0012±0,0003

Tabla 7.5: Índices de desempeño promedio (ACC, ESP, SEN y DOR) y tiempo de validación promedio obtenidos con los datos de validación en los 22 modelos CART sin podar y con podas de nivel al 25%, 50% y 75%.

	ACC	ESP	SEN	DOR	tiempo validación [s]
CHAID sin podar	0,734±0,009	0,777±0,014	0,670±0,019	7,151±0,675	0,00200±0,00020
CHAID podados al 75%	0,7160±0,010	0,777±0,014	0,680±0,028	6,034±0,799	0,00060±0,00005
CHAID podados al 50%	0,734±0,010	0,777±0,012	0,670±0,018	7,213±0,779	0,00080±0,00005
CHAID podados al 25%	0,741±0,009	0,784±0,011	0,650±0,015	7,765±0,725	0,00120±0,00010

Tabla 7.6: Índices de desempeño promedio (ACC, ESP, SEN y DOR) y tiempo de validación promedio obtenidos con los datos de validación en los 22 modelos CHAID sin podar y con podas de nivel al 25%, 50% y 75%.

La tabla 7.7 muestra los tiempos de entrenamiento y validación promedio para CART, CHAID y ambos algoritmos con poda del 25% dado que esta poda resultó arrojar el valor más

alto de ACC según tablas 7.5 y 7.6. Para poder realizar la poda es necesario contar con el árbol completo por lo que el tiempo de entrenamiento de los modelos podados es la suma de los tiempos de entrenamiento del árbol sin podar más el que demandó el algoritmo de poda.

	CART	CART podados al 25%	CHAID	CHAID podados al 25%
Tiempo de Entrenamiento promedio [s]	0,282 ±0,027	0,288±0,027	0,340±0,024	0,346±0,025

Tabla 7.7: Tiempos promedio para el entrenamiento y la validación con CART, CHAID y ambos algoritmos con poda del 25%.

Se observa que con el proceso de poda del 25% es el que mejora levemente en promedio todos los indicadores de desempeño tanto para CART como para CHAID, en ambos casos con una sustancial mejora en el tiempo de validación el cual se reduce un 40% en promedio.

7.5 ANÁLISIS ESTADÍSTICO Y DE RESULTADOS DE LOS CLASIFICADORES DEL PRESENTE CAPÍTULO

Se realizaron las pruebas estadísticas según se describió en la sección 6.2.1 para comparar el desempeño de los modelos que por grupo se detallan en la tabla 7.8 considerando los indicadores de desempeño exactitud (ACC), especificidad (ESP), sensibilidad (SEN), índice de capacidad de diagnóstico (DOR) y los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val.).

Grupos	Clasificadores a comparar estadísticamente
GRUPO 1	CART sin podar, CART podado 25%, CART podado 50% y CART podado 75%
GRUPO 2	CHAID sin podar, CHAID podado 25%, CHAID podado 50% y CHAID podado 75%
GRUPO 3	KNN, seleccionado del Grupo 1 y seleccionado del Grupo 2

Tabla 7.8: Clasificadores que integran cada grupo que se estudió estadísticamente.

Para el grupo 1: el test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores por lo que se procedió a realizar Nemenyi como test *post hoc* para la comparación

de a pares de clasificadores. Los resultados completos se presentan en la tabla 7.9 y las figuras 7.13 y 7.14 representan la comparación de rangos para ACC y ESP respectivamente.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
CART vs CART_Pod25	0,816	0,004	0,871	0,897	<0,01	0,050
CART vs CART_Pod50	0,571	0,852	0,188	0,573	<0,01	<0,01
CART vs CART_Pod75	<0,01	1,000	0,005	<0,01	0,050	<0,01
CART_Pod25 vs CART_Pod50	0,133	0,045	0,605	0,194	0,050	0,050
CART_Pod25 vs CART_Pod75	<0,01	0,004	0,048	<0,01	<0,01	<0,01
CART_Pod50 vs CART_Pod75	<0,01	0,852	0,530	<0,01	0,050	0,050

Tabla 7.9: Comparación estadística de a pares de clasificadores del grupo 1 como resultado de los test de Friedman y Nemenyi. Valores en negrita indican significancia estadística para $\alpha=0,05$.

El clasificador tipo CART sin podar presentó una exactitud promedio con el grupo de validación de $0,738\pm 0,01$, el mismo algoritmo con una poda del 25% arrojó una exactitud promedio $0,741\pm 0,09$ y el CART con poda del 50% de $0,732\pm 0,090$. Por lo que la ACC, tomando como referencia CART es levemente superior en el algoritmo con poda del 25% (ver figura 7.13) y levemente inferior en el algoritmo con poda del 50%, sin representar en ambos casos diferencias estadísticamente significativa según se observa en la tabla 7.9 ($p=0,816$ y $p=0,571$ respectivamente). Los indicadores SEN y DOR para la comparación de CART y CART podado al 25% no indicaron diferencias estadísticamente significativas ($p=0,871$ y $p=0,897$ respectivamente). Lo mismo ocurrió cuando se comparó SEN y DOR para CART y su versión con poda del 50% ($p=0,188$ y $p=0,573$ respectivamente). Sin embargo, para ESP la diferencia fue significativa a favor del algoritmo con poda del 25% al compararlo con CART (ver figura 7.14) con $p=0,004$. CART presentó una curva ROC aceptable (ver figura 7.10 superior) con un área bajo la curva promedio de $0,783\pm 0,01$.

El tiempo de validación promedio de CART fue de $0,0020\pm 0,0002$ s y el de CART con poda del 25% de $0,0012\pm 0,0005$ s, resultando una diferencia con valor $p=0,05$ al límite de significancia estadística. No existió una diferencia significativa entre los tiempos de validación de CART con poda del 25% y CART con poda del 50% ($p=0,05$).

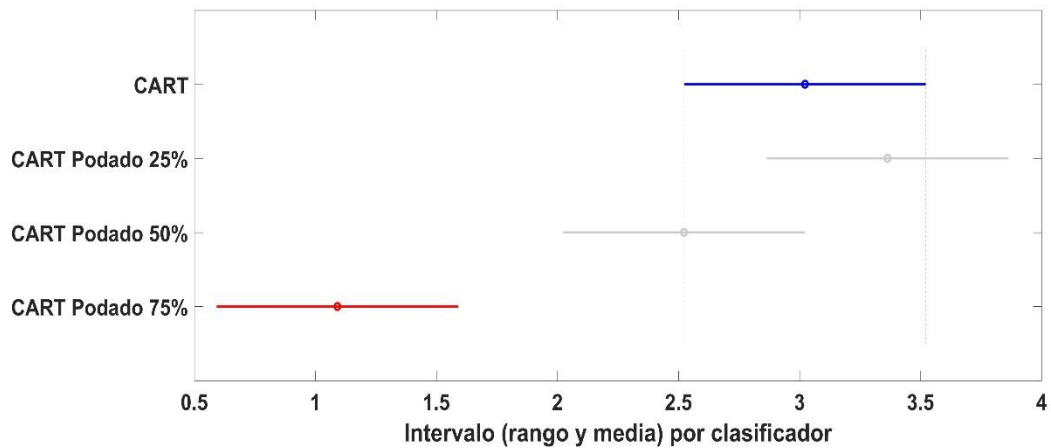


Figura 7.13: Comparación de clasificadores del grupo 1 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, las líneas grises los clasificadores sin deferencia significativa con CART, la línea roja indica diferencia significativa de CART podado al 75% con CART.

En función de lo antes detallado podemos asumir que el modelo de clasificación **CART con poda del 25%** tuvo estadísticamente el mejor rendimiento relativo de este grupo.

Para el grupo 2: el test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 7.10 y las figuras 7.15 y 7.16 representan la comparación de rangos para ACC y tiempo de validación.

El clasificador tipo CHAID sin podar presentó una exactitud promedio con el grupo de validación de $0,734 \pm 0,009$, el mismo algoritmo con una poda del 25% arrojó una exactitud promedio $0,741 \pm 0,009$ y el CHAID con poda del 50% de $0,734 \pm 0,010$.

La ACC, tomando como referencia CHAID, resultó levemente superior en el algoritmo con poda del 25% (ver figura 7.15) e igual en el algoritmo con poda del 50%, sin representar en ambos casos diferencias estadísticamente significativas según se observa en la tabla 7.10 ($p = 0,057$ y $p = 0,918$ respectivamente).

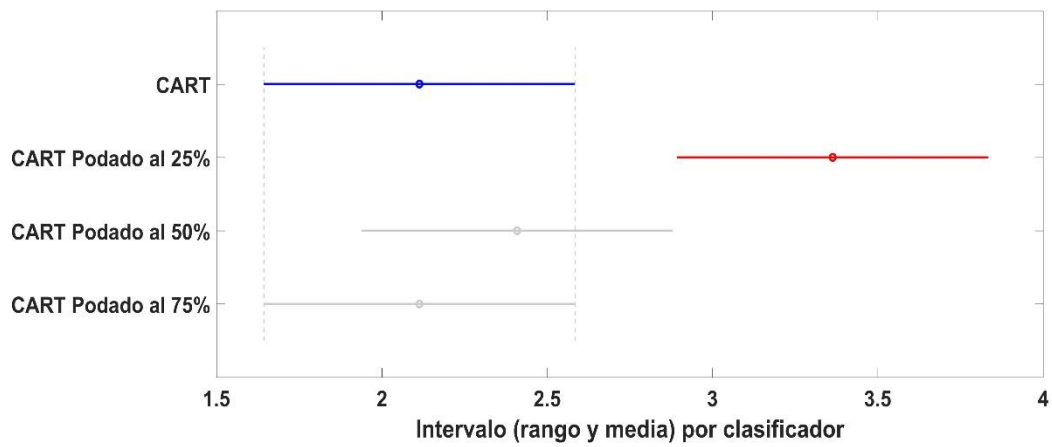


Figura 7.14: Comparación de clasificadores del grupo 1 para la especificidad (ESP) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, las líneas grises los clasificadores sin diferencia significativa con CART, la línea roja indica diferencia significativa de CART podado al 25% con CART.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
CHAID vs CHAID_Pod25	0,057	0,265	0,162	0,050	<0,01	0,152
CHAID vs CHAID_Pod50	0,918	0,989	0,934	0,996	<0,01	<0,01
CHAID vs CHAID_Pod75	<0,01	1,000	0,206	<0,01	0,050	<0,01
CHAID_Pod25 vs CHAID_Pod50	<0,01	0,143	0,449	0,026	0,050	0,018
CHAID_Pod25 vs CHAID_Pod75	<0,01	0,265	<0,01	<0,01	<0,01	<0,01
CHAID_Pod50 vs CHAID_Pod75	<0,01	0,989	0,053	<0,01	0,050	0,242

Tabla 7.10: Comparación estadística de a pares de clasificadores del grupo 2 como resultado de los test de Friedman y Nemenyi. Valores en negrita indican significancia estadística para $\alpha = 0,05$.

Los indicadores ESP, SEN y DOR para la comparación de CHAID y CHAID podado al 25% no indican diferencias estadísticamente significativas ($p = 0,265$, $p = 0,162$ y $p = 0,050$ respectivamente). Lo mismo ocurrió cuando se comparó ESP, SEN y DOR y su versión con poda del 50% ($p = 0,989$, $p = 0,934$ y $p = 0,996$ respectivamente). CHAID podado al 25% y el podado al 50% presentaron diferencias muy significativa para ACC ($p = 0,008$) y significativa

en DOR ($p=0,026$), como se observa en la figura 7.15 y tabla 7.10. CHAID presentó una curva ROC aceptable (ver figura 7.10 inferior) con un área bajo la curva promedio de $0,781\pm 0,010$.

El tiempo de validación promedio de CHAID fue de $0,0020\pm 0,0002$ s, el de CHAID con poda del 25% de $0,0012\pm 0,0005$ s y el de CHAID con poda del 50% de $0,0008\pm 0,0001$ s, resultando una diferencia muy significativa a favor del CHAID con poda del 50% ($p<0,01$) el que insumió un tiempo promedio de validación menor al 50% que CHAID con poda del 25%. Se observa asimismo una diferencia significativa entre los tiempos de validación de CHAID con poda del 25% y CHAID con poda del 50% ($p=0,018$) a favor de este último.

En base a lo detallado en el párrafo anterior existe una contraposición para la selección de un clasificador en este grupo, ya que CHAID con poda del 25% presenta mejor exactitud y CHAID con poda del 50% menor tiempo de entrenamiento, en ambos casos validados estadísticamente. La mejora porcentual de la exactitud promedio a favor del CHAID con poda del 25% es menor al 1% y la mejora porcentual promedio del tiempo de validación a favor del CHAID con poda del 50% es del 50%. Por lo tanto, se decide optar por **CHAID con poda del 50%** como el que tiene mejor rendimiento relativo de este grupo.

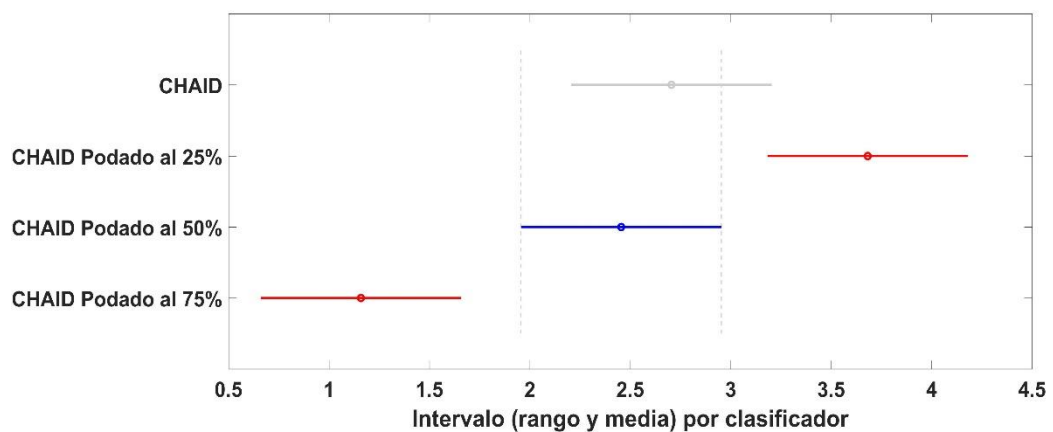


Figura 7.15: Comparación de clasificadores del grupo 2 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p<0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, la línea gris el clasificadores sin deferencia significativa con CHAID podado al 50%, las líneas rojas indican diferencias significativas de esos clasificadores con CHAID podado al 50%.

Para el grupo 3: se comparó el desempeño relativo mediante un análisis estadístico para KNN, CART con poda del 25% y CHAID con poda del 50%. El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 7.12 y las figuras 7.17 y 7.18 representan la comparación de rangos para ACC y tiempo de validación respectivamente.

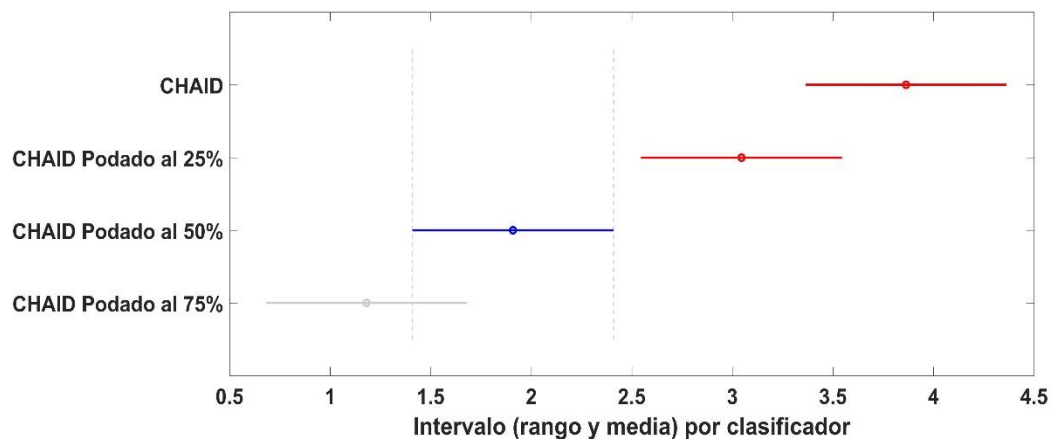


Figura 7.16: Comparación de clasificadores del grupo 2 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, la línea gris el clasificador sin deferencia significativa con CHAID podado al 50%, las líneas rojas indican diferencias significativas de CHAID y su versión podada al 25% con CHAID podado al 50%.

La exactitud promedio de los clasificadores del grupo 3 fue: KNN de $0,751 \pm 0,013$, el algoritmo CART con una poda del 25% de $0,741 \pm 0,090$ y CHAID con poda del 50% de $0,734 \pm 0,010$. KNN presentó mejor exactitud con diferencias estadísticamente muy significativas ($p < 0,01$) al compararlo con la ACC de CART podado al 25% y CHAID podado al 50%, como puede observarse de la tabla 7.12 y la figura 7.17. En los demás indicadores de desempeño KNN presentó ventajas significativas o muy significativas, excepto para la sensibilidad con CART podado al 25% ($p = 0,631$). KNN presentó una curva ROC (ver figura 7.7) con una buena área bajo la curva, cuyo valor promedio fue de $0,816 \pm 0,013$.

	Valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
KNN vs CART_Pod25	<0,01	0,631	0,027	<0,01	<0,01	<0,01
KNN vs CHAID_Pod50	<0,01	0,032	<0,01	<0,001	<0,01	<0,01
CART_Pod25 vs CHAID_Pod50	0,542	0,245	0,320	0,542	<0,01	<0,01

Tabla 7.12: Comparación estadística de a pares de clasificadores del grupo 3 como resultado de los test de Friedman y Nemenyi. Valores en negrita indican significancia estadística para $\alpha=0,05$.

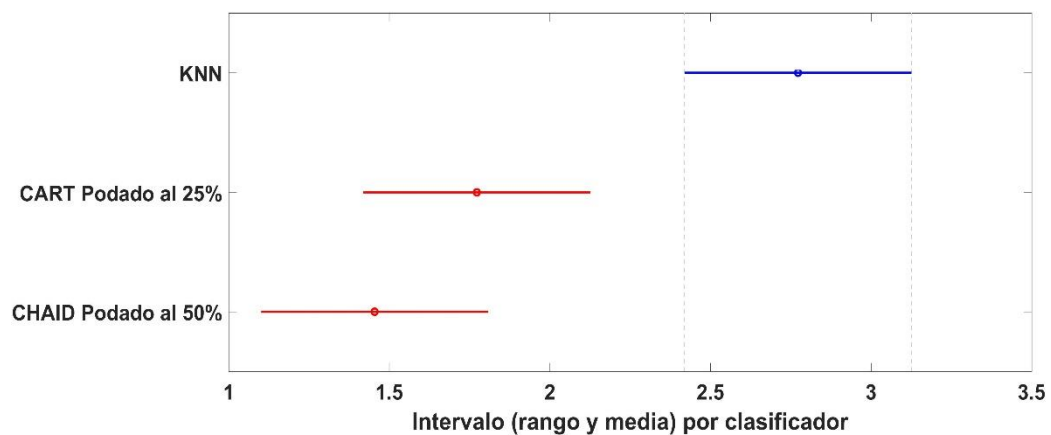


Figura 7.17: Comparación de clasificadores del grupo 3 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p<0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, las líneas rojas indican diferencias significativas de los otros clasificadores del grupo con KNN.

CART podado al 25% y CHAID podado al 50% no presentaron diferencias significativas al comparar todos los indicadores de desempeño (valor p de ACC=0,542; valor p de ESP=0,245; valor de p de SEN=0,320 y valor de p de DOR=0,542). El tiempo de validación promedio de KNN fue de $0,1860\pm 0,0180$ s, el de CART con poda del 25% de $0,0012\pm 0,0005$ s y el de CHAID con poda del 50% de $0,0008\pm 0,0001$ s, resultando una diferencia muy significativa ($p<0,01$) a favor de este último. Si se analizan los tiempos de entrenamiento, KNN es el más rápido con diferencias muy significativas ($p<0,01$) comparado con los otros clasificadores del grupo 3.

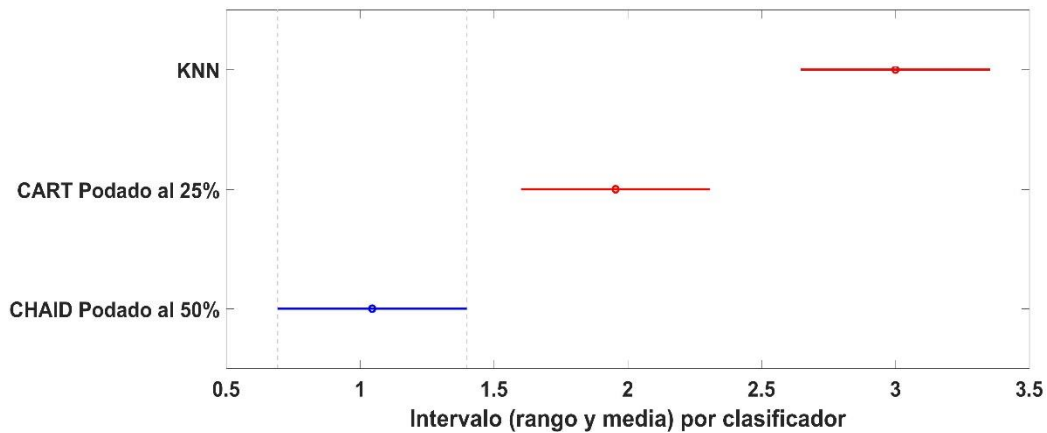


Figura 7.18: Comparación de clasificadores del grupo 2 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, las líneas rojas indican diferencias significativas de los otros clasificadores del grupo con CHAID podado al 50%.

Del grupo 3 se tiene entonces a KNN con un desempeño estadísticamente mejor a los otros clasificadores y por otro lado CHAID con poda del 50% que presenta el menor tiempo de validación de datos. Al analizar los valores promedio de los indicadores de desempeño para los 3 clasificadores de este grupo la variación porcentual como máximo entre el mejor y el peor es del 6%, lo que no representa funcionalmente una superioridad de uno sobre otro. Considerando que los tiempos promedio de validación que se presentan a lo largo de la tesis corresponden a un paciente completo, se evidencia que el tiempo de clasificación de todos los algoritmos analizados en el presente capítulo, aún los más lentos, podrían utilizarse para un adecuado funcionamiento del sistema de clasificación durante una cirugía. Por lo que inicialmente se propone considerar a **KNN** y **CHAID con poda del 50%** como los clasificadores individuales del presente capítulo con mejor rendimiento bajo las condiciones experimentales de la presente tesis.

Capítulo 8 Experiencia 2: Clasificación de registros MER mediante combinación de clasificadores para la detección del STN

En el capítulo anterior, se han estudiado los clasificadores individuales que arrojaron mayor exactitud global sobre el conjunto de datos utilizados en la presente tesis. El error de generalización de estos clasificadores individuales en la suma de tres términos no negativos [40]: el riesgo de Bayes, el sesgo y la varianza. El primero de ellos, manifiesta el error inherente en un conjunto de datos. El sesgo mide el error persistente del método de clasificación, es decir, el error que se mantendría incluso si se tuviese un conjunto infinito de clasificadores entrenados de manera independiente. El término correspondiente a la varianza mide el error debido a las fluctuaciones que se producen al generar un clasificador individual.

La inestabilidad de un clasificador representa los cambios que se producen en el clasificador ante pequeños cambios en el conjunto de entrenamiento, si el clasificador sufre grandes cambios, se dice que es inestable [44]. En general, se consideran algoritmos más estables el análisis discriminante lineal y el método de vecinos más cercanos, mientras que los árboles de decisión y las redes neuronales son considerados más inestables.

La agregación de modelos consigue clasificadores más estables, basándose en la idea de combinar varios modelos, cada uno de los cuales resuelve la misma tarea original, con el fin de obtener un mejor modelo global compuesto, con estimaciones o decisiones más precisas y confiables que las que se pueden obtener usando un solo modelo [40].

Dietterich (2000) [55] justifica el mejor desempeño de clasificadores combinados sobre los individuales por tres aspectos. La razón estadística se basa en la eliminación del riesgo asociado a la elección de un clasificador frente al resto. La razón computacional plantea la posibilidad de que los sistemas de clasificación que realizan una búsqueda local, puedan quedar atrapados en un óptimo local y verse imposibilitados para alcanzar el óptimo global. Finalmente, la razón de representación aduce la imposibilidad de representar la verdadera función con las hipótesis disponibles.

Asimismo se debe considerar la perspectiva bayesiana, que también defiende la agregación de modelos cuando el objetivo no es la selección explícita de un modelo, sino extraer el mejor rendimiento posible al conjunto de datos disponible para la predicción que se desea realizar [40].

En los últimos años, los estudios experimentales realizados por la comunidad científica dedicada al aprendizaje automático han demostrado que la combinación de múltiples clasificadores reduce el error de generalización (Domingos, 1996; Quinlan, 1996; Bauer y Kohavi, 1999; Opitz y Maclin, 1999) [40], [56]. Los métodos combinados son muy efectivos, principalmente debido al fenómeno de que varios tipos de clasificadores tienen diferentes "sesgos inductivos" (Geman y col., 1995; Mitchell, 1997). De hecho, los métodos de conjunto pueden utilizar de manera efectiva dicha diversidad para reducir el error de varianza (Turner y Ghosh, 1999; Ali y Pazzani, 1996) sin aumentar el error de sesgo.

Este capítulo aborda los métodos de combinación para la función de clasificación si bien se ha demostrado (Dimitriadou y col., 2003) [40] que también se pueden utilizar para mejorar la exactitud y robustez de los algoritmos de *clustering*.

En los últimos años se han desarrollado numerosos algoritmos para la combinación de clasificadores y existen diversos factores que los caracterizan, los cuales se describen a continuación:

1.- Relación entre los clasificadores: ¿Cómo afecta cada clasificador a los otros clasificadores? Los métodos combinados se pueden dividir en dos tipos principales: secuenciales y concurrentes.

2. Método de combinación: la estrategia de combinar los clasificadores es generada por un algoritmo de inducción. El combinador más simple determina la salida únicamente a partir de las salidas de los inductores individuales. Ali y Pazzani (1996) han comparado varios métodos combinados: votación uniforme, combinación Bayesiana, suma de distribución y combinación de verosimilitud. Además, se ha desarrollado un análisis teórico para estimar la mejora de la clasificación (Turner y Ghosh, 1999). Junto con los combinadores simples existen otros métodos más sofisticados, como el apilamiento o *stacking* (Wolpert, 1992) y el arbitraje (Chan y Stolfo, 1995) [40].

3. Diversidad: para hacer que el conjunto sea eficiente, debería haber algún tipo de diversidad entre los clasificadores. La diversidad se puede obtener a través de diferentes conjuntos de los datos de entrada, variaciones en el diseño del aprendizaje, o agregando una penalización a las salidas [44].

4. El número de clasificadores que conforman la combinación.

8.1 METODOLOGÍA SECUENCIAL O SERIE

En los enfoques secuenciales para el aprendizaje de clasificadores combinados, existe una interacción entre las secuencias de aprendizaje. Por lo tanto, es posible aprovechar el conocimiento generado en iteraciones previas para guiar el aprendizaje en las siguientes iteraciones. Se distinguen dos enfoques principales para el aprendizaje secuencial, como se describe en las siguientes secciones (Provost y Kolluri, 1997).

8.1.1 MODELOS GUIADOS POR LA SELECCIÓN DE INSTANCIAS

En este enfoque secuencial, los clasificadores que se construyeron en iteraciones previas se usan para manipular el conjunto de entrenamiento para la siguiente iteración. Uno puede incorporar este proceso dentro del algoritmo de aprendizaje básico. Estos métodos, que también se conocen como métodos constructivos o conservadores, generalmente ignoran todas las instancias de datos en las que su clasificador inicial es correcto y solo aprenden de instancias mal clasificadas [40].

8.1.1.1 *Muestreo de incertidumbre.*

Este método es útil en escenarios donde los datos no etiquetados son abundantes y el proceso de etiquetado es costoso. Se puede definir el muestreo de incertidumbre como un proceso iterativo de etiquetado manual de ejemplos, ajuste del clasificador a partir de esos ejemplos, y el uso del clasificador para seleccionar nuevos ejemplos cuya pertenencia a la clase no está clara (Lewis y Gale, 1994) [57].

Se ha demostrado que el uso del método de muestreo de incertidumbre en las tareas de categorización de texto puede reducir en un factor de hasta 500 la cantidad de datos que deben etiquetarse para obtener un determinado nivel de exactitud (Lewis y Gale, 1994) [57]. El muestreo de incertidumbre simple requiere la construcción de muchos clasificadores, por lo que es imprescindible contar con un clasificador que implique un bajo costo computacional y reducido tiempo de ejecución. Este clasificador selecciona instancias y luego usa esas instancias para entrenar a otro inductor de mayor costo computacional.

Desafortunadamente, un muestreo de incertidumbre tiende a crear un conjunto de entrenamiento que contiene una cantidad desproporcionadamente grande de alguna de las clases. Para equilibrar este efecto, se desarrolló una versión modificada de un árbol de decisión

C4.5 (Lewis y Catlett, 1994) [57]. Este algoritmo acepta un parámetro llamado tasa de pérdida (LR por sus siglas en inglés). El parámetro especifica el costo relativo de dos tipos de errores: falsos positivos (donde la instancia negativa se clasifica como positiva) y falsos negativos (donde la instancia positiva se clasifica como negativa). Elegir una razón de pérdida mayor que uno indica que los errores de falsos positivos son más costosos que los falsos negativos. Por lo tanto, establecer LR mayor a 1 compensará la sobrerrepresentación de instancias positivas. Elegir el valor exacto de LR requiere un análisis de sensibilidad del efecto del valor específico sobre la exactitud del clasificador producido [57].

El C4.5 original determina el valor de clase en las hojas al verificar si la división disminuye la tasa de error. El valor final de la clase está determinado por el voto mayoritario. En un C4.5 modificado, la clase de la hoja se determina por comparación con un umbral de probabilidad de $LR / (LR + 1)$ (o su recíproco apropiado). Lewis y Catlett (1994) muestran que su método conduce a una exactitud significativamente mayor que en el caso del uso de muestras aleatorias diez veces más grandes [57].

8.1.1.2 *Boosting*

Boosting es un método general para mejorar el rendimiento de cualquier algoritmo de aprendizaje. El método funciona ejecutando repetidamente un clasificador individual débil (como árboles de decisión) en varios datos de entrenamiento distribuidos. Los clasificadores producidos se combinan en un único clasificador compuesto con el fin de lograr una mayor exactitud [40]. En general se utilizan clasificadores individuales inestables como árboles de decisión ya que se ha demostrado que en estos casos *boosting* realmente mejora la exactitud general [55].

Schapiro introdujo el primer algoritmo *boosting* en 1990 el cual sólo trabaja con 3 clasificadores $C_i(x)$, el primero de ellos se entrena con un sub conjunto del conjunto total de entrenamiento, el segundo también se entrena con otro sub conjunto pero éste debe incluir la mitad de las muestras que fueron mal clasificadas por el primer clasificador. El tercer clasificador se entrena con las muestras en las cuales el clasificador 1 y 2 no tienen concordancia. Finalmente el clasificador final se combinaba de acuerdo a la ecuación 8.1 [44], donde *sgn* es la función signo.

$$H(x) = \text{sgn} \left[\sum_{i=1}^3 C_i(x) \right] \quad (8.1)$$

Luego en 1996, Freund y Schapire [58] introdujeron el algoritmo AdaBoost (por su denominación en inglés de *Adaptative Boosting*). La idea principal de este algoritmo es asignar un peso a cada conjunto de entrenamiento. Inicialmente todos los pesos son iguales, pero en cada iteración, los pesos de todas las posiciones mal clasificadas aumentan mientras que los pesos de las instancias correctamente clasificadas se reducen. Como consecuencia, el aprendizaje se adapta en base a las instancias difíciles del clasificar del conjunto de entrenamiento. Este procedimiento proporciona una serie de clasificadores que se complementan entre sí. Adaboost puede ser binario o multiclase.

En la figura 8.1 podemos observar gráficamente como es el principio general del algoritmo Adaboost [58]. El pseudocódigo del algoritmo del mismo se describe en la figura 8.2. El algoritmo supone que el conjunto de entrenamiento S consta de m instancias con características denominadas x siendo y la etiqueta correspondiente con valores $\{-1, 1\}$. La clasificación de una nueva instancia se realiza por el voto mayoritario de todos los clasificadores C_t , cada uno con un peso diferente α_t según la ecuación 8.2 [58].

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t \cdot C_t(x) \right) \quad (8.2)$$

En el ejemplo representado en la figura 8.3 se construye un clasificador $H(x)$ por AdaBoost a partir de 3 clasificadores individuales C_1 , C_2 y C_3 y un conjunto de validación S con 10 muestras. Inicialmente los pesos para cada elemento de S son todos iguales ($D_1=0,1$). Luego los datos son testeados en el primer clasificador C_1 resultando 3 muestras mal clasificadas (indicadas con un círculo en la gráfica de C_1) por lo que el error es la sumatoria de los pesos individuales de cada uno de esos datos ($\alpha_1=0,3$). Se procede a continuación a recalcular los pesos ($D_2(x)$) y normalizarlos para todos los datos resultando un incremento del valor a los que fueron erróneamente clasificados (pasan de 0,1 a 0,166) y una disminución a los que fueron bien clasificados (pasan de 0,1 a 0,07). De esta forma, si el siguiente clasificador C_2 vuelve a clasificar mal los datos erróneamente clasificados por su antecesor tendrá un mayor error por ese dato (y una disminución de su α) y si los clasifica bien tendrá menor error. Este proceso se repite para C_3 para finalmente calcular $H(x)$ como un clasificador combinado según la ecuación

8.2. Como se puede observar en la figura 8.3 el clasificador individual que tuvo menor error es el que tiene mayor peso y viceversa.

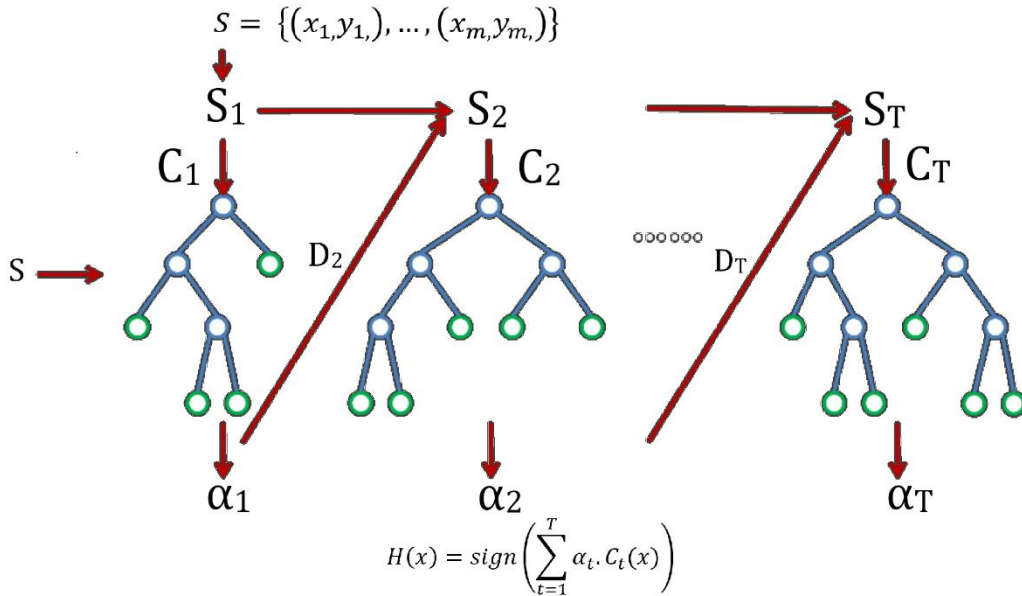


Figura 8.1: Esquema gráfico de la forma de trabajo del algoritmo Adaboost.

ENTRADA: I (algoritmo de clasificación), T (número de iteraciones), S (datos de entrenamiento)

SALIDA: C_t y α_t con $t = 1, \dots, T$

$t=1$

$$D_1(i) = \frac{1}{m}, \text{ con } i = 1, \dots, m$$

repeat

Construir el clasificador C_t utilizando I y la distribución D_t

$$\varepsilon_t = \sum_{i: C_t(x_i) \neq y_i} D_t(i)$$

if $\varepsilon_t > 0,5$ **then**

$$T = t - 1$$

exit loop

end if

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

$$D_{t+1}(i) = D_t(i) e^{-\alpha_t y_t C_t(x_i)}$$

Normalizar D_{t+1}

Incrementar t

until $t > T$

Figura 8.2: Pseudocódigo del algoritmo Adaboost.M1.

El proceso de clasificación $H(x)$ se indica en la parte inferior de la figura 8.3 con un ejemplo, resultando un valor negativo de -0,69 por lo que el resultado de la función signo será -1 y el dato es correctamente clasificado como negativo.

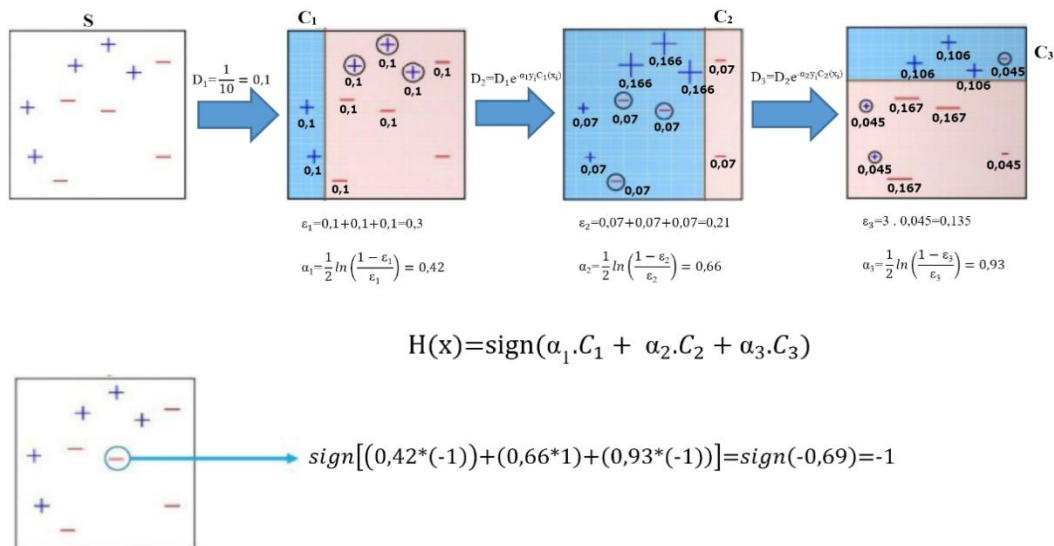


Figura 8.3: Ejemplo de la forma de cálculo y clasificación del algoritmo AdaBoost.

El algoritmo básico de AdaBoost aborda la clasificación binaria, Freund y Schapire (1996) describen dos versiones del algoritmo AdaBoost: AdaBoost.M1 para 2 clases (binario) y AdaBoost.M2 para problemas de clasificación multiclase [58].

Boosting mejora el rendimiento de clasificación por dos razones principales:

1. Genera un clasificador final cuyo error en el conjunto de entrenamiento es pequeño al combinar muchas hipótesis cuyo error puede ser grande.

2. Produce un clasificador combinado cuya varianza es significativamente más baja que las producidas por el clasificador individual.

Por otro lado, *boosting* a veces presenta inconvenientes en la generalización del modelo. Según Quinlan (1996) [56], la razón principal de este problema es el *overfilling*. El objetivo de *boosting* es construir un clasificador compuesto que rinda bien en los datos, pero un gran número de iteraciones puede crear un clasificador compuesto muy complejo, que es significativamente menos preciso que un clasificador único. Una forma posible de evitar el *overfilling* es manteniendo el número de iteraciones lo más pequeño posible [56].

Otro inconveniente de *boosting* es que el conjunto resultante se considera menos comprensible o entendible ya que el usuario debe capturar varios clasificadores en lugar de un solo clasificador [56]. A pesar de los inconvenientes anteriores, Breiman (1996) se refiere a la idea de *boosting* como el desarrollo más significativo en el diseño clasificador de los años noventa.

Existen muchas versiones del algoritmo Adaboost además de AdaBoost.M1 y AdaBoost.M2, entre las más utilizadas en el mundo científico se encuentran [40]:

a) *Adaptive logistic regression* (LogitBoost): trabaja en clasificación binaria en forma similar a AdaBoost.M1 excepto que minimiza la desviación binomial. Se asigna menos peso a las observaciones mal clasificadas según la ecuación 8.3. LogitBoost puede proporcionar una precisión promedio mejor que AdaBoostM1 para datos con clases poco separables.

$$D_{t+1}(i) = D_t(i) \ln(1 + e^{-2 \cdot y_t c_t(x_i)}) \quad (8.3)$$

b) *Gentle adaptive boosting* (GentleBoost): este algoritmo combina las características de AdaBoost.M1 y LogitBoost. Al igual que AdaBoost.M1 minimiza los errores exponencialmente. Pero su optimización numérica está configurada de manera diferente ya que al igual que LogitBoost, cada clasificador débil se ajusta a un modelo de regresión según los valores de la respuesta asignado. Esto hace que GentleBoost sea un buen candidato para la clasificación binaria de datos con características categóricas multinivel.

c) *Random undersampling boosting* (RUSBoost): este algoritmo se utiliza tanto para clasificación binaria como en multiclase. Es especialmente eficaz para clasificar datos con clases desbalanceadas. RUS significa muestreo aleatorio (*Random Under Sampling* por su denominación en inglés). El algoritmo calcula un valor N que representa la cantidad de datos de la clase con el menor número de muestras de entrenamiento. Conformar sub conjuntos para todas las clases con un máximo de N observaciones. El procedimiento para establecer los pesos y construir el clasificador es igual a AdaBoost.M2.

d) *Gradient Boosting*: es una generalización del algoritmo AdaBoost el cual permite emplear cualquier función de coste, siempre que esta sea diferenciable. La flexibilidad de este algoritmo ha hecho posible aplicar *boosting* a multitud de problemas (regresión, clasificación con más de dos clases, etc). Para cada uno de ellos, el algoritmo de *Gradient Boosting* es

ligeramente distinto, pero, para todos, la idea es la misma: dada una función de coste (por ejemplo, residuos cuadrados para regresión) y un clasificador débil individual (por ejemplo, árboles), el algoritmo trata de encontrar el modelo que minimiza la función de coste. Suele iniciarse con la mejor aproximación de la variable respuesta (la media en el caso de regresión), se calculan los residuos y con ellos se ajusta un nuevo clasificador individual débil que intente minimizar la función de coste. Este proceso se repite sucesivamente de forma que cada nuevo modelo minimiza los residuos (errores) del anterior.

Dado que el objetivo de *Gradient Boosting* es minimizar los residuos iteración a iteración, es susceptible de *overfitting*. Una forma de evitar este problema es empleando un valor de regularización, también conocido como tasa de aprendizaje (λ) que limite la influencia de cada modelo en el conjunto combinado.

8.2 MÉTODOS CONCURRENTES O PARALELOS

En la metodología concurrente o paralela, el conjunto de datos original se divide en varios subconjuntos a partir de los cuales se inducen múltiples clasificadores simultáneamente. Los subconjuntos creados a partir del conjunto de entrenamiento original pueden ser disjuntos (mutuamente excluyentes) o superpuestos. Luego los métodos concurrentes aplican un procedimiento de combinación para producir una única clasificación para una instancia determinada [40]. Como el método para combinar los resultados de los clasificadores inducidos suele ser independiente de los algoritmos de inducción, se puede usar con diferentes inductores en cada subconjunto. Estos métodos tienen como objetivo mejorar el poder predictivo de los clasificadores o disminuir el tiempo total de ejecución. A continuación se describen los más relevantes y que han demostrado un desempeño robusto en una gran variedad de aplicaciones.

8.2.1 BAGGING

Uno de los problemas más habituales a la hora de establecer un clasificador para un conjunto de datos, es el tamaño limitado del conjunto de ejemplos de entrenamiento. Aunque este problema afecte especialmente a los métodos paramétricos, supone un reto para cualquier clasificador. Cuanto más pequeño sea el conjunto de datos disponibles, menos seguro se puede estar que este conjunto represente fielmente a la población total. En general, los clasificadores

construidos en conjuntos pequeños pueden estar sesgados y presentarán una elevada varianza en la probabilidad de clasificación errónea. Se dice en este caso que el clasificador es inestable.

En muchos casos no se puede disponer de más observaciones y, por tanto, el conjunto de entrenamiento está limitado. Existen diversas técnicas que intentan obtener clasificadores más estables, y actualmente éste es uno de los campos de investigación abiertos en el ámbito de los sistemas de clasificación.

Una posible solución es utilizar *bagging* (por la contracción gramatical del inglés *bootstrap aggregating*) que se basa en la idea de promediar un conjunto de observaciones para reducir la varianza y aumentar la exactitud [40]. Para esto es posible obtener múltiples muestras de la población, ajustar un modelo distinto T_n con cada una de ellas, y hacer la media (la moda en el caso de variables cualitativas) de las predicciones resultantes [59]. La figura 8.4 muestra un esquema gráfico de la forma de trabajo de *bagging* y en la figura 8.5 se presenta el pseudocódigo del algoritmo propuesto inicialmente por Breiman en 1996 [59].

Cada clasificador está entrenado con un conjunto muestreado con reemplazo del conjunto total de entrenamiento (*bootstrapping*). Dado que se utiliza el muestreo con reemplazo, algunas de las posiciones originales de S pueden aparecer más de una vez en S_n y algunas pueden no estar incluidas en absoluto. Entonces, los conjuntos de entrenamiento S_n son diferentes entre sí, pero ciertamente no son independientes.

Cuando se extrae un sub conjunto mediante *bootstrapping* del conjunto de entrenamiento inicial $S=\{(x_1, y_1), \dots, (x_m, y_m)\}$, la probabilidad de que la observación i -ésima x_i ($i=1, 2, \dots, m$) sea incluida n veces ($n=0, 1, 2, \dots, m$) en esa muestra *bootstrap*, S_n , vendrá dada por la distribución binomial $B(m, 1/m)$, donde $1/m$ es la probabilidad que tiene x_i de ser seleccionada en cada extracción y m es el número de extracciones con reposición que se efectúan, según la ecuación 8.4 [59].

$$P(m) = \binom{m}{n} \left(\frac{1}{m}\right)^n \left(1 - \frac{1}{m}\right)^{m-n} \quad (8.4)$$

Cuando $1/m < 0,1$, es decir, para conjuntos de más de diez observaciones, se puede aproximar la distribución binomial a través de la de Poisson, resultando la probabilidad según se indica en la ecuación 8.5.

$$P(n) = \frac{e^{-1}}{n!} \quad (8.5)$$

Si quiere obtener la probabilidad de que una observación no sea incluida en la muestra se hace $n=0$ en la ecuación 8.5, resultando aproximadamente un valor de $\frac{1}{e}=0,367$. Por lo tanto, se puede esperar que, en término medio, aproximadamente el 37% de las observaciones queden afuera de una muestra *bootstrap* [59]. De esta forma, las posibles observaciones ruidosas del conjunto de entrenamiento no aparecerán en algunas de esas muestras. En ese caso, el clasificador construido bajo esas condiciones obtendrá mejor tasa de error que el construido en el conjunto de entrenamiento original con observaciones ruidosas.

Para clasificar una nueva instancia se utiliza cada clasificador T_n que devuelve la predicción de clase para la instancia desconocida. El clasificador compuesto por *bagging*, $F(x)$, devuelve la clase que se ha predicho con mayor frecuencia mediante el método de votación mayoritaria [59].

Existen otras funciones de combinación final además de votación mayoritaria basadas en probabilidades a posteriori: mínimo, máximo, media, mediana y producto. Otra posibilidad, es promediar los coeficientes de los clasificadores básicos, para lo que se utiliza la función promedio.

Bagging produce un modelo combinado que en general ha demostrado mejor desempeño que un clasificador individual construido a partir de los datos totales de entrenamiento [40]. Breiman (1996) señala que esto es cierto especialmente para los inductores inestables porque *bagging* puede eliminar su inestabilidad [59]. En este contexto, un inductor se considera inestable si cambios el conjunto de aprendizaje pueden causar cambios significativos en el clasificador construido. Realizar *bagging* para clasificadores individuales estables como KNN no aporta mayor exactitud general al modelo combinado por lo que es generalmente utilizado con árboles de decisión [59].

En el proceso de *bagging*, el número de clasificadores individuales utilizados no es un parámetro crítico en cuanto a que, por mucho que se incremente el número, no se aumenta el riesgo de *overfitting* [60]. Alcanzado un determinado número de árboles, la reducción del error se estabiliza. A pesar de ello, cada clasificador individual ocupa memoria, por lo que no es conveniente más de los necesarios.

El proceso de *bagging* consigue mejorar la capacidad predictiva en comparación a los modelos basados en un único clasificador pero esto tiene un coste asociado: la interpretabilidad del modelo se reduce. Al tratarse de una combinación de múltiples clasificadores, no es posible obtener una representación gráfica sencilla del modelo y no es inmediato identificar de forma visual que predictores son más importantes [60]. Sin embargo, el tener múltiples clasificadores, abre la posibilidad de nuevas estrategias para cuantificar la importancia de los predictores que hacen de los modelos de *bagging* una herramienta muy potente, no solo para predecir, sino también para el análisis exploratorio [40]. Dos de estas medidas son: el incremento del error cuadrático medio o del error medio de clasificación y el incremento de la pureza de nodos en los casos en que se utilicen los árboles de decisión como clasificador individual.

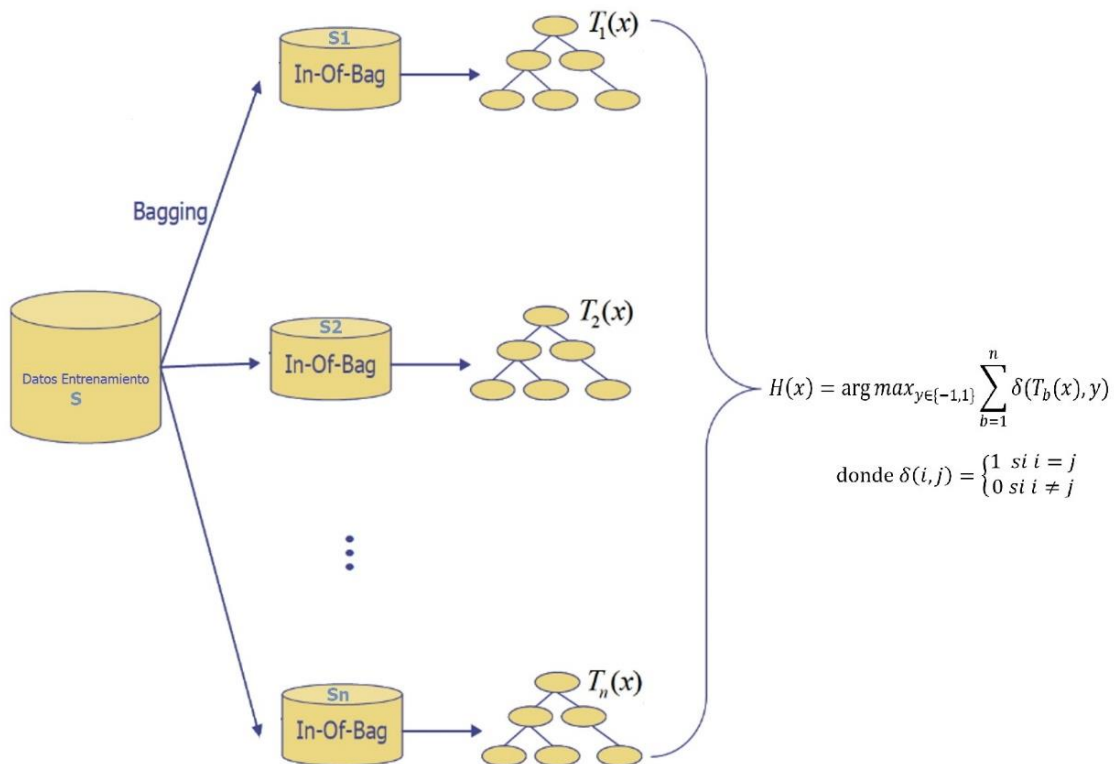


Figura 8.4: Ejemplo de la forma de cálculo y clasificación del algoritmo Bagging. Adaptado de [60].

ENTRADA: I (algoritmo de clasificación), n (número de clasificadores que integran la combinación), S (datos de entrenamiento)

SALIDA: T_n con $n= 1, \dots, n$ y $H(x)$

$t=1$

repeat

Realizar *bootstrapping* con el conjunto S para obtener S_n para entrenamiento

Construir el clasificador T_n utilizando I y S_n

Incrementar t

until $t > n$

Combinar los clasificadores individuales T_n para obtener un único clasificador $H(x)$ que clasifica cada nueva muestra por voto mayoritario

$$H(x) = \arg \max_{y \in \{-1,1\}} \sum_{b=1}^n \delta(T_b(x), y) \text{ donde } \delta(i, j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Figura 8.5: Pseudocódigo del algoritmo *bagging*.

8.2.2 ÁRBOLES ALEATORIOS (*RANDOM FOREST* POR SU DENOMINACIÓN EN INGLÉS)

El método de *random forest* es una modificación del proceso de *bagging* que consigue en varios casos mejores resultados gracias a que decorrelaciona los árboles generados en el proceso [40]. Como se ha descrito, *bagging* se basa en el hecho de que promediando un conjunto de modelos, se consigue reducir la varianza. Esto es cierto siempre y cuando los modelos agregados no estén correlacionados. Si la correlación es alta, la reducción de varianza que se puede lograr es pequeña.

En el caso de un conjunto de datos en el que hay un predictor es muy influyente junto con otros moderadamente influyentes, todos o casi todos los árboles creados en el proceso de *bagging* estarán dominados por el mismo predictor y serán muy parecidos entre ellos. Como consecuencia de la alta correlación entre los árboles, el proceso de *bagging* apenas conseguirá disminuir la varianza y, por lo tanto, tampoco mejora el modelo final combinado. *Random forest* evita este problema haciendo una selección aleatoria de m predictores antes de evaluar cada división. Si se considera a p el número de predictores totales, un promedio de $(p-m)/p$ divisiones no contemplarán el predictor influyente, permitiendo que otros predictores puedan

ser seleccionados. Solo con añadir este paso extra se consigue decorrelacionar los árboles, por lo que su agregación consigue una mayor reducción de la varianza [61].

Los métodos de *random forest* y *bagging* siguen el mismo algoritmo con la única diferencia que en *random forest* antes de cada división se seleccionan aleatoriamente m predictores. La diferencia en el resultado dependerá del valor m escogido. Si $m=p$ los resultados de *random forest* y *bagging* son equivalentes. Un valor recomendado del valor m es $m = \sqrt{p}$ [40]. Sin embargo, la mejor forma para encontrar el valor óptimo de m es evaluar el error cuadrático medio para diferentes valores de m . Por lo general, si los predictores están muy correlacionados, valores pequeños de m consiguen mejores resultados.

Al igual que ocurre con *bagging*, *random forest* no sufre problemas de sobreajuste por aumentar el número de árboles creados en el proceso. Alcanzado un determinado número, la reducción de error se estabiliza.

8.3 COMBINACIÓN DE CLASIFICADORES DE DIFERENTE TIPO

La forma de combinar los clasificadores se puede dividir en dos grupos principales: combinaciones de múltiples clasificadores mediante métodos no generadores y meta combinaciones. Los primeros son más adecuados para los problemas donde los clasificadores individuales realizan la misma tarea y tienen un éxito comparable [40]. Sin embargo, tales combinaciones son más vulnerables a valores atípicos y a clasificadores de rendimiento desigual. Por otro lado, los meta combinadores son teóricamente más poderosos pero son susceptibles a todos los problemas asociados con el aprendizaje agregado, como el sobreajuste y el tiempo de entrenamiento prolongado [44].

8.3.1 MÉTODOS DE COMBINACIÓN NO GENERADORES

En esta metodología los clasificadores entrenados se unen mediante un procedimiento de combinación que depende de su capacidad de adaptación a las observaciones de entrada y de las necesidades de la salida que proporcionan los sistemas de aprendizaje individuales. Es decir, el tipo de combinación depende del tipo de salida.

8.3.1.1 Voto uniforme

Esta metodología es la más sencilla como esquema de combinación, donde cada clasificador tiene el mismo peso. La clasificación de una instancia no etiquetada se realiza de acuerdo con la clase que obtiene el mayor número de votos (función moda estadística). Su descripción matemática se detalla en la ecuación 8.6.

$$\text{Clase}(x) = \text{moda}\{C_1(x), C_2(x), \dots, C_m(x)\} \quad (8.6)$$

Dónde $C_m(x)$ denota la clase resultante de cada clasificador dada una instancia x y m la cantidad total de clasificadores combinados. En esta metodología el problema reside en que todos los clasificadores de base tienen la misma importancia y no se tiene en cuenta la mayor o menor precisión del mismo a la hora de la generalización. Asimismo, el número de clasificadores incluido es un factor crítico ya que demasiados clasificadores poco precisos pueden llevar a una decisión final errónea y por tanto a disminuir la precisión del conjunto.

Este procedimiento puede mejorarse mediante la votación mayoritaria con peso asignando un peso a cada clasificador individual de tal forma que se optimiza el comportamiento del clasificador combinado en el conjunto de entrenamiento. Normalmente este peso se asigna de acuerdo al error de clasificación individual con los datos de validación.

8.3.1.2 Combinaciones de clase consistente sin entrenamiento

Dentro de este grupo se analiza y combina la probabilidad a posteriori de cada clasificador individual que conforma el conjunto combinado para una determinada instancia de evaluación. La combinación se realiza a través de una operación de máximo, mínimo, promedio, mediana o producto, entre los más utilizados, para obtener una probabilidad final y la definición a la clase a la que pertenecerá la muestra evaluada. La clase seleccionada se elige de acuerdo con el valor más alto de probabilidad obtenida en la combinación de probabilidades individuales. Matemáticamente estas operaciones se detallan en las siguientes ecuaciones: 8.7 para máximo, 8.8 para mínimo, 8.9 para promedio, 8.10 para mediana y 8.11 para producto, en las cuales L representa el número de clasificadores individuales combinados, $\mu_j(x)$ la probabilidad final del grupo combinado para la clase w_j y $d_{i,j}$ la probabilidad a posteriori del clasificador individual i para la clase w_j .

$$\mu_j(x) = \max_i \{d_{i,j}(x)\} \quad (8.7)$$

$$\mu_j(x) = \min_i \{d_{i,j}(x)\} \quad (8.8)$$

$$\mu_j(x) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(x) \quad (8.9)$$

$$\mu_j(x) = \text{mediana}_i \{d_{i,j}(x)\} \quad (8.10)$$

$$\mu_j(x) = \prod_{i=1}^L d_{i,j}(x) \quad (8.11)$$

8.3.1.3 Dempster-Shafer.

La idea de utilizar la teoría de la evidencia de Dempster-Shafer (Buchanan y Shortliffe, 1984) para combinar modelos ha sido sugerida por Shilen (1990; 1992). Este método usa la noción de asignación de probabilidad básica definida para cierta clase C_i dada la instancia x según se detalla en la ecuación 8.12.

$$bpa(c_i, x) = 1 - \prod_k (1 - \hat{P}_{M_k}(y = c_i | x)) \quad (8.12)$$

En consecuencia, la clase seleccionada es la que maximiza el valor de la función de creencia según se expresa en la ecuación 8.13.

$$Bel(c_i, x) = \frac{1}{A} \frac{bpa(c_i, x)}{(1 - bpa(c_i, x))} \quad (8.13)$$

Dónde A es un factor de normalización definido según la ecuación 8.14.

$$Bel(c_i, x) = \frac{1}{A} \frac{bpa(c_i, x)}{(1 - bpa(c_i, x))} \quad (8.14)$$

8.3.1.4 Naive Bayes.

Usando la regla de Bayes, uno puede extender la idea de Naive Bayes para combinar varios clasificadores según se expresa en la ecuación 8.15. Este método presenta una variante con corrección de cero.

$$clase(x) = \underset{c_j \in \text{dom}(y)}{\text{argmax}} \hat{P}(y = c_j) \cdot \prod_{k=1} \frac{\hat{P}_{M_k}(y = c_i | x)}{\hat{P}(y = c_i |)} \quad (8.15)$$

8.3.1.5 Fuzzy integral.

Basada en la teoría de lógica difusa (Zadeh, 1965) este método de combinación tiene como filosofía medir el desempeño no solo para cada clasificador individual, sino también para todos los subconjuntos de clasificadores. Cada subconjunto tiene una medida de fortaleza que expresa cuán bueno es este grupo de expertos para la entrada x dada que se debe clasificar. Para la decisión final se utiliza la *fusificación Sugeno* como se expresa en la ecuación 8.16. Donde g representa la medición difusa y H_α es un sub conjunto de los clasificadores individuales para un determinado α de corte.

$$\mu_j(x) = \max_\alpha \{\min\{\alpha, g(H_\alpha)\}\} \quad (8.16)$$

8.3.2 MÉTODOS DE META COMBINACIÓN

Meta aprendizaje significa aprender de los clasificadores producidos por los inductores y de las clasificaciones de estos clasificadores en los datos de entrenamiento. A continuación se describe la técnica de *stacking* que es la más utilizada pero existen otras como *grading* o árboles árbitros que también se encuentran dentro del grupo de meta combinación.

8.3.2.1 Stacking

Stacking es una técnica cuyo propósito es lograr la mayor exactitud de generalización. Al usar meta aprendizaje, se intenta inducir cuáles de los clasificadores son confiables y cuáles no. *Stacking* generalmente se emplea para combinar modelos construidos por diferentes inductores. La idea es crear un meta *dataset* que contenga una tupla para cada tupla en el conjunto de datos original. Sin embargo, en lugar de utilizar los atributos de entrada originales, usa la clasificación predicha de los clasificadores como atributos de entrada. El atributo de destino permanece como en el conjunto de entrenamiento original [40].

La instancia de prueba primero se clasifica por cada uno de los clasificadores básicos. Estas clasificaciones se introducen en un conjunto de entrenamiento de meta nivel a partir del cual se produce un meta clasificador. Este clasificador combina las diferentes predicciones en una final. Se recomienda que el conjunto de datos original se divida en dos subconjuntos. El primer subconjunto está reservado para formar el meta *dataset* y el segundo subconjunto se usa para construir los clasificadores de nivel base. En consecuencia, las preferencias del meta clasificador reflejan el verdadero rendimiento de los algoritmos de aprendizaje de nivel base [40]. Los rendimientos de *stacking* podrían mejorarse utilizando las probabilidades de salida para cada etiqueta de clase de los clasificadores de nivel base. En tales casos, el número de entradas en los atributos en el meta *dataset* se multiplica por el número de clases.

Dzeroski y Zenko (2004) [62] han evaluado varios algoritmos para construir conjuntos de clasificadores con *stacking* y muestran que el conjunto funciona en el mejor de los casos de forma comparable a la selección del mejor clasificador del conjunto por validación cruzada. Para mejorar el enfoque de *stacking*, proponen emplear un nuevo árbol de modelo de múltiples respuestas para aprender en el meta nivel y muestran empíricamente que se comporta mejor que los enfoques de *stacking* existentes y es mejor que seleccionar el mejor clasificador por validación cruzada.

8.4 TAMAÑO DE CLASIFICADOR COMBINADO

Un aspecto importante de los métodos combinados es definir cuántos clasificadores se deben utilizar. Esta decisión se basa generalmente en alguna de las siguientes consideraciones [40]:

- **Exactitud deseada:** Hansen (1990) propuso que los conjuntos que contienen diez clasificadores son suficientes para reducir la tasa de error. Sin embargo, hay evidencia empírica que indica que en el caso de AdaBoost usando árboles de decisión, la reducción de errores se observa incluso en conjuntos relativamente grandes que contienen 25 clasificadores (Opitz y Maclin, 1999). En los enfoques de participación disjunta, puede haber una compensación entre el número de subconjuntos y la exactitud final. El tamaño de cada subconjunto no puede ser demasiado pequeño porque deben estar disponibles datos suficientes para que cada proceso de aprendizaje produzca un clasificador eficaz.

- **Preferencias del usuario:** el aumento del número de clasificadores generalmente aumenta la complejidad computacional y disminuye la comprensión. Por esa razón, los usuarios pueden establecer sus preferencias predefiniendo el límite de tamaño del conjunto.

- **Número de procesadores disponibles:** en enfoques concurrentes, el número de procesadores disponibles para el aprendizaje paralelo podría ponerse como un límite superior en la cantidad de clasificadores que se tratan en dicho proceso.

Caruana y col. (2004) presentó un método para construir clasificadores combinados a partir de bibliotecas con miles de modelos. Sugieren utilizar la selección progresiva hacia adelante para seleccionar los modelos que maximiza el rendimiento del clasificador combinado. Esta selección permite que los modelos combinados mejoren sus métricas de rendimiento tales como exactitud, entropía cruzada, área ROC, etc.

8.5 RESULTADOS OBTENIDOS CON *BOOSTING*

Dado que existen diversos algoritmos que implementan *AdaBoost* se indagó mediante optimización Bayesiana cuáles presentaban mejor desempeño cuando el clasificador base es un árbol de decisión tipo CART. Se utilizó CART como clasificador individual dado que es un algoritmo de base inestable ampliamente sugerido para *boosting*, el cual demandó como algoritmo individual menor tiempo de entrenamiento ante similar desempeño y tiempo de validación que CHAID, cómo se observó en la sección 7.4. Una característica importante para los algoritmos combinados como *boosting* es la cantidad de clasificadores base por lo que se estableció dicha cantidad midiendo el error de clasificación cuando la cantidad árboles CART utilizados varió de 1 a 500. Los resultados luego de implementar la optimización en Matlab® se presentan en la figura 8.6, donde se puede observar que el algoritmo que mejor desempeño presenta es *GentleBoost* para 500 árboles base dado que minimizan la función de error.

Optimizado los parámetros para realizar *boosting* se procedió a entrenar los 22 clasificadores según datos de entrenamiento descritos en la sección 6.2.1 con un tiempo promedio de entrenamiento por clasificador de $23,10 \pm 7,84$ s.

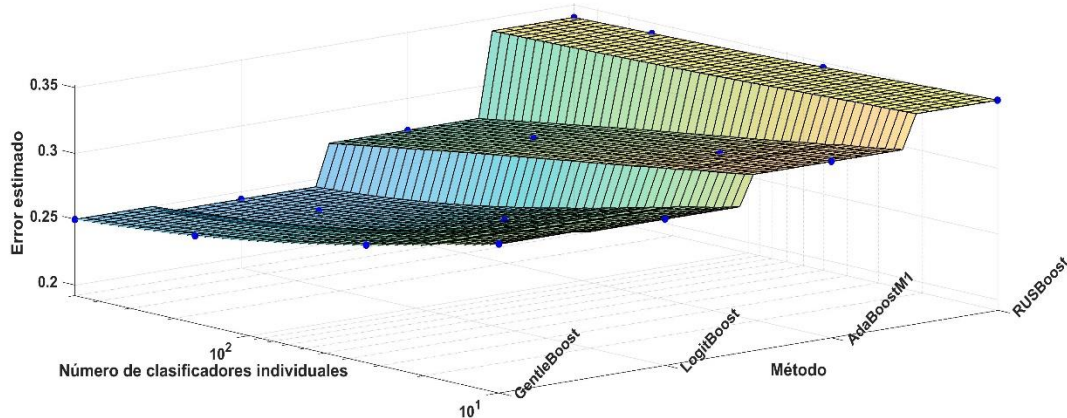


Figura 8.6: Gráfica del estudio de optimización Bayesiana para determinar el algoritmos *AdaBoost* a implementar y la cantidad de clasificadores CART base que presenten menor error de clasificación para los datos de la tesis.

Luego se validó cada clasificador *GentleBoost* con su conjunto de validación obteniéndose los valores exactitud (ACC), especificidad (ESP), sensibilidad (SEN), área bajo la curva ROC (AUC), índice de capacidad de diagnóstico (DOR) y tiempo de entrenamiento según se detalla en la tabla 8.1. El tiempo promedio de validación por paciente fue de $0,536 \pm 0,131$ s. En la figura 8.7 se observa la curva ROC.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,808	0,834	0,771	0,893	16,922	0,476
Paciente 2	0,796	0,839	0,734	0,886	14,408	0,492
Paciente 3	0,814	0,844	0,771	0,906	18,292	0,831
Paciente 4	0,814	0,838	0,779	0,904	18,192	0,573
Paciente 5	0,803	0,844	0,743	0,883	15,714	0,949
Paciente 6	0,798	0,834	0,747	0,891	14,848	0,495
Paciente 7	0,791	0,822	0,745	0,880	13,553	0,467
Paciente 8	0,792	0,807	0,770	0,873	13,956	0,526
Paciente 9	0,802	0,849	0,734	0,890	15,582	0,515
Paciente 10	0,783	0,812	0,740	0,869	12,287	0,487
Paciente 11	0,814	0,846	0,767	0,901	18,046	0,611
Paciente 12	0,804	0,822	0,778	0,890	16,265	0,466
Paciente 13	0,814	0,833	0,786	0,891	18,263	0,456
Paciente 14	0,793	0,813	0,764	0,881	14,071	0,477
Paciente 15	0,811	0,846	0,760	0,897	17,316	0,594
Paciente 16	0,814	0,861	0,745	0,888	18,086	0,469

Paciente 17	0,795	0,826	0,749	0,880	14,161	0,754
Paciente 18	0,786	0,804	0,760	0,875	13,005	0,470
Paciente 19	0,789	0,834	0,724	0,883	13,203	0,474
Paciente 20	0,793	0,831	0,737	0,876	13,850	0,484
Paciente 21	0,807	0,847	0,749	0,884	16,465	0,494
Paciente 22	0,814	0,835	0,784	0,887	18,417	0,486
Valor medio	0,801	0,833	0,756	0,887	15,551	0,536
Desvío Estándar	0,010	0,015	0,018	0,010	2,032	0,131

Tabla 8.1: Medidas de desempeño de los 22 clasificadores *GentleBoost* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

En particular el entrenamiento de cada clasificador *GentleBoost* permite también analizar, en base al error y los pesos calculados en cada etapa de construcción del modelo, la importancia de las características, es decir, aquellas que tienen mayor peso según los datos de entrenamiento. El resultado de importancia de características para *GentleBoost* que resulta del promedio para los 22 clasificadores entrenados se presenta en la figura 8.8, los que serán retomados en el capítulo 9 de selección de características (sección 9.5.1) donde se aborda el tema desde diversas perspectivas.

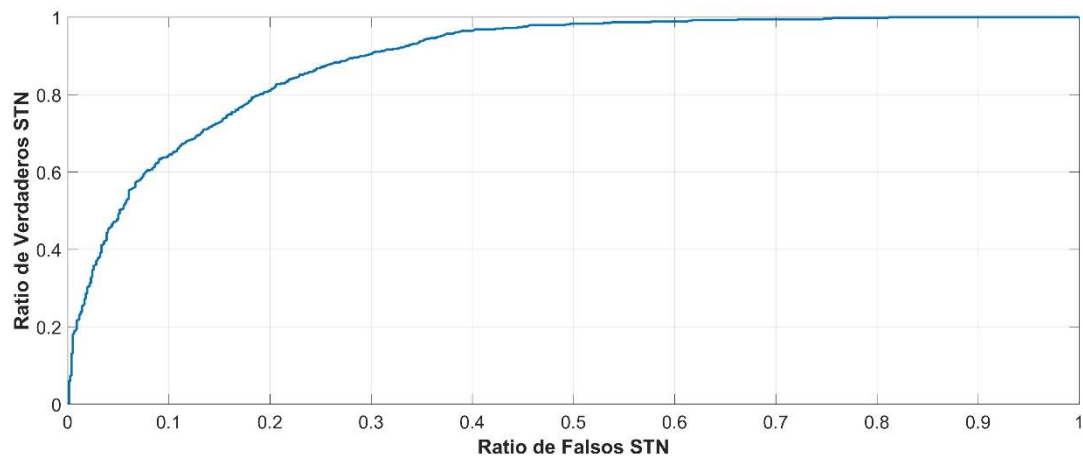


Figura 8.7: Curva ROC del desempeño del clasificador *GentleBoost*.

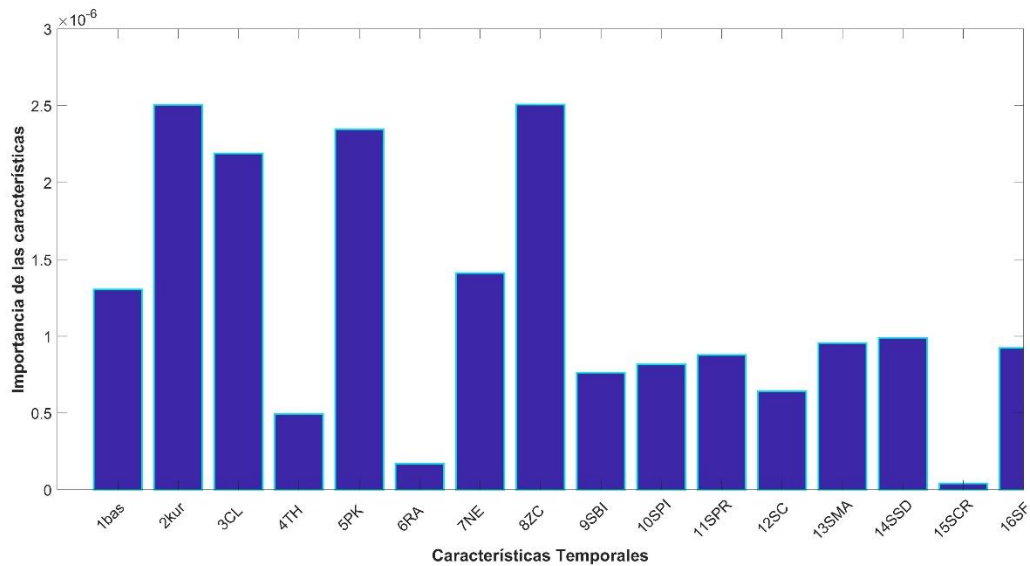


Figura 8.8: Importancia promedio de las características temporales que mayor peso tienen en el algoritmo *GentleBoost* a partir de los datos de entrenamiento.

8.5.1 RESULTADOS OBTENIDOS CON ÁRBOLES DE DECISIÓN PODADOS

En la sección 7.5 los árboles tipo CHAID con poda del 50% preservaron el desempeño general del clasificador al compararlo con el árbol CART con poda del 25%, presentando menor tiempo de validación. Dado que el tiempo de validación es de interés para el trabajo en tiempo real se indagó realizar *GentleBoost* utilizando como algoritmo de base árboles CHAID con poda del 50% y conservando los demás parámetros de configuración del algoritmo como se detalló en la sección 8.5. Los resultados promedio de los indicadores de desempeño para los 22 clasificadores se presentan en la tabla 8.2. El tiempo de entrenamiento promedio por clasificador fue de $33,48 \pm 13,60$ s.

	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
<i>GentleBoost</i> con CHAID podados al 50%	0,801±0,010	0,833±0,015	0,756±0,018	0,887±0,010	15,680±2,030	0,465±0,040

Tabla 8.2: Medidas promedio de desempeño de los 22 clasificadores *GentleBoost* con CHAID como algoritmos base podados al 50% y entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

8.6 RESULTADOS OBTENIDOS CON *BAGGING*

Como se refirió en la sección 8.2.1 es recomendable realizar *bagging* con algoritmos individuales inestables, como son los árboles de decisión. Dado que el menor tiempo de entrenamiento de los dos tipos de AD ensayados sin poda en la sección 7.4 resultó para el tipo CART el cual a su vez presentó un rendimiento similar que CHAID sin podar, se adoptó ese clasificador como base para realizar el proceso de *bagging* en la presente tesis.

Una característica para configurar el algoritmo *bagging* es la cantidad de clasificadores base que componen el *ensemble*, por lo que se estableció dicha cantidad mediante optimización Bayesiana minimizando la función del error de clasificación cuando la cantidad árboles CART varió de 1 a 500. Los resultados luego de implementar la optimización en Matlab® se presentan en la figura 8.6, donde se puede observar que el algoritmo que mejor desempeño presenta es 261 árboles base dado que minimizan la función de error.

Se procedió a entrenar para *bagging* los 22 clasificadores según datos de entrenamiento descriptos en la sección 6.2.1 con un tiempo promedio de entrenamiento de $32,50 \pm 1,13$ s.

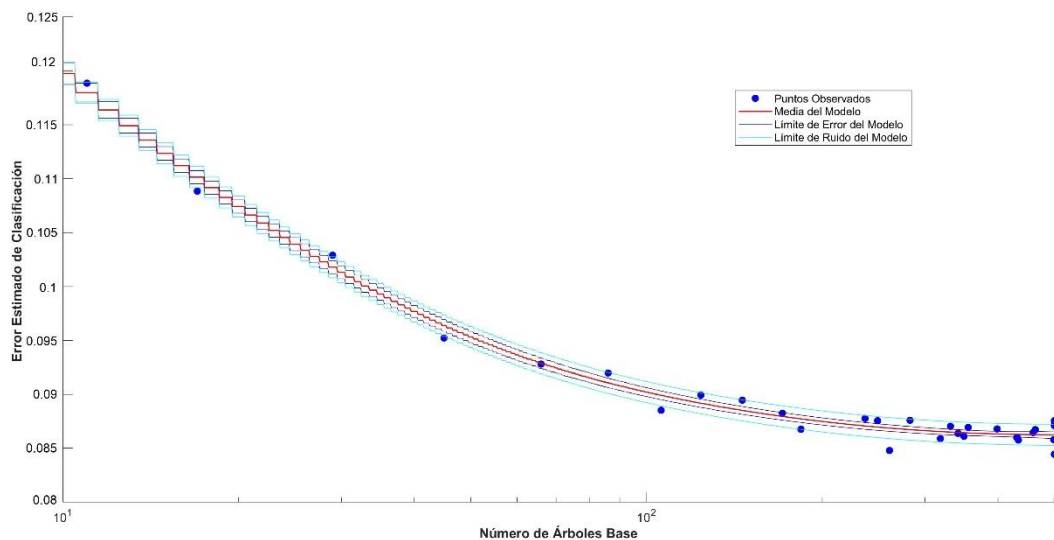


Figura 8.9: Gráfica del estudio de optimización Bayesiana para configurar en forma óptima el algoritmo *bagging*.

Luego se ensayó cada clasificador *bagging* con su conjunto de validación obteniéndose los valores exactitud (ACC), especificidad (ESP), sensibilidad (SEN), área bajo la curva ROC (AUC), índice de capacidad de diagnóstico (DOR) y tiempo de entrenamiento según se detalla

en la tabla 8.3. El tiempo promedio de validación por paciente fue de $0,60 \pm 0,03$ s lo que representa un valor medio de 0,02 s por nivel de profundidad. En la figura 8.10 se observa la curva ROC para *bagging*.

En particular, el entrenamiento de cada clasificador *bagging* permite también analizar, en base al error y los pesos calculados en cada etapa de construcción del modelo, la importancia de las características, es decir aquellas que tienen mayor peso según los datos de entrenamiento. El resultado de importancia de características para *bagging* que resultan del promedio para los 22 clasificadores entrenados se presenta en la figura 8.11, los que serán retomados en el capítulo 9 de selección de características (sección 9.5.1) donde se aborda el tema desde diversas perspectivas.

8.6.1 RESULTADOS OBTENIDOS CON ÁRBOLES DE DECISIÓN PODADOS

Con el mismo fundamento de la sección 8.5.1 se entrenó utilizando *bagging* y algoritmos de base árboles CHAID con poda del 50%, conservando los demás parámetros de configuración del algoritmo como se detalló en la sección 8.5. Los resultados promedio de los indicadores de desempeño para los 22 clasificadores se presentan en la tabla 8.4. El tiempo de entrenamiento por clasificador promedio fue de $42,950 \pm 2,740$ s.

8.7 RESULTADOS PARA ÁRBOLES ALEATORIOS

En esta sección se presentan los resultados de los clasificadores entrenados mediante *random forest* utilizando 261 árboles CART base y una cantidad de características muestreadas al azar de $m=4$. El valor de m se estimó mediante optimización bayesiana tomando como función objetivo el error de clasificación. Se implementaron los *scripts* correspondientes en Matlab® para entrenar los 22 clasificadores según datos de entrenamiento descritos en la sección 6.2.1 con un tiempo promedio de entrenamiento de $83,90 \pm 4,30$ s.

Luego se ensayó cada clasificador *random forest* con su conjunto de validación obteniéndose los valores exactitud (ACC), especificidad (ESP), sensibilidad (SEN), área bajo la curva ROC (AUC), índice de capacidad de diagnóstico (DOR) y tiempo de entrenamiento según se detalla en la tabla 8.4. El tiempo promedio de validación por paciente fue de

0,615±0,050 s lo que representa un valor medio de 0,020 s por nivel de profundidad. En la figura 8.12 se observa la curva ROC obtenida.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,929	0,942	0,911	0,950	165,829	0,555
Paciente 2	0,921	0,941	0,892	0,950	131,514	0,561
Paciente 3	0,928	0,940	0,911	0,957	158,555	0,555
Paciente 4	0,927	0,932	0,920	0,958	157,253	0,578
Paciente 5	0,922	0,934	0,905	0,950	135,933	0,572
Paciente 6	0,929	0,934	0,922	0,953	168,112	0,578
Paciente 7	0,926	0,936	0,913	0,954	151,897	0,586
Paciente 8	0,916	0,923	0,907	0,942	116,632	0,587
Paciente 9	0,922	0,940	0,898	0,955	136,398	0,605
Paciente 10	0,919	0,937	0,894	0,948	125,373	0,612
Paciente 11	0,922	0,940	0,896	0,948	133,408	0,587
Paciente 12	0,924	0,932	0,912	0,947	142,417	0,604
Paciente 13	0,924	0,932	0,912	0,950	142,417	0,601
Paciente 14	0,917	0,931	0,898	0,949	117,335	0,624
Paciente 15	0,925	0,934	0,912	0,953	148,411	0,598
Paciente 16	0,927	0,945	0,901	0,948	155,882	0,656
Paciente 17	0,919	0,931	0,903	0,949	124,877	0,643
Paciente 18	0,922	0,929	0,912	0,952	136,859	0,645
Paciente 19	0,922	0,937	0,901	0,954	135,677	0,630
Paciente 20	0,926	0,943	0,901	0,958	152,132	0,599
Paciente 21	0,922	0,936	0,903	0,947	135,613	0,635
Paciente 22	0,921	0,929	0,909	0,950	130,823	0,622
Valor medio	0,923	0,935	0,906	0,951	140,343	0,601
Desvío Estándar	0,004	0,005	0,008	0,004	14,602	0,030

Tabla 8.3: Medidas de desempeño de los 22 clasificadores *bagging* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
<i>Bagging</i> con CHAID podados al 50%	0,901±0,005	0,917±0,007	0,878±0,011	0,938±0,004	80,060±8,850	0,635±0,110

Tabla 8.4: Medidas promedio de desempeño de los 22 clasificadores *bagging* con CHAID podado al 50% como algoritmos base y entrenados utilizando como datos de validación.

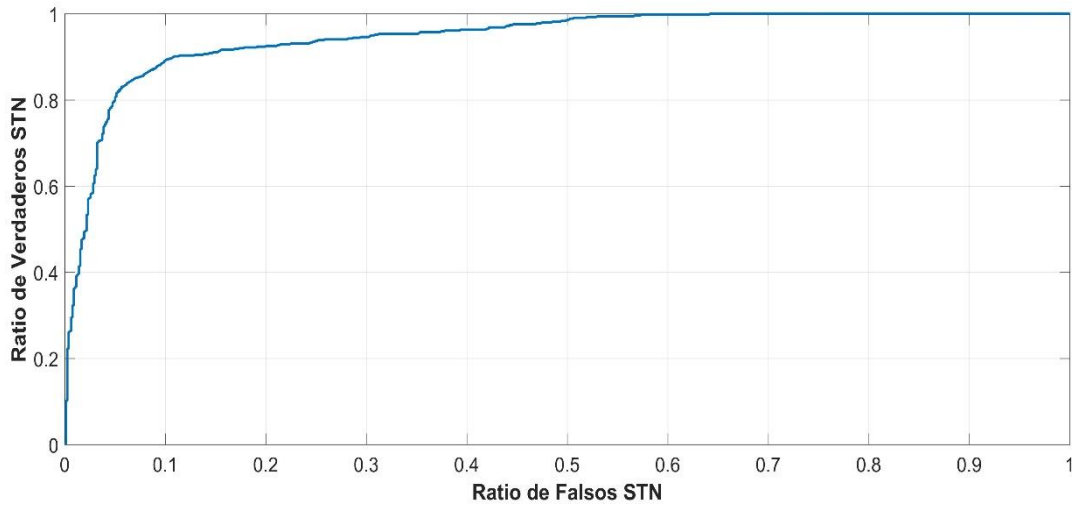


Figura 8.10: Curva ROC del desempeño del clasificador *bagging*.

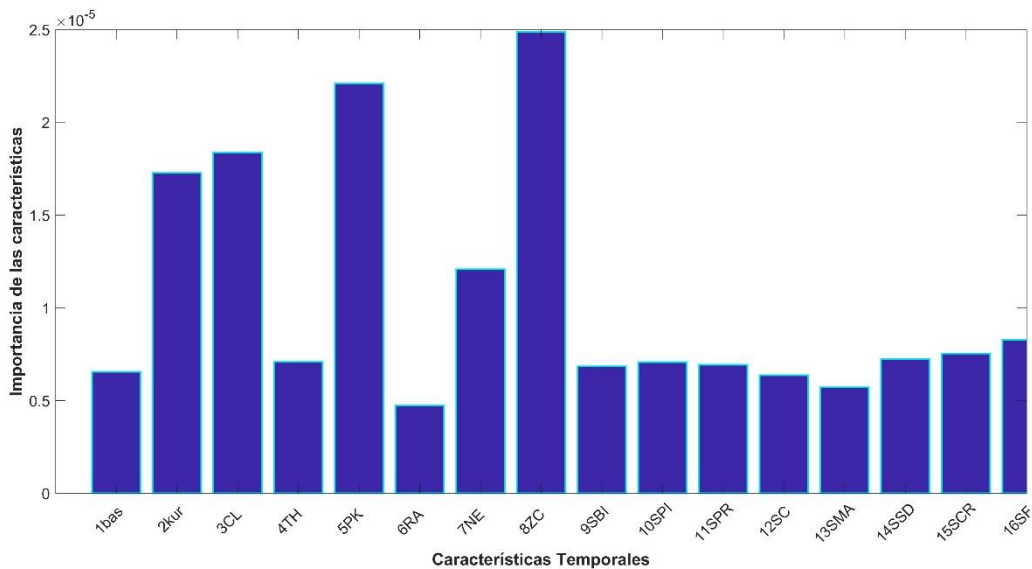


Figura 8.11: Importancia promedio de las características temporales que mayor peso tienen en el algoritmo *bagging* a partir de los datos de entrenamiento.

En particular el entrenamiento de los clasificadores *random forest* permite también analizar, en base al error y los pesos calculados en cada etapa de construcción del modelo, la importancia de las características, es decir aquellas que tienen mayor peso según los datos de entrenamiento. El resultado de importancia de características que resultan del promedio para los 22 clasificadores entrenados se presenta en la figura 8.13, los que serán retomados en el capítulo

9 de selección de características (sección 9.5.1) donde se aborda el tema desde diversas perspectivas.

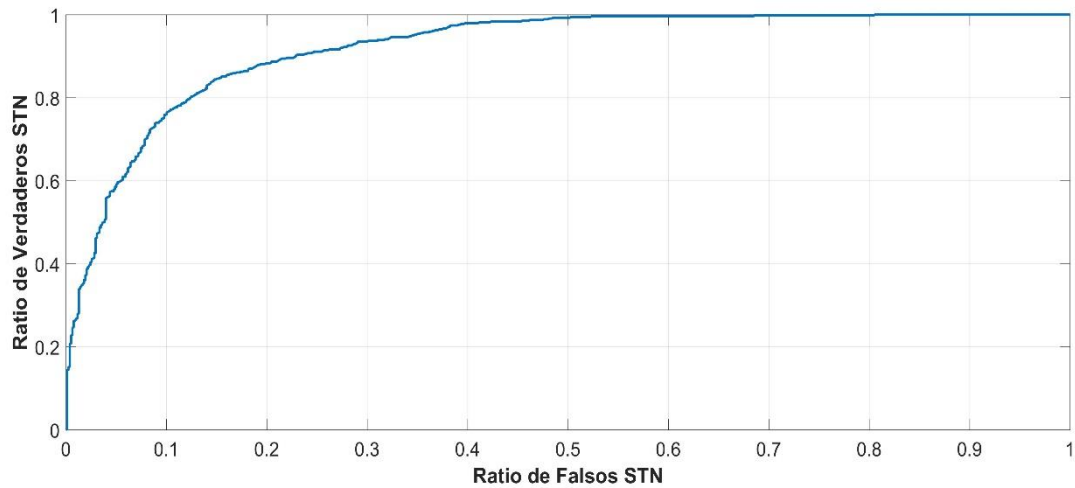


Figura 8.12: Curva ROC del desempeño del clasificador *random forest*.

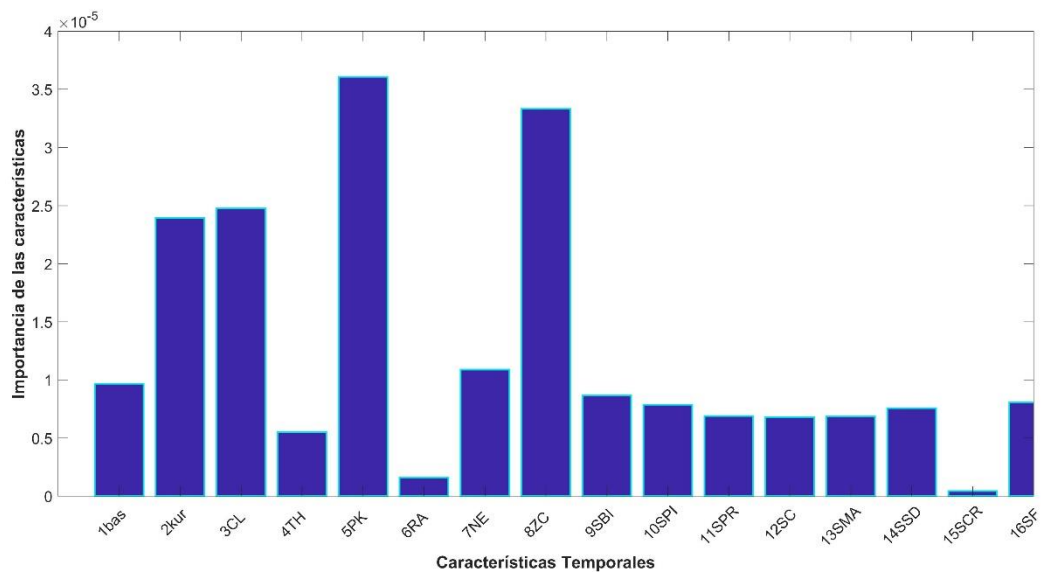


Figura 8.13: Importancia promedio de las características temporales que mayor peso tienen en el algoritmo *random forest* a partir de los datos de entrenamiento.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,865	0,884	0,838	0,921	39,513	0,633
Paciente 2	0,856	0,885	0,812	0,913	33,447	0,594
Paciente 3	0,855	0,878	0,822	0,921	33,053	0,663
Paciente 4	0,850	0,860	0,836	0,924	31,339	0,566
Paciente 5	0,859	0,874	0,838	0,915	35,917	0,609
Paciente 6	0,861	0,873	0,844	0,920	37,014	0,597
Paciente 7	0,844	0,862	0,818	0,913	28,114	0,620
Paciente 8	0,834	0,839	0,827	0,901	24,958	0,584
Paciente 9	0,847	0,882	0,797	0,916	29,304	0,567
Paciente 10	0,847	0,875	0,807	0,916	29,255	0,552
Paciente 11	0,849	0,880	0,804	0,917	30,260	0,608
Paciente 12	0,853	0,864	0,838	0,911	32,742	0,661
Paciente 13	0,851	0,867	0,827	0,911	31,241	0,625
Paciente 14	0,849	0,865	0,827	0,912	30,555	0,578
Paciente 15	0,849	0,860	0,834	0,914	30,849	0,582
Paciente 16	0,853	0,885	0,806	0,914	32,185	0,554
Paciente 17	0,846	0,870	0,812	0,910	28,893	0,778
Paciente 18	0,846	0,860	0,825	0,909	28,882	0,679
Paciente 19	0,847	0,882	0,797	0,917	29,236	0,630
Paciente 20	0,855	0,885	0,812	0,923	33,371	0,650
Paciente 21	0,848	0,875	0,808	0,912	29,539	0,629
Paciente 22	0,847	0,862	0,825	0,909	29,532	0,608
Valor medio	0,851	0,871	0,821	0,914	31,174	0,615
Desvío Estándar	0,007	0,012	0,014	0,005	3,227	0,051

Tabla 8.4: Medidas de desempeño de los 22 clasificadores *random forest* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

8.8 RESULTADO DE LA COMBINACIÓN DE CLASIFICADORES DIFERENTES MEDIANTE MÉTODOS NO GENERADORES

Se presenta en esta sección los resultados obtenidos de la combinación de clasificadores individuales de diferente tipo mediante los métodos descritos en la sección 8.3.1. Se implementaron los scripts correspondientes en Matlab® entrenando, por cada método de combinación, 22 clasificadores compuestos de 3 clasificadores de base diferentes según se detalla en la tabla 8.5. El entrenamiento y la validación para todos los casos se realizó con el mismo conjunto de datos descritos en la sección 6.2.1. Los clasificadores que se seleccionaron

como base para luego realizar la fusión de sus resultados individuales fueron aquellos que en forma individual arrojaron mejor exactitud y que a su vez presentaron una baja correlación entre sí: KNN (sección 7.3), CART (sección 7.4) y *bagging* (sección 8.6).

Luego se ensayó el desempeño de cada clasificador combinado utilizando para cada caso los 22 conjuntos de validación y midiendo los valores promedio de exactitud (ACC), especificidad (ESP), sensibilidad (SEN), índice de capacidad de diagnóstico (DOR) y tiempo de validación según se detalla en la tablas 8.6.

Denominación	Método de combinación	Clasificadores individuales utilizados
Combi_Voto	Voto Mayoritario	KNN, CART, Bagging
Combi_Max	Máximo	KNN, CART, Bagging
Combi_Min	Mínimo	KNN, CART, Bagging
Combi_Promedio	Promedio	KNN, CART, Bagging
Combi_Mediana	Mediana	KNN, CART, Bagging
Combi_Producto	Producto	KNN, CART, Bagging
Combi_DS	Dempster-Shafer	KNN, CART, Bagging
Combi_Bayes	Naive Bayes	KNN, CART, Bagging
Combi_Fuzzy	Fuzzy Integral	KNN, CART, Bagging

Tabla 8.5: Detalle de cada clasificador combinado con su método de combinación y algoritmos individuales que lo integran.

Denominación	ACC	ESP	SEN	DOR	tiempo validación [s]
Combi_Voto	0,810±0,008	0,840±0,012	0,770±0,020	17,78±1,93	0,998±0,146
Combi_Max	0,780±0,008	0,817±0,010	0,734±0,016	12,43±1,22	0,884±0,038
Combi_Min	0,780±0,008	0,817±0,010	0,734±0,016	12,43±1,22	0,960±0,110
Combi_Promedio	0,800±0,009	0,830±0,010	0,750±0,019	14,66±1,68	0,933±0,050
Combi_Mediana	0,810±0,008	0,840±0,012	0,770±0,020	17,78±1,93	0,880±0,040
Combi_Producto	0,790±0,009	0,820±0,010	0,740±0,017	13,38±1,40	0,870±0,020
Combi_DS	0,800±0,007	0,820±0,020	0,750±0,030	15,34±1,75	1,235±0,210
Combi_Bayes	0,790±0,008	0,815±0,020	0,735±0,022	14,21±1,87	0,920±0,120
Combi_Fuzzy	0,830±0,009	0,857±0,010	0,803±0,016	19,32±2,23	1,220±0,270

Tabla 8.6: Medidas de desempeño promedio de los clasificadores combinados mediante voto mayoritario, máximo, mínimo, promedio, mediana, producto, Dempster-Shafer, Naive Bayes y fuzzy integral. Los valores en negrita indican los mejores desempeños.

8.9 RESULTADO DE LA COMBINACIÓN DE CLASIFICADORES MEDIANTE *STACKING* COMO META COMBINADOR

Los clasificadores que se seleccionaron como nivel 0 para luego entrenar el meta clasificador de nivel 1 fueron aquellos que en forma individual arrojaron mejor exactitud y que a su vez presentaron una baja correlación entre sí: KNN (sección 7.3), CART (sección 7.4) y *Bagging* (sección 8.6). Para seleccionar los clasificadores de nivel 1 del *stacking* se realizó una prueba mediante validación cruzada con los datos de la tesis utilizando los clasificadores más reportados en el estado del arte: regresión lineal, árboles de decisión, máquinas de soporte vectorial, KNN, *boosting*, *bagging* y *random forest*. De los resultados exploratorios se obtuvieron mejores valores de exactitud promedio con estos últimos 3 como clasificadores de nivel 1, por lo que se reportan los resultados obtenidos con los mismos. En la tabla 8.7 se presentan las configuraciones de nivel 0 y nivel 1 de los clasificadores *stacking* seleccionados. Para los clasificadores de nivel 1 se utilizaron las mismas configuraciones que las adoptadas en las secciones 8.5, 8.6 y 8.7 y para los de nivel 0 las definidas por optimización en las secciones 7.3 (para KNN), 7.4 (para CART) y 8.6 (para *bagging*).

Denominación	Clasificador Nivel 1	Clasificadores nivel 0
Stack1	<i>GentleBoost</i>	KNN
		CART
		<i>bagging</i>
Stack2	<i>Random Forest</i>	KNN
		CART
		<i>bagging</i>
Stack3	<i>Bagging</i>	KNN
		CART
		<i>bagging</i>

Tabla 8.7: Detalle de cada clasificador *stacking* con su composición a nivel 1 y nivel 0.

Se implementaron los scripts correspondientes en Matlab® entrenando, por cada *satacking* propuesto, 22 clasificadores compuestos de 3 clasificadores de nivel 0 y un clasificador de nivel 1 según se detalla en la tabla 8.7. El entrenamiento y la validación para todos los casos se realizó con el mismo conjunto de datos descriptos en la sección 6.2.1

Luego se ensayó el desempeño de cada clasificador *stacking* utilizando para cada caso los 22 conjuntos de validación y midiendo los valores promedio de exactitud (ACC), especificidad (ESP), sensibilidad (SEN), área bajo la curva ROC (AUC), índice de capacidad de diagnóstico (DOR) y tiempo de validación según se detalla en la tabla 8.8 (para Stack1), 8.9 (para Stack2) y 8.10 (para Stack3). El tiempo de entrenamiento promedio para Stack1 fue de $49,57 \pm 1,11$ s , para Stack2 de $52,48 \pm 2,77$ s y para Stack3 de $49,66 \pm 1,31$ s como resultado de la suma de los tiempos de entrenamiento de los clasificadores individuales de base más el tiempo propio de entrenamiento para realizar *stacking*.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,867	0,888	0,836	0,862	40,498	1,429
Paciente 2	0,862	0,885	0,829	0,857	37,475	1,384
Paciente 3	0,868	0,879	0,853	0,866	42,216	1,236
Paciente 4	0,865	0,879	0,844	0,861	39,271	1,398
Paciente 5	0,860	0,884	0,825	0,855	36,055	1,051
Paciente 6	0,872	0,883	0,857	0,870	45,133	1,028
Paciente 7	0,873	0,892	0,846	0,869	45,226	1,049
Paciente 8	0,856	0,870	0,835	0,852	33,766	1,043
Paciente 9	0,853	0,875	0,822	0,848	32,277	1,029
Paciente 10	0,864	0,884	0,835	0,859	38,510	1,027
Paciente 11	0,862	0,888	0,825	0,856	37,377	1,410
Paciente 12	0,861	0,884	0,829	0,856	36,922	1,119
Paciente 13	0,872	0,888	0,849	0,869	44,649	1,101
Paciente 14	0,870	0,891	0,840	0,865	42,694	1,049
Paciente 15	0,860	0,878	0,834	0,856	36,137	1,040
Paciente 16	0,869	0,888	0,842	0,865	42,174	1,017
Paciente 17	0,849	0,871	0,818	0,844	30,327	1,045
Paciente 18	0,865	0,878	0,845	0,862	39,268	0,995
Paciente 19	0,861	0,880	0,834	0,857	37,022	1,034
Paciente 20	0,865	0,894	0,823	0,859	39,434	1,052
Paciente 21	0,860	0,885	0,823	0,854	35,966	1,038
Paciente 22	0,848	0,865	0,823	0,844	29,800	1,024
Valor medio	0,863	0,882	0,835	0,859	38,034	1,110
Desvío Estándar	0,007	0,007	0,011	0,007	4,394	0,147

Tabla 8.8: Medidas de desempeño de los 22 clasificadores *Stack1* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,947	0,956	0,935	0,946	313,635	1,187
Paciente 2	0,946	0,955	0,933	0,944	295,622	1,241
Paciente 3	0,948	0,952	0,942	0,947	327,097	1,202
Paciente 4	0,947	0,952	0,939	0,946	306,061	1,225
Paciente 5	0,945	0,955	0,931	0,943	287,059	1,176
Paciente 6	0,950	0,954	0,944	0,949	348,544	1,218
Paciente 7	0,950	0,958	0,939	0,948	345,014	1,168
Paciente 8	0,943	0,949	0,935	0,942	264,794	1,236
Paciente 9	0,942	0,951	0,929	0,940	255,886	1,211
Paciente 10	0,947	0,955	0,935	0,945	304,674	1,204
Paciente 11	0,946	0,956	0,931	0,944	295,310	1,250
Paciente 12	0,946	0,955	0,933	0,944	295,033	1,199
Paciente 13	0,950	0,956	0,940	0,948	344,867	1,232
Paciente 14	0,948	0,956	0,937	0,946	323,295	1,208
Paciente 15	0,945	0,952	0,935	0,944	286,857	1,230
Paciente 16	0,948	0,956	0,937	0,946	323,295	1,199
Paciente 17	0,941	0,950	0,927	0,939	241,633	1,217
Paciente 18	0,947	0,952	0,939	0,945	305,455	1,190
Paciente 19	0,945	0,952	0,935	0,944	286,857	1,194
Paciente 20	0,947	0,959	0,931	0,945	314,611	1,255
Paciente 21	0,945	0,955	0,931	0,943	286,486	1,238
Paciente 22	0,940	0,946	0,931	0,939	236,637	1,212
Valor medio	0,946	0,954	0,935	0,944	297,911	1,213
Desvío Estándar	0,003	0,003	0,004	0,003	30,911	0,023

Tabla 8.9: Medidas de desempeño de los 22 clasificadores *Stack2* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

Datos de Validación	ACC	ESP	SEN	AUC	DOR	tiempo validación [s]
Paciente 1	0,865	0,884	0,838	0,921	39,513	0,633
Paciente 2	0,856	0,885	0,812	0,913	33,447	0,594
Paciente 3	0,855	0,878	0,822	0,921	33,053	0,663
Paciente 4	0,850	0,860	0,836	0,924	31,339	0,566
Paciente 5	0,859	0,874	0,838	0,915	35,917	0,609
Paciente 6	0,861	0,873	0,844	0,920	37,014	0,597
Paciente 7	0,844	0,862	0,818	0,913	28,114	0,620
Paciente 8	0,834	0,839	0,827	0,901	24,958	0,584
Paciente 9	0,847	0,882	0,797	0,916	29,304	0,567
Paciente 10	0,847	0,875	0,807	0,916	29,255	0,552
Paciente 11	0,849	0,880	0,804	0,917	30,260	0,608
Paciente 12	0,853	0,864	0,838	0,911	32,742	0,661
Paciente 13	0,851	0,867	0,827	0,911	31,241	0,625
Paciente 14	0,849	0,865	0,827	0,912	30,555	0,578
Paciente 15	0,849	0,860	0,834	0,914	30,849	0,582
Paciente 16	0,853	0,885	0,806	0,914	32,185	0,554
Paciente 17	0,846	0,870	0,812	0,910	28,893	0,778
Paciente 18	0,846	0,860	0,825	0,909	28,882	0,679
Paciente 19	0,847	0,882	0,797	0,917	29,236	0,630
Paciente 20	0,855	0,885	0,812	0,923	33,371	0,650
Paciente 21	0,848	0,875	0,808	0,912	29,539	0,629
Paciente 22	0,847	0,862	0,825	0,909	29,532	0,608
Valor medio	0,851	0,871	0,821	0,914	31,174	0,615
Desvío Estándar	0,007	0,012	0,014	0,005	3,227	0,051

Tabla 8.10: Medidas de desempeño de los 22 clasificadores *Stack3* entrenados utilizando como datos de validación las 16 características temporales calculadas por paciente.

8.10 ANÁLISIS ESTADÍSTICO Y DE RESULTADOS DE LOS CLASIFICADORES DEL PRESENTE CAPÍTULO

En la presente sección se realiza el estudio estadístico de los clasificadores abordados en el capítulo. En particular se comparan los índices de desempeño de los clasificadores que combinan algoritmos base del mismo tipo: *boosting* implementado mediante el algoritmo GentleBoost con árboles tipo CART sin podar como clasificadores de base (sección 8.5), GentleBoost utilizando árboles base tipo CHAID podados al 50% (sección 8.5.1), *bagging* con

árboles tipo CART sin podar como clasificador de base (sección 8.6), *bagging* con árboles base tipo CHAID podados al 50% (sección 8.6.1) y *Random forest* (sección 8.7).

Analizando los resultados obtenidos del proceso de clasificación que resulta de utilizar los métodos no generadores, que permiten combinar los resultados de clasificadores de distinto tipo (sección 8.8), puede observarse en la tabla 8.6 que combi fuzzy presentó la mejor exactitud promedio de $0,83\pm 0,009$ con los datos de validación. De los algoritmos de base utilizados para las combinaciones de los métodos no generadores cuyos resultados se presentaron en la tabla 8.6, el de mejor desempeño fue *bagging* con una exactitud promedio de $0,923\pm 0,004$ para los mismos datos de validación. Una combinación de clasificadores tiene sentido si al combinarlos se logra un desempeño superior al de los clasificadores base. Dado que no se obtuvieron esas mejoras según se desprende de comparar la tabla 8.6 con la tabla 8.3 de *bagging* no se selecciona ninguno de estos clasificadores combinados obtenidos por métodos no generadores para realizar un estudio estadístico comparativo.

Para el caso de los meta clasificadores, los cuales para todas las opciones propuestas en la sección 8.9 los clasificadores de nivel 0 fueron KNN, CART y *bagging*, se presenta una situación similar a la descrita en el párrafo anterior, al comparar las tablas con los resultados de Stack1 (tabla 8.8) y Stack 3 (tabla 8.10) con la tabla 8.3 de los resultados de *bagging*. Es decir, con sólo utilizar *bagging* (exactitud promedio de $0,923\pm 0,004$) se obtiene mejor desempeño que utilizando Stack1 (exactitud promedio $0,863\pm 0,007$) o Stack3 (exactitud promedio $0,851\pm 0,007$). Diferente es el caso de la combinación denominada Stack2, donde el clasificador de nivel 1 es *Random forest* el cual, según se desprende de la tabla 8.9, dio una exactitud promedio de $0,946\pm 0,003$ y un desempeño general superior a *bagging*. Por lo tanto, se selecciona Stack2 para conformar el grupo de clasificadores del presente capítulo a los cuales se les realizó un estudio estadístico comparativo.

Los clasificadores seleccionados del presente capítulo para comparar sus desempeños estadísticamente se detallan en la tabla 8.11, en concordancia con lo analizado en los párrafos precedentes. Tomando las variables exactitud (ACC), especificidad (ESP), sensibilidad (SEN), índice de capacidad de diagnóstico (DOR) y los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val.) se analizó estadísticamente el rendimiento de todos los clasificadores siguiendo la metodología propuesta en la sección 6.2.1.

Denominación	Clasificadores a comparar estadísticamente
Gentle_CART	GentleBoost utilizando árboles base tipo CART sin podar como clasificadores de base
Gentle_CHAID50	GentleBoost utilizando árboles base tipo CHAID podados al 50% como clasificadores de base
Bag_CART	<i>Bagging</i> utilizando árboles base tipo CART sin podar como clasificadores de base
Bag_CHAID50	<i>Bagging</i> utilizando árboles base tipo CHAID podados al 50% como clasificadores de base
RForest	<i>Random forest</i> utilizando árboles base tipo CART sin podar como clasificadores de base
Stack2	<i>Stacking</i> utilizando KNN, CART sin podar y <i>Bagging</i> como clasificadores de nivel 0 y <i>Random Forest</i> como clasificador de nivel 1

Tabla 8.11: Clasificadores seleccionados del capítulo 8 y su codificación para el estudio estadístico comparativo.

El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 8.12 y las figuras 8.14 y 8.15 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

De los clasificadores comparados Stack2 presentó la mejor exactitud promedio con los datos de validación de $0,946 \pm 0,003$, de la tabla 8.12 y la gráfica 8.14 se observa que para ACC presenta una diferencia muy significativa ($p < 0,01$) con los otros clasificadores comparados, excepto con Bag_CART con el cual no presentó diferencia significativa ($p = 0,39$). Stack2 evidenció diferencias muy significativas ($p < 0,01$) también para los otros indicadores de desempeño ESP, SEN y DOR cuando se los compara con todos los otros clasificadores analizados de a pares, excepto para Bag_CART con el cual no tubo diferencias significativas para ninguno de esos indicadores (p para ESP=0,23; p para SEN=0,31 y p para DOR=0,39). Se puede asumir que Stack2 y Bag_CART tienen desempeños estadísticamente similares en base a los indicadores antes analizados.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Valid.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Gentle_CART vs Gentle_CHAID50	1,000	1,000	1,000	1,000	0,100	0,990
Gentle_CART vs Bag_CART	<0,01	<0,01	<0,01	<0,01	0,050	0,070
Gentle_CART vs Bag_CHAID50	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Gentle_CART vs Rforest	0,075	0,073	0,074	0,075	<0,01	<0,01
Gentle_CART vs Stack2	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Gentle_CHAID50 vs Bag_CART	<0,01	<0,01	<0,01	<0,01	0,990	0,250
Gentle_CHAID50 vs Bag_CHAID50	<0,01	<0,01	<0,01	<0,01	0,250	<0,01
Gentle_CHAID50 vs RForest	0,075	0,073	0,074	0,075	<0,01	<0,01
Gentle_CHAID50 vs Stack2	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART vs Bag_CHAID50	0,630	0,920	0,780	0,630	0,380	0,640
Bag_CART vs RForest	<0,01	0,017	0,01	<0,01	<0,01	0,100
Bag_CART vs Stack2	0,390	0,230	0,310	0,390	<0,01	<0,01
Bag_CHAID50 vs RForest	0,390	0,230	0,310	0,390	<0,01	0,900
Bag_CHAID50 vs Stack2	<0,01	<0,01	<0,01	<0,01	0,330	<0,01
RForest vs Stack2	<0,01	<0,01	<0,01	<0,01	0,380	0,100

Tabla 8.12: Comparación estadística de a pares de clasificadores entre los seleccionados del capítulo 8 según tabla 8.11 como resultado de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación. Valores en negrita indican significancia estadística para $\alpha=0,05$.

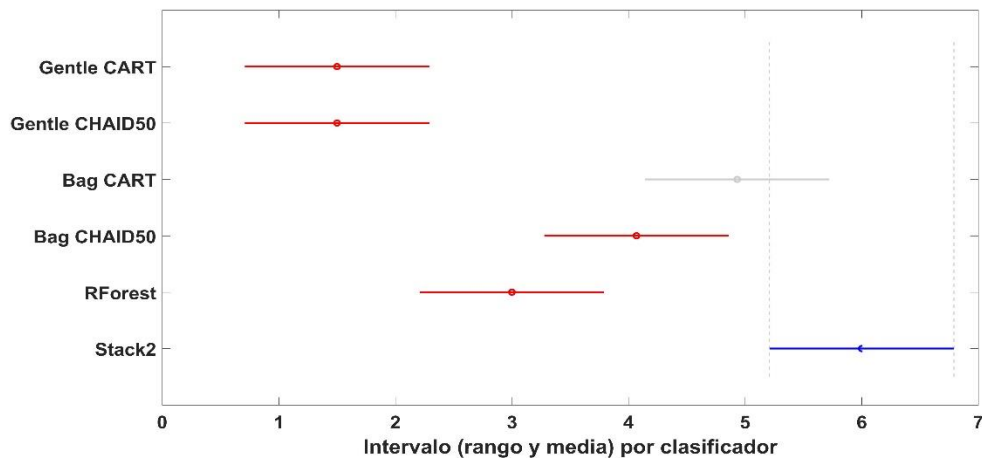


Figura 8.14: Comparación de clasificadores detallados en la tabla 8.11 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, la línea gris el clasificador *Bag_CART* que no presenta diferencia significativa con *Stack2*, las líneas rojas indican diferencias significativas de los clasificadores respectivos con *Stack2*.

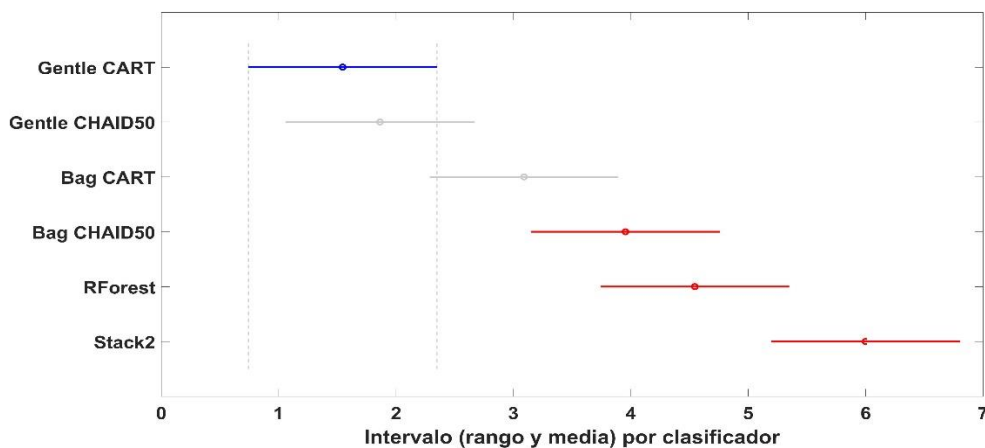


Figura 8.15: Comparación de clasificadores detallados en la tabla 8.11 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado, las líneas grises los clasificadores *Bag_CART* y *Gentle_CHAID50* que no presentan diferencia significativa con *Gentle_CART*, las líneas rojas indican diferencias significativas de los clasificadores respectivos con *Gentle_CART*.

Bag_CART presentó una exactitud promedio para los datos de validación de $0,923\pm 0,004$ (tabla 8.3), una curva ROC (figura 8.10) con una excelente área promedio de $0,951\pm 0,004$ según tabla 8.3. Esta AUC resultó superior en promedio a la del algoritmo Gentle_CART ($0,887\pm 0,010$; figura 8.7), a la AUC promedio de Gentle_CHAID50 ($0,887\pm 0,010$; tabla 8.2) y la AUC promedio de RForest ($0,914\pm 0,005$; figura 8.12). BAG_CART también demuestra un muy buen desempeño general, presentando diferencias significativas o muy significativas para ACC, ESP, SEN y DOR con los otros clasificadores excepto para Stack2 y Bag_CHAID50. Para este último un valor de $p=0,63$ en ACC, de $p=0,92$ para ESP, de $p=0,78$ para SEN y de $p=0,63$ para DOR. Sin embargo Stack2 y Bag_CHAID50 presentan entre si diferencias muy significativas para todos los indicadores de desempeño. Lo analizado anteriormente permite inferir que Bag_CART presenta muy buen desempeño general en el proceso de clasificación de los datos de validación de la presente tesis.

El menor tiempo de validación promedio fue de Gentle_CART con $0,536\pm 0,131$ s (tabla 8.1 y figura 8.15, lo que representa un valor medio de 0,017 s por nivel de profundidad) el cual presentó diferencias muy significativas con Bag_CHAID50 (tiempo de validación promedio de $0,635\pm 0,11$ s; $p<0,01$), con RForest (tiempo de validación promedio de $0,615\pm 0,050$; $p<0,01$) y con Stack2 (tiempo de validación promedio de $1,213\pm 0,023$; $p<0,01$). Sin embargo Gentle_CART con Bag_CART y con Gentle_CHAID50 no presentó para el tiempo de validación diferencias estadísticamente significativas ($p=0,05$ y $p=0,10$ respectivamente).

En base a lo detallado anteriormente Stack2 es claramente el clasificador con mejor desempeño de los analizados en el presente capítulo, presentando como desventaja comparativa con los otros clasificadores analizados su tiempo de validación de $1,213\pm 0,023$ s. Este valor representa un tiempo medio de 0,040 s por nivel de profundidad lo cual no presenta dificultades según los tiempos definidos en la sección 6.2.2. Bagging utilizando árboles base tipo CART sin podar como clasificadores de base (Bag_CART) es una alternativa con un muy buen desempeño y un tiempo de validación aproximadamente la mitad de Stack2. Por lo tanto se decide optar por Stack2 y Bag_CART como los clasificadores que tienen mejor rendimiento relativo, de los analizados en este capítulo, bajo las condiciones experimentales de la presente tesis.

Capítulo 9 Experiencia 3: Selección de características

En el presente capítulo se presentan los distintos métodos de selección de características y se ensayan los mismos con el conjunto de las 16 características temporales calculadas de los registros MER de la base de datos utilizada. El objetivo final de la selección de características es reducir la dimensión de dicho espacio, preservando o mejorando el desempeño de los clasificadores, con el objetivo final de reducir los tiempos de cálculo para el trabajo en tiempo real del sistema de clasificación. Algunos autores proponen realizar el proceso de reducción del orden de dimensión de las características previo al entrenamiento de los clasificadores, en la presente tesis se decidió realizar dicho procesos luego del entrenamiento con todas la características dado que algunos métodos de reducción de dimensionalidad surgen del propio entrenamiento del clasificador y otros ajustan la reducción según el tipo de clasificador, tal como se desarrolla en la siguientes secciones.

9.1 REDUCCIÓN DE LA DIMENSIÓN DEL ESPACIO DE LAS CARACTERÍSTICAS

La reducción de la dimensionalidad se realiza utilizando diversas técnicas para eliminar las características con ruido o irrelevantes y las redundantes [63]. Las técnicas de reducción de la dimensionalidad se pueden clasificar principalmente en la extracción de características y la selección de características. La extracción de características obtiene nuevas características en un espacio de menor dimensión. Las nuevas características construidas suelen ser combinaciones de características originales siendo las técnicas más utilizadas el análisis de componentes principales (PCA), el análisis discriminante lineal (LDA) y el análisis de correlación canónica (CCA) [63]. Por otro lado, la selección de características elige un subconjunto de características que minimizan la redundancia y maximizan la exactitud del clasificador [63].

Tanto la extracción de características como la selección de características son capaces de mejorar el rendimiento del proceso de entrenamiento de un clasificador, reducir la complejidad computacional, crear modelos generalizables y disminuir el almacenamiento de datos. La extracción de características al generar nuevas características como resultado de la combinación de las originales dificulta su análisis e interpretación, ya que no hay un significado físico para

las características transformadas. Mientras que la selección de características obtiene un subconjunto de características del conjunto de características original sin ninguna transformación, y mantiene el significado físico original. Por este motivo en la presente tesis se abordan técnicas de selección de características que se desarrollarán en las próximas secciones.

9.2 SELECCIÓN SUPERVISADA DE CARACTERÍSTICAS

La selección de características para clasificación se puede definir como un problema de optimización combinatoria mediante el cual se selecciona un conjunto de características que maximiza la calidad de la hipótesis aprendida de estas características. Los métodos supervisados de selección de características se pueden categorizar en forma general en: modelos de filtros, modelos de envoltura y modelos integrados [63].

Los modelos de filtro separan la selección de características del aprendizaje del clasificador, de modo que el sesgo de un algoritmo de aprendizaje no interactúe con el sesgo de un algoritmo de selección de características [41]. Se basa en medidas de las características generales de los datos de entrenamiento tales como distancia, consistencia, dependencia, información y correlación. *Relief* y los métodos basados en la ganancia de información se encuentran entre los algoritmos más representativos de los modelos de filtro [64].

Los modelos de envoltura utilizan la precisión predictiva de un algoritmo de aprendizaje predeterminado para determinar la calidad de las características seleccionadas. Estos métodos son costosos computacionalmente y con poca aplicabilidad en los casos donde hay que procesar muchos datos con una gran cantidad de características [41].

Los modelos integrados incorporan criterios estadísticos, como lo hace el modelo de filtro, para seleccionar varios subconjuntos de características candidatas y luego selecciona el subconjunto con la mayor exactitud de clasificación y k clases respectivamente y $X = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^{m \times n}$ representan los datos a clasificar, donde n es el número de instancias correspondiendo a x_i la clase y_i . Por lo tanto, el modelo integrado generalmente logra una precisión comparable a los algoritmos de envoltura y una eficiencia comparable al modelo del filtro [63]. El modelo integrado realiza la selección de características simultáneamente con el proceso de entrenamiento.

En las próximas secciones se denomina $F = \{f_1, f_2, \dots, f_m\}$ y $C = \{c_1, c_2, \dots, c_k\}$ a los conjuntos de m características independientes

9.2.1 MODELOS DE FILTRO

Los métodos de filtro reducen la cantidad de características utilizando las propiedades intrínsecas de los datos independientemente de qué algoritmo de aprendizaje se use finalmente [63]. Los métodos pueden ser univariados si no toman en consideración los valores de otros atributos al producir la selección o multivariado si consideran la interacción de las otras características [41]. La mayoría de los métodos de filtro desarrollados son univariados.

En las últimas dos décadas, se han propuesto una serie de algoritmos para la selección de características basadas en filtros siendo los más utilizados ganancia de información [63] y ReliefF [65].

9.2.1.1 Método de ganancia de información

En el algoritmo de ganancia de información (IG) se clasifican las características según un puntaje de relevancia que se basa en cada atributo individual. Al no considerarse la correlación entre otros atributos lo convierte en un método univariable, por lo que no puede discriminar características redundantes.

Debido a su eficiencia computacional e interpretación simple es uno de los métodos utilizados frecuentemente en la selección de características, si bien se ha demostrado inestabilidad para datos con una importante cantidad de características [64]. Básicamente mide la ganancia de información (IG) entre la i -ésima función f_i y la clase etiquetada C según las ecuaciones 9.1, 9.2 y 9.3. Dónde $H(f_i)$ es la entropía de f_i y $H(f_i|C)$ es la entropía de f_i después de observar C . Una característica es relevante si tiene una gran ganancia de información.

$$IG(f_i, C) = H(f_i) - H(f_i|C) \quad (9.1)$$

$$H(f_i) = - \sum_j p(x_j) \cdot \log_2(p(x_j)) \quad (9.2)$$

$$H(f_i|C) = - \sum_j p(c_k) \sum_j p(x_j|c_k) \cdot \log_2(p(x_j|c_k)) \quad (9.3)$$

9.2.1.2 ReliefF

Relief y su extensión de clase múltiple ReliefF y fue propuesto por Kononenko y col. en 1994. Consiste básicamente en seleccionar las características al azar y luego, en función de los vecinos más cercanos, otorgan más peso a las características que discriminan más entre las clases [65]. Estas características a su vez se clasifican según su relevancia siendo por lo tanto Relief es un método multivariable.

Considerando que las instancias se muestrean aleatoriamente a partir de los datos y luego la puntuación de la i -ésima característica S_i es definida según la ecuación 9.4, donde M_k denota los valores en la i -ésima característica de las instancias más cercanas a x_k con la misma etiqueta de clase, mientras que H_k denota los valores en la función i -ésima de las instancias más cercanas a x_k con etiquetas de clase diferentes, siendo $d(\cdot)$ una medida de distancia [65].

$$S_i = \frac{1}{2} \sum_{k=1}^l d(X_{ik} - X_{iM_k}) - d(X_{ik} - X_{iH_k}) \quad (9.4)$$

En la ecuación 9.5 se expresa ReliefF para multiclase donde, M_k y H_{ky} denota los conjuntos de puntos más cercanos a x_k con la misma clase y la clase y con tamaños de m_k y h_{ky} respectivamente, y $p(y)$ es la probabilidad de una instancia de la clase y . Robnik-Sikonj y col [65] relacionaron el criterio de evaluación de relevancia de ReliefF con la hipótesis de maximización de márgenes, concluyendo que ReliefF proporciona un rendimiento superior en muchas aplicaciones siendo un algoritmo más estable que el de ganancia de información [64].

$$S_i = \frac{1}{K} \sum_{k=1}^l \left(-\frac{1}{m_k} \sum_{x_j \in M_k} d(X_{ik} - X_{ij}) + \sum_{y \neq y_k} \frac{1}{h_{ky}} \frac{p(y)}{1 - p(y)} \sum_{x_j \in H_k} d(X_{ik} - X_{ij}) \right) \quad (9.5)$$

9.2.2 MODELOS DE ENVOLTURA O ENVOLVENTES

En un método de envoltura (*wrapper* por su denominación del inglés) el procedimiento de búsqueda de subconjuntos de atributos consiste en generar varios subconjuntos obtenidos del conjunto original que son evaluados usando un algoritmo clasificador [64]. La dificultad de

estos métodos se presenta en que el número de subconjuntos crece exponencialmente al aumentar el número de características siendo necesario utilizar métodos heurísticos para guiar la búsqueda [64]. Además, los métodos envolventes son propensos al sobre entrenamiento, factor que deberá ser tenido en cuenta al trabajar con ellos.

Los métodos de envoltura tienen generalmente mejor desempeño que los métodos de filtro [63] ya que al mejorar el algoritmo de clasificación también se está mejorando el proceso de selección de atributos. Sin embargo, los métodos de envoltura pueden ser más costosos para problemas con una alta dimensionalidad (mayor que 50 características) desde el punto de vista del coste computacional ya que cada subconjunto de características considerado debe ser evaluado por el algoritmo de clasificación [64].

Dado un clasificador predefinido, un modelo de envoltura típico realizará los siguientes pasos:

- Paso 1: buscar un subconjunto de características,
- Paso 2: evaluar el subconjunto de características seleccionado por el rendimiento del clasificador,
- Paso 3: repetir el Paso 1 y el Paso 2 hasta alcanzar el desempeño deseado.

La metodología de búsqueda de características producirá un sub conjunto de características que se utilizan para entrenar el clasificador. El clasificador resultante se evalúa luego con un conjunto de datos independiente no utilizados en el proceso de entrenamiento [64].

Se utilizan varias estrategias de búsqueda heurísticas, tales como: eliminación hacia atrás (*backward* por su denominación del inglés), selección hacia adelante (*forward* por su denominación del inglés) y *branch and bound* [63]. En la técnica *forward* la búsqueda comienza con un conjunto vacío de características y luego se van incorporando progresivamente en subconjuntos cada vez más grandes; en *backward* se inicia con un conjunto conformado con todas las características y se van eliminando progresivamente aquellas que no aporten en el desempeño del clasificador. En *branch and bound* el algoritmo comienza desde el conjunto completo y elimina las características mediante una primera búsqueda en profundidad con una estrategia de retroceso. En esta última estrategia la búsqueda se lleva a cabo sistemáticamente por medio de una estructura de árbol considerando que los nodos cuya función objetivo son más bajos que los mejores actuales no se exploran, bajo la suposición de monotonía que asegura

que sus hijos no contendrán una mejor solución. El método *branch and bound* ha demostrado superioridad ante *backward* y *forward* en una gran cantidad de aplicaciones, dado que por la naturaleza de búsqueda del algoritmo no cae en mínimos locales [63].

9.2.3 MODELOS INTEGRADOS

Los modelos de filtro seleccionan características en forma independientes del clasificador y por lo tanto el algoritmo no realiza la evaluación del desempeño del clasificador como ocurre con los modelos de envoltura. Esto implica que si bien son computacionalmente eficientes no tienen en cuenta los sesgos de los clasificadores. Por ejemplo, la medida de relevancia de Relief no sería apropiada como selectores de subconjuntos de características para Naive-Bayes porque en muchos casos el rendimiento de Naive-Bayes mejora con la eliminación de características relevantes [41]. Los modelos *wrapper* utilizan un clasificador predefinido para evaluar la calidad de las características y el proceso de selección de características evita los sesgos de representación del clasificador. Sin embargo, tienen que ejecutar el entrenamiento y validación del clasificador muchas veces para evaluar la calidad de los subconjuntos de características seleccionados, lo cual es muy costoso desde el punto de vista computacional [41].

En los modelos integrados la selección de características es intrínseca a la construcción del clasificador. Hay tres tipos de métodos integrados: los primeros son los métodos de poda que utilizan primero todas las características para entrenar un modelo y luego intentan eliminar algunas características estableciendo los coeficientes correspondientes en 0, mientras mantienen el rendimiento del modelo tal como ocurre en la eliminación recursiva de características usando la máquina de soporte vectorial (SVM) [41]. El segundo son modelos con un mecanismo incorporado para la selección de características, como en árboles de decisión CART o CHAID [63]. El tercero son modelos de regularización con funciones objetivo que minimizan los errores de ajuste los cuales han demostrado muy buen desempeño [63]. Dentro de los modelos de regularización se encuentra *Adaptive Lasso* [63] y *Elastic Net* [63] los que se desarrollan en la siguientes sub secciones.

Dado un clasificador lineal w en los que la clasificación de Y puede basarse en una combinación lineal de X como ocurre en SVM y en regresión logística. En los métodos de regularización, la inducción del clasificador y la selección de características se logran

simultáneamente al estimar w con penalizaciones ajustadas adecuadamente [63]. Dado que cada coeficiente de w corresponde a una característica como w_i para f_i , solo las características con coeficientes distintos de cero en w se utilizarán en el clasificador. Específicamente, lo antes descrito queda definido en la ecuación 9.6, donde $c(\cdot)$ es la función objetivo de clasificación, la *penalización*(w) es un término de regularización, y α es el parámetro de regularización que controla el intercambio entre $c(\cdot)$ y la penalización. Las elecciones habituales de $c(\cdot)$ incluyen funciones de pérdida cuadrática, mínimos cuadrados y pérdida logística entre otras [63].

$$\hat{w} = \min_w c(w, X) + \alpha \cdot \text{penalización}(w) \quad (9.6)$$

9.2.3.1 LASSO (por el acrónimo del inglés *Least Absolute Shrinkage and Selection Operator*)

El algoritmo de regularización LASSO queda definido cuando la penalización de la ecuación 9.6 se obtiene mediante la ecuación 9.7 [63].

$$\text{penalización}(w) = \sum_{i=1}^m |w_i| \quad (9.7)$$

Una propiedad importante de la regularización LASSO es que puede generar una estimación de w con coeficientes exactamente igual a cero, en aquellos casos donde el algoritmo determina que las características correspondientes se deben eliminar durante el proceso de aprendizaje del clasificador. Esto permite utilizar el resultado del algoritmo para la selección de características.

La selección de la función LASSO es consistente si el modelo subyacente satisface una condición no trivial, que puede no ser así en la práctica [41]. LASSO produce estimaciones sesgadas para los coeficientes de gran valor, por lo tanto, podría en algunos casos, tener soluciones sub óptimas [41]. Para estos casos una versión LASSO denominada adaptativa según se detalla en la ecuación 9.8 puede mejorar el rendimiento, donde la única diferencia es que la versión adaptativa emplea un ajuste ponderado b_i para cada coeficiente w_i .

$$\text{penalización}(w) = \sum_{i=1}^m \frac{1}{b_i} |w_i| \quad (9.8)$$

9.2.3.2 Elastic Net

En la práctica, es común que algunas características estén altamente correlacionadas. En esta situación, LASSO tiende a seleccionar solo una de las características correlacionadas cuando es deseable conservar ambas [63]. Para lograr esto se propone trabajar con *Elastic net* según la ecuación 9.9 con $0 < \gamma \leq 1$ y $\lambda \geq 1$ [63].

$$\text{penalización}(w) = \sum_{i=1}^n |w_i|^\gamma + \left(\sum_{i=1}^n w_i^2 \right)^\lambda \quad (9.9)$$

9.3 RESULTADOS OBTENIDOS CON MODELOS DE FILTRO PARA LA SELECCIÓN DE CARACTERÍSTICAS

Dado que el método de ganancia de información resulta más inestable y que ReliefF ha demostrado muy buenos resultados, se procedió a obtener mediante este último algoritmo la selección de características mediante métodos de filtro. Los resultados se presentan en la figura 9.1. Se adoptó como criterio de selección elegir las características que tengan como peso de al menos la mitad de la que tiene mayor importancia. Esto arrojó como resultado las siguientes características de mayor peso según ReliefF: 2kur ,8ZC, 12SC, 15SCR y 16SF.

Se procedió a entrenar con este sub conjunto de características al clasificador que mejor desempeño tuvo en forma individual, en forma combinada utilizando un solo tipo de clasificador y combinada utilizando distintos tipos de clasificadores: KNN (sección 7.3), *Bagging* (sección 8.6) y *Stack2* (sección 8.9), con el objetivo de disminuir sus tiempos de validación conservando su desempeño general. Los resultados promedio de los indicadores de desempeño para cada clasificador con el sub conjunto de características adoptado como resultado de ReliefF se presentan en la tabla 9.1. Los tiempos de entrenamiento y validación promedio para cada caso se presentan en la tabla 9.2.

Clasificadores con ReliefF	ACC	ESP	SEN	AUC	DOR
KNN_ReliefF	0,712±0,011	0,77±0,014	0,628±0,016	0,758±0,012	5,670±0,635
Bag_CART_ReliefF	0,882±0,005	0,904±0,006	0,85±0,009	0,897±0,007	53,420±4,710
Stack2_ReliefF	0,937±0,003	0,95±0,003	0,92±0,004	0,934±0,003	215,830±19,210

Tabla 9.1: Medidas de desempeño promedio de los 22 clasificadores para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo ReliefF. En negrita se indican los mayores valores de cada índice de desempeño.

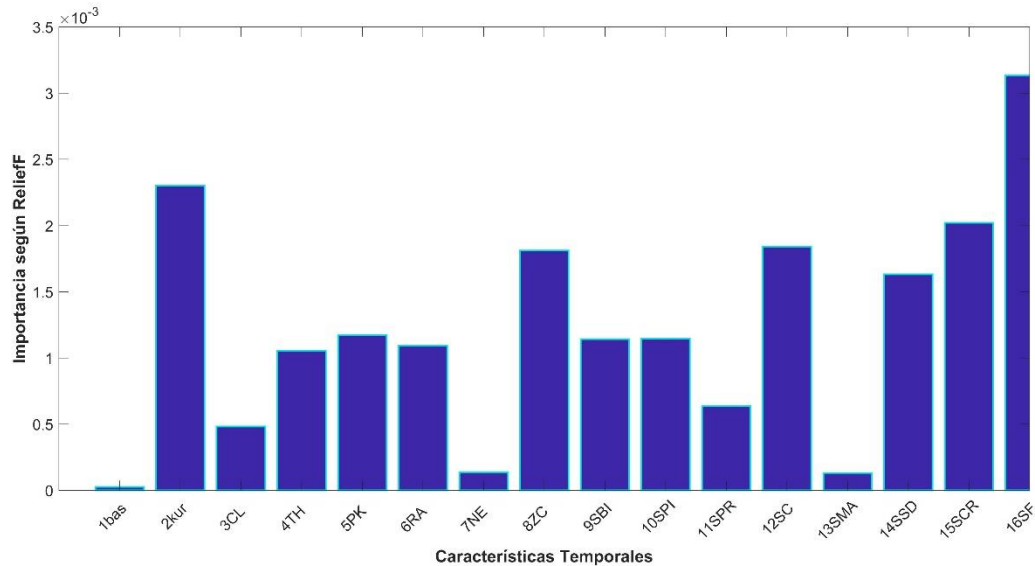


Figura 9.1: Importancia promedio de las características según el algoritmo ReliefF.

Clasificadores con ReliefF	tiempo entrenamiento [s]	tiempo validación [s]
KNN_ReliefF	0,059±0,004	0,011±0,006
Bag_CART_ReliefF	39,800±10,700	0,63±0,026
Stack2_ReliefF	60,120±11,700	1,05±0,048

Tabla 9.2: Tiempos de entrenamiento y clasificación promedio para KNN, Bagging (Bag_CART) y Stack2 entrenados con el sub conjunto de características seleccionado por el algoritmo ReliefF. En negrita se indican los menores valores de cada tiempo.

Se observa en la figura 9.1 que en la importancia relativa el algoritmo ReliefF da mayor peso a las características vinculadas a la presencia de espigas y en menor peso a las vinculadas con la actividad de fondo.

Las características seleccionadas por el algoritmo ReliefF disminuyen aproximadamente un 5% en promedio el desempeño de los clasificadores evaluados. Por ejemplo la ACC de KNN es de $0,751 \pm 0,013$ pasa en KNN_ReliefF $0,712 \pm 0,011$; Bag_CART con $0,935 \pm 0,005$ pasa a Bag_CART_ReliefF $0,882 \pm 0,005$ y Stack2 con $0,954 \pm 0,003$ disminuye en Stack2_ReliefF a $0,937 \pm 0,003$. En contraposición los tiempos de entrenamiento y validación disminuyen en distintos porcentajes según el clasificador.

Al ser ReliefF un algoritmo tipo filtro que analiza las características y las selecciona con un enfoque estadístico independiente del clasificador evidentemente no logra para los datos y clasificadores propuestos mantener o mejorar el desempeño de los mismos.

9.4 RESULTADOS OBTENIDOS CON MODELOS DE ENVOLTURA PARA LA SELECCIÓN DE CARACTERÍSTICAS

Branch and bound ha sido reportado con mejores resultados dentro de los algoritmos de envoltura dado que la estrategia de búsqueda no cae en mínimos locales y realiza una búsqueda combinada hacia adelante y hacia atrás. En la presente tesis se ensayó la selección de características de modelos de envoltura con *branch and bound*, ajustando el algoritmo las características resultantes al tipo de clasificador.

Para aplicar el método de *branch and bound* se seleccionaron los siguientes clasificadores con el objetivo de disminuir sus tiempos de validación conservando su desempeño general: en forma individual a KNN (sección 7.3), en forma combinada utilizando un solo tipo de clasificador de base a *Bagging* (sección 8.6) y combinados que utilizan distintos tipos de clasificadores a Stack2 (sección 8.9).

Los resultados de *branch and bound* fueron para KNN, Bagging y Stack2: 2kur, 3CL, 5PK, 8ZC, 12SC y 15SCR. Se procedió a entrenar los 22 clasificadores de cada tipo con este sub conjunto de características y luego se los validó con el conjunto de validación propuesto en la sección 6.2.1. Los resultados promedio de los indicadores de desempeño para cada clasificador con el sub conjunto de características adoptado como resultado de *branch and bound* se presentan en la tabla 9.3. Los tiempos de entrenamiento y validación promedio para cada caso figuran en la tabla 9.4.

Branch and bound para los 3 clasificadores estudiados selecciona las mismas características las cuales caracterizan en mayor proporción a la actividad de fondo y con menor peso a las espigas presentes. Es importante resaltar que *branch and bound* selecciona 6 características de las 16 totales, lo que representa una disminución porcentual de características utilizadas del 62,5%. Si se compara las características que seleccionó ReliefF versus las de *branch and bound* se observa que este último incorporó a 3CL y 5PK y quitó la 16SF reforzando aquellas vinculadas a la actividad de fondo.

En todos los casos *branch and bound* al ser un algoritmo tipo envoltura selecciona las características de acuerdo al clasificador propuesto y para los datos y algoritmos analizados mantiene o mejora el desempeño de los clasificadores como se desprende de la tabla 9.3 y disminuye los tiempos de entrenamiento y validación (tabla 9.4). Por ejemplo para la ACC de KNN que es de $0,751\pm 0,013$ pasa en KNN_BBoud a $0,780\pm 0,010$; Bag_CART de $0,923\pm 0,004$ pasa con Bag_CART_BBoud a $0,920\pm 0,004$ y Stack2 con $0,946\pm 0,003$ se incrementa en Stack2_BBoud a $0,950\pm 0,002$.

Clasificadores con <i>branch and bound</i>	ACC	ESP	SEN	AUC	DOR
KNN_BBoud	$0,780\pm 0,010$	$0,820\pm 0,015$	$0,720\pm 0,014$	$0,840\pm 0,009$	$11,820\pm 1,560$
Bag_CART_BBoud	$0,920\pm 0,004$	$0,930\pm 0,005$	$0,900\pm 0,007$	$0,950\pm 0,004$	$113,160\pm 11,480$
Stack2_BBoud	$0,950\pm 0,002$	$0,960\pm 0,003$	$0,940\pm 0,003$	$0,950\pm 0,002$	$383,800\pm 43,200$

Tabla 9.3: Medidas de desempeño promedio de los 22 clasificadores para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo *branch and bound*. En negrita se indican los mayores valores de cada índice de desempeño.

Clasificadores con <i>branch and bound</i>	tiempo entrenamiento [s]	tiempo validación [s]
KNN_BBoud	$0,056\pm 0,004$	$0,012\pm 0,001$
Bag_CART_BBoud	$32,043\pm 2,390$	$0,623\pm 0,138$
Stack2_BBoud	$43,947\pm 5,320$	$1,047\pm 0,280$

Tabla 9.4: Tiempos de entrenamiento y clasificación promedio para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo *branch and bound*. En negrita se indican los menores valores de cada tiempo.

9.5 RESULTADOS OBTENIDOS CON MODELOS INTEGRADOS PARA LA SELECCIÓN DE CARACTERÍSTICAS

Dentro de del grupo de modelos integrados se presentan como resultado la selección de características de los clasificadores vistos en los capítulos 7 y 8 en los cuales la propia naturaleza de entrenamiento de algunos clasificadores permite determinar el peso de cada característica en el clasificador final. También se abordarán los métodos LASSO y *Elastic Net* vistos en la sección 9.2.1.1 y 9.2.1.2 respectivamente.

9.5.1 AUTO SELECCIÓN DE CARACTERÍSTICAS DE LOS CLASIFICADORES ENTRENADOS EN LOS CAPÍTULOS 7 Y 8

En los capítulos 7 y 8 se abordaron los siguientes clasificadores: árboles de decisión tipo CART y CHAID (sección 7.4), *Boosting* implementado con *GentleBoost* y CART como clasificador base (sección 8.5), *Bagging* con CART (sección 8.6) y *random forest* con CART (sección 8.6). Para todos los algoritmos antes listados se presentó en cada sección respectiva el peso promedio que cada característica tiene en los 22 clasificadores entrenados de cada tipo. Se adoptó como criterio de selección, elegir las características que tengan como peso promedio al menos la mitad de la que tiene mayor peso. Con este criterio los resultados para cada tipo de clasificador se presentan en la tabla 9.5.

Clasificador	Denominación	Características seleccionadas en el proceso de entrenamiento del propio clasificador
CART (figura 7.11)	CART_Autosel	2kur , 3CL, 5PK, 8ZC
CHAID (figura 7.12)	CHAID_Autosel	2kur , 3CL, 5PK, 8ZC
GentleBoost (figura 8.8)	GentleBoost_Autosel	1bas, 2kur , 3CL, 5PK, 7NE, 8ZC
Bagging (figura 8.11)	Bagging_Autosel	2kur , 3CL, 5PK, 8ZC
Random Forest (figura 8.13)	RForest_Autosel	2kur , 3CL, 5PK, 8ZC

Tabla 9.5: Selección de características realizada en el proceso de entrenamiento del propio clasificador para los algoritmos CART, CHAID, *GentleBoost*, *bagging* y *Random forest*.

Se procedió a entrenar y validar con cada sub conjunto de características a los clasificadores detallados en la tabla 9.5. Los resultados promedio de los indicadores de desempeño para cada caso se presentan en la tabla 9.6. Los tiempos de entrenamiento y validación promedio figuran en la tabla 9.7.

Clasificadores	ACC	ESP	SEN	AUC	DOR
CART_Autosel	0,720±0,001	0,754±0,020	0,670±0,020	0,770±0,010	6,230±0,700
CHAID_Autosel	0,710±0,010	0,760±0,020	0,640±0,010	0,770±0,020	5,880±0,710
GentleBoost_Autosel	0,760±0,010	0,800±0,020	0,710±0,010	0,850±0,008	9,730±1,120
Bagging_Autosel	0,910±0,003	0,920±0,005	0,890±0,006	0,940±0,004	95,580±6,830
RForest_Autosel	0,820±0,010	0,840±0,010	0,780±0,010	0,880±0,010	18,830±1,310

Tabla 9.6: Medidas de desempeño promedio de los 22 clasificadores para CART, CHAID, *GentleBoost*, *Bagging* y *Random forest* entrenados con el sub conjunto de características auto seleccionado por cada algoritmo. En negrita se indican los mayores valores de cada índice de desempeño.

Clasificador con Autoselección	tiempo entrenamiento [s]	tiempo validación [s]
CART_Autosel	0,150±0,030	0,003±0,020
CHAID_Autosel	0,190±0,020	0,002±0,003
GentleBoost_Autosel	10,314±0,760	0,405±0,020
Bagging_Autosel	27,230±0,990	0,600±0,030
RForest_Autosel	33,020±0,720	0,690±0,040

Tabla 9.7: Tiempos de entrenamiento y validación promedio para CART, CHAID, *GentleBoost*, *Bagging* y *Random forest* entrenados con el sub conjunto de características auto seleccionado por cada algoritmo. En negrita se indican los menores valores de cada tiempo.

Se observa que los métodos de autoselección para todos los casos sólo dan importancia a las características vinculadas a la actividad de fondo. Todos comparten las características 2kur, 3CL, 5PK y 8ZC y *GentleBoost* agrega 1bas y 7NE. Los indicadores de desempeño de todos los clasificadores se reducen levemente entre un 2% y un 5% con una disminución porcentual promedio en los tiempos de entrenamiento del 40%. Los tiempos de validación disminuyen sin cambios notorios al compararlos con los mismos clasificadores entrenados con las 16 características.

9.5.2 RESULTADOS DE LA SELECCIÓN DE CARACTERÍSTICAS CON LASSO

Como se desarrolló en la sección 9.2.3.1 LASSO utiliza la regresión regularizada de mínimos cuadrados con un factor de penalidad, este proceso lo hace un algoritmo que además de la regresión pueda utilizarse para la selección de características. En la figura 9.2 se presenta la evolución de error cuadrático medio de la correlación cruzada de los 22 pacientes para distintos valores de alfa, parámetro de penalización según ecuaciones 9.6 y 9.7. La línea verde de la gráfica 9.2 indica el valor de alfa ($\alpha=0,00086$) para error cuadrático medio mínimo, mientras que la línea azul indica un alfa ($\alpha=0,093$) para el mismo error mínimo más un desvío estándar. En la presente tesis se adopta $\alpha=0,093$ para seleccionar las características por LASSO.

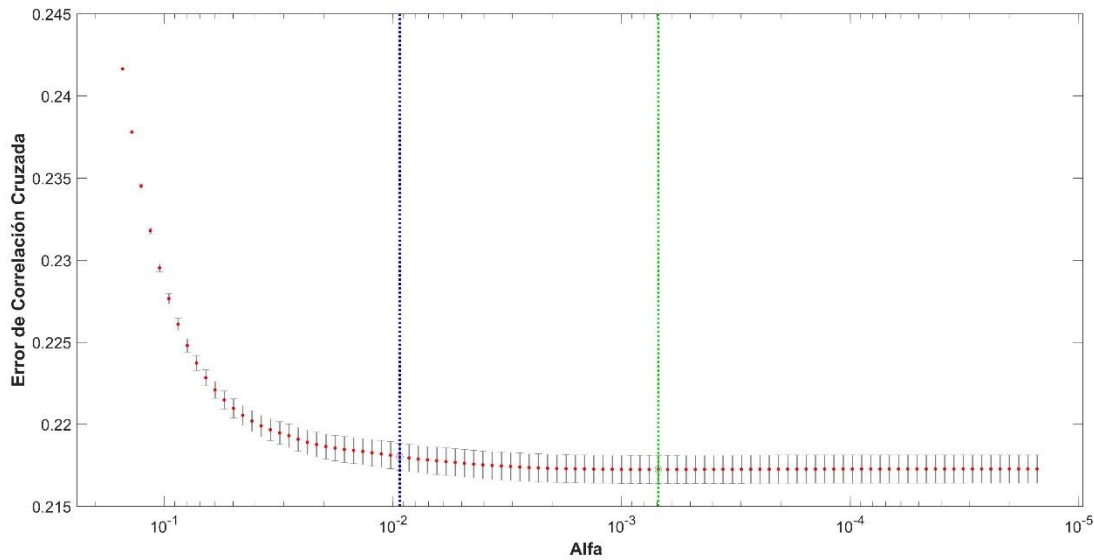


Figura 9.2: Error cuadrático medio de clasificación resultante de la correlación cruzada para los 22 pacientes. La línea en verde punteada indica el valor de corte de alfa para el algoritmo Lasso para obtener mínimo error. La línea azul punteada indica el valor de corte de alfa para obtener mínimo error mas un desvío estándar.

El sub conjunto de características seleccionas por algoritmo LASSO fueron: 1bas, 5PK, 6RA, 8ZC y 14SSD. Se procedió a entrenar con este sub conjunto de características al clasificador que mejor desempeño tuvo en forma individual (KNN, sección 7.3), en forma combinada utilizando un solo tipo de clasificador (*bagging*, sección 8.6) y combinada utilizando distintos tipos de clasificadores (*Stack2*, sección 8.9). Los resultados promedio de los indicadores de desempeño con los datos de validación para cada clasificador con el sub conjunto de características adoptado como resultado de *LASSO* se presentan en la tabla 9.8. Los tiempos de entrenamiento y validación promedio para cada caso figuran en la tabla 9.9.

Clasificadores	ACC	ESP	SEN	AUC	DOR
KNN_LASSO	0,720±0,010	0,760±0,012	0,660±0,020	0,780±0,010	6,270±0,670
Bagg_CART_LASSO	0,890±0,004	0,910±0,006	0,870±0,009	0,920±0,005	65,350±5,730
Stack2_LASSO	0,940±0,002	0,950±0,002	0,930±0,005	0,940±0,002	230,070±18

Tabla 9.8: Medidas de desempeño promedio de los 22 clasificadores para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo LASSO. En negrita se indican los mayores valores de cada índice de desempeño.

Clasificador con LASSO	tiempo entrenamiento [s]	tiempo validación [s]
KNN_LASSO	0,026±0,004	0,180±0,030
Bag_CART_LASSO	29,250±1,020	0,614±0,042
Stack2_LASSO	43,360±1,830	1,190±0,130

Tabla 9.9: Tiempos de entrenamiento y clasificación promedio para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo LASSO. En negrita se indican los menores valores de cada tiempo.

LASSO selecciona 5 características de las 16 totales, lo que representa una disminución porcentual del 69% en el sub conjunto propuesto. En general se observa que 4 de las características están vinculadas a caracterizar la actividad de fondo y sólo una representa a las espigas. Es importante resaltar que solo comparte la característica 8ZC con las resultantes de ReliefF y *branch and bound*.

Las característica seleccionadas por LASSO disminuyen levemente el desempeño de KNN y Bag_CART en sus versiones LASSO pero logra mantener igual el desempeño de Stack2. El tiempo de entrenamiento promedio para las versiones LASSO de los clasificadores analizados disminuye un 15% sin cambios sustanciales en el tiempo de validación.

9.5.3 RESULTADOS DE LA SELECCIÓN DE CARACTERÍSTICAS CON ELASTIC NET

Se analizó la relación de las características mediante la correlación lineal de Pearson, dado que son numerosas variables se presenta una representación gráfica de dicha correlación en la figura 9.3. Se observa que existe correlación fuerte entre algunas características: 1bas- 7NE ($r=0,99$, $p<0,01$), 4TH-6RA ($r=0,99$, $p<0,01$) y 5PK-8ZC ($r=0,99$, $p<0,01$).

Cuando se presenta correlación importante entre alguna de las características, complementariamente a LASSO se realizó la selección mediante ELASTIC NET. Se trabajó este algoritmo con KNN, *Bagging* y Stack2 bajo los mismos fundamentos utilizados en las secciones anteriores. El sub conjunto de características para obtener mínimo error cuadrático medio promedio de los 22 clasificadores más un desvío estándar fue: 1bas, 4TH, 5PK, 6RA, 8ZC y 14SSD.

Se procedió a entrenar con este sub conjunto de características a los clasificadores propuestos. Los resultados promedio de los indicadores de desempeño con los datos de

validación para cada clasificador con el sub conjunto de características adoptado como resultado de *ELASTIC NET* se presentan en la tabla 9.10. Los tiempos de entrenamiento y validación promedio para cada caso figuran en la tabla 9.11.

Clasificadores	ACC	ESP	SEN	AUC	DOR
KNN_ELASTIC	0,720±0,010	0,760±0,015	0,670±0,019	0,780±0,010	6,400±0,623
Bag_CART_ELASTIC	0,890±0,004	0,910±0,006	0,880±0,006	0,920±0,004	71,730±6,850
Stack2_ELASTIC	0,940±0,002	0,950±0,003	0,930±0,004	0,940±0,002	233,140±17,580

Tabla 9.10: Medidas de desempeño promedio de los 22 clasificadores para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo ELASTIC NET. En negrita se indican los mayores valores de cada índice de desempeño.

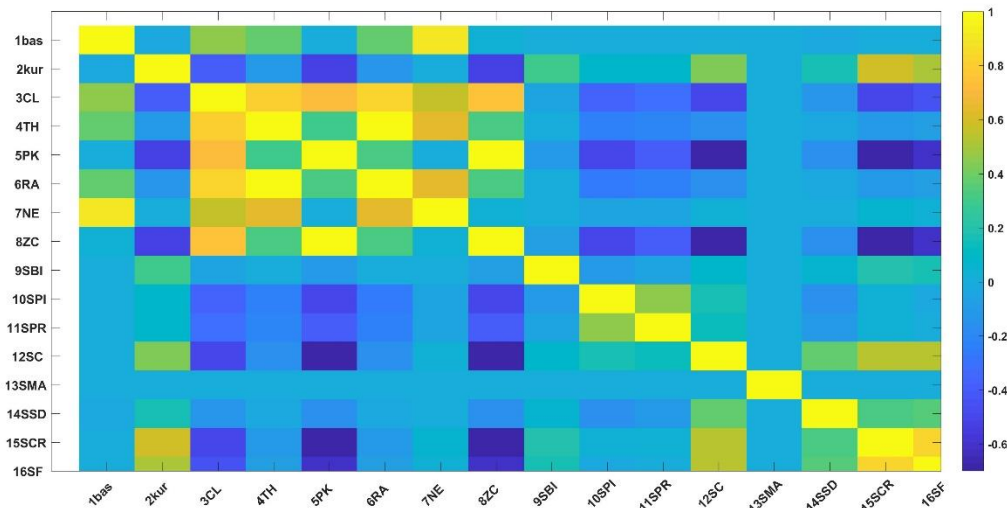


Figura 9.3: Correlación lineal de Pearson entre las 16 características temporales, la escala de color de la derecha indica los valores de correlación de gráfica de 0 (azul oscuro) a 1 (amarillo).

Clasificadores con ELASTIC NET	tiempo entrenamiento [s]	tiempo validación [s]
KNN_ELASTIC	0,070±0,030	0,018±0,008
Bag_CART_ELASTIC	28,561±1,437	0,642±0,058
Stack2_ELASTIC	43,952±1,999	1,227±0,087

Tabla 9.11: Tiempos de entrenamiento y clasificación promedio para KNN, *Bagging* (Bag_CART) y *Stack2* entrenados con el sub conjunto de características seleccionado por el algoritmo ELASTIC NET. En negrita se indican los menores valores de cada tiempo.

ELASTIC NET selecciona las mismas características que LASSO agregando 4TH la cual se encuentra altamente correlacionada con la 6RA. Refuerza de esta forma las características vinculadas a caracterizar la actividad de fondo y sólo una representa a las espigas. Es importante resaltar que solo comparte la característica 8ZC con las resultantes de ReliefF y *branch and bound*.

Las característica seleccionadas por *ELASTIC NET* disminuyen levemente el desempeño de KNN y Bag_CART pero logran mantener igual el desempeño de Stack2, repitiendo la misma situación de LASSO. De hecho, no existen mejoras en los desempeños promedio al agregar la característica 4TH, por lo que para los clasificadores analizados es recomendable trabajar con el sub conjunto seleccionado por LASSO y no por *ELASTIC NET*. El tiempo de entrenamiento promedio para las versiones *ELASTIC NET* de los clasificadores analizados disminuye un 12% sin cambios sustanciales en el tiempo de validación.

9.6 ANÁLISIS ESTADÍSTICO Y DE RESULTADOS DE LA SELECCIÓN DE CARACTERÍSTICAS DEL PRESENTE CAPÍTULO

El estudio estadístico se organizó de la siguiente forma para facilitar el análisis final:

a) primero se compararon estadísticamente los resultados de la autoselección de características (sección 9.5.1) que producen los clasificadores individuales cuyos algoritmos de entrenamiento permiten determinar un peso relativo de cada característica (Árboles de decisión tipo CART y CHAID) y luego con los resultados obtenidos de la autoselección de características de los clasificadores combinados que utilizan como base un mismo tipo de clasificador (*boosting*, *bagging* y *Random forest*).

b) estudio estadístico del mismo tipo clasificador comparados estadísticamente con distintos sub conjuntos de características según ReliefF (sección 9.3), *branch and bound* (sección 9.4), LASSO (sección 9.5.2) y Elastic Net (sección 9.5.3)

c) Comparación final de los algoritmos seleccionados en los ítems anteriores.

9.6.1 ESTUDIO ESTADÍSTICO Y ANÁLISIS DE RESULTADOS DE LOS MÉTODOS DE AUTOSELECCIÓN DE CARACTERÍSTICAS

En la sección 7.5 se compararon estadísticamente y se analizaron los resultados arrojados por los árboles de decisión, entrenados con las 16 características temporales, tipo CART sin podar, CART podados al 25%, 50% y 75%, como así también, los árboles tipo CHAID y sus versiones podadas, resultando de la comparación final entre todos los algoritmos de árboles de decisión que CHAID con poda del 50% presentaba el mejor rendimiento.

Por lo tanto en la presente sección se consideró CHAID podado al 50% entrenado con 16 características (CHAID50) como referencia y se comparó su desempeño con los clasificadores entrenados con el sub conjunto de características de la auto selección (sección 9.5.1): CART utilizando las características 2kur , 3CL, 5PK, 8ZC (CART_Autosel) y CHAID con 2kur , 3CL, 5PK, 8ZC (CHAID_Autosel).

Tomando las variables exactitud (ACC), especificidad (ESP), sensibilidad (SEN), índice de capacidad de diagnóstico (DOR) y los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val.) se analizó estadísticamente el rendimiento de todos los clasificadores siguiendo la metodología propuesta en la sección 6.2.1.

El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 9.12 y las figuras 9.4 y 9.5 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

Se observa en la figura 9.4 que CHAID50 presenta la mejor exactitud con un valor promedio de $0,734 \pm 0,010$ (tabla 7.6) y diferencias muy significativas ($p < 0,01$) en ACC tanto con CART_Autosel como con CHAID_Autosel (tabla 9.12).

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
CHAID50 vs CART_Autosel	<0,01	<0,01	0,73	0,73	<0,01	<0,01
CHAID50 vs CHAID_Autosel	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
CART_Autosel vs CHAID_Autosel	0,12	0,08	<0,01	<0,01	0,06	0,89

Tabla 9.12: Comparación estadística de a pares de clasificadores entre CHAID50, CART_Autosel y CHAID_Autosel como resultado de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación. Valores en negrita indican significancia estadística para $\alpha=0,05$.

Lo mismo ocurre cuando se analiza el resto de los indicadores de desempeño (ESP, SEN y DOR) entre CHAID50 y CHAID_Autosel (figura 9.4). Al comparar CHAID50 con CART_Autosel se observa que no hay diferencia estadísticamente significativa para SEN cuyo valor promedio es $0,670 \pm 0,020$ y $0,754 \pm 0,020$ y DOR con valor promedio de $7,180 \pm 0,670$ y $6,230 \pm 0,700$ respectivamente. CHAID50 presenta el mayor valor promedio de ESP con una diferencia muy significativa ($p < 0,01$) con CART_Autosel (tabla 9.12).

De la figura 9.5 se observa que CHAID50 presenta el menor tiempo de validación, con un valor promedio de $0,0013 \pm 0,0007$ s con diferencias muy significativas ($p < 0,01$) tanto con CART_Autosel como con CHAID_Autosel (tabla 9.12). Si bien estos últimos utilizan menos características no se encuentran podados y no logran mejorar los tiempos de validación de un árbol de decisión (CHAID50) entrenado con todas las características y podado al 50%. Se observa un mayor peso en la disminución de tiempos de validación de la poda sobre la selección de características.

Por lo tanto, del análisis de los párrafos anteriores puede inferirse que CHAID con poda del 50% y entrenado con todas las características (**CHAID50**) presenta el mejor rendimiento relativo de este grupo (tabla 9.12) utilizando los datos de la presente tesis.

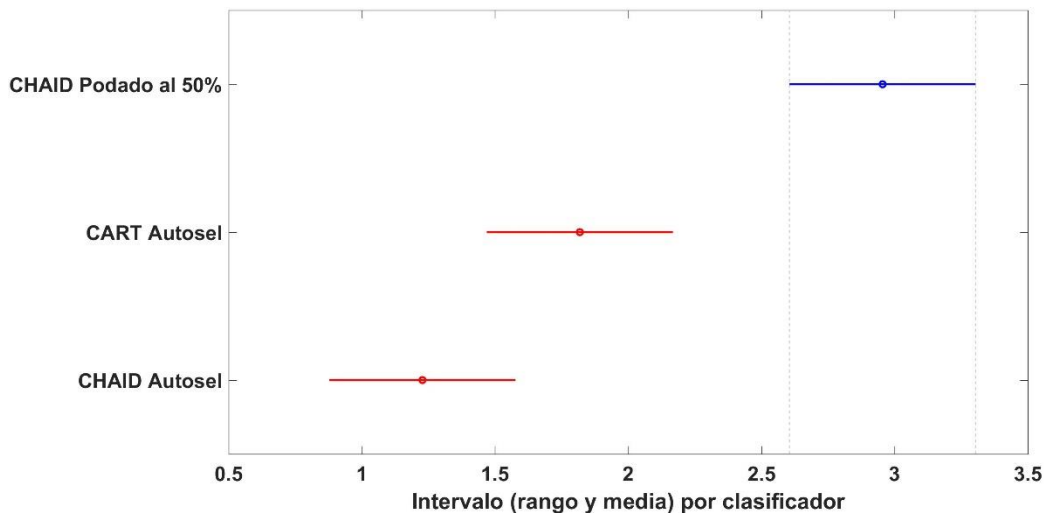


Figura 9.4: Comparación de clasificadores detallados en la tabla 9.12 para la exactitud (ACC) como resultados del test de Nemenyi. La grafica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado CHAID50 de mayor ACC y las líneas rojas indican diferencias significativas de los clasificadores CART_Autosel y CHAID_Autosel.

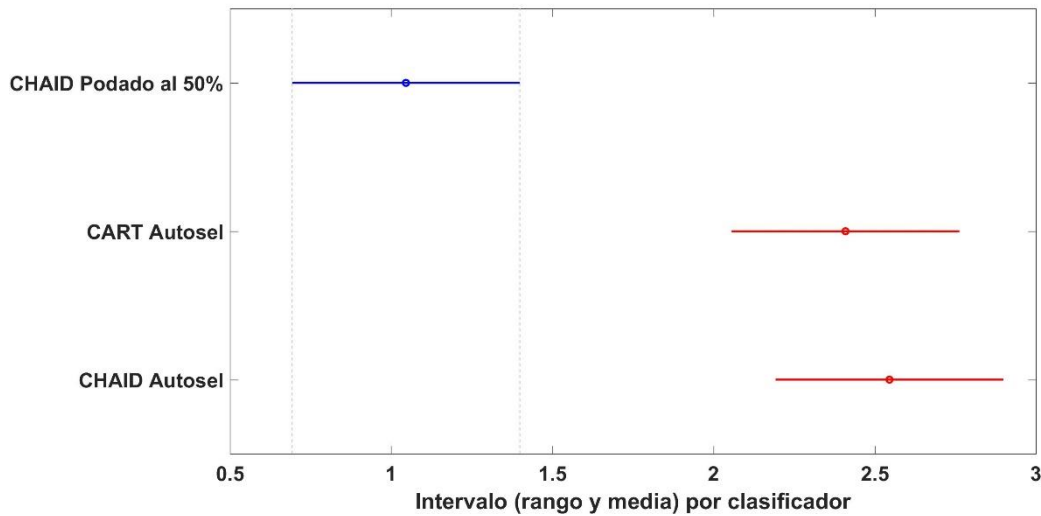


Figura 9.5: Comparación de clasificadores detallados en la tabla 9.12 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado CHAID50 de menor tiempo de validación y las líneas rojas indican diferencias significativas de los clasificadores CART_Autosel y CHAID_Autosel.

En la sección 8.10 se compararon estadísticamente y se analizaron los resultados arrojados por los algoritmos que combinan clasificadores. En particular *bagging* utilizando árboles base tipo CART sin podar y entrenado con las 16 características (Bag_CART) resultó el de mejor rendimiento dentro de los algoritmos que combinan un mismo tipo de clasificadores de base, por lo tanto se lo tomó como referencia y se comparó su desempeño con los clasificadores entrenados con el sub conjunto de características de la auto selección (sección 9.5.1 y tabla 9.5): *GentleBoost* utilizando las características 1bas, 2kur, 3CL, 5PK, 7NE, 8ZC (GentleBoost_Autosel), *bagging* entrenado con 2kur, 3CL, 5PK, 8ZC (Bagging_Autosel) y *Random forest* con 2kur, 3CL, 5PK, 8ZC (RForest_Autosel).

Se estudió estadísticamente mediante el test de Friedman las variables de desempeño que fueron definidas anteriormente: ACC, ESP, SEN y DOR junto con los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val). El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se

presentan en la tabla 9.13 y las figuras 9.6 y 9.7 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART vs Bagging_Autosel	0,04	0,04	0,09	0,04	<0,01	<0,01
Bag_CART vs GentleBoost_Autosel	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART vs RForest_Autosel	<0,01	<0,01	<0,01	<0,01	0,05	0,15
Bagging_Autosel vs GentleBoost_Autosel	<0,01	<0,01	<0,01	<0,01	0,05	0,05
Bagging_Autosel vs RForest_Autosel	0,05	0,05	0,03	0,05	0,05	0,03
GentleBoost_Autosel vs RForest_Autosel	0,05	0,05	0,04	0,05	<0,01	<0,01

Tabla 9.13: Comparación estadística de a pares de clasificadores entre Bag_CART, Bagging_Autosel, GentleBoost_Autosel y RForest_Autosel como resultado de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación. Valores en negrita indican significancia estadística para $\alpha=0,05$.

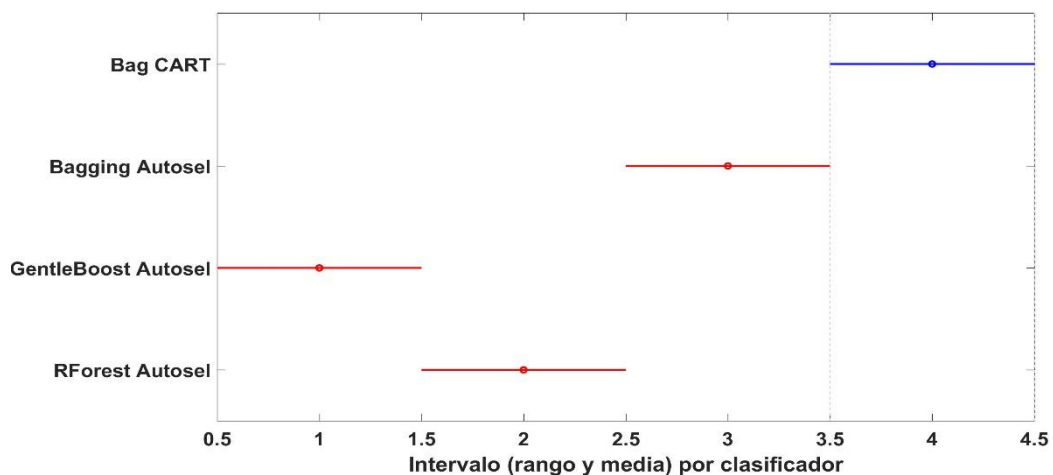


Figura 9.6: Comparación de clasificadores detallados en la tabla 9.13 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p<0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Bag_CART de mayor ACC y las líneas rojas indican diferencias significativas de los clasificadores Bagging_Autosel, GentleBoost_Autosel y RForest_Autosel.

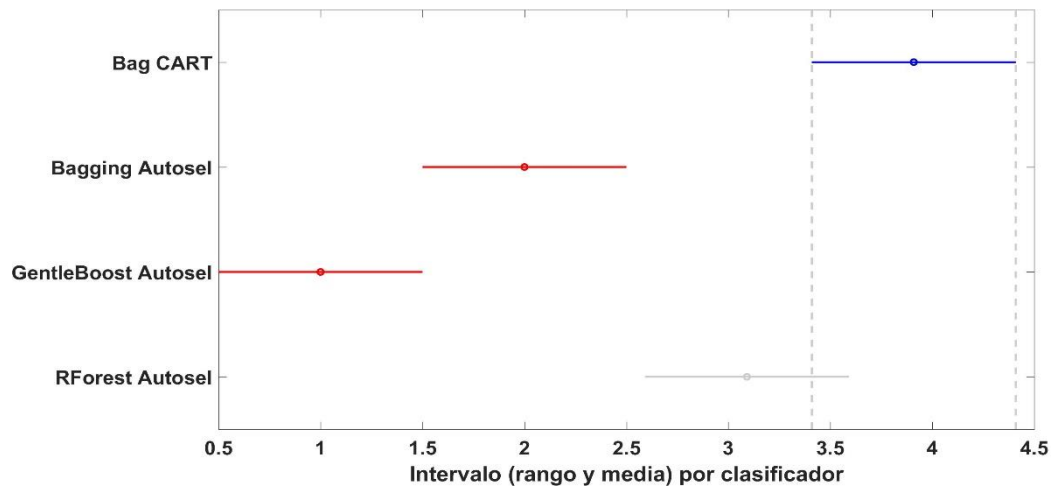


Figura 9.7: Comparación de clasificadores detallados en la tabla 9.12 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Bag_CART de mayor tiempo de validación, las líneas rojas indican diferencias significativas de los clasificadores Bagging_Autoselección y GentleBoost_Autoselección. La línea gris indica que no presenta diferencia significativa RForest_Autoselección.

Se observa en la figura 9.6 que Bag_CART presenta la mejor exactitud al compararlo con el resto de los clasificadores de este grupo con un valor promedio de $0,923 \pm 0,004$ (tabla 8.3), siendo una diferencia estadísticamente significativa ($p = 0,04$) con Bagging_Autoselección y muy significativa ($p < 0,01$) con GentleBoost_Autoselección y RForest_Autoselección (tabla 9.13). En los indicadores de desempeño ESP y DOR ocurre la misma situación (tabla 9.13). Para el caso de la especificidad (ESP) Bag_CART presenta una diferencia no significativa con Bagging_Autoselección ($p = 0,09$) y muy significativa ($p < 0,01$) con GentleBoost_Autoselección y RForest_Autoselección (tabla 9.13). En general puede asumirse que Bag_CART tiene mejor desempeño que el resto de los clasificadores de este grupo.

De la figura 9.7 se observa que Bag_CART presenta el mayor tiempo de validación, con un valor promedio de $0,601 \pm 0,030$ s (tabla 8.3) resultando una diferencia estadísticamente muy significativa ($p < 0,01$) al compararlo con el de menor tiempo de validación promedio de $0,405 \pm 0,020$ s que resultó GentleBoost_Autoselección (tabla 9.7). En este caso GentleBoost al trabajar con 6 características presenta un tiempo menor estadísticamente significativo que Bag_CART. En este caso el clasificador más rápido es el de peor desempeño y viceversa (figura 9.6). Al

analizar los otros clasificadores no se puede seleccionar ninguno que tenga una solución de compromiso entre el desempeño y el tiempo de validación.

Se elige de este grupo a **Bag_CART** bajo las condiciones experimentales de la presente tesis.

9.6.2 ESTUDIO ESTADÍSTICO Y ANÁLISIS DE RESULTADOS DE LOS MÉTODOS RELIEFF, BRANCH AND BOUND, LASSO Y ELASTIC NET

En la presente sección se comparó un mismo tipo de algoritmo de clasificación frente a sus versiones entrenadas con distintos sub grupos de características según se detalla en la tabla 9.14.

Se inició con KNN comparando el desempeño del clasificador entrenado con las 16 características temporales, frente a sus versiones KNN_ReliefF, KNN BBound, KNN_LASSO y KNN_ELASTIC según se detalla en la tabla 9.14. Siguiendo la metodología descrita en la sección 6.2.1, se estudió estadísticamente mediante el test de Friedman las variables de desempeño que fueron definidas anteriormente: ACC, ESP, SEN y DOR junto con los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val). El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 9.15 y las figuras 9.8 y 9.9 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

Se observa en la figura 9.8 que KNN_BBound presenta la mejor exactitud al compararlo con el resto de los clasificadores de este grupo, con un valor promedio de $0,78 \pm 0,010$ (tabla 9.3), siendo una diferencia estadísticamente muy significativa ($p < 0,01$) con KNN_ReliefF, KNN_LASSO y KNN_ELASCTIC tanto para ACC como para el resto de los indicadores desempeño ESP, SEN y DOR (tabla 9.13). Al comparar KNN_BBound con KNN se observa que no presenta diferencias significativas en ACC ($p = 0,2$), SEN ($p = 0,18$) y DOR ($p = 0,18$). Con ESP, la comparación de estos mismos clasificadores, arroja una diferencia significativa ($p = 0,03$) a favor de KNN_BBound con un valor promedio de $0,820 \pm 0,015$ (tabla 9.3). Puede considerarse que KNN_BBound tiene el mejor desempeño de los clasificadores de este grupo.

Clasificador	Algoritmo Selector de características	Denominación	Características de entrenamiento
KNN	-	KNN	las 16 características
	ReliefF	KNN_ReliefF	2kur ,8ZC, 12SC, 15SCR, 16SF
	<i>Branch and bound</i>	KNN_BBoud	2kur, 3CL, 5PK, 8ZC, 12SC, 15SCR
	LASSO	KNN_LASSO	1bas, 5PK, 6RA, 8ZC, 14SSD
	Elastic Net	KNN_ELASTIC	1bas, 4TH, 5PK, 6RA, 8ZC y 14SSD
Bagging	-	Bag_CART	las 16 características
	ReliefF	Bag_CART_ReliefF	2kur ,8ZC, 12SC, 15SCR, 16SF
	<i>Branch and bound</i>	Bag_CART_BBoud	2kur, 3CL, 5PK, 8ZC, 12SC,15SCR
	LASSO	Bag_CART_LASSO	1bas, 5PK, 6RA, 8ZC, 14SSD
	Elastic Net	Bag_CART_ELASTIC	1bas, 4TH, 5PK, 6RA, 8ZC, 14SSD
Stack2	-	Stack2	las 16 características
	ReliefF	Stack2_ReliefF	2kur ,8ZC, 12SC, 15SCR, 16SF
	<i>Branch and bound</i>	Stack2_BBoud	2kur, 3CL, 5PK, 8ZC, 12SC,15SCR
	LASSO	Stack2_LASSO	1bas, 5PK, 6RA, 8ZC ,14SSD
	Elastic Net	Stack2_ELASTIC	1bas, 4TH, 5PK, 6RA, 8ZC, 14SSD

Tabla 9.14: Clasificadores, métodos de selección de características aplicados, denominación y características utilizadas para el entrenamiento de los clasificadores analizados estadísticamente en la sección 9.6.2.

De la figura 9.9 se observa que KNN_ReliefF presenta el menor tiempo de validación, con un valor promedio de $0,011 \pm 0,006$ s (tabla 9.2) y diferencias estadísticamente muy significativas ($p < 0,01$) al compararlo con KNN, KNN_LASSO y KNN_ELASTIC. Con KNN_BBoud arroja una diferencia significativa ($p = 0,03$) siendo el segundo menor tiempo promedio de validación ($0,012 \pm 0,001$ s, tabla 9.4). Asimismo, si bien KNN_ReliefF es el algoritmo más rápido en validar de este grupo, presenta el menor desempeño general.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
KNN vs KNN_ReliefF	<0,01	0,22	<0,01	<0,01	<0,01	<0,01
KNN vs KNN_BBound	0,20	0,03	0,18	0,18	<0,01	<0,01
KNN vs KNN_LASSO	<0,01	<0,01	0,04	<0,01	0,03	<0,01
KNN vs KNN_ELASTIC	<0,01	<0,01	0,31	0,01	<0,01	0,10
KNN_ReliefF vs KNN_BBound	<0,01	<0,01	<0,01	<0,01	0,10	0,03
KNN_ReliefF vs KNN_LASSO	0,73	0,54	0,02	0,61	<0,01	<0,01
KNN_ReliefF vs KNN_ELASTIC	0,54	0,42	<0,01	0,22	1,00	<0,01
KNN_BBound vs KNN_LASSO	<0,01	<0,01	<0,01	<0,01	0,61	0,94
KNN_BBound vs KNN_ELASTIC	<0,01	<0,01	<0,01	<0,01	0,05	0,03
KNN_LASSO vs KNN_ELASTIC	1,00	1,00	0,89	0,96	<0,01	0,18

Tabla 9.15: Resultados de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación cuando se compara de a pares el clasificador KNN entrenado con 16 características (KNN) y entrenado con sub conjunto de características según algoritmos ReliefF, *branch and bound*, LASSO y Elastic Net, Valores en negrita indican significancia estadística para $\alpha=0,05$.

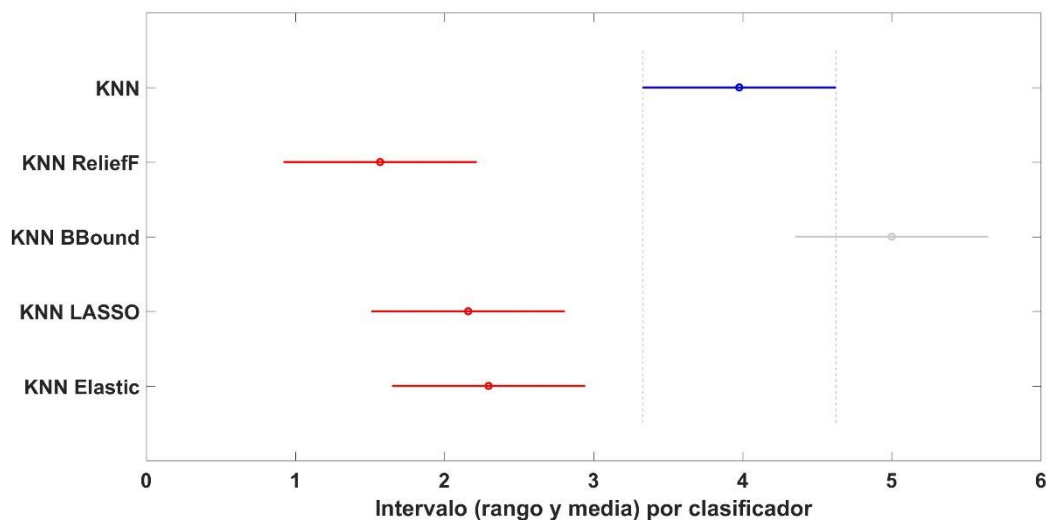


Figura 9.8: Comparación de clasificadores KNN detallados en la tabla 9.15 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p<0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado KNN, las líneas rojas indican diferencias significativas de KNN con los clasificadores KNN_ReliefF, KNN_LASSO y KNN_ELASCTIC. La línea gris indica que KNN no presenta diferencias significativas con KK_BBound de de mayor ACC.

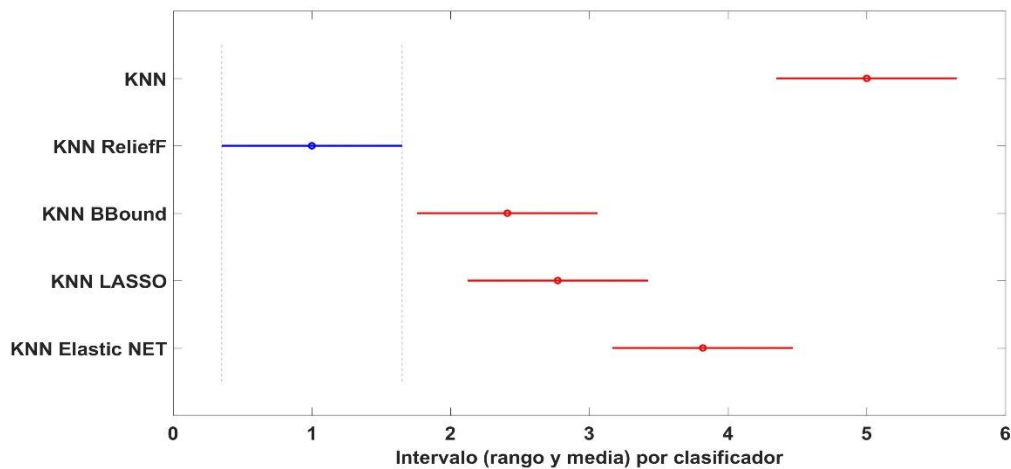


Figura 9.9: Comparación de clasificadores KNN detallados en la tabla 9.15 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado con menor tiempo KNN_ReliefF y las líneas rojas indican diferencias significativas con KNN, KNN_BBound, KNN_LASSO y KNN_Elastic NET.

Por lo tanto, se elige de este grupo a **KNN_BBound** como el de mejor rendimiento de este grupo con el mejor desempeño relativo y el segundo mejor tiempo de validación.

Se continuó el estudio con *Bagging* comparando el desempeño del clasificador entrenado con las 16 características temporales, frente a sus versiones Bag_CART_ReliefF, Bag_CART_BBound, Bag_CART_LASSO y Bag_CART_ELASTIC según se detalla en la tabla 9.14. Siguiendo la metodología descrita en la sección 6.2.1, se estudió estadísticamente mediante el test de Friedman las variables de desempeño que fueron definidas anteriormente: ACC, ESP, SEN y DOR junto con los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val).

El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 9.16 y las figuras 9.10 y 9.11 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART vs Bag_CART_ReliefF	<0,01	<0,01	<0,01	<0,01	0,31	0,99
Bag_CART vs Bag_CART_BBound	0,31	0,21	1,00	0,31	<0,01	0,01
Bag_CART vs Bag_CART_LASSO	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART vs Bag_CART_ELASTIC	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART_ReliefF vs Bag_CART_BBound	<0,01	<0,01	<0,01	<0,01	0,42	<0,01
Bag_CART_ReliefF vs Bag_CART_LASSO	0,18	1,00	0,12	0,18	<0,01	<0,01
Bag_CART_ReliefF vs Bag_CART_ELASTIC	<0,01	0,85	<0,01	<0,01	<0,01	<0,01
Bag_CART_BBound vs Bag_CART_LASSO	<0,01	<0,01	<0,01	<0,01	<0,01	0,37
Bag_CART_BBound vs Bag_CART_ELASTIC	0,09	<0,01	0,01	0,10	0,01	0,94
Bag_CART_LASSO vs Bag_CART_ELASTIC	0,48	0,78	0,54	0,48	1,00	0,83

Tabla 9.16: Resultados de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación cuando se compara de a pares el clasificador *Bagging* entrenado con 16 características (Bag_CART) y entrenado con sub conjunto de características según algoritmos ReliefF, *branch and bound*, LASSO y Elastic Net, Valores en negrita indican significancia estadística para $\alpha=0,05$.

Se observa en la figura 9.10 que Bag_CART presenta la mejor exactitud al compararlo con el resto de los clasificadores de este grupo, con un valor promedio de $0,923 \pm 0,004$ (tabla 8.3), siendo una diferencia estadísticamente muy significativa ($p < 0,01$) con Bag_CART_ReliefF, Bag_CART_LASSO y Bag_CART_ELASCTIC, tanto para ACC como para el resto de los indicadores desempeño ESP, SEN y DOR (tabla 9.14). Al comparar Bag_CART con Bag_CART_BBound se observa que no presenta diferencias significativas con todos los indicadores de desempeño (ACC con $p=0,31$; ESP con $p=0,20$; SEN con $p=1,00$ y DOR con $p=0,31$). Por lo tanto, puede asumirse que Bag_CART y Bag_CART_BBound tienen desempeños estadísticamente similares.

De la figura 9.11 se observa que Bag_CART_LASSO presenta el menor tiempo de validación, con un valor promedio de $0,623 \pm 0,138$ s (tabla 9.9) y diferencias estadísticamente muy significativas ($p < 0,01$) al compararlo con Bag_CART y Bag_CART_ReliefF. Con

Bag_CART_BBound y Bag_CART_ELASTIC no presenta diferencias significativas ($p=0,94$ y $p=0,83$ respectivamente).

Por lo antes analizado se elige de este grupo a **Bag_CART_BBound** como una opción intermedia entre buen desempeño y velocidad de clasificación de los datos de validación bajo las condiciones experimentales de la presente tesis.

Finalmente se realizó la comparación del algoritmo meta clasificador Stack2, desarrollado en la sección 8.9, comparando el desempeño del mismo entrenado con las 16 características temporales, frente a sus versiones Stack2_ReliefF, Stack2_BBound, Stack2_LASSO y Stack2_ELASTIC según se detalla en la tabla 9.14. Siguiendo la metodología descrita en la sección 6.2.1, se estudió estadísticamente mediante el test de Friedman las variables de desempeño que fueron definidas anteriormente: ACC, ESP, SEN y DOR junto con los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val).

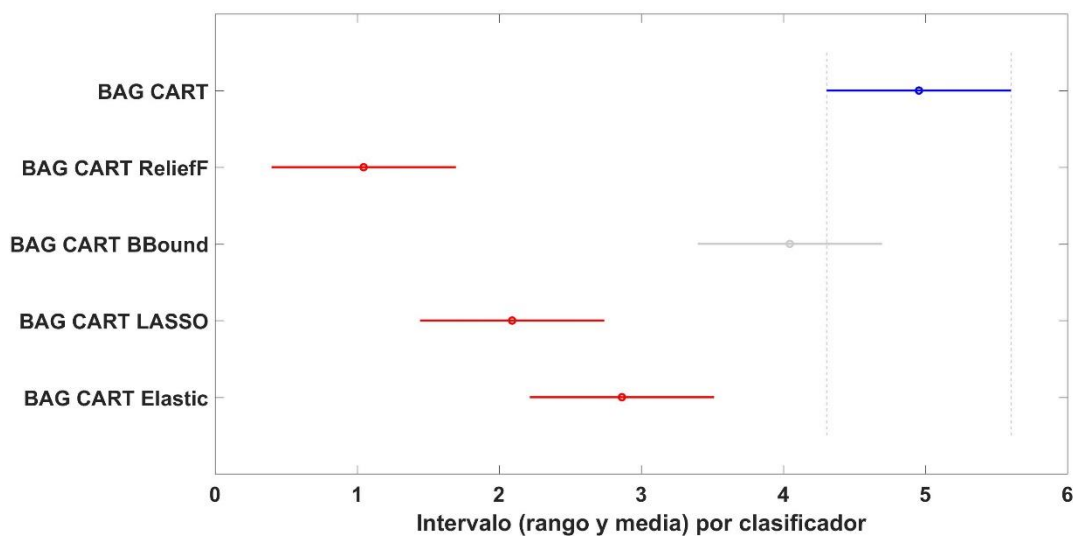


Figura 9.10: Comparación de clasificadores *Bagging* detallados en la tabla 9.16 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p<0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Bag_CART de mayor ACC, las líneas rojas indican diferencias significativas de Bag_CART con los clasificadores Bag_CART_ReliefF, Bag_CART_LASSO y Bag_CART_ELASTIC. La línea gris indica que Bag_CART no presenta diferencias significativas con Bag_CART_BBound.

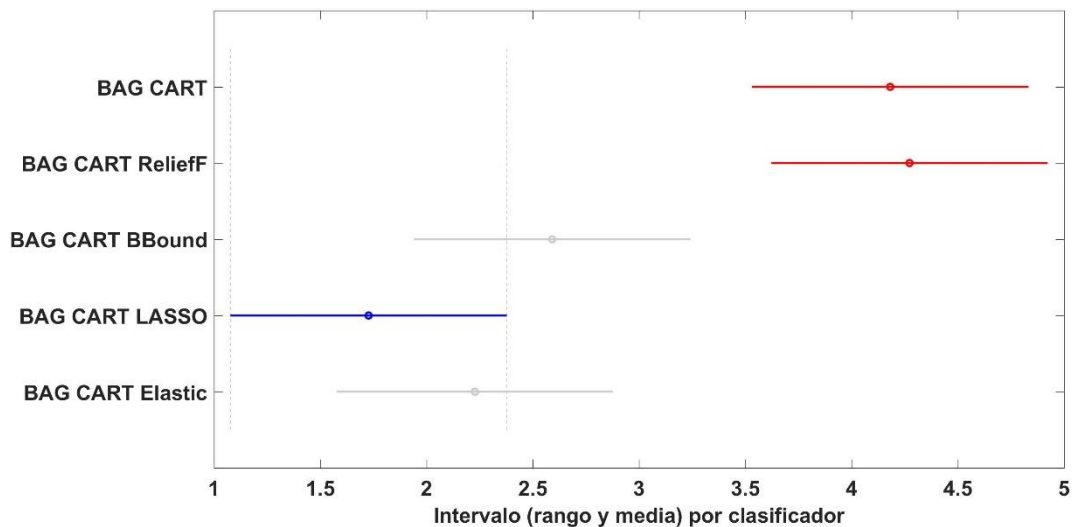


Figura 9.11: Comparación de clasificadores *Bagging* detallados en la tabla 9.16 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Bag_CART_LASSO de menor tiempo, las líneas rojas indican diferencias significativas del seleccionado con Bag_CART y Bag_CART_ReliefF. La línea gris indica que Bag_CART_LASSO no presenta diferencias significativas con Bag_CART_BBound y Bag_CART_ELASTIC.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Stack2 vs Stack2_ReliefF	<0,01	0,14	<0,01	<0,01	0,61	1,00
Stack2 vs Stack2_BBound	0,20	0,04	0,17	0,20	<0,01	<0,01
Stack2 vs Stack2_LASSO	<0,01	<0,01	0,04	<0,01	<0,01	<0,01
Stack2 vs Stack2_ELASTIC	<0,01	<0,01	0,19	<0,01	<0,01	<0,01
Stack2_ReliefF vs Stack2_BBound	<0,01	<0,01	<0,01	<0,01	0,12	<0,01
Stack2_ReliefF vs Stack2_LASSO	0,69	0,53	0,02	0,61	<0,01	<0,01
Stack2_ReliefF vs Stack2_ELASTIC	0,39	0,53	<0,01	0,39	<0,01	<0,01
Stack2_BBound vs Stack2_LASSO	<0,01	<0,01	<0,01	<0,01	<0,01	0,18
Stack2_BBound vs Stack2_ELASTIC	<0,01	<0,01	<0,01	<0,01	0,08	1,00
Stack2_LASSO vs Stack2_ELASTIC	0,99	1,00	0,96	1,00	0,83	0,10

Tabla 9.17: Resultados de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación cuando se compara de a pares el clasificador *Stack2* entrenado con 16 características y entrenado con sub conjunto de características según algoritmos ReliefF, *branch and bound*, LASSO y Elastic Net, Valores en negrita indican significancia estadística para $\alpha = 0,05$.

El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 9.17 y las figuras 9.12 y 9.13 representan la comparación de rangos para ACC y tiempos de validación respectivamente.

Se observa en la figura 9.10 que Stack2_BBoud presenta la mejor exactitud al compararlo con el resto de los clasificadores de este grupo, con un valor promedio de $0,95 \pm 0,002$ (tabla 9.3), siendo una diferencia estadísticamente muy significativa ($p < 0,01$) con Stack2_ReliefF, Stack2_LASSO y Stack2_ELASCTIC, tanto para ACC como para el resto de los indicadores desempeño ESP, SEN y DOR (tabla 9.17). Al comparar Stack2 con Stack2_BBoud se observa que no presenta diferencias significativas con la mayoría de los indicadores de desempeño (ACC con $p = 0,20$; SEN con $p = 0,17$ y DOR con $p = 0,20$). En ESP la diferencia es significativa ($p = 0,04$) a favor Stack2_BBoud que presenta un valor de $0,960 \pm 0,003$ (tabla 9.3). Por lo tanto Stack2_BBoud tienen un desempeño superior al compararlo estadísticamente con los otros clasificadores de este grupo.

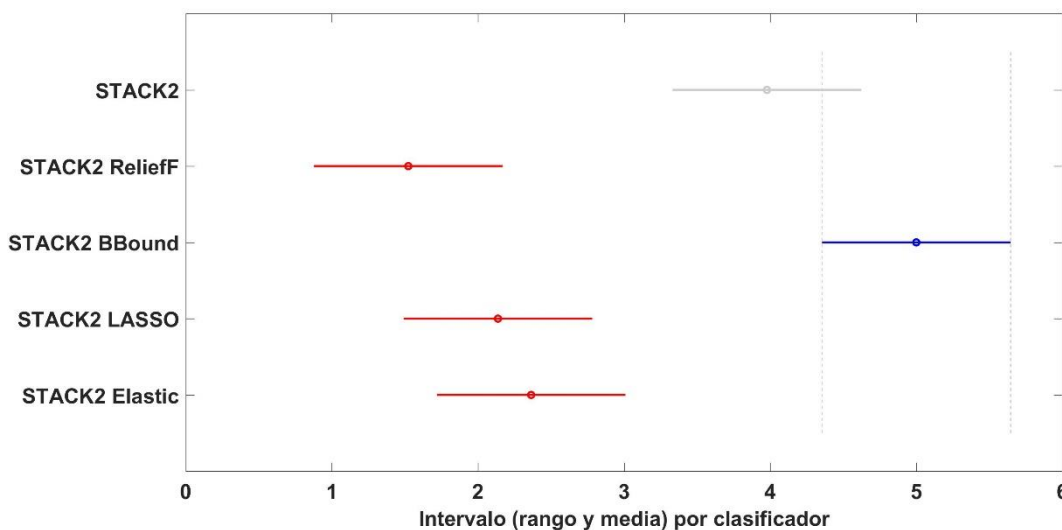


Figura 9.12: Comparación de clasificadores *Stack2* detallados en la tabla 9.17 para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Stack2_BBoud de mayor ACC, las líneas rojas indican diferencias significativas de Stack2_BBoud con los clasificadores Stack2_ReliefF, Stack2_LASSO y Stack2_ELASTIC. La línea gris indica que Stack2_BBoud no presenta diferencias significativas con Stack2.

De la figura 9.13 se observa que Stack2_LASSO presenta el menor tiempo de validación, con un valor promedio de $1,260 \pm 0,130$ s (tabla 9.9) y diferencias estadísticamente muy significativas ($p < 0,01$) al compararlo con Stack2 y Stack2_ReliefF. Con Stack2_BBound y Stack2_ELASTIC no presenta diferencias significativas ($p = 0,18$ y $p = 0,10$ respectivamente).

Por lo antes analizado y bajo las condiciones experimentales de la presente tesis se considera que **Stack2_BBound** resulta la mejor opción de este grupo, ya que tiene el mayor desempeño y no presenta diferencias significativas con el más rápido en clasificar de este grupo.

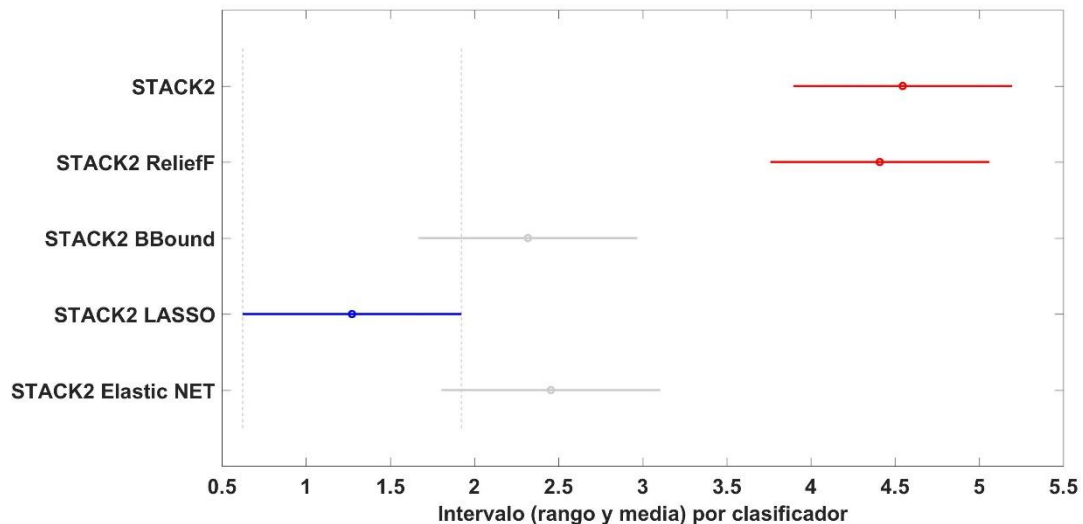


Figura 9.13: Comparación de clasificadores *Stack2* detallados en la tabla 9.17 para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Stack2_LASSO de menor tiempo, las líneas rojas indican diferencias significativas del seleccionado con Stack2 y Stack2_ReliefF. La línea gris indica que Stack2_LASSO no presenta diferencias significativas con Stack2_BBound y Stack2_ELASTIC.

9.6.3 COMPARACIÓN DE LOS CLASIFICADORES CON MEJOR RENDIMIENTO

En la presente sección se estudiará comparativamente los clasificadores que fueron seleccionados en las secciones 9.6.1 y 9.6.2: árbol de decisión tipo CHAID podado al 50% y entrenado con todas las características al que se ha denominado CHAID50 y 3 clasificadores entrenados con el sub conjunto de características que seleccionó el algoritmo *branch and bound*: KNN al que se ha denominado KNN_BBound (tabla 9.14 y tabla 9.15), Bagging con

árboles base tipo CART sin podar al que se ha denominado Bag_CART_BBoud (tabla 9.14 y tabla 9.14) y Stack2 al que se ha denominado Stack2_BBoud (tabla 9.14 y tabla 9.17).

Siguiendo la metodología descrita en la sección 6.2.1, se estudió estadísticamente mediante el test de Friedman las variables de desempeño que fueron definidas anteriormente: ACC, ESP, SEN y DOR junto con los tiempos de entrenamiento (tiempo Entre.) y tiempo de validación (tiempo Val). El test de Friedman arrojó significancia global ($p < 0,01$) para todos los indicadores, por lo que se procedió a realizar Nemenyi para la comparación de a pares de clasificadores. Los resultados completos se presentan en la tabla 9.18 y las figuras 9.14, 9.15 y 9.16 representan el diagrama de cajas, la comparación de rangos para ACC y tiempos de validación respectivamente.

	valor p ACC	valor p ESP	valor p SEN	valor p DOR	valor p tiempo Entre.	valor p tiempo Val.
Significancia global Friedman	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
CHAID50 vs KNN_BBoud	0,04	0,04	0,04	0,04	0,04	0,04
CHAID50 vs Bag_CART_BBoud	<0,01	<0,01	<0,01	<0,01	0,05	<0,01
CHAID50 vs Stack2_BBoud	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
KNN_BBoud vs Bag_CART_BBoud	0,04	0,04	0,04	0,04	<0,01	0,04
KNN_BBoud vs Stack2_BBoud	<0,01	<0,01	<0,01	<0,01	<0,01	<0,01
Bag_CART_BBoud vs Stack2_BBoud	0,04	0,04	0,04	0,04	0,04	0,09

Tabla 9.18: Resultados de los test de Friedman y Nemenyi para cada indicador de desempeño y tiempos de entrenamiento y validación cuando se compara de a pares los clasificadores con mejor rendimiento que han resultado de la selección de características: CHAID50, KNN_BBoud, Bag_CART_BBoud y Stack2_BBoud. Valores en negrita indican significancia estadística para $\alpha=0,05$.

Se observa en la figura 9.14 que Stack2_BBoud presenta la mejor exactitud al compararlo con el resto de los clasificadores de este grupo con poca dispersión de datos y un valor medio de ACC de $0,950 \pm 0,002$, siendo una diferencia estadísticamente muy significativa ($p < 0,01$) o significativa con CHAID50, KNN_BBoud y Bag_CART_BBoud, tanto para ACC como para el resto de los indicadores de desempeño ESP, SEN y DOR (tabla 9.18). El segundo clasificador con mejor desempeño promedio es Bag_CART_BBoud (figura 9.14) con un valor promedio de ACC de $0,920 \pm 0,004$ presentando, según la tabla 9.18, diferencias estadísticamente significativas con KNN_BBoud ($p=0,04$) y muy significativas con CHAID40 ($p < 0,01$). Siguiendo en el orden de desempeño continúan KNN BBoud con un valor promedio de ACC de $0,780 \pm 0,010$ y CHAID50 con $0,734 \pm 0,010$ (figura 9.14), presentando entre ellos

tanto para ACC como para el resto de los indicadores de desempeño diferencias significativas ($p=0,04$). Por lo tanto, Stack2_BBound tienen un desempeño superior al compararlo estadísticamente con los otros clasificadores de este grupo.

Analizando al desempeño en general es claro que el incremento de la complejidad de los clasificadores analizados va logrando mejores índices de desempeño. Por ejemplo, si se considera ACC se inicia con un promedio de exactitud del 73% utilizando CHAID50 hasta llegar al 95% de Stack2_BBound, representando un incremento porcentual del 30%.

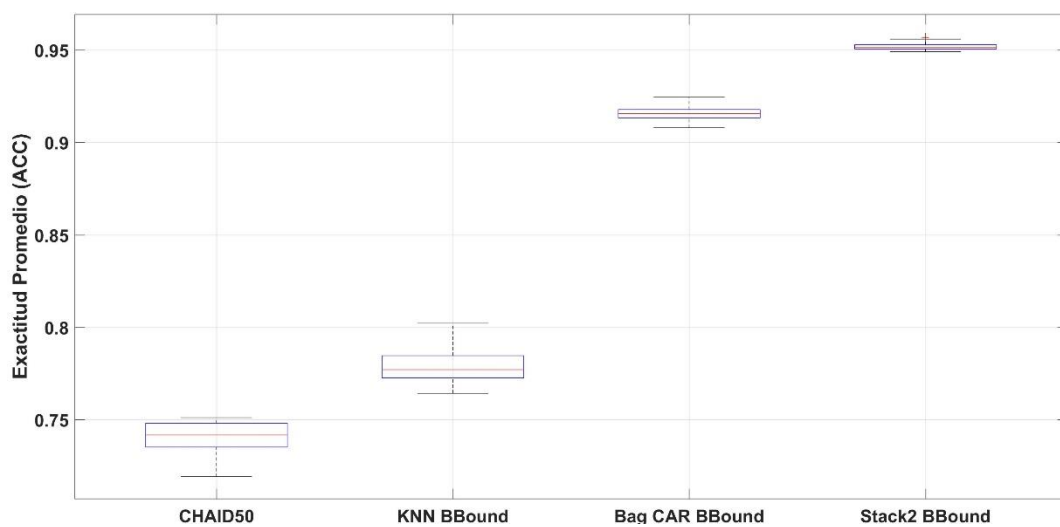


Figura 9.14: Diagrama de cajas para la exactitud promedio (ACC) de CHAID50, KNN_BBound, Bag_CART_BBound y Stack2_BBound.

Al incrementar la complejidad algorítmica de los clasificadores también crecen sus tiempos de entrenamiento y validación como puede observarse claramente en el orden de los rangos de la figura 9.16. Para los clasificadores que se están analizando, el proceso de clasificación de los datos de validación que demandó menos tiempos fue CHAID50 con $0,0008 \pm 0,0001$ s, continuando en orden creciente KNN_BBound con $0,012 \pm 0,001$ s, Bag_CART_BBound con $0,623 \pm 0,138$ s y el de mejor desempeño con el mayor tiempo de validación de $1,047 \pm 0,280$ s. Todos estos tiempos son admisibles para los objetivos de la presente tesis según se definió en la sección 6.2.2. De la figura 9.13 y la tabla 9.18 se observa que CHAID50 presenta diferencias estadísticamente muy significativas ($p < 0,01$) al compararlo con Bag_CART_BBound y Stack2_BBound y significativa ($p = 0,04$) con KNN_BBound.

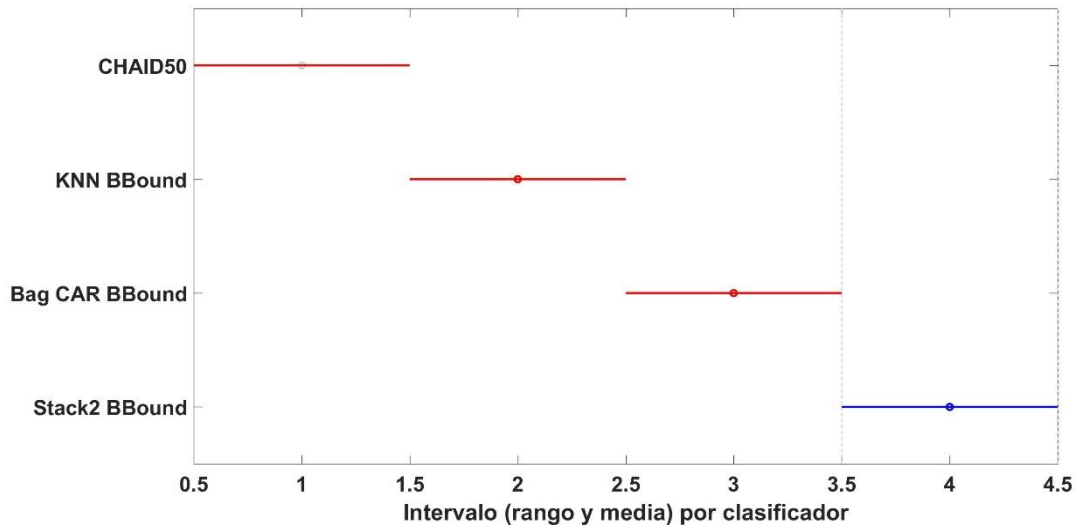


Figura 9.15: Comparación de los clasificadores CHAID50, KNN_BBound, Bag_CART_BBound y Stack2_BBound para la exactitud (ACC) como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado Stack2_BBound de mayor ACC, las líneas rojas indican diferencias significativas de Stack2_BBound con todos los otros clasificadores.

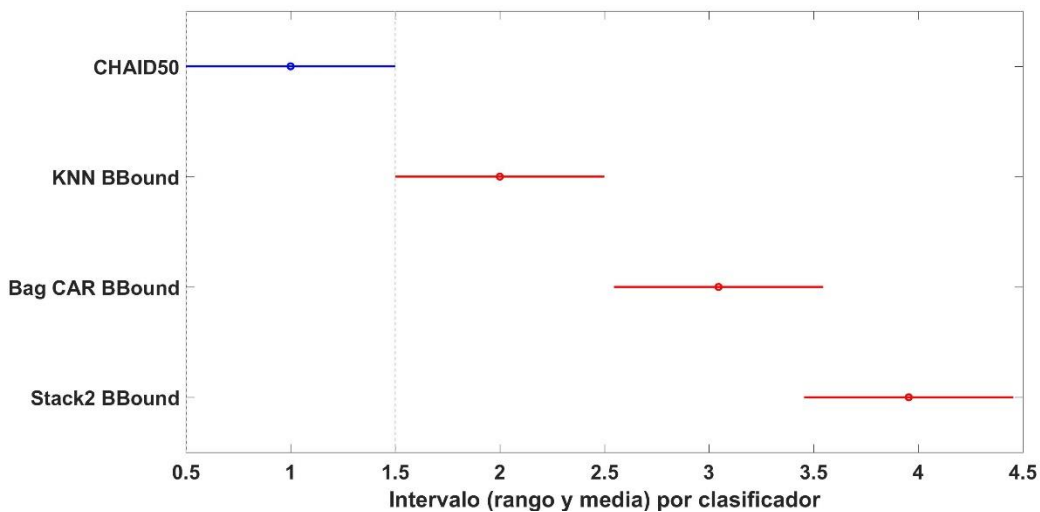


Figura 9.16: Comparación de los clasificadores CHAID50, KNN_BBound, Bag_CART_BBound y Stack2_BBound para el tiempo de validación como resultados del test de Nemenyi. La gráfica representa las estimaciones y los intervalos de comparación donde cada media del grupo está representada por un círculo y el intervalo representado por una línea. Dos medias grupales son significativamente diferentes ($p < 0,05$) si sus intervalos son disjuntos. La línea azul indica el clasificador seleccionado CHAID50 de menor tiempo, las líneas rojas indican diferencias significativas de CHAID50 con todos los otros clasificadores.

Stack2_BBound con el mejor desempeño y Bag_CART_BBound como el segundo mejor pueden resultar opciones interesantes bajo las condiciones experimentales de la presente tesis.

En el capítulo 10 de Discusiones y Conclusiones se abordará un análisis general de los clasificadores, por los que los resultados, comparaciones y discusiones del capítulo 9 se retomaran para enriquecer la discusión final con eje en la aplicación para lo cual fueron diseñados los clasificadores de la presente tesis.

Capítulo 10 Discusiones y conclusiones

10.1 CLASIFICADORES INDIVIDUALES

De los clasificadores individuales analizados se observa que KNN presentó mayor desempeño promedio con valores porcentuales de exactitud del 75,1%, especificidad 78,6%, sensibilidad 70% y valor promedio de DOR de 8,713. KNN tuvo menor capacidad para clasificar como STN los registros etiquetados STN en la base de datos, dada la menor sensibilidad al compararla con la especificidad. Este clasificador presentó el menor tiempo de entrenamiento promedio $0,031 \pm 0,006$ s y un tiempo de validación muy aceptable de $0,196 \pm 0,018$ s. En general se observa que el algoritmo KNN es mucho más veloz para entrenar que para clasificar nuevos datos.

Branch and bound permitió seleccionar las características 2kur, 3CL, 5PK, 8ZC, 12SC y 15SCR que incrementaron el rendimiento de KNN logrando índices de desempeño porcentuales de ACC del 78%, ESP 82%, SEN 72% y DOR con un valor de 11,82. Para el caso de DOR que representa la capacidad de discriminación de clases el incremento fue de aproximadamente un 21%. Si bien las características que maximizaron su rendimiento corresponden a una mezcla de las que describen la actividad de base (2kur, 3CL, 5PK, 8ZC) y las vinculadas a las espigas (12SC y 15SCR) es claro que la proporción de las primeras es superior e indicaría que tienen mayor capacidad para discriminar las clases con el algoritmo KNN. La versión KNN_BBond presentó tiempos de entrenamiento y validación menores que KNN.

Art Chaovallitwongse y col. [4] trabajaron con 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) y señales MER provenientes de 17 pacientes. Reportaron para KNN dos versiones: una con $K=3$ que arrojó una exactitud promedio de 82,25%, una sensibilidad promedio para la clase STN de 78,82% y especificidad para STN del 81,96%; y otra con $K=9$ con una exactitud promedio de 79,22%, una sensibilidad promedio para la clase STN de 77,94% y especificidad de 82,81%. En ambos casos los resultados obtenidos están en el orden de los de la presente tesis pero son superiores. En [4] se optimizó K variando el valor de 1 a 9 siempre con distancia Euclidiana. En [4] se realizó la validación de los clasificadores de la misma forma que en la presente tesis mediante un método de validación cruzada *leave one patient out*. En el trabajo no se realiza un estudio estadístico para comparar ambos desempeños y analizar si los resultados obtenidos

pudieran haber sido obtenidos por azar, tampoco se informan tiempos de entrenamiento y validación.

En un trabajo del autor de la tesis y otros coautores [66] se publicaron resultados parciales del desarrollo de la presente investigación reportado sobre el algoritmo KNN el cual fue entrenado con las mismas 16 características temporales de la sección 6.1.1 y con datos provenientes de 15 pacientes. En el trabajo la optimización de KNN arrojó un $K=9$ y medida de distancia *Cityblock*. Se reportó una ACC promedio del 85,8% y aplicando selección de características con *branch and bound* se entrenó al clasificador KNN con un sub conjunto de 6 características mejorando la ACC al 86,13%. En el trabajo los grupos de entrenamiento y validación se dividieron mediante la estrategia de validación cruzada con 10 pliegues y no validación cruzada *leave one patient out*, por lo que los resultados de desempeño pueden resultar sobre dimensionados. Otra variable a considerar en la diferencias de resultados entre el trabajo [66] ($n=15$) y la presente tesis ($n=22$) es la cantidad de pacientes lo que indicaría que es un factor de peso al comparar una misma estrategia de clasificación.

En la publicación de Rajpurohit y col. [10] también reportaron el desempeño del algoritmo KNN entrenando el clasificador con 13 características temporales (6 vinculadas a la actividad de fondo e independientes de las espigas y 7 dependientes de las espigas) con datos provenientes de 26 pacientes. En [10] se optimizó K variando el valor de 1 a 10 siempre con distancia Euclidiana. Cuando se utilizaron las 13 características estandarizadas con los valores de todos los pacientes KNN presentó un error promedio de 0,2 (por lo que se asume que la exactitud fue del 80%) similar pero superior al obtenido en la presente tesis. En el trabajo también se reportó la selección de características por 2 métodos de envoltura: *backward* seleccionando 7 características y *forward* seleccionando 6 características en ambos casos 5 de ellas vinculadas a la actividad de fondo. El error promedio de clasificación en ambos casos disminuye a 0,1. Esta reducción produce un incremento de la exactitud del 12,5% superior a la obtenida en la presente tesis que representó el 4% de ACC con el sub conjunto de características seleccionadas por *branch and bound*. En [10] se realiza un estudio estadístico comparativo y utiliza la validación de los clasificadores de la misma forma que en la tesis (validación cruzada *leave one patient out*) sin informar los tiempos de entrenamiento y validación.

En la presente tesis para KNN se utilizó un valor de $K=11$ y distancias tipo *Cityblock* resultantes de la optimización Bayesiana, ensayando una combinación de rangos mucho más

amplio de K que los trabajos antes reportados (de 1 a 14049) junto con distintas medidas de distancia, lo que permitió elegir los valores óptimos que minimizan el error de clasificación con la base de datos utilizada. Definir un valor óptimo de K es importante ya que dicho parámetro regula la relación entre el sesgo y la varianza del algoritmo de clasificación.

Se observa que con KNN en [4] la mejor ACC es de 81,96%, en [66] del 86,13%, en [10] se logró una ACC del 90,00% y en esta investigación la mejor ACC fue del 78,00%. Bajo las condiciones experimentales de la presente tesis se analizaron muchas configuraciones posibles y óptimas para KNN junto con diversos métodos de selección de características pero no fue posible superar una ACC promedio del 78,00%. Seguramente esta diferencia se deba a la base de datos y la cantidad de pacientes con la que se entrenó y validó a los clasificadores en cada trabajo por lo que en la sección 10.3 se realizará una discusión sobre este aspecto.

Los dos tipos de árboles de decisión considerados en la tesis como clasificadores individuales fueron CART y CHAID que en general no presentaron un rendimiento muy superior uno sobre el otro. Para ambos casos y para los datos de la tesis se seleccionaron los parámetros de configuración de cada árbol mediante optimización Bayesiana que permitió minimizar el error de clasificación de CART y CHAID mediante la variación conjunta de los parámetros cantidad de hojas mínimas (adoptando 2 y 3 respectivamente), cantidad de divisiones máxima (seleccionado 2221 y 7091 respectivamente) y criterio de división (seleccionado Gini para ambos tipos de árboles).

La versión podada al 50% de CHAID (CHAID50) fue seleccionada estadísticamente al compararla con las versiones de árboles considerados en las secciones 7.4 y 7.5. Se obtuvo con CHAID50 los siguientes índices de desempeño porcentuales promedio: ACC del 73,40%, ESP 77,70%, SEN 67,00% y DOR con un valor de 7,21. Este clasificador presentó un tiempo de entrenamiento promedio de $0,346 \pm 0,025$ s y el tiempo de validación de $0,00080 \pm 0,00005$ s.

Los métodos de selección de características abordados en el capítulo 9 no ofrecieron para los árboles de decisión mejoras en el desempeño. Se pudo apreciar que tiene mayor impacto sobre el rendimiento de los árboles la poda que la selección de características.

En general CHAID50 presentó un rendimiento inferior a KNN_BBoud si bien es de resaltar su tiempo de validación, que resultó ser el menor de todos los clasificadores analizados en la presente tesis.

Art Chaovalitwongse y col. [4], cuyo trabajo se contextualizó en párrafos previos, reportaron para árboles de decisión binarios dos versiones: una utilizando Gini como criterio de división que arrojó una exactitud promedio de 82,25%, una sensibilidad promedio para la clase STN de 87,06% y especificidad para STN del 88,89%; y otra con criterio de división de ganancia de información con una exactitud promedio de 89,61%, una sensibilidad promedio para la clase STN de 91,47% y especificidad de 89,88%. No se explicita en el trabajo si se realizó una optimización para definir la configuración de árbol y no se realiza una validación estadística de los resultados.

Se observa que la mejor exactitud en [4] fue de 89,61% superior a la encontrada en la presente tesis de 73,40%. Otra diferencia entre ambos trabajos es que para [4] los árboles funcionaron mejor que KNN y en la tesis resultó de manera inversa. Bajo las condiciones experimentales del presente trabajo de investigación se analizaron muchas configuraciones posibles y óptimas para los árboles de decisión junto con métodos de selección de características como se ha reportado en las secciones 7.4, 7.5 y 9.5.1 pero no fue posible superar una ACC promedio porcentual del 73,40%.

Se concluye que los clasificadores individuales estudiados con mejores desempeños (KNN y CHAID50) tuvieron un rendimiento aceptable pero no superior al reportado en otros trabajos, por lo que su aplicación es limitada bajo las condiciones experimentales y los objetivos de la presente tesis.

10.2 CLASIFICADORES COMBINADOS

El estudio detallado de las configuraciones que optimizaron el desempeño de los árboles de decisión permitió su selección y utilización en los algoritmos que combinan un mismo tipo de clasificadores de base. La combinación de clasificadores individuales que varíen su estructura ante pequeños cambios de los datos de entrenamiento (inestables) y que no se encuentren altamente correlacionados, producen al combinarse, una disminución del sesgo y la varianza del error total en los resultados de la clasificación combinada.

De los clasificadores combinados las estrategias de *bagging* con Bag_CART y *stacking* con Stack2 presentaron los mejores desempeños. Bag_CART obtuvo los siguientes indicadores porcentuales promedio: ACC del 92,3%, especificidad 93,5%, sensibilidad 90,6% y valor

promedio de DOR de 140,34. Este clasificador presentó un tiempo de entrenamiento promedio $32,50 \pm 1,13$ s y un tiempo de validación de $0,601 \pm 0,030$ s.

Branch and bound permitió seleccionar 6 características 2kur, 3CL, 5PK, 8ZC, 12SC y 15SCR que mantuvieron el rendimiento (desempeño y tiempos) de Bag_CART. Las características seleccionadas corresponden a una mezcla de las que describen la actividad de fondo (2kur, 3CL, 5PK, 8ZC) y las vinculadas a las espigas (12SC y 15SCR) con mayor proporción de las primeras, indicando que *bagging* para los datos de la presente tesis es más sensible a las características vinculadas a la actividad de fondo.

Stack2 obtuvo los siguientes indicadores porcentuales promedio: ACC del 94,6%, especificidad 95,6%, sensibilidad 93,5% y valor promedio de DOR de 297,91. Presentó un tiempo de entrenamiento promedio de $52,480 \pm 2,770$ s y un tiempo de validación de $1,213 \pm 0,023$ s.

Branch and bound permitió seleccionar 6 características 2kur, 3CL, 5PK, 8ZC, 12SC y 15SCR que mejoraron el rendimiento de Stack2 en lo que se denominó Stack2_BBound, pasando ACC al 95%, ESP al 96%, SEN al 94% y el valor promedio de DOR a 383,8. Este clasificador tuvo el mejor desempeño de todos los analizados en la presente tesis. Las características seleccionadas se corresponden a una mezcla de las que describen la actividad de base (2kur, 3CL, 5PK, 8ZC) y las vinculadas a las espigas (12SC y 15SCR) con mayor proporción de las primeras, indicando que Stack2 para los datos de la presente tesis es más sensible a las características vinculadas a la actividad de fondo.

No se han reportado publicaciones que utilicen clasificadores supervisados combinados para detectar la zona STN entrenados con características temporales de las señales MER, por lo que se comparará a manera de referencia los desempeños de Stack2_BBound con los mejores reportados con otras estrategias para la misma aplicación.

El trabajo de Art Chaovaitwongse y col. [4], el cual ya se ha caracterizado y sintetizado en los párrafos precedentes, obtuvo con árboles de decisión binarios entrenado con 13 características temporales y criterio de división de ganancia de información una exactitud promedio de 89,61%, una sensibilidad promedio para la clase STN de 91,47% y especificidad de 89,88% en todos los casos inferior a la lograda con Bag_CART_BBound y Stack2_BBound de la presente tesis. En [4] se evalúan asimismo KNN y Naive Bayes.

Rajpurohit y col. [10], también ya antes detallado, reporta el menor error de clasificación de 0,1056 (exactitud de 89,4%) con KNN cuando se selecciona un sub conjunto de 7 características fuertemente relacionadas con la actividad de fondo. En este trabajo también se evalúan regresión logística, Naive Bayes Gaussiano y máquinas de soporte vectorial (SVM). También este trabajo logra menor exactitud promedio que Bag_CART_BBoud y Stack2_BBoud.

En Cagan y col. [9] proponen un método de clasificación propio con una estructura similar a un árbol de decisión binario, trabajan con 2 características temporales (una relacionada a la actividad de fondo y otra a la cantidad de espigas) y 2 frecuenciales vinculadas a la potencia en el rango de frecuencias beta y gamma, con el objetivo de detectar la entrada y salida del electrodo en STN. Los datos provienen del registro MER de 48 pacientes procesados sin preselección de datos y tratando de representar lo que ocurría en el quirófano. Logran un acuerdo porcentual promedio con las clasificaciones que realizan los neurofisiólogos del 87-88%, que puede asumirse como la exactitud porcentual lograda en ese trabajo. La validación se realiza separando al azar trayectorias completas, lo cual puede arrojar un porcentaje sobredimensionado de aciertos ya que no se asegura que el clasificador pueda haber estado entrenado con algunas otras trayectorias del mismo paciente. Realizaron un estudio estadístico comparativo y no reportaron tiempos de entrenamiento y validación. También en este caso los resultados de desempeño obtenidos con Bag_CART_BBoud y Stack2_BBoud de la presente tesis, ambos entrenados con 6 características temporales, son superiores a los reportados en [9].

Stack2_BBoud como meta clasificador está conformado a nivel 0 por KNN_BBoud y Bag_CART_BBoud además de con árboles CART sin podar, por lo que la acción conjunta de estos algoritmos con el *Random forest* entrenado con las características seleccionadas por *branch and bound* como nivel 1 le otorgan robustez y un incremento sustancial y funcional para la clasificación de los registros MER.

10.3 BASE DE DATOS

En los trabajos científicos antes analizados se utilizaron bases de datos de registros MER diferentes, por lo que no es posible hacer una comparación directa de resultados. Se evidenció de los análisis precedentes que un factor clave es la cantidad de pacientes de los cuales se

obtienen los registros MER. En la presente tesis se utilizaron métodos estadísticos no paramétricos dado que en general los datos no presentaron una distribución normal y provenían de 22 pacientes.

La eliminación automática de registros asociados a ruidos de movimiento, mecánicos o eléctricos según se detalló en la sección 6.1.1 fue el único pre procesamiento que se aplicó a los registros MER utilizados en la presente tesis. No se realizó una preselección o limpieza de datos ni depuración de ninguna otra naturaleza de los registros, proporcionando de esta forma una base de datos similar a lo que ocurriría durante una cirugía y asegurado fiabilidad de los resultados obtenidos a situaciones semejantes de la vida real.

Se observa que para las mismas estrategias de clasificación como por ejemplo KNN y árboles de decisión en los trabajos de Art Chaovaitwongse y col. [4] (17 pacientes) y Rajpurohit y col. [10] (26 pacientes) utilizando cantidades similares de características temporales, logran un mejor desempeño que esas mismas estrategias en la presente tesis. Por lo que los resultados obtenidos con Bag_CART_BBound y Stack2_BBound son muy alentadores ya que superan los máximos desempeños reportados por otros autores aún bajo las condiciones de trabajar con una base de datos similar a lo que ocurriría en el quirófano.

10.4 SELECCIÓN DE CARACTERÍSTICAS

Inicialmente se adoptaron en la tesis 16 características temporales que fueron las reportadas en el estado del arte [4], [9], [10] con buenos resultados en la clasificación supervisada, con la idea de indagar posteriormente mediante algoritmos de selección de características aquellas que pudieran optimizar el proceso de clasificación con los datos disponibles en la base de datos utilizada.

Los métodos de selección de características abordados en el presente trabajo han permitido obtener clasificadores con un rendimiento similar o superior a los clasificadores entrenados con todas las características, eliminando aquellas redundantes o ruidosas. En general se observa que los métodos y los criterios de selección propuestos arrojaron una cantidad similar de características que osciló entre las 4 como mínimo y 6 como máximo. Esto indica la existencia de características redundantes o ruidosas en el conjunto total de las 16 consideradas inicialmente.

Asimismo, se observa que todos los métodos de selección del capítulo 9 dieron más importancia a las características asociadas a la actividad de fondo. La característica 8.ZC, que representa los cruces por cero de la señal obtenida con los registros MER, es la única que fue elegida por todos los métodos de selección de características, posiblemente sea una de las que mejor representa la actividad de fondo.

El método *branch and bound* resultó el mejor algoritmo para seleccionar las características dado que fue el único que logró mantener el desempeño en el caso de Bag_CART_BBoud y mejoró el rendimiento el caso de KNN_BBoud y Stack2_BBoud. Esto indicaría que logra seleccionar un sub conjunto de características óptimo con una adecuada combinación de 4 características vinculadas a la actividad de fondo (2kur, 3CL, 5PK y 8ZC) y 2 vinculadas a las espigas (2SC y 15SCR). Este resultado es concordante con lo obtenido por Rajpurohit y col. [10] que al aplicar la selección de características por métodos de envoltura (*backward* y *forward*) lograron mejorar el desempeño de los clasificadores KNN, regresión logística, SVM y Bayesiano desarrollados en el trabajo. En particular KNN, que fue el de menor error en dicho trabajo, resultó entrenado con 7 características de las 13 totales, siendo 5 de ellas relacionadas con la actividad de fondo independiente de las espigas y 2 características vinculadas a la actividad de las espigas.

Otros autores como Novak y col [67] reportaron que la actividad de fondo en el STN de 15 pacientes presentó una diferencia estadísticamente muy significativa entre el STN y el tálamo y la zona incerta (estructuras circundantes al STN por su zona dorsal) como así también con la sustancia negra (estructura vecina al STN por la zona ventral). En ese mismo trabajo se reporta que los niveles de amplitud de fondo del tálamo y la sustancia negra son similares. Przybyszewski y col. [68] también reportaron, en un estudio con 10 pacientes con validación estadística, que la actividad de fondo de los MUA junto con la potencia media en el rango beta de los potenciales de campo local (LPF por sus siglas en inglés) tenían un poder discriminante para detectar el STN de sus estructuras vecinas. En ambos trabajos se concluye que tener características vinculadas a la actividad de fondo es sustancial para la detección del STN.

La amplitud de fondo representa potenciales de acción pequeños, con niveles del orden del ruido, que son generados por poblaciones neuronales alejadas del electrodo de registro los cuales resultan una característica distintiva del STN que permite, mediante la clasificación supervisada, detectar los registros provenientes de ese núcleo cerebral. Los neurofisiólogos

también identifican dicha zona en base a las espigas, analizando tanto su amplitud como frecuencias de disparos, lo cual se corresponde con los potenciales de acción del grupo de neuronas en contacto con el electrodo. Se evidencia menor capacidad de los algoritmos de clasificación utilizados en la tesis así como los reportados por otros autores [9], [10] para diferenciar el STN en base a esta información.

Es importante remarcar que obtener clasificadores entrenados con un número menor de características no sólo mejora los tiempos de clasificación de los datos de validación sino también los tiempos de cálculo de las características a partir de los registros MER, por lo que se mejora el tiempo completo de todo el proceso.

10.5 LIMITACIONES Y PERSPECTIVAS FUTURAS DE LA INVESTIGACIÓN

En esta sección se analizan las limitaciones de la investigación desarrollada en la tesis junto con las perspectivas futuras de trabajo.

Un aspecto limitante ya discutido previamente en este capítulo está vinculado al número de pacientes del cual provinieron los registros MER. En este sentido se continúa trabajando con el Hospital Universitario y Politécnico La Fe de Valencia y la Universidad de Valencia mediante un proyecto de investigación financiado por la Universidad Nacional de Entre Ríos (Argentina) en el cual entre otros aspectos prevé incrementar el número de pacientes. Por lo que se espera a mediano plazo contar con mayor cantidad de registros MER de nuevos pacientes en la base de datos.

Asimismo, sería enriquecedor obtener registros MER obtenidos durante las cirugías para la implantación de un DBS en STN de otras instituciones y etiquetados por otros profesionales con el fin de probar Stack2_BBoud. Esto permitiría poder generalizar los resultados obtenidos con menor error.

En particular en la presente tesis se trabajó en la detección de señales MER provenientes del STN, sin identificar una zona específica de dicho núcleo basal del cerebro. Como se desarrolló en la sección 4.4.2 el STN presenta un dominio sensorimotor que se localiza en la porción dorsolateral y diversos estudios han demostrado que es la mejor zona de implantación del electrodo activo del sistema de estimulación DBS.

Particularmente la porción dorsolateral del STN suele presentar mayor energía en la banda beta. De los trabajos de [4] y [45] no se han reportado resultados alentadores al incluir características vinculadas a la potencia de la banda beta de frecuencias en algoritmos de clasificación supervisados. Por los que se propone trabajar a futuro con un sistema en cascada. De tal forma que Stack2_BBoud identifique como clasificador las señales provenientes del STN y sólo a esas señales se les realice la estimación de potencia en la banda beta para precisar la zona dorsolateral.

10.6 CONCLUSIÓN GENERAL

Con el desarrollo de las investigaciones realizadas en la presente tesis doctoral se pudo obtener un modelo combinado de clasificación supervisada mediante la metodología *stacking* al que se ha denominado Stack2_BBoud, cuyos indicadores de desempeño y tiempos de validación indican que puede ser utilizado, con excelente rendimiento, en un proceso de clasificación automática para detectar señales del STN a partir de procesar señales eléctricas cerebrales provenientes de micro electrodos de registros durante cirugías para implantar un estimulador cerebral profundo en paciente con Parkinson.

Dado que se trabajó conjuntamente con el equipo de neurocirugía para definir el protocolo de adquisición de datos y que el tesista asistió a numerosas cirugías de las cuales se obtuvieron los registros MER utilizados en la tesis, se puede asegurar que se cuenta con una base de datos de calidad. Asimismo, al poder interactuar con el equipo de médicos y observar el procedimiento que se realiza durante una cirugía para implantar un DBS se ha podido proponer un proceso de clasificación de modo que se adapte al modo en que realiza la misma, como así también recibir la realimentación de los médicos para optimizar el sistema.

Los resultados de la presente tesis contribuyen con un modelo novedoso (Stack2_BBoud) que constituye el primer paso para un sistema de clasificación automático que trabaje en el quirófano como herramienta de soporte a los neurofisiólogos y neurocirujanos al momento de definir la localización óptima del electrodo de estimulación de un sistema DBS en pacientes con Parkinson. Un sistema de éstas características permitirá reducir los tiempos de una cirugía de esta naturaleza además de brindar un resultado de clasificación objetivo.

En virtud de las discusiones y conclusiones que se fueron desarrollando en este capítulo se considera que bajo las condiciones experimentales de la presente tesis se ha verificado la hipótesis original y se han cumplido los objetivos propuestos.

Referencias

- [1] A. H. Schapira and P. Jenner, “Etiology and pathogenesis of Parkinson’s disease,” *Mov. Disord.*, vol. 26, no. 6, pp. 1049–1055, May 2011.
- [2] E. Peñas Domingo, *El libro blanco del Parkinson en España*. Federación Española de Parkinson, 2015.
- [3] E. R. Dorsey *et al.*, “Projected number of people with Parkinson disease in the most populous nations, 2005 through 2030,” *Neurology*, vol. 68, no. 5, pp. 384–386, Jan. 2007.
- [4] W. Chaovaitwongse, Y. Jeong, M. K. Jeong, S. Danish, and S. Wong, “Pattern Recognition Approaches for Identifying Subcortical Targets during Deep Brain Stimulation Surgery,” *IEEE Intell. Syst.*, vol. 26, no. 5, pp. 54–63, Sep. 2011.
- [5] T. Itakura, Ed., *Deep Brain Stimulation for Neurological Disorders: Theoretical Background and Clinical Application*, 2015 edition. New York: Springer, 2014.
- [6] S. J. Groiss, L. Wojtecki, M. Südmeyer, and A. Schnitzler, “Deep Brain Stimulation in Parkinson’s Disease,” *Ther. Adv. Neurol. Disord.*, vol. 2, no. 6, pp. 20–28, Nov. 2009.
- [7] J. S. Perlmutter and J. W. Mink, “Deep Brain Stimulation,” *Annu. Rev. Neurosci.*, vol. 29, pp. 229–257, 2006.
- [8] A. Moran and I. Bar-Gad, “Revealing neuronal functional organization through the relation between multi-scale oscillatory extracellular signals,” *J. Neurosci. Methods*, vol. 186, no. 1, pp. 116–129, Jan. 2010.
- [9] H. Cagnan *et al.*, “Automatic subthalamic nucleus detection from microelectrode recordings based on noise level and neuronal activity,” *J. Neural Eng.*, vol. 8, no. 4, p. 046006, 2011.
- [10] V. Rajpurohit, S. F. Danish, E. L. Hargreaves, and S. Wong, “Optimizing computational feature sets for subthalamic nucleus localization in DBS surgery with feature selection,” *Clin. Neurophysiol.*, vol. 126, no. 5, pp. 975–982, May 2015.
- [11] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [12] S. C. Pandey, “Data mining techniques for medical data: A review,” in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs)*, 2016, pp. 972–982.
- [2] «The Michael J. Fox Foundation for Parkinson’s Research», The Michael J. Fox Foundation for Parkinson’s Research | Parkinson’s Disease. [En línea]. Disponible en: <https://www.michaeljfox.org/>.
- [14] Á. Tepper and M. C. Henrich, “Diseño de un sistema de soporte para la implantación de electrodos en terapias DBS en pacientes con Parkinson,” Universidad Nacional de Entre Ríos, Facultad de Ingeniería, 2016.
- [15] J. L. Lanciego, N. Luquin, and J. A. Obeso, “Functional Neuroanatomy of the Basal Ganglia,” *Cold Spring Harb. Perspect. Med.*, vol. 2, no. 12, Dec. 2012.

- [16] J. A. Obeso, M. C. Rodriguez-Oroz, M. Stamelou, K. P. Bhatia, and D. J. Burn, "The expanding universe of disorders of the basal ganglia," *The Lancet*, vol. 384, no. 9942, pp. 523–531, Aug. 2014.
- [17] J. A. Obeso, M. C. Rodriguez-Oroz, M. Rodriguez, M. R. DeLong, and C. W. Olanow, "Pathophysiology of levodopa-induced dyskinesias in Parkinson's disease: problems with the current model," *Ann. Neurol.*, vol. 47, no. 4 Suppl 1, pp. S22–32; discussion S32–34, Apr. 2000.
- [18] Á. Tepper *et al.*, "Comparación de métodos de estimación espectral para señales neurales de microelectrodos obtenidas de pacientes con Parkinson," *Cogn. Area Netw.*, vol. 4, no. 1, pp. 55–60, Jun. 2017.
- [19] J. Jankovic, "Parkinson's disease: clinical features and diagnosis," *J. Neurol. Neurosurg. Psychiatry*, vol. 79, no. 4, pp. 368–376, Apr. 2008.
- [20] A. H. Schapira *et al.*, "Perspectives on recent advances in the understanding and treatment of Parkinson's disease," *Eur. J. Neurol.*, vol. 16, no. 10, pp. 1090–1099, Oct. 2009.
- [21] R. Pahwa and K. E. L. Ph.D, Eds., *Handbook of Parkinson's Disease, Fifth Edition*, 5 edition. Boca Raton, FL: CRC Press, 2013.
- [22] M. C. Rodriguez-Oroz, J. M. Matsubara, P. Clavero, J. Guridi, and J. A. Obeso, "Deep Brain Stimulation and Parkinson's Disease A2 - Squire, Larry R.," in *Encyclopedia of Neuroscience*, Oxford: Academic Press, 2009, pp. 375–384.
- [23] T. M. Herrington, J. J. Cheng, and E. N. Eskandar, "Mechanisms of deep brain stimulation," *J. Neurophysiol.*, vol. 115, no. 1, pp. 19–38, Jan. 2016.
- [24] superintendencia de Servicios de Salud, "Estimulación Cerebral Profunda en el tratamiento de la Enfermedad de Parkinson refractaria al tratamiento médico.," Buenos Aires, Feb. 2012.
- [25] "Estimulación cerebral profunda." [Online]. Available: https://www.parkinsongaliciacoruna.org/tratamiento_quirurgico_parkinson.html. [Accessed: 09-Nov-2017].
- [18] D. Anderson, G. Beecher, y F. Ba, «Deep Brain Stimulation in Parkinson's Disease: New and Emerging Targets for Refractory Motor and Nonmotor Symptoms», Parkinson's Disease, 2017. Disponible en: <https://www.hindawi.com/journals/pd/2017/5124328/>.
- [27] P. Krack, R. Martinez-Fernandez, M. del Alamo, and J. A. Obeso, "Current applications and limitations of surgical treatments for movement disorders," *Mov. Disord.*, vol. 32, no. 1, pp. 36–52, Jan. 2017.
- [28] M. Clerc, "Review of 'Brain-Computer Interfaces, principles and practise', edited by Jonathan R. Wolpaw and Elizabeth Winter Wolpaw," *Biomed. Eng. OnLine*, vol. 12, p. 22, Mar. 2013.
- [29] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents--EEG, ECoG, LFP and spikes," *Nat. Rev. Neurosci.*, vol. 13, no. 6, pp. 407–420, May 2012.
- [30] P. Novak, A. W. Przybyszewski, A. Barborica, P. Ravin, L. Margolin, and J. G. Pilitsis, "Localization of the Subthalamic Nucleus in Parkinson Disease Using Multi Unit Activity," *J. Neurol. Sci.*, vol. 310, no. 1–2, pp. 44–49, Nov. 2011.

- [31] K. Dolan, H. C. F. Martens, P. R. Schuurman, and L. J. Bour, “Automatic noise-level detection for extra-cellular micro-electrode recordings,” *Med. Biol. Eng. Comput.*, vol. 47, no. 7, pp. 791–800, Jul. 2009.
- [32] S. Wong, G. H. Baltuch, J. L. Jaggi, and S. F. Danish, “Functional localization and visualization of the subthalamic nucleus from microelectrode recordings acquired during DBS surgery with unsupervised machine learning,” *J. Neural Eng.*, vol. 6, no. 2, p. 026006, 2009.
- [33] J. Guerrero-Martínez *et al.*, “Characterization of Microelectrode Records in Deep Brain Stimulation Applied to Parkinson’s Disease Patients,” in *VI Latin American Congress on Biomedical Engineering CLAIB 2014, Paraná, Argentina 29, 30 & 31 October 2014*, A. Braidot and A. Hadad, Eds. Springer International Publishing, 2015, pp. 647–650.
- [34] A. Zaidel, A. Spivak, L. Shpigelman, H. Bergman, and Z. Israel, “Delimiting subterritories of the human subthalamic nucleus by means of microelectrode recordings and a Hidden Markov Model,” *Mov. Disord. Off. J. Mov. Disord. Soc.*, vol. 24, no. 12, pp. 1785–1793, Sep. 2009.
- [35] S. Little and P. Brown, “What brain signals are suitable for feedback control of deep brain stimulation in Parkinson’s disease?,” *Ann. N. Y. Acad. Sci.*, vol. 1265, pp. 9–24, Aug. 2012.
- [36] H. Bronte-Stewart, C. Barberini, M. M. Koop, B. C. Hill, J. M. Henderson, and B. Wingeier, “The STN beta-band profile in Parkinson’s disease is stationary and shows prolonged attenuation after deep brain stimulation,” *Exp. Neurol.*, vol. 215, no. 1, pp. 20–28, Jan. 2009.
- [37] T. E. Özkurt *et al.*, “High frequency oscillations in the subthalamic nucleus: A neurophysiological marker of the motor state in Parkinson’s disease,” *Exp. Neurol.*, vol. 229, no. 2, pp. 324–331, Jun. 2011.
- [38] A. Zaidel, A. Spivak, B. Grieb, H. Bergman, and Z. Israel, “Subthalamic span of beta oscillations predicts deep brain stimulation efficacy for patients with Parkinson’s disease,” *Brain J. Neurol.*, vol. 133, no. Pt 7, pp. 2007–2021, Jul. 2010.
- [39] G. Giannicola *et al.*, “The effects of levodopa and ongoing deep brain stimulation on subthalamic beta oscillations in Parkinson’s disease,” *Exp. Neurol.*, vol. 226, no. 1, pp. 120–127, Nov. 2010.
- [40] O. Maimon and L. Rokach, Eds., *Data Mining and Knowledge Discovery Handbook*, 2nd ed. 2010 edition. New York: Springer, 2010.
- [41] J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: A new perspective,” *Neurocomputing*.
- [42] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, “Efficient Machine Learning for Big Data: A Review,” *Big Data Res.*, vol. 2, no. 3, pp. 87–93, Sep. 2015.
- [43] G. Hoang, A. Bouzerdoum, and S. Lam, “Learning Pattern Classification Tasks with Imbalanced Data Sets,” in *Pattern Recognition*, P.-Y. Yin, Ed. InTech, 2009.
- [44] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2011.
- [45] J. V. Francés-Víllora *et al.*, “Clasificación de registros de microelectrodo para localización de zonas de estimulación en pacientes de Parkinson,” in *Actas del XXXIV Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB)*, Valencia, España, 2016, pp. 536–539.

- [46] P. Eusebi, "Diagnostic Accuracy Measures," *Cerebrovasc. Dis.*, vol. 36, no. 4, pp. 267–272, 2013.
- [47] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, no. Jan, pp. 1–30, 2006.
- [48] F. van der Heijden, R. P. Duin, D. de Ridder, and D. M. J. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, 1 edition. Chichester, West Sussex, Eng. ; Hoboken, NJ: Wiley, 2004.
- [49] W. L. Martinez and A. R. Martinez, *Computational Statistics Handbook with MATLAB, Third Edition*, 3 edition. Boca Raton: Chapman and Hall/CRC, 2015.
- [50] F. J. Martínez de Pisón Ascacóbar, "Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado," Universidad de La Rioja, Logroño, España, 2003.
- [51] "K nearest neighbors algorithm," *Wikipedia*. 24-Feb-2018.
- [52] S. Taneja, C. Gupta, K. Goyal, and D. Gureja, "An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering," in *2014 Fourth International Conference on Advanced Computing Communication Technologies*, 2014, pp. 325–329.
- [53] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, Jan. 2011.
- [54] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, 1 edition. New York: Springer, 2018.
- [55] T. G. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, Aug. 2000.
- [56] J. R. Quinlan, "Bagging, Boosting, and C4.S," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, Portland, Oregon, 1996, pp. 725–730.
- [57] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *In Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 148–156.
- [58] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," p. 9.
- [59] L. Breiman, "Bagging Predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
- [60] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
- [61] H. B. Li, W. Wang, H. W. Ding, and J. Dong, "Trees Weighting Random Forest Method for Classifying High-Dimensional Noisy Data," in *2010 IEEE 7th International Conference on e-Business Engineering (ICEBE)*, 2010, pp. 160–163.
- [62] S. Džeroski and B. Ženko, "Is Combining Classifiers with Stacking Better than Selecting the Best One?," *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, Mar. 2004.

- [63] Jiliang Tang, Salem Alelyani, and Huan Liu, “Feature Selection for Classification: A Review,” in *Data Classification*, 0 vols., Chapman and Hall/CRC, 2014, pp. 37–64.
- [64] A. Kalousis, J. Prados, and M. Hilario, “Stability of feature selection algorithms: a study on high-dimensional spaces,” *Knowl. Inf. Syst.*, vol. 12, no. 1, pp. 95–116, May 2007.
- [65] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF,” *Mach. Learn.*, vol. 53, no. 1–2, pp. 23–69, Oct. 2003.
- [66] L. Schiaffino *et al.*, “Feature selection for KNN classifier to improve accurate detection of subthalamic nucleus during deep brain stimulation surgery in Parkinson’s patients,” *Springer IFMBE Proc.*, vol. 60, pp. 441–444, 2017.
- [67] P. Novak, S. Daniluk, S. A. Ellias, and J. M. Nazzaro, “Detection of the subthalamic nucleus in microelectrographic recordings in Parkinson disease using the high-frequency (> 500 Hz) neuronal background. Technical note,” *J. Neurosurg.*, vol. 106, no. 1, pp. 175–179, Jan. 2007.
- [68] A. W. Przybyaszewski, P. Ravin, J. G. Pilitsis, A. Szymanski, A. Barborica, and P. Novak, “Multi-parametric analysis assists in STN localization in Parkinson’s patients,” *J. Neurol. Sci.*, vol. 366, pp. 37–43, Jul. 2016.

ANEXO 1: Informe de actividades y publicaciones vinculadas a la tesis Doctoral

1) En el marco del trabajo de investigación de la tesis doctoral se han realizado las siguientes publicaciones:

L. Schiaffino, A. Rosado Muñoz, J. Guerrero Martínez, J. Francés Villora, A. Gutiérrez, I. Martínez Torres and D. R. Kohan, “**STN area detection using K-NN classifiers for MER recordings in Parkinson patients during neurostimulator implant surgery,**” *J. Phys.*, vol. 705, no. 1, 2016, Doi: 10.1088/1742-6596/705/1/012050, ISSN 1742-6588.

L. Schiaffino, A. Rosado Muñoz, J. Francés Villora, M. Bataller, A. Gutiérrez, I. Martínez Torres and J. Guerrero Martínez, “**Feature selection for KNN classifier to improve accurate detection of subthalamic nucleus during deep brain stimulation surgery in Parkinson’s patients,**” *Springer IFMBE*, vol. 60, pp. 441–444, 2017.

Á. Tepper, M. C. Henrich, **L. Schiaffino**, A. Rosado Muñoz, A. Gutiérrez, and J. Guerrero Martínez, “**Comparación de métodos de estimación espectral para señales neurales de microelectrodos obtenidas de pacientes con Parkinson,**” *Cogn. Area Netw.*, vol. 4, no. 1, pp. 55–60, Jun. 2017.

Á. Tepper, M. C. Henrich, **L. Schiaffino**, A. Rosado Muñoz, J. Guerrero Martínez, A. Gutiérrez and I. Martínez Torres, “**Selection of the Optimal Algorithm for Real-Time Estimation of Beta Band Power during DBS Surgeries in Patients with Parkinson’s Disease,**” *Comput. Intell. Neurosci.*, vol. 2017, no. Article ID 1512504, 9 pages, Dec. 2017.

Se ha enviado en febrero de 2018 la siguiente publicación la cual se encuentra en proceso de revisión:

L. Schiaffino, A. Rosado Muñoz, J. Francés Villora, M. Bataller, A. Gutiérrez, I. Martínez Torres, V. Teruel and J. Guerrero Martínez, “**High accuracy classification of microelectrode recordings acquired during DBS surgery for localization of the subthalamic nucleus in Parkinson patients,**” *Biomedical Journal*, ISSN: 2319-4170.

2) Premio al mejor trabajo de Estudiante de Doctorado otorgado por la Federación Internacional de Ingeniería en Medicina y Biología (IFMBE) por el trabajo “Feature selection for KNN classifier to improve accurate detection of subthalamic nucleus during deep brain stimulation surgery in Parkinson’s patients” presentado en el VII Congreso Latinoamericano de Ingeniería Biomédica (CLAIB 2016) realizado del 26 al 28 de octubre de 2016 en Bucaramanga, Colombia.

3) El doctorando ha participado de los siguientes proyectos vinculados al tema de tesis:

Integrante de 2 Proyectos de I+D+i competitivos: “Sistema de soporte analítico en la cirugía del Parkinson mediante estimulación cerebral profunda” (año 2015) y “Sistema Integral de Soporte Analítico en la cirugía y seguimiento postoperatorio del Parkinson mediante Estimulación Cerebral Profunda” (año 2016). Ambos proyectos fueron financiados por la Universidad de Valencia y la Fundación para la Investigación Hospital la Fe dentro del VLC-Campus de Excelencia.

4) Con el objetivo de continuar las investigaciones vinculados con el tema de tesis se ha presentado el siguiente proyecto de investigación en el cual el Doctorando es el director:

Proyecto de investigación (PID-UNER 6169) “Mejoras tecnológicas en estimuladores cerebrales profundos y su proceso de implantación en pacientes con Parkinson”. Monto asignado: \$54.500. Financiado por la Universidad Nacional de Entre Ríos. Lugar de ejecución Laboratorio de Ingeniería en Rehabilitación e Investigaciones Neuromusculares y Sensoriales (LIRINS - FIUNER). Inicio: febrero 2018 - Finalización: agosto 2020. Resolución “C.S.” 401/17. Actualmente en ejecución.

5) Asistencia y/o exposición de trabajos en Congresos durante el período de desarrollo de la tesis doctoral:

Asistente y expositor del trabajo “Aplicaciones de lógica Difusa en Bioingeniería” en el Simposio Argentino de Sistemas Embebidos (SASE 2014), Año 2014, Facultad de Ingeniería de Universidad de Buenos Aires, ciudad Autónoma de Buenos Aires - Argentina.

Asistente al 24 Congreso Argentino de Control Automático (AADECA 2104), Año 2014, Universidad Tecnológica regional Buenos Aires.

Asistente al Open Source Brain Workshop (OSB 2015) en Alghero, Año 2015, Cerdeña, Italia.

Asistente y expositor del trabajo “Detección de zona STN mediante minería de datos con registros MER de cirugías para la implantación de neuroestimuladores en pacientes con Parkinson” en el XX Congreso de la Sociedad Argentina de Ingeniería Biomédica (SABI 2015) del 28 al 30 de octubre de 2015 en San Nicolás de los Arroyos, Argentina.

Asistente y expositor del trabajo “Determinación mediante minería de datos del núcleo subtalámico utilizando micro electrodos de registro (MER) en cirugías para la implantación de neuroestimuladores en pacientes con Parkinson” en las 3ª Jornada de Divulgación de la Investigación en Ingeniería de los alumnos de Doctorado de la Escuela Técnica Superior de Ingeniería (ETSE) de la Universidad de Valencia (UV), realizada en la ETSE el 4 de julio de 2016.

Asistente y expositor del trabajo “Selección de características para lograr la detección precisa del núcleo subtalámico utilizando clasificadores KNN durante cirugías para implantar estimuladores cerebrales profundos en pacientes con Parkinson” realizado en coautoría de Muñoz, A. R., Villora, J. F., Bataller, M., Gutiérrez, A., Torres, I. M., Martínez, J. G. en el VII Congreso Latinoamericano de Ingeniería Biomédica (CLAIB 2016) realizado del 26 al 28 de octubre de 2016 en Bucaramanga, Colombia.

Asistente y expositor del trabajo “Comparación de métodos de estimación espectral para señales neurales de microelectrodos obtenidas de pacientes con Parkinson” realizado en coautoría de Tepper, Á., Henrich, M. C., Schiaffino, L., Rosado Muñoz, A., Guerrero Martinez, J., Gutiérrez, A., & Martínez Torrez, I. en el 9 Simposio CEA de Bioingeniería realizado del 7 al 8 de julio de 2017 en el Instituto Guttmann, Barcelona, España.

6) Formación de recursos humanos realizadas por el doctorando:

Director de una Beca de Formación para el PID 6169 “Mejoras tecnológicas en estimuladores cerebrales profundos y su proceso de implantación en pacientes con Parkinson”.

Facultad de Ingeniería. Universidad Nacional de Entre Ríos. Becario: Gabriel Bellino. Duración 24 meses. Resolución Decano FIUNER No 9205/17.

Director de 2 tesis de grado de la carrera de Bioingeniería de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (Argentina) en el período septiembre 2014 a junio 2018:

Tesis: "Diseño de un sistema de soporte para la implantación de electrodos en terapias DBS en pacientes con Parkinson"

Estudiantes: Ángeles TEPPER y Mauricio C. HENRICH

Aprobación: 25 de noviembre de 2016.

Resolución Decano FI-UNER No 8704/16.

Tesis: "Diseño e implementación de un control de humedad y temperatura para incubadoras neonatales"

Estudiantes: Sebastián A. ILI FLORES y Hisashi J. KONNO

Aprobación: 22 de diciembre de 2017.

Resolución Decano FI-UNER No 9334/17

Evaluador de 6 tesis de grado de la carrera de Bioingeniería de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (Argentina) en el período septiembre 2014 a junio 2018.