

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

APRENDIZAJE AUTOMÁTICO PARA LA ANOTACIÓN DE RITMOS EN PARADA CARDIORRESPIRATORIA

Alumno	<i>López Manibardo, Eric</i>
Directores	<i>Irusta Zarandona, Unai y Del Ser Lorente, Javier</i>
Departamento	<i>Tecnología de Telecomunicaciones y Procesado de Señal</i>
Curso académico	<i>2018/2019</i>

Bilbao, 23, septiembre, 2019

Resumen trilingüe	5
Resumen.....	5
Summary	6
Laburpena	7
Índice de tablas	8
Índice de figuras	9
Lista de acrónimos.....	11
1. Introducción.....	13
2. Contexto	14
3. Objetivos y alcance del proyecto.....	19
3.1 Base de datos auditada y anotada	19
3.2 Clasificador 5 clases mediante señal ECG	19
3.3 Clasificador 5 clases multimodal: señal ECG y señal impedancia	19
3.4 Clasificadores mediante técnicas Deep Learning.....	20
4. Beneficios del proyecto	21
4.1 Beneficios técnicos.....	21
4.2 Beneficios económicos.....	21
4.3 Beneficios sociales.....	21
5. Análisis del estado del arte	23
5.1 Clasificación de ritmos PCREH.....	23
5.2 Validación cruzada	24
5.3 Evaluación del algoritmo.....	26
5.4 Características señal ECG	27
5.4.1 Detección de ritmos desfibrilables.....	28
5.4.2 Detección de circulación sanguínea.....	28
5.4.3 Clasificación en 5 ritmos.....	29
5.5 Algoritmo de clasificación	29
5.5.1 Árboles de decisión	29
5.5.2 Random Forest	30
5.6 Técnicas de Deep Learning.....	31
5.6.1 Capa convolucional	31
5.6.2 Capa función de activación	32
5.6.3 Capa MaxPooling.....	33
5.6.4 Capa recurrente	33
5.6.5 Capa Densa.....	33

6. Análisis de alternativas	35
6.1 Lenguaje de programación.....	35
6.1.1 Matlab	35
6.1.2 Octave	35
6.1.3 Python	36
6.1.4 R.....	36
6.2 Criterios de selección: lenguaje de programación.....	36
6.2.1 Sencillez de uso	36
6.2.2 Procesado y gestión de datos.....	36
6.2.3 Desarrollo de algoritmos.....	37
6.2.4 Eficiencia.....	37
6.2.5 Precio licencia.....	37
6.3 Selección de alternativas: lenguaje de programación	37
6.4 Algoritmos de clasificación.....	38
6.4.1 KNN.....	38
6.4.2 SVM	39
6.4.3 Random Forest	40
6.5 Criterios de selección: lenguaje de programación.....	40
6.5.1 Velocidad de aprendizaje	40
6.5.2 Interpretabilidad	41
6.5.3 Flexibilidad del modelo	41
6.5.4 Precisión	41
6.6 Selección de alternativas: algoritmo de clasificación.....	41
6.7 Librerías modelos Deep Learning.....	42
6.7.1 TensorFlow	42
6.7.2 Theano.....	43
6.7.3 Keras	43
6.7.4 PyTorch.....	43
6.8 Criterios de selección: librerías modelos Deep Learning	43
6.8.1 Visualización de la estructura de red	43
6.8.2 Soporte	44
6.8.3 Interpretabilidad del código.....	44
6.9 Selección de alternativas: librerías modelos Deep Learning	44
7. Análisis de riesgos	45
A. Riesgos de gastos	46
B. Riesgos de horario.....	46

C.	Riesgos técnicos	46
D.	Riesgos tecnológicos	47
E.	Riesgos de operación	47
F.	Riesgos de mercado y competencia.....	47
G.	Riesgos de cambio de leyes.....	48
8.	Descripción de la solución propuesta. Diseño (de alto nivel)	50
8.1	Base de datos auditada y anotada	50
8.1.1	Análisis de los datos	50
8.1.2	Anotación de intervalos RP y AESP	51
8.1.3	Preprocesado de los datos	52
8.1.4	Segmentación de los datos	53
8.1.5	Reserva de datos para test.....	54
8.1.6	Particiones de datos	54
8.2	Clasificador 5 clases mediante señal ECG	55
8.2.1	Extracción de características de la señal ECG	56
8.2.2	Diseño clasificador 5 clases	57
8.2.3	Resultados obtenidos para 10 ejecuciones del algoritmo	59
8.2.4	Ranking de características señal ECG	61
8.2.5	Optimización del preprocesado de la señal ECG.....	63
8.3	Clasificador 5 clases multimodal: señal ECG y señal impedancia	64
8.3.1	Diseño clasificador 4 clases	64
8.3.2	Ranking características clasificador 4 clases	65
8.3.3	Filtrado casos ORG fallidos.....	66
8.3.4	Cálculo componente de circulación	66
8.3.5	Extracción características especializadas	67
8.3.6	Diseño clasificador RP/AESP.....	68
8.3.7	Ranking características clasificador RP/AESP.....	69
8.3.7	Integración clasificador 5 clases.....	70
8.4	Clasificador 5 clases mediante técnicas de Deep Learning.....	71
8.4.1	Diseño red convolucional preliminar	71
8.4.2	Búsqueda de mejor configuración mediante optimización Bayesiana	72
8.5	Resumen de resultados.....	73
9.	Metodología	75
9.1	Descripción de los recursos materiales.....	75
9.2	Descripción de los paquetes de trabajo.....	75
9.2.1	Grupo de trabajo	75

9.2.2 Paquetes de trabajo	76
9.2.3 Hitos y entregables.....	79
9.3 Diagrama de Gantt	80
10. Presupuesto del Proyecto	82
10.1 Recursos humanos	82
10.2 Recursos materiales	83
10.2.1 Materiales amortizables.....	83
10.2.2 Materiales fungibles.....	83
10.3 Coste total del Proyecto.....	83
11. Conclusiones y trabajo futuro	84
12. Bibliografía.....	85

Resumen trilingüe

Resumen

Las paradas cardiorrespiratorias extrahospitalarias (PCREH) se posicionan como una de las principales causas de defunción en los países desarrollados. Ante dicho evento, existen ciertos factores determinantes para la supervivencia del sujeto, incluyendo la reanimación cardio pulmonar, una pronta desfibrilación y la calidad del tratamiento ofrecido por el Servicio de Emergencias Médicas. El corazón del paciente puede presentar hasta cinco tipos de ritmos distintos. Puesto que cada estado clínico precisa un tratamiento diferente, es de vital importancia para el personal médico, la pronta y correcta identificación del ritmo/estado del paciente. Por consiguiente, existen numerosos estudios dedicados al entendimiento de dichas patologías, los cuales emplean grabaciones de la señal electrocardiograma (ECG) durante episodios PCREH. Dichas grabaciones deben ser anotadas manualmente por un grupo de expertos clínicos. Por lo tanto, resulta una tarea dispendiosa, lo cual ocasiona escasez de bases de datos debidamente caracterizadas y anotadas.

Con el objetivo de facilitar el acceso a colecciones de datos correctamente anotadas, existen algoritmos de anotación semiautomáticos. Estos algoritmos permiten identificar con elevada certeza, las patologías presentes en distintos intervalos temporales de la señal ECG. De esta forma, los expertos clínicos se focalizan en repasar las decisiones del algoritmo, ahorrando tiempo y coste. Por todo ello, los algoritmos de anotación facilitan los estudios de enfermedades cardíacas, mejorando la calidad del tratamiento realizado y, de esta forma, la probabilidad de supervivencia del paciente.

En este trabajo se presentan cuatro clasificadores de ritmos de pacientes en PCREH. Para su desarrollo, primero se prepara una colección de episodios PCREH con los que entrenar los algoritmos. El primer clasificador extrae información únicamente de la señal ECG. El segundo añade la información presente en la impedancia transtorácica del paciente. Después, se desarrolla un tercer clasificador mediante técnicas de Deep Learning, puesto que recientemente ha demostrado su potencial en este campo. El cuarto clasificador lo conforma una versión optimizada del anterior modelo. Finalmente, se analizan los resultados y se compara el rendimiento de las distintas soluciones propuestas.

Summary

Out-of-hospital cardiac arrest (OHCA) is one of the leading causes of death in developed countries. There are several key factors to survive an OHCA event, including cardiopulmonary resuscitation, early defibrillation and the overall quality of treatment given by the Emergency Medical System. The patient's heart can present up to five different types of rhythms. Since each clinical condition requires a different treatment, a fast and precise identification of the patient's rhythm/status is crucial for the medical staff. Therefore, there are numerous studies that focus on the understanding of these pathologies, using electrocardiogram signals (ECG) recorded during OHCA events. These recordings must be manually annotated by a group of clinical experts. Because the high costs associated to manual annotation, there is a lack of properly characterized and annotated databases.

In order to facilitate access to correctly annotated data collections, there are semi-automatic annotation algorithms. These algorithms identify with high accuracy the pathologies present in different time intervals of the ECG signal. In this way, clinical experts would focus on reviewing the algorithm's decisions, saving time and money. All these considerations make annotation algorithms a key factor to develop studies on OHCA, improving the quality of the treatment performed and the probability of patient survival.

In this work, four classifiers of OHCA rhythms are presented. For their development, first a collection of OHCA episodes is prepared, in order to train the algorithms. The first classifier extracts information only from the ECG signal. The second one, adds the information present in the patient's transthoracic impedance. Then, a third classifier is developed using Deep Learning techniques, since it has recently demonstrated its potential in this field. After that, a fourth classifier is made optimizing the previous model. Lastly, the results are analysed and the performance of the different proposed solutions is compared.

Laburpena

Hospitalez kanpoko bihotz geldiketa (HKBG) mundo garatuko heriotza kausa handienetariko bat dira. Geldiketa bat ematen denean zenbait gertakari gako dira pazientearen biziraupenerako, adibidez bihotz biriketako masajea, desfibrilazio goiztiarra edota emergentzia zerbitzuek emandako tratamendua. Pazientearen bihotzak bost erritmo desberdin aurkez ditzazke HKGB batean. Egoera kliniko bakoitzak tratamendu desberdina behar duenez, pazientearen erritmoa/egoera goiz eta zehatz detektatzea oso garrantzitsua da. Ondorioz, lan asko egin dira patologia horiek ulertzeko eta identifikatzeko, orokorrean pazientearen grabatutako elektrokardiograma (EKG) erabiliz. Grabaketa horietan aditu klinikoek erritmoa identifikatu eta anotatu behar dute. Azken hau kostu handiko lana da, eta ondorioz oso HKGB datubase gutxi dago erritmo anotazio egokiekin.

Erritmo anotazioak dituzten HKGB datubaseak sortzeko badira erritmoa modu erdiautomatikoan anotatzeko algoritmoak. Algoritmo hauek modu nahiko zehatzean identifika dezaketa HKGB pazientearen erritmo/egoera, horretarako grabatutako EKG erabiliz. Horrela aditu klinikoek emandako diagnostikoa baino ez dute berrikusi behar, denbora eta kostuak aurreztuz. Horregatik anotaziorako algoritmoek HKGBaren inguruko ikerkuntza errazteu eta hobetuko lituzkete, emandako tratamendua hobetuz, eta pazienteen biziraupen aukerak handituz.

Lan honetan lau algoritmo garatu dira HKGB erritmoak modu automatikoan sailkatzeko. Algoritmoak garatzeko lehendabizi HKGB kasuen datubase bat prestatu da, algoritmoak entrenatzeko. Lehen sailkatzailea EKG-tik soilik lortzen du informazioa. Bigarrenak bular inpedantziako informazioa ere erabiltzen du. Gero, ikasketa sakonean oinarritutako sailkatzailea garatu da, esparru honetan teknika hauek oso emaitza onak eman izan baitituzte. Azkenik laugarren sailkatzailea aurrekoaren bertsio hobetua da. Bukatzeko, emaitzak aztertu eta sailkatzaileen errendimenduak alderatu dira.

Índice de tablas

Tabla 1. Matriz de confusión para clasificación de ritmos PCREH.	27
Tabla 2. Comparativa lenguajes de programación.....	37
Tabla 3. Comparativa algoritmos de clasificación.....	41
Tabla 4. Comparativa librerías Deep Learning.....	44
Tabla 5. Descripción de criterios de medición de probabilidad de riesgos.....	45
Tabla 6. Descripción de criterios de medición del impacto ocasionado por los riesgos.....	45
Tabla 7. Descripción de los campos contenidos en la base de datos original.....	50
Tabla 8. Descripción de los campos contenidos en 'analysis'.....	51
Tabla 9. Representación datos entrenamiento.....	55
Tabla 10. Valor UMS para distintas configuraciones del filtro de señal.	63
Tabla 11. Valor UMS para distintas configuraciones de mother wavelet.....	64
Tabla 12. Sensibilidad de cada clase obtenida para el clasificador de 4 clases.	65
Tabla 13. Proceso de selección ritmos RP/AESP.	66
Tabla 14. Comparativa sensibilidad sin aplicar pesos y aplicando pesos.....	69
Tabla 15. Comparativa sensibilidad al emplear todas las características y empleando las nueve mejores.....	70
Tabla 16. Comparativa sensibilidad entre clasificador mediante señal ECG y clasificador multimodal.	70
Tabla 17. Comparativa valores UMS entre distintos clasificadores.....	74
Tabla 18. Recursos hardware y software utilizados.....	75
Tabla 19. Grupo de trabajo del proyecto.....	75
Tabla 20. Primer paquete de trabajo.	76
Tabla 21. Segundo paquete de trabajo.	76
Tabla 22. Tercer paquete de trabajo.....	77
Tabla 23. Cuarto paquete de trabajo.	78
Tabla 24. Organización de tiempos de trabajo.	79
Tabla 25. Hitos del proyecto.	79
Tabla 26. Entregables del proyecto.....	79
Tabla 27. Coste unitario del grupo de trabajo.	82
Tabla 28. Coste total de los recursos humanos.	82
Tabla 29. Coste total de los materiales amortizables.	83
Tabla 30. Coste total de los materiales fungibles.	83
Tabla 31. Resumen del coste total del proyecto.....	83

Índice de figuras

Figura 1. DEA conectado para simular una descarga, junto con ensayo de RCP.	14
Figura 2. Anatomía del corazón.	15
Figura 3. Ritmo sinusal típico.	16
Figura 4. Ejemplo del ECG para los cinco ritmos/estados clínicos del paciente en PCREH.....	17
Figura 5. Ejemplo de interfaz creada para inspección de ritmos cardiacos.....	18
Figura 6. Comparativa entre señales ECG procedentes de base de datos frente a episodios PCREH (adaptada de [33]).	24
Figura 7. Validación cruzada de K particiones.....	25
Figura 8. Representación visual de la clasificación en dos clases (positiva y negativa), un clasificador binario.	26
Figura 9. Esquema funcionamiento detector de circulación. (adaptada de [15]).	28
Figura 10. Representación 2D y 3D de las regiones de decisión definidas por X_1 y X_2	30
Figura 11. Diagrama de un árbol de decisión.....	30
Figura 12. Ejemplo capa Conv1D de $M=5$ filtros con kernel de $N=32$ muestras.....	32
Figura 13. Comparativa entre distintas funciones de activación.	32
Figura 14. Capa LSTM descompuesta en N unidades fundamentales	33
Figura 15. Ejemplo capa densa de 4 neuronas.....	34
Figura 16. Ejemplo clasificador perros/gatos basado en KNN. Obsérvese la importancia del parámetro K en la decisión tomada por el algoritmo.	39
Figura 17. Selección de hiperplano óptimo por parte del algoritmo SVM. Vectores de soporte (S_1 y S_2) permiten calcular el margen óptimo del hiperplano B'	39
Figura 18. Ejemplo de 'kernel trick' aplicado sobre un espacio bidimensional (dos características X e Y), para convertirlo en tridimensional. La tercera dimensión se obtiene como $Z = X^2 + Y^2$. De esta forma es posible separar ambas clases mediante un plano lineal.	40
Figura 19. Ejemplo sobreajuste del algoritmo a los datos de entrenamiento (verde), frente al ajuste óptimo deseado (negro).	42
Figura 20. Matriz probabilidad/impacto de los riesgos presentes en el proyecto.	48
Figura 21. Ejemplo caso RP. Variaciones de impedancia en fase con complejos QRS.....	52
Figura 22. Ejemplo caso AESP. Sin variaciones de impedancia.	52
Figura 23. Respuesta frecuencial filtros paso alto y paso bajo.	53
Figura 24. Casos posibles durante la extracción de tramos de señal.....	54
Figura 25. Descomposición SWT de ocho niveles (adaptada de [79]). En el primer nivel se obtiene el primer detalle d_1 y se propaga el coeficiente de aproximación a_1 , el cual se corresponde con la señal de entrada del segundo nivel. Progresivamente se obtienen los distintos detalles $d_1 - d_8$ y un último coeficiente de aproximación a_8	56
Figura 26. Influencia del número de árboles en el rendimiento del algoritmo, expresado en UMS y F1-score.	58
Figura 27. Influencia del número de predictores en el rendimiento del algoritmo, expresado en UMS y F1-score.	59
Figura 28. Matriz de confusión para 10 ejecuciones del clasificador RF ($N_t = 150$, $N_p = 8$, $Prior = 'Empirical'$) sobre los datos de test, empleando 54 características de la señal ECG.	60

Figura 29. Ejemplo de segmentos ECG de la colección de datos reservada para test. Arriba se muestra el ritmo clasificado por los expertos clínicos. A la izquierda las predicciones del algoritmo. Los casos acertados se muestran en azul, los errores en rojo.	61
Figura 30. Se muestran 54 características de la señal ECG, ordenadas según su importancia a la hora de clasificar 5 ritmos cardiacos distintos.	62
Figura 31. Rendimiento del algoritmo expresado en UMS, empleando las K mejores características de la señal ECG, para entrenamiento y test.....	62
Figura 32. Características empleadas en el clasificador de 4 clases, ordenadas por importancia.	65
Figura 33. Esquema extractor señal CCI (adaptada de [15]), siendo $s[n]$ la señal de impedancia transtorácica.	67
Figura 34. Características empleadas en el clasificador RP/AESP, ordenadas por importancia.	69
Figura 35. Diseño clasificador 5 clases multimodal.....	70
Figura 36. Arquitectura CNN: capa convolucional de 24 filtros y 16 muestras, con MaxPooling cada 3 muestras. Capa recurrente formada por 20 células LSTM, seguida por dos capas densas de 8 neuronas. Finalmente, capa Softmax de 5 neuronas. En todas las capas se emplea Relu como función de activación.	72
Figura 37. Resultados función objetivo para 100 iteraciones del algoritmo de optimización Bayesiana.	73
Figura 38. Tareas e hitos del proyecto.	80
Figura 39. Diagrama de Gantt del proyecto.	81

Lista de acrónimos

BioRes	-----	Bioengineering and Resuscitation
JRL	-----	Joint Research Lab
PCREH	-----	Parada Cardiorrespiratoria Extrahospitalaria
SEM	-----	Servicio de Emergencias Médicas
ML	-----	Machine Learning
IA	-----	Inteligencia Artificial
ECG	-----	Electrocardiograma
CAPV	-----	Comunidad autónoma del País Vasco
SVB	-----	Soporte Vital Básico
SVA	-----	Soporte Vital Avanzado
DEA	-----	Desfibrilador Externo automático
RCP	-----	Reanimación Cardio Pulmonar
FV	-----	Fibrilación Ventricular
RP	-----	Ritmo con Pulso
AESP	-----	Actividad Eléctrica Sin Pulso
TV	-----	Taquicardia Ventricular
AS	-----	Asistolia
ORG	-----	Ritmo Organizado
AHA	-----	American Heart Association
MITDB	-----	MIT-BIH Arrhythmia Database
VFDB	-----	MIT-BIH Malignant Ventricular Arrhythmia Database
CUDB	-----	Creighton University Ventricular Tachycardia Database
Se	-----	Sensibilidad
P	-----	Precisión
F1	-----	F1-score
UMS	-----	Unweighted Mean of Sensitivities
IT	-----	Impedancia Transtorácica
CCI	-----	Componente de Circulación de Impedancia

DWT	-----	Discrete Wavelet Transform
RF	-----	Random Forest
DL	-----	Deep Learning
Conv-1D	-----	Capa Convolutiva de una Dimensión
ReLU	-----	Rectified linear activation unit
RNN	-----	Recurrent Neural Network
LSTM	-----	Long Short-Term Memory
GRU	-----	Gated Recurrent Layer
CNN	-----	Convolutional Neural Network
IDE	-----	Entorno de Desarrollo Integrado
GUI	-----	Interfaz Gráfica de Usuario
KNN	-----	K-Nearest Neighbors
SVM	-----	Support Vector Machine
TF	-----	TensorFlow
IIR	-----	Filtro de Respuesta Infinita
SWT	-----	Stationary Wavelet Transform

1. Introducción

El presente trabajo de Fin de Master de la Escuela de Ingeniería de Bilbao se ha realizado bajo la supervisión y apoyo del Grupo de investigación Bioengineering and Resuscitation (BioRes) de la UPV/EHU. A su vez se ha contado con la ayuda del Joint Research Lab (JRL) de Tecalia, grupo de investigación privado sin ánimo de lucro, dirigido por Javier Del Ser, segundo tutor de este proyecto.

Las enfermedades cardiovasculares son la principal causa de defunción en todo el mundo. Cada año mueren más personas por alguna de estas enfermedades que por cualquier otra causa. Solo en 2012, murieron 17.5 millones de personas por enfermedades cardiovasculares, representando el 30% de las defunciones registradas en el mundo. Destaca la parada cardiorrespiratoria, definida como la pérdida de actividad mecánica del corazón y la ausencia de circulación sanguínea, como representante principal de dichas defunciones. Más en concreto aún, aquella producida fuera de los centros médicos, denominada, parada cardiorrespiratoria extrahospitalaria (PCREH), conforma un problema sanitario de envergadura, puesto que pese a ser un proceso potencialmente reversible, las probabilidades de recuperación son escasas. La pronta intervención del personal médico, juega un papel crucial, así como la correcta identificación del estado del paciente. Se manifiesta la importancia de clasificar con certeza el ritmo cardiovascular del paciente, tanto durante la intervención del Servicio de Emergencias Médicas (SEM), de cara a mejorar el tratamiento al paciente y con ello incrementar su probabilidad de supervivencia, como en análisis posteriores, con el objetivo de estudiar los mejores procedimientos durante un episodio PCREH, como duración y frecuencia de compresiones torácicas o la administración de fármacos.

Para realizar estudios retrospectivos que contribuyan a determinar los efectos de la terapia en la tasa de supervivencia de episodios PCREH, y por lo tanto a determinar las mejores prácticas y terapias, se precisa de bases de datos anotadas, que recojan una colección de grabaciones de pacientes. El problema surge por el método de anotación, intrínsecamente tedioso y costoso, al requerir el análisis y clasificación manual de cientos de horas de registros de múltiples pacientes. La subjetividad humana al observar y distinguir los ritmos cardíacos, genera diferencias y errores entre distintos revisores, haciendo mandatorio aplicar un consenso entre múltiples profesionales. Estas características ocasionan que las bases de datos debidamente anotadas y consensuadas resulten escasas dado su elevado coste.

Por tanto, el objetivo de este trabajo consiste en generar una solución automática para la anotación del ritmo cardíaco de grandes colecciones de datos PCREH. Los datos han sido provistos por BioRes en colaboración con Emergentziak-Osakidetza, así como el aprendizaje de técnicas de procesamiento de señales biomédicas. Por su parte, el JRL aporta métodos de Machine Learning (ML) e Inteligencia Artificial (IA), con los que diseñar un algoritmo de clasificación que permita distinguir una señal de electrocardiograma (ECG) entre 5 patologías cardíacas distintas.

2. Contexto

La parada cardiorrespiratoria extrahospitalaria consiste en un cese inesperado de la función cardiaca, respiración y la consciencia. Este trastorno es generalmente producido como resultado de una alteración eléctrica que interrumpe la acción de bombeo del órgano, deteniendo así el flujo sanguíneo del cuerpo. Solo en España se estima que se producen anualmente 18.6 casos de PCREH por cada 100.000 habitantes [1]. Solo en la comunidad autónoma del País Vasco (CAPV) se producen alrededor de 900 casos/año con una supervivencia al alta hospitalaria del 10% [2].

Al producirse fuera del entorno médico, el paciente es tratado por el SEM. En la CAPV el SEM se organiza en dos niveles. Por un lado, el soporte vital básico (SVB), con una cobertura más amplia, dispone de 48 ambulancias sin médico. Proporcionan atención más rápida pero menos especializada, hasta la llegada del soporte vital avanzado (SVA). El SVA está constituido por 11 ambulancias con médico, y proporciona atención médica avanzada al paciente en PCREH. El personal técnico del SVB conecta el paciente a un desfibrilador externo automático (DEA), o a un monitor-desfibrilador con funcionalidad avanzada en el caso del SVA. Estos equipos monitorizan el ECG del paciente mediante dos parches colocados en el pecho y comprueban si presenta un ritmo desfibrilable (ver Figura 1). La supervivencia del paciente durante un episodio PCREH depende de una desfibrilación temprana, junto con la aplicación de reanimación cardio pulmonar (RCP), puesto que permite alargar la oportunidad de desfibrilación. El DEA analiza el ECG en intervalos regulares (cada dos minutos) y si detecta un ritmo desfibrilable, aconseja realizar una descarga eléctrica.



Figura 1. DEA conectado para simular una descarga, junto con ensayo de RCP.

El DEA analiza el ECG para determinar el ritmo (arritmia) cardíaco del paciente. Para entender las distintas morfologías del ECG, se explica brevemente la anatomía del corazón, que se muestra en la Figura 2. El órgano cardíaco es el encargado de bombear sangre oxigenada por el sistema circulatorio, mediante unas contracciones rítmicas. El corazón dispone de cuatro cámaras: las dos de la derecha se encargan de bombear sangre a los pulmones para recoger el oxígeno, mientras que las dos de la izquierda envían esta sangre oxigenada al resto del cuerpo. En cada lado, se encuentra una cámara superior denominada aurícula, que recibe la sangre del sistema circulatorio y una cámara inferior o ventrículo, encargada de bombear la sangre procedente de la aurícula fuera del corazón.

El ciclo cardíaco se compone de dos fases: durante la diástole se llenan las aurículas, mientras que en la sístole se contraen los ventrículos. Para realizar este proceso de forma eficiente, dichas contracciones se realizan sincronizadamente. El ciclo se inicia mediante una señal eléctrica generada por el nodo sinoauricular, el cual actúa como marcapasos natural, generando latidos a una frecuencia de 60 a 100 latidos por minuto.

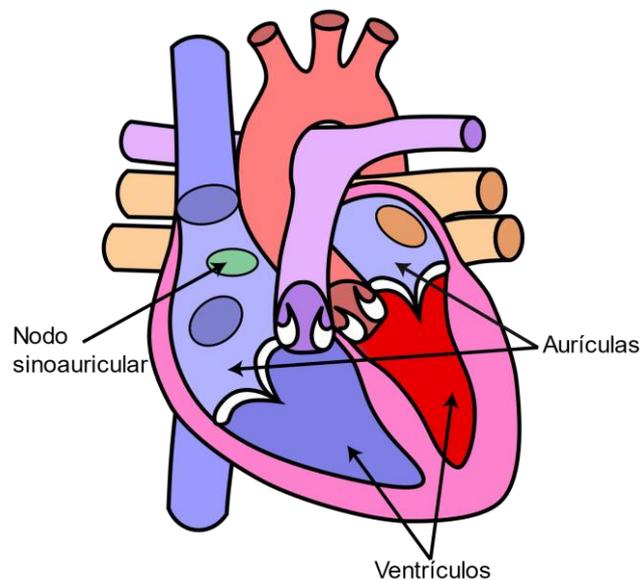


Figura 2. Anatomía del corazón.

Esta señal eléctrica se propaga por las paredes internodales, activando las aurículas (despolarización de las células cardíacas), de forma que pase la sangre a los ventrículos. Tras una pequeña pausa necesaria para llenar completamente los ventrículos, se produce la despolarización de los mismos, realizando una contracción que bombea la sangre fuera del órgano. Por último, las células cardíacas se repolarizan, quedando preparadas para el siguiente ciclo.

La actividad eléctrica ECG, es recogida por los electrodos del DEA colocados sobre la superficie del cuerpo del paciente. La Figura 3 muestra el ECG típico de un ciclo cardíaco normal, denominado ritmo sinusal. Consiste en una onda P, un complejo QRS y una onda T, las cuales caracterizan la despolarización auricular, despolarización ventricular y repolarización ventricular, respectivamente. La repolarización auricular queda enmascarada por el complejo QRS, por lo que no resulta visible en el ECG.

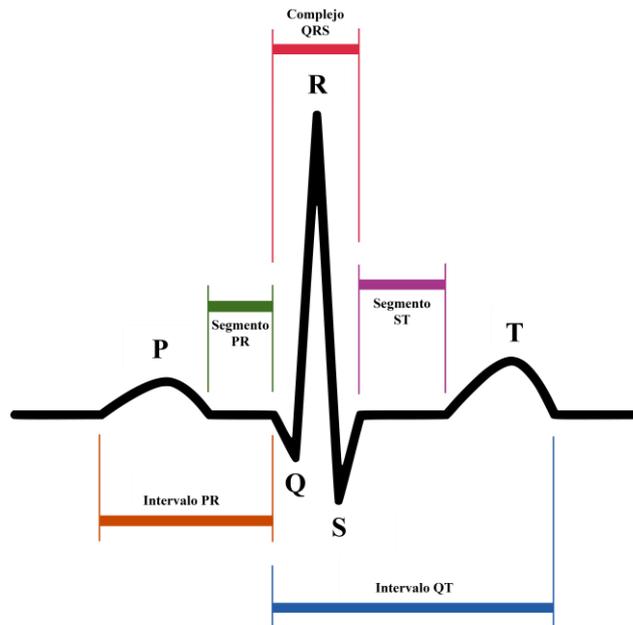


Figura 3. Ritmo sinusal típico.

Una de las causas principales de la PCREH son las arritmias ventriculares malignas, y en particular la fibrilación ventricular (FV). En este estado, el traspaso de sangre a los tejidos del organismo se reduce drásticamente, ocasionando la muerte al cabo de pocos minutos. La única forma de revertir la FV es mediante la desfibrilación, y en ese caso una pronta intervención es esencial.

Durante la PCREH el ritmo cardíaco del paciente puede variar entre cinco ritmos o estados clínicos: ritmo con pulso (RP), actividad eléctrica sin pulso (AESP), taquicardia ventricular (TV), FV y asistolia (AS) [3]. El estado objetivo, el cual indica la recuperación del paciente es RP, en el que se recupera la circulación de sangre. Durante AESP el corazón dispone de actividad eléctrica organizada, pero no posee actividad muscular del miocardio, ni pulso alguno. FV y TV son arritmias ventriculares letales, siendo TV una actividad mucho más regular que FV, que generalmente la precede. Por último, la AS no posee actividad eléctrica ni muscular alguna y conlleva la muerte del paciente.



Figura 4. Ejemplo del ECG para los cinco ritmos/estados clínicos del paciente en PCREH.

Los algoritmos de decisión de los desfibriladores [4]–[8] requieren únicamente una simple clasificación binaria entre ritmos desfibrilables (TV,FV) y no desfibrilables (RP,AESP,AS). Sin embargo, el conocimiento más detallado del ritmo del paciente durante la PCREH mediante su clasificación en 5 tipos permitiría estudios más elaborados como la observación del efecto de la calidad del RCP en la evolución del paciente [9], [10], el análisis de las transiciones entre ritmos asociados a terapia RCP o farmacológica [3], [11], [12], o la detección de pulso [13]–[15] y la transición a los cuidados post-resucitación.

Actualmente, la clasificación de ritmos cardiacos con el fin de analizar episodios de resucitación, precisan de una anotación de manual por parte de expertos clínicos. Las colecciones de datos poseen a menudo cientos de casos y numerosas horas de grabación por caso, tornando el proceso ineficiente.

Para este trabajo se parte ya de una base de datos de PCREH anotada y auditada. En el periodo Junio 2013 – Diciembre 2015, se realizó un estudio con el objetivo de determinar las causas de diagnósticos fallidos por parte del DEA utilizados por el personal del SVB en la atención a la PCREH [16]. Se recolectaron 855 archivos electrónicos de pacientes PCREH mediante DEAs modelos LP100, LP500 y LPCR+ de PhysioControl, y se transformaron a formato MATLAB, de cara a su posterior anotación por parte de seis médicos expertos del SVA. De esta forma se obtuvieron 4409 intervalos ECG anotados bajo cuatro categorías: AS, ritmo organizado (RP y AESP), TV y FV [17]. Posteriormente en un trabajo de fin de máster [18] se desarrollaron las herramientas Matlab para auditar los intervalos de análisis y desarrollar un primer clasificador de ritmos en 3 categorías: desfibrilable, organizado y asistolia. Para desarrollar los algoritmos se empleó el ECG durante los intervalos de análisis, pero la base de datos disponía además de la herramienta Matlab, para auditar los intervalos de análisis DEA, como se muestra en la Figura 5.

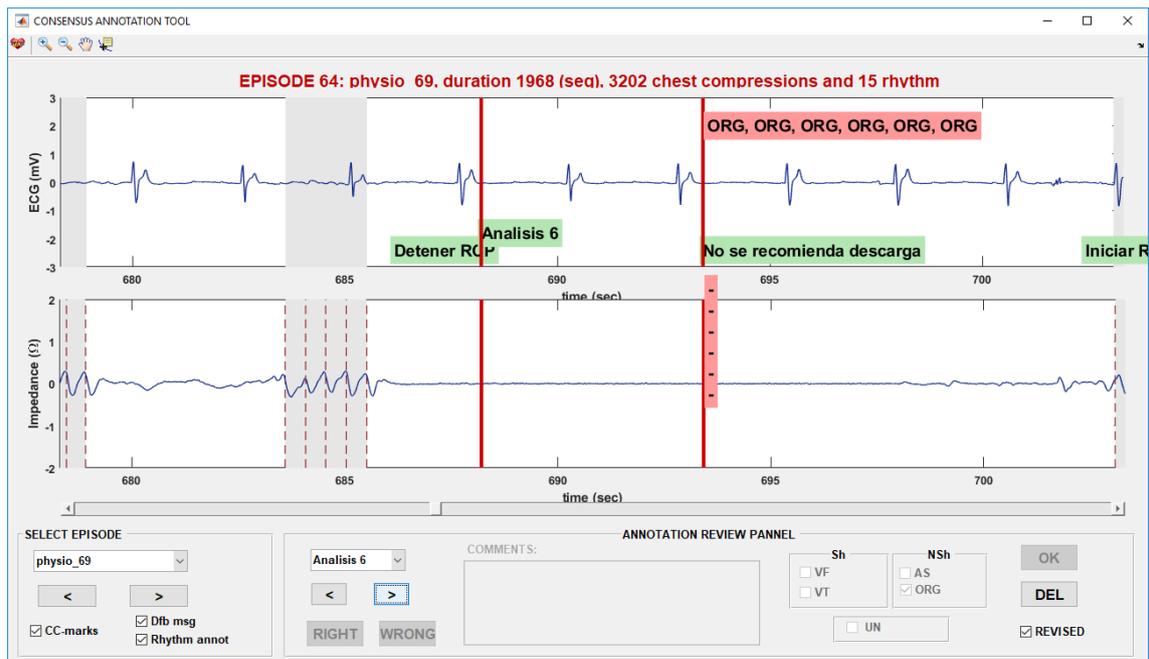


Figura 5. Ejemplo de interfaz creada para inspección de ritmos cardiacos.

La extensión natural de dicho trabajo, partiendo ya de una base de datos semi-annotada (no se dispone de anotación RP/AESP), es desarrollar algoritmos completos de clasificación de ritmos en las 5 categorías que ocurren durante la PCREH. En la actualidad existen numerosos estudios enfocados a la clasificación binaria de ritmos cardiacos como los algoritmos DEA (shock/no shock) o la detección de pulso (RP/AESP), pero escasean aquellos destinados a la clasificación sobre 5 clases. Además, las propuestas de clasificación de ritmo en 5 clases se basan en el procesado del ECG, pero añadir información de señales adicionales como la impedancia podría mejorar el desempeño de los algoritmos. Como se ha explicado previamente, resulta interesante disponer de una colección de datos anotados, para poder estudiar el efecto de los distintos tratamientos aplicados por el SEM ante casos PCREH. Por ello, esta memoria documenta el diseño de un clasificador de ritmos de PCREH de 5 clases, con el que poder anotar de forma automática o semiautomática grandes bases de datos e impulsar así las investigaciones de este ámbito.

3. Objetivos y alcance del proyecto

El objetivo del presente proyecto es el diseño de algoritmos de clasificación de ritmos cardíacos de PCREH, los cuales permitan anotar grandes bases de datos de PCREH de forma automática o semiautomática, disminuyendo el tiempo requerido, así como la carga de trabajo que supone para los expertos contratados la anotación manual.

Para alcanzar el objetivo principal del proyecto deberán completarse algunos objetivos secundarios o más específicos del proyecto que se describen a continuación.

3.1 Base de datos auditada y anotada

La anotación de los datos por parte de los expertos clínicos generó 3035 segmentos de duración variable, clasificados en cuatro clases: AS, ORG, TV y FV. Puesto que el objetivo del proyecto consiste en realizar un clasificador de cinco clases, es necesario distinguir ritmos RP y AESP del conjunto clasificado como ORG (organizado). Para ello, se emplea una segunda señal disponible en las grabaciones: la impedancia transtorácica. El oxígeno presente en la sangre varía la conductividad del pecho (proporciona un camino de menor resistencia), y por lo tanto la impedancia transtorácica cambia con cada latido efectivo. De esta forma, se dispone de un buen indicador de presencia de actividad mecánica en el corazón.

Puesto que los segmentos anotados presentan una duración variable, será necesario definir los intervalos de análisis comunes, y es posible que dichos intervalos de análisis sean considerablemente más numerosos que el número de segmentos anotados. Es decir, se aumentará el número de ejemplos disponibles.

3.2 Clasificador 5 clases mediante señal ECG

En este apartado se trata de diseñar un clasificador de ritmos cardíacos empleando la señal ECG. A partir de ella, se extraen características que modelan la señal, como amplitud, pendiente, frecuencia o complejidad. Se aplican algoritmos de ML clásicos y se trata de mejorar los resultados presentes en el estado del arte.

Se realiza una clasificación de las características en base a la importancia de las mismas durante la clasificación de los ritmos cardíacos. El objetivo consiste en proponer un clasificador que precise de un menor tiempo de operación (al calcular menos características), manteniendo un rendimiento similar.

Por último, se prueban distintas configuraciones del preprocesado de la señal ECG, buscando mejorar el rendimiento del algoritmo.

3.3 Clasificador 5 clases multimodal: señal ECG y señal impedancia

Además de la señal ECG, se extraen también características de la señal de impedancia transtorácica, con el objetivo de mejorar los resultados obtenidos al clasificar los ritmos AESP y RP.

Todos los desfibriladores registran la señal ECG, pero la impedancia transtorácica no está siempre disponible. Sin embargo, se espera mejorar los niveles de precisión de un algoritmo basado únicamente en el ECG debido a la información de actividad mecánica que se observa en la impedancia y de la que no se dispone en el ECG. Este segundo algoritmo no será tan universal,

pero mejorará la precisión del sistema, cuando la impedancia esté disponible en los datos. Se propone un clasificador formado por dos clasificadores en cascada. El primero distingue entre AS, ORG, TV y FV, empleando únicamente la señal ECG. Seguido, las señales marcadas como ORG se clasifican entre AESP y RP, haciendo uso esta vez, de la señal ECG junto a la señal de impedancia transtorácica.

Esta configuración resulta útil para aquellos equipos en los que se dispone de la señal de impedancia transtorácica, los cuales representan la mayor parte de la oferta actual.

3.4 Clasificadores mediante técnicas Deep Learning

Alternativamente, se propone diseñar un clasificador mediante técnicas de Aprendizaje Profundo o Deep Learning. El objetivo se basa en lograr una máquina que permita anotar los 5 ritmos empleando únicamente la señal original. El clasificador toma directamente por entrada la señal ECG, de forma que no es necesario extraer características de ella, puesto que es el propio sistema el encargado de aprender los detalles que identifican cada arritmia.

A su vez, se diseña un segundo clasificador multimodal, el cual toma como señales de entrada la señal ECG y la señal de impedancia transtorácica.

Se compara el rendimiento de los clasificadores Deep Learning con sus correspondientes versiones basadas en ML, estudiando las fortalezas y debilidades de ambos métodos.

4. Beneficios del proyecto

La consecución de los objetivos planteados en este proyecto supondría una serie de beneficios que se pueden agrupar en beneficios técnicos, económicos y sociales. A continuación, se detallan dichos beneficios.

4.1 Beneficios técnicos

En este trabajo se aplicarán nuevas técnicas de procesado del ECG, y procesado de ECG e impedancia, y se estudiarán su poder predictivo para la clasificación en 5 ritmos de PCREH. A su vez se desarrollarán nuevos algoritmos de clasificación en 5 clases, con los que se espera mejorar los resultados del estado del arte. Especialmente al añadir el procesado multimodal de ECG e impedancia. Finalmente, se desarrollarán modelos de clasificación usando los últimos paradigmas de aprendizaje automático basados en aprendizaje profundo.

Todos los aspectos anteriores suponen claras innovaciones técnicas, que permitirán reducir los errores de anotación de ritmos en las bases de datos de PCREH. Además, los desarrollos técnicos asociados al proyecto podrán ser empleados tanto por el grupo BioRes como por el JRL en futuros proyectos del ámbito de la ingeniería biomédica aplicada a la PCREH, o del aprendizaje máquina.

4.2 Beneficios económicos

Actualmente, la clasificación de ritmos cardiacos con el objetivo de realizar un posterior análisis de los episodios de resucitación, conlleva un proceso de anotación tedioso, lento y propenso a errores. Para anotar los volúmenes de datos necesarios para estudios retrospectivos de PCREH se necesitan muchas horas de personal especializado, por lo que en muchas ocasiones los proyectos no son viables o tienen un elevado coste de personal.

Al disponer de sistemas automáticos de anotación de ritmos, el proceso de anotación se puede (semi)-automatizar y agilizar. De esta forma los algoritmos producirían ya una base de datos anotada, y el personal especializado simplemente revisaría el trabajo. Así, mediante un sistema de clasificación automático o semiautomático se ahorrarían muchas horas de inspección manual. La tarea puede reducirse a supervisar las anotaciones realizadas por la máquina, inspeccionar los casos más problemáticos, etc.

Por lo tanto, cuanto mejor sea el rendimiento del algoritmo de clasificación de ritmo, mayor puede ser el ahorro en el presupuesto del estudio, al recortar notablemente las horas de contratación de expertos, de cara a crear la base de datos.

4.3 Beneficios sociales

Los estudios que permiten esclarecer los factores terapéuticos relevantes para la supervivencia a la PCREH son de claro interés social. Actualmente los índices de supervivencia están alrededor del 10%, y solo en la CAPV se producen cerca del millar de PCREH. Por ello, mejoras en la terapia que incrementasen la supervivencia en 2 puntos porcentuales supondrían salvar 20 vidas/año más en la CAPV, lo que tiene un enorme impacto social.

El disponer de nuevos algoritmos de anotación de ritmos PCREH (respuesta del paciente) permitirá abordar estudios retrospectivos, con amplias bases de datos, sobre la influencia de las distintas intervenciones terapéuticas en la respuesta del paciente. Se espera por lo tanto contribuir a mejorar la terapia, y por ende la supervivencia de los pacientes en PCREH.

A su vez, debido a la reducción del coste de anotación de la base de datos, con los presupuestos disponibles ahora se podrían abordar proyectos con mayor volumen de datos. Esto permitiría por ejemplo analizar distintos subgrupos poblacionales (edad, sexo...) o características fisiológicas (dimensiones torácicas), y podrían evaluarse los efectos de la terapia en dichos subgrupos. Es decir, se abriría el camino a personalizar la terapia de PCREH lo que probablemente mejoraría los índices de supervivencia. Actualmente las guías de resucitación solamente distinguen entre pacientes adultos y pediátricos.

5. Análisis del estado del arte

Aprendizaje Máquina o Machine Learning, identifica un campo de las Ciencias de la Computación, en el cual se consigue otorgar a una máquina la habilidad de aprender una tarea sin la necesidad de programarla explícitamente [19]. Surgido a partir del estudio de reconocimiento de patrones y la teoría de aprendizaje computacional, ML crea algoritmos que pueden aprender a partir de una gran colección de datos y hacer predicciones sobre nuevos datos [20]. Actualmente, ML es aplicado a multitud de campos como ciberseguridad, pronóstico de eventos naturales, estudios demográficos o ciencias de la salud. En muchas de estas áreas de conocimiento, en las que diseñar un algoritmo con buen rendimiento resulta muy difícil o inviable, ML ha demostrado ser capaz de ofrecer resultados excepcionales. Ha pasado de ser una disciplina concreta a representar un fuerte motor económico [21]. A su vez, día a día se mejora el rendimiento de los algoritmos hasta el punto de que los seres humanos ya se han visto superados en tareas como reconocimiento de imagen [22] o jugar una partida de Go [23].

5.1 Clasificación de ritmos PCREH

En el caso de las ciencias de la salud, se disponen de amplias colecciones de datos, herencia de la revolución digital. El aprendizaje automático permite analizar esa información y generar sistemas de diagnóstico como sistemas de apoyo para el personal sanitario [24]. Aunque son múltiples sus aplicaciones, en el ámbito médico, el ML es empleado mayoritariamente en la predicción de enfermedades, como puede ser la identificación de arritmias cardíacas. En el caso concreto de la clasificación de ritmos PCREH, existen distintos problemas y por tanto enfoques posibles.

Los algoritmos de los dispositivos DEA se especializan en la detección de arritmias ventriculares letales (TV y FV), ya que son los que se tratan mediante la descarga de desfibrilación [25]. Además, se observan arritmias ventriculares como ritmo inicial en el 75% de los casos en los que se graba un episodio PCREH mediante un DEA público (aquellos disponibles en centros comerciales, aeropuertos, oficinas, etc.) [26]. En estos casos, el único tratamiento efectivo consiste en una pronta desfibrilación que revierta la TV/FV y restaure un ritmo normal. Es por ello, que los DEA incorporan un algoritmo de clasificación binaria entre ritmos desfibrilables (TV y FV) y no desfibrilables (AS, RP y AESP) [27]. Ante la presencia de un ritmo pernicioso, el DEA aconseja la desfibrilación del paciente. De lo contrario recomienda realizar RCP. La American Heart Association (AHA), establece los requerimientos mínimos de precisión por parte del algoritmo, a la hora de clasificar los ritmos cardíacos, con la finalidad de garantizar un correcto uso del dispositivo [27].

Además de la terapia aplicada, existen otros dos factores clave que influyen en la probabilidad de supervivencia del paciente: la identificación del paro cardíaco (ausencia de pulso) y la identificación del retorno espontáneo de circulación (pulso efectivo) durante la terapia [28]. En este campo, el principal desafío reside en poder discriminar ritmos ORG entre ritmos con pulso (RP) y ritmos organizados con función eléctrica disociada respecto de la actividad mecánica (AESP). Las últimas directrices del European Resuscitation Council enfatizan la necesidad de disponer de métodos efectivos de detección de pulso, para lograr mejorar las expectativas de supervivencia ante episodios PCREH [29].

Por último, los distintos tratamientos aplicados durante la resucitación, afectan a la evolución de los ritmos cardiacos. Realizar análisis sobre los patrones de cambio entre distintas arritmias, puede aportar información importante sobre la calidad de la terapia [3], [12], contribuyendo a identificar las mejores intervenciones para distintas circunstancias. Así pues, la clasificación de ritmos resulta fundamental para poder realizar estudios que mejoren los tratamientos durante un episodio PCREH [9]. Actualmente se investiga el diseño de algoritmos de anotación automáticos que permitan enriquecer la cantidad de datos disponibles para estudios retrospectivos, los cuales emplean tradicionalmente anotación manual por parte de expertos clínicos.

A su vez, muchos de estos estudios se basan en datos obtenidos de bases de datos públicas, como MIT-BIH Arrhythmia Database (MITDB) [30], MIT-BIH Malignant Ventricular Arrhythmia database (VFDB) [31] o Creighton University Ventricular Tachycardia database (CUDB) [32], los cuales contienen grabaciones de señal ECG mediante dispositivos Holter. Las señales grabadas mediante este método, presentan unas características entre clases más marcadas respecto de grabaciones de episodios PCREH, las cuales son obtenidas entre 5 y 10 minutos después de producirse el evento, debido al tiempo que tardan los SEM en llegar a la escena de la PCREH. Como se puede apreciar en la Figura 6, las grabaciones PCREH presentan características muy distintas de los ritmos obtenidos mediante dispositivos Holter [33], tanto para ritmos desfibrilables como no desfibrilables. Clasificar ritmos de PCREH es más difícil [33], es por ello importante entrenar los algoritmos con datos obtenidos de PCREH, es decir, en las condiciones en los que se encuentran los pacientes cuya terapia se intenta mejorar..

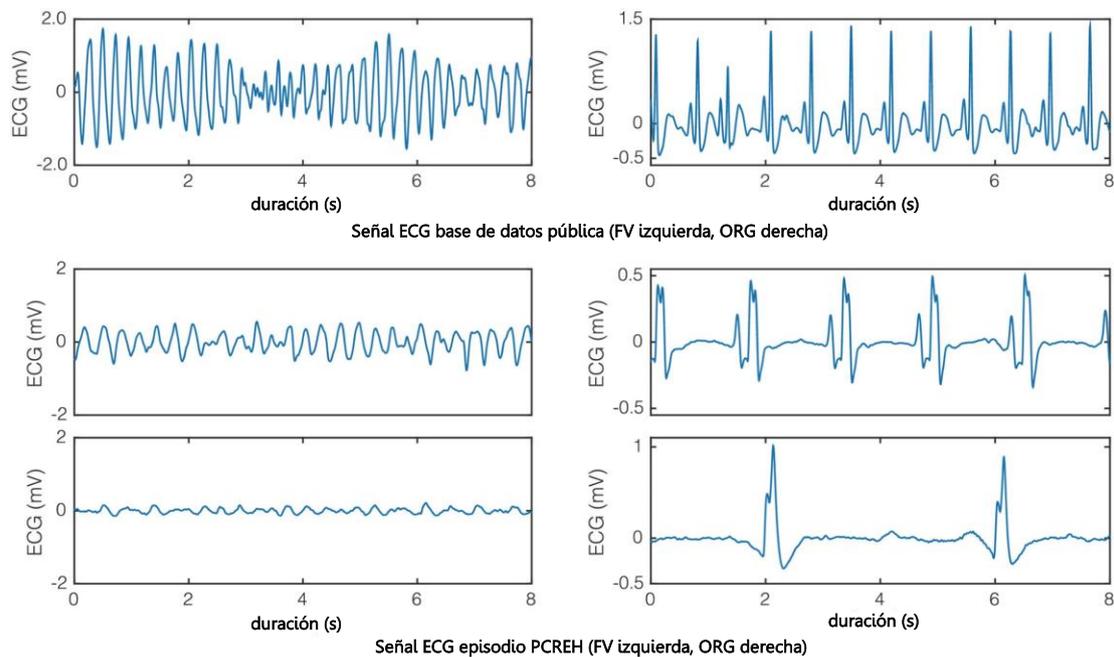


Figura 6. Comparativa entre señales ECG procedentes de base de datos frente a episodios PCREH (adaptada de [33]).

5.2 Validación cruzada

Para poder evaluar el rendimiento del algoritmo de clasificación, es necesario reservar una porción de los datos disponibles como conjunto de test. De esta forma, se elabora un modelo en base a los patrones aprendidos a partir de los datos de entrenamiento, y se comprueba su desempeño al clasificar los datos reservados para test, los cuales son completamente nuevos para el algoritmo. Una proporción típica puede ser un 70-80 % de la colección de datos dedicado a entrenamiento y un 30-20 % restante para test. De esta forma se consigue un compromiso que

permite entrenar el modelo con suficientes datos como para representar la mayoría de escenarios posibles, sin descuidar la verificación del rendimiento obtenido.

Ahora bien, durante la fase de entrenamiento, es preciso validar el rendimiento del algoritmo, de forma que este sea capaz de discernir cuales son los patrones que mejor modelizan las características de las distintas clases. Para realizar este proceso, resulta una mala praxis validar un entrenamiento con los mismos datos empleados para entrenar. De hacerlo así, se obtienen resultados satisfactorios, puesto que el algoritmo trata de clasificar datos que ya ha visto antes y, por tanto, puede inferir estructuras y patrones con mayor facilidad. Sin embargo, estos resultados de validación no reflejan la realidad, ya que el cometido del algoritmo es clasificar datos nuevos. Es por ello que no servirían como indicadores del rendimiento durante la fase de entrenamiento.

Con el objetivo de respetar los datos reservados para test, se aplica una técnica denominada validación cruzada [34]. Este método consiste en realizar un número finito de particiones K , sobre los datos de entrenamiento. Entonces, se entrena el algoritmo con $K-1$ particiones (Figura 7) y se obtienen el rendimiento del clasificador sobre los datos de validación de la partición restante. Repitiendo este proceso K veces, y promediando los K resultados obtenidos, se obtiene un rendimiento más realista. Este permite inferir el verdadero comportamiento del algoritmo, al haber dedicado datos independientes para validación respecto de los empleados durante la fase de entrenamiento. De esta forma, es posible entrenar con todos los datos reservados para entrenamiento, pero obteniendo una estimación no sesgada (validación) del comportamiento del algoritmo para datos nuevos.

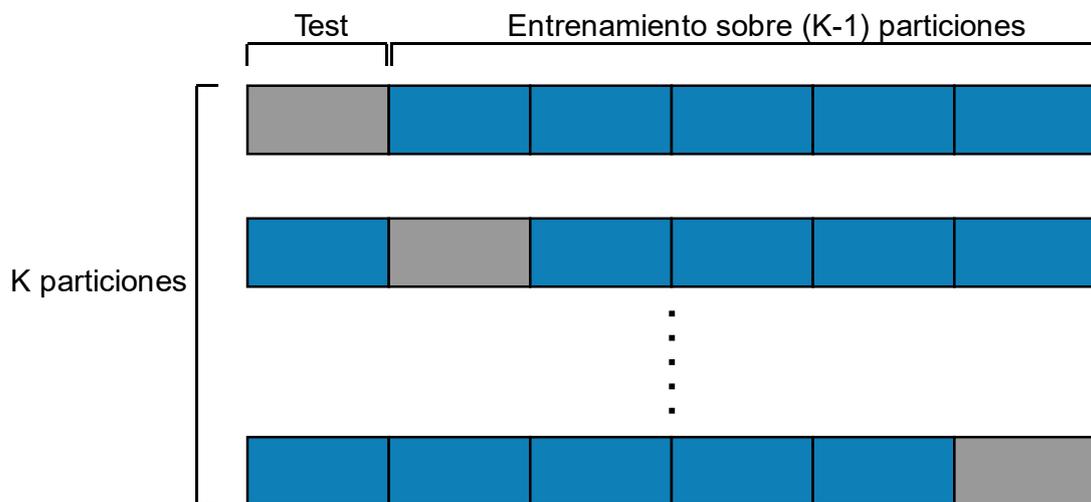


Figura 7. Validación cruzada de K particiones.

Las particiones de datos deben guardar cierto equilibrio respecto a la representación de clases. Si se agrupa una proporción alta de una clase concreta en una partición, se producen dos efectos negativos. Por un lado, el resto de particiones pueden no disponer de datos suficientes de esa clase como para aprender sus características y poder clasificarlas. Además, durante el proceso de validación el rendimiento obtenido será mucho menor de lo esperado, puesto que se producirían una elevada cantidad de errores por pertenecer la mayoría de datos de validación a dicha clase infrarrepresentada. Resulta, por tanto, buena praxis establecer un porcentaje mínimo de representación de cada clase existente, y mantenerlo para todas las particiones, es decir realizar K -particiones estratificadas.

Por último, los datos de pacientes médicos tienen una característica importante: los datos pertenecientes a un mismo paciente guardan una fuerte semejanza, es decir, están correlados. Esta cualidad origina un problema durante el proceso de formación de las distintas particiones necesarias para realizar validación cruzada. De la misma forma que no se deben usar datos de test para entrenar el modelo, la partición de datos reservada para validación debe contener datos sin relación respecto al resto de datos. Volviendo a la correlación presente entre datos de un mismo paciente, es preciso agrupar todas las grabaciones disponibles bajo una misma partición, es decir hacer K-particiones estratificadas con separación de pacientes.

5.3 Evaluación del algoritmo

Los algoritmos de clasificación, disponen de múltiples métricas que permiten evaluar su rendimiento. En la Figura 8, se muestra gráficamente las métricas básicas de un clasificador binario, a partir de las cuales se obtienen indicadores más complejos. La clase de interés (en lo que engloba este proyecto el ritmo cardiaco que se quiere detectar), se denomina elemento relevante o positivo. Los elementos seleccionados son aquellos clasificados por el algoritmo como elementos positivos, pertenezcan realmente a la clase (verdaderos positivos) o no (falsos positivos). De la misma forma se obtienen los elementos clasificados como negativos de forma correcta (verdaderos negativos) e incorrecta (falsos negativos).

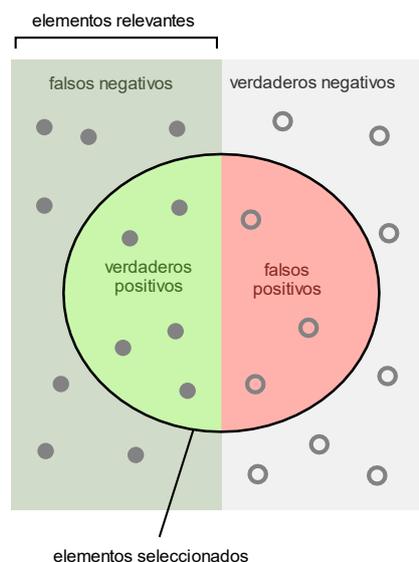


Figura 8. Representación visual de la clasificación en dos clases (positiva y negativa), un clasificador binario.

Las métricas para medir el desempeño de los algoritmos se calculan comparando los diagnósticos del algoritmo con la clasificación (ground truth) de consenso de un grupo de expertos. Se puede medir así el número N de instancias reales de una clase j , clasificada como ritmo de clase i . Y dichos números N_{ij} pueden agruparse en una matriz denominada matriz de confusión. Por tanto, en este trabajo en el que se aborda un algoritmo de clasificación de 5 clases se tiene $i, j \in \{1,2,3,4,5\}$ y una matriz de confusión 5×5 . Se ha designado los ritmos AS (1), RP (2), AESP (3), FV (4) y TV (5), tal y como se muestra en la Tabla 1.

Tabla 1. Matriz de confusión para clasificación de ritmos PCREH.

		<u>Consenso de los expertos</u>				
		AS	RP	AESP	FV	TV
Algoritmo	AS	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}
	RP	N_{21}	N_{22}	N_{23}	N_{24}	N_{25}
	AESP	N_{31}	N_{32}	N_{33}	N_{34}	N_{35}
	FV	N_{41}	N_{42}	N_{43}	N_{44}	N_{45}
	TV	N_{51}	N_{52}	N_{53}	N_{54}	N_{55}

Verdaderos positivos (VP), falsos positivos (FP) y falsos negativos (FN) de cada clase i son calculados tal que:

$$VP_i = N_{ii} \quad FP_i = \sum_{\substack{j=1 \\ (j \neq i)}}^5 N_{ji} \quad FN_i = \sum_{\substack{j=1 \\ (j \neq i)}}^5 N_{ij}$$

En el ámbito médico, se emplean tradicionalmente las siguientes métricas: sensibilidad (Se), precisión (P) y F₁-score (F₁):

- **Se.** Indica la capacidad del clasificador para clasificar el ritmo correctamente. Para un ritmo concreto, se computa como el número de casos correctamente clasificados, Verdaderos Positivos (VP), entre el número total de casos clasificados como tal, sean VP o ritmos de otros tipos, denominados Falsos Negativos (FN).
- **P.** Concreta el número de casos clasificados correctamente VP, respecto del número total de predicciones de esa clase, sean verdaderas (VP) o falsas (FP).
- **F₁.** Considera por igual Se y P, calculándose como la media armónica de ambas métricas.

Las métricas de rendimiento del algoritmo son obtenidas para cada clase tal que:

$$Se_i = \frac{VP_i}{VP_i + FN_i}$$

$$P_i = \frac{VP_i}{VP_i + FP_i}$$

$$F_{1i} = \frac{2Se_i P_i}{Se_i + P_i} = \frac{2TP_i}{2TP_i + FN_i + FP_i}$$

La clasificación de ritmos PCREH es un problema con clases fuertemente desbalanceadas, dadas las prevalencias de ciertos ritmos, como AS o AESP, en la base de datos. Por tanto, se escogen dos métricas como indicadores del rendimiento del clasificador, de forma que todas las clases se encuentren igualmente representadas: unweighted mean of sensitivities (UMS) y unweighted mean of F1-scores, es decir, el promedio para todas las clases de Se y P:

$$UMS = \frac{1}{5} \sum_{i=1}^5 Se_i \quad mF_1 = \frac{1}{5} \sum_{i=1}^5 F_{1i}$$

5.4 Características señal ECG

Para entrenar un modelo de ML, es necesario extraer características de la señal ECG. Por ello, en este apartado se realiza un análisis de las características que mejor se adaptan al marco de este proyecto.

5.4.1 Detección de ritmos desfibrilables

Primero, se buscan características que permitan distinguir ritmos desfibrilables. En la literatura existen numerosos estudios en este campo, puesto que los DEA funcionan mediante algoritmos binarios que clasifican ritmos letales y no letales. Si se detecta un ritmo letal, se aplica un shock que desfibrile al paciente. Por tanto, las características están diseñadas para identificar ritmos letales.

Un buen ejemplo es el estudio de *Figuera et al.* [33], trabajo que sirve como compendio de las mejores características de cara a la detección de ritmos desfibrilables. *Figuera et al.* analiza 30 características, que modelan la amplitud, pendiente, distribución de las muestras de la señal ECG y el ritmo cardiaco del paciente [4], [35]–[40], concentración de la señal en el espectro y potencia de las distintas bandas frecuenciales [4], [40]–[42], relaciones tiempo-frecuencia [8] y medidas que representan la complejidad de la señal [43]–[46]. En esta investigación comparan el rendimiento de estas características, para algoritmos entrenados con datos de bases de datos públicas (monitorización Holter) frente a datos de tipo PCREH. Gracias a este estudio es posible seleccionar las características de mayor importancia que permitan diferenciar ritmos desfibrilables de tipo PCREH.

5.4.2 Detección de circulación sanguínea

Con el fin de identificar RP y AESP, existen publicaciones que resaltan la utilidad de emplear características extraídas de la señal de impedancia transtorácica (IT), puesto que está directamente relacionada con la presencia de pulso del paciente. Habitualmente, los DEA son capaces de captar únicamente el ECG y la señal de impedancia transtorácica, adquiridas mediante los parches de desfibrilación. Ambas señales han demostrado ser de utilidad a la hora de detectar el pulso en ritmos organizados [14],[47]. La señal IT muestra fluctuaciones de menos de 100 mΩ con cada latido del corazón. Ciertos trabajos obtuvieron buenos resultados clasificando RP y AESP, mediante una combinación de características obtenidas de las señales ECG e IT [13],[47].

En este campo, resalta el trabajo de *Elola et al.* [48], en el cual se obtiene una señal adicional, a partir del ECG y la impedancia, mediante una versión modificada del detector de complejos QRS o latidos del corazón de Hamilton-Tompinks [49]. Esta tercera señal se denomina componente de circulación de impedancia (CCI) y asocia la señal IT con los latidos presentes en el ECG. El detector, cuya arquitectura se muestra en la Figura 9, fue definido en *Alonso et al.* [15]. Dicho detector solo se activa ante la presencia de un ritmo ORG y está diseñado para operar en conjunto con el algoritmo de aviso de desfibrilación del DEA, de forma que añada nueva información útil al técnico sanitario.

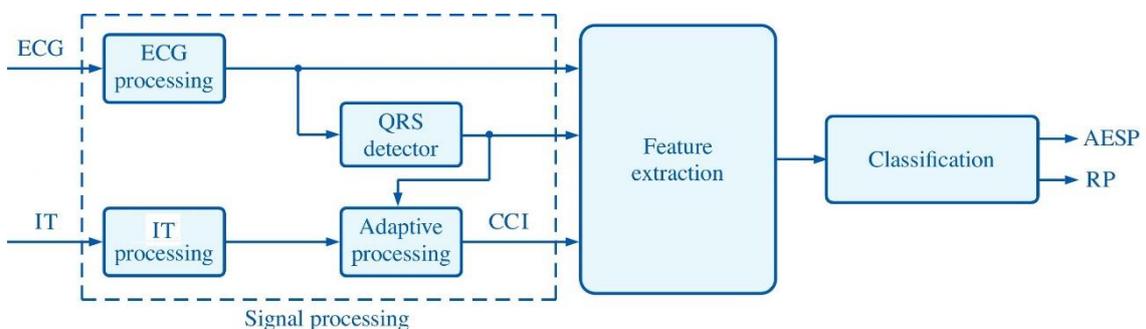


Figura 9. Esquema funcionamiento detector de circulación. (adaptada de [15]).

Mediante la señal ECG se obtienen las posiciones de los complejos QRS (contracción ventricular), que se corresponden con los latidos del corazón. Ahora, a partir de los instantes de los latidos y la señal IT, se calcula CCI.

En conjunto, las tres señales permiten extraer un total de 9 características que modelizan los complejos QRS (más estables y de mayor amplitud en RP) [13], [50], pendiente del ECG (más brusca en RP), distribución de la energía en los armónicos (RP tiende a ser más periódico) [39], [51] y regularidad de la señal [52].

5.4.3 Clasificación en 5 ritmos

Debido al carácter de este proyecto, la línea de investigación que más se asemeja es la seguida por *Rad et al.* [53]–[55]. En su última publicación, define un total de 47 características extraídas únicamente a partir del ECG [56], con la que elabora un anotador automático de ritmos ECG de 5 clases.

La señal ECG es descompuesta mediante la Discrete Wavelet Transform (DWT) [57] en distintas sub-bandas, representadas en forma de coeficientes, a partir de los cuales se obtienen las primeras 32 características. Luego, se añaden 12 características clásicas en la detección de ritmos letales, aquellas que obtuvieron un mejor desempeño en el trabajo de *Figuera et al.* [33]. Por último, las 3 características restantes miden el número de latidos, la duración media entre ellos y la dispersión del ciclo cardíaco.

Todos estos trabajos conforman el punto de partida del proyecto, asentando las bases del conocimiento necesario para desarrollar un anotador automático de 5 ritmos cardíacos.

5.5 Algoritmo de clasificación

Tras calcular las características de cada segmento de señal ECG, se disponen de los datos necesarios para elaborar un modelo con el que definir el clasificador de ritmos cardíacos. Para este proyecto se ha escogido el algoritmo Random Forest [58].

Random Forest (RF), es un algoritmo formado por una combinación de múltiples árboles de decisión. Cada árbol de decisión proporciona una predicción de cuál es la clase a la que pertenece el ritmo cardíaco. Para el caso de problemas de clasificación, Random Forest obtiene una clase como voto de cada árbol y selecciona como salida la clase con más votos.

5.5.1 Árboles de decisión

Los métodos basados en árboles, particionan el espacio de características X_i en un conjunto de áreas, asociando cada una a una salida, por ejemplo, cual es la clase predicha.

Consideremos el caso de un problema de clasificación. Para una respuesta Y (ritmos cardíacos posibles) y dos entradas X_1 y X_2 (dos características cualesquiera de las definidas en el apartado anterior), se realizan particiones binarias de forma recursiva, en función de los valores de X_1 y X_2 . Se asigna entonces, para cada región un valor de salida para Y . Se repite el proceso hasta alcanzar algún término impuesto. Por ejemplo, se limita a 5 regiones (Figura 10), identificando cada región a cada ritmo cardíaco posible.

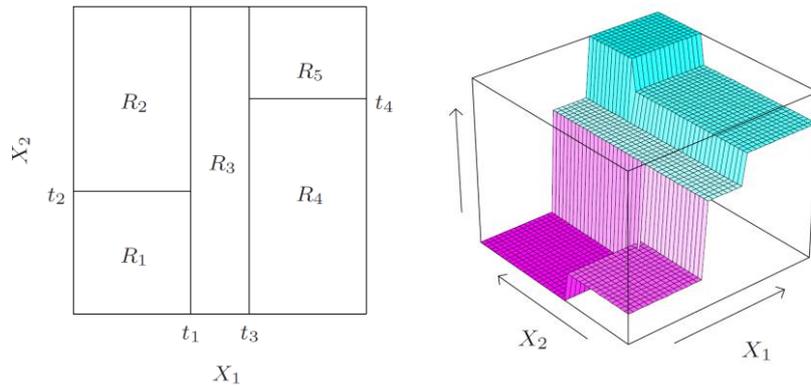


Figura 10. Representación 2D y 3D de las regiones de decisión definidas por X_1 y X_2 .

Los árboles de decisión, necesitan un mecanismo automático que permita escoger la mejor partición binaria en cada nodo. Este criterio varía en función del objetivo del algoritmo, como puede ser reducir el error cuadrático medio, disminuir el número de falsos positivos, etc.

El modelo, a su vez puede representarse mediante un árbol binario (Figura 11). Las observaciones que cumplen la primera condición (umbral t_1) se asignan a la rama izquierda y el resto a la derecha. Se repite el proceso hasta completar el árbol. Los nodos terminales u hojas del árbol se corresponden con las regiones R_1, R_2, \dots, R_5 .

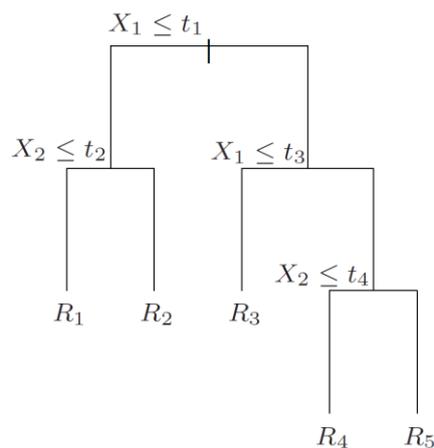


Figura 11. Diagrama de un árbol de decisión

Ante la duda sobre cuanto dejar crecer el árbol, se debe tener en cuenta que cuanto mayor sea su profundidad, mayor será la tendencia a obtener un modelo sobreajustado. Si por el contrario se escoge un árbol demasiado pequeño, quizás el modelo no consiga representar la estructura de los datos. Por lo tanto, la profundidad del árbol es uno de los parámetros a optimizar, puesto que gobierna la complejidad del modelo.

5.5.2 Random Forest

La idea esencial de RF es promediar muchos modelos ruidosos, pero notablemente imparciales, de forma que se reduzca la varianza del resultado obtenido. Los árboles son buenos candidatos para esta técnica, puesto que son capaces de modelar interacciones complejas entre datos. La técnica de Random Forest genera varios árboles de decisión de forma que dichos árboles se crean empleando particiones de datos aleatorias y variables de forma aleatoria en cada rama de decisión. Se consigue de esta forma que los árboles que generan el modelo final estén incorrelados.

Los parámetros de optimización de Random Forest son:

- **Número de predictores m .** Número de variables de entrada (características de la señal ECG) empleadas en las bifurcaciones. El valor por defecto es $m = \sqrt{n^{\circ} \text{ de entradas}}$
- **Número mínimo de nodos.** Valor por defecto es 1
- **Numero de árboles.** Cuanto mayor sea el número de árboles menor será la varianza de las decisiones tomadas por el algoritmo, pero se incrementa el coste computacional por cada nuevo árbol entrenado. Una práctica común consiste en buscar el número mínimo de árboles para el cual no se mejora el rendimiento del algoritmo.

Cuando el número de variables de entrada es grande, pero la proporción de variables relevantes es pequeña, Random Forest ofrece un rendimiento pobre para valores pequeños de m . Para cada partición, la probabilidad de emplear las características relevantes es pequeña. Sin embargo, al aumentar el número de variables relevantes, el rendimiento de Random Forest se vuelve muy robusto frente a un aumento del número de variables ruidosas. Esta robustez se debe en gran medida a la relativa insensibilidad del costo de clasificación errónea con respecto al sesgo y la varianza de las estimaciones de probabilidad en cada árbol.

Como ejemplo, para 6 variables relevantes y 100 variables ruidosas, $m = \sqrt{6 + 100} \approx 10$, de forma que, para cada partición del árbol, la probabilidad de que al menos una de esas 10 variables sea relevante es del 46 %.

Otra ventaja de Random Forest es que la probabilidad de sobreajustar el modelo (Figura 19) a los datos de entrenamiento se reduce al aumentar el número de árboles, puesto que se escoge el valor más votado por todos los árboles y estos están incorrelados al escogerse aleatoriamente.

5.6 Técnicas de Deep Learning

El rendimiento de soluciones basadas en ML, dependen fuertemente del diseño de las características empleadas. Si dichas características no concretan las diferencias entre distintos ritmos, no se consigue un clasificador eficaz.

Una alternativa a ML, consiste en permitir que el algoritmo se encargue de extraer y seleccionar de forma autónoma las características de la señal. Este es el principio fundamental detrás de Deep Learning (DL) o Aprendizaje Profundo. Esta técnica, consiste en elaborar una red con distintas capas, las cuales realizan distintas operaciones matemáticas, cuya entrada consiste únicamente en la señal ECG.

5.6.1 Capa convolucional

Las capas convolucionales son las encargadas de extraer los descriptores de alto nivel de la señal ECG. Primero, se define un número N de muestras, que abarca el filtro convolucional (kernel). Luego se indica un número M de filtros aplicados. Cada filtro se inicializa con unos pesos aleatorios y se multiplica el valor de dichos pesos con el de las muestras correspondientes. Recorriendo la señal completa se consigue aumentar la dimensionalidad de la entrada por el número de filtros aplicados (Figura 12). Los pesos de los filtros se ajustan en cada etapa de entrenamiento, buscando reducir el error producido en validación. Para señales unidimensionales como el ECG, se emplean capas Conv-1D (convolución de una dimensión).

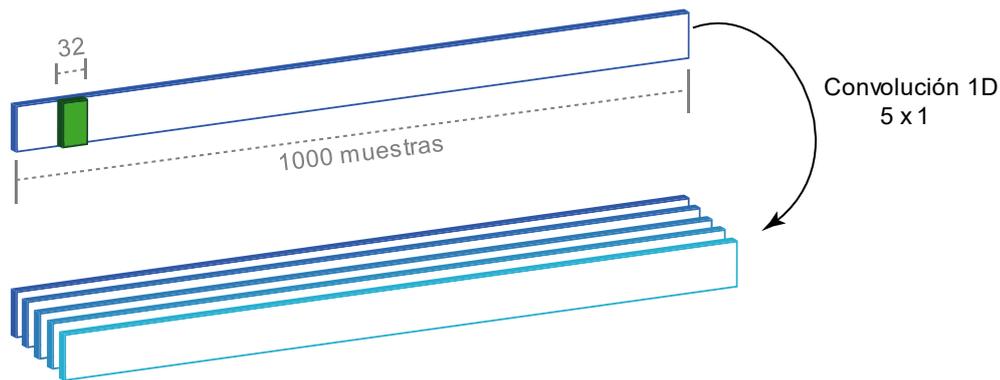


Figura 12. Ejemplo capa Conv1D de M=5 filtros con kernel de N=32 muestras

5.6.2 Capa función de activación

Las redes artificiales precisan de funciones de activación, con el objetivo de convertir dichas redes en funciones no lineales. Una función de activación conecta dos nodos sucesivos de la red (por ejemplo, dos capas convoluciones) y modifica el valor de entrada del segundo nodo en relación al valor de salida del primero. En caso de no aplicar funciones de activación, la red neuronal podría ser expresada como una función lineal. Este tipo de funciones no son las más adecuadas para resolver problemas complejos, puesto que los procesos presentes en la naturaleza no son habitualmente lineales. Por ello, las funciones de activación suelen emplear algún tipo de no-linealidad.

Durante la fase de aprendizaje de la red, se compara la salida obtenida con la salida real y se calcula el error obtenido. Entonces, para reducir dicho error, se ajustan los pesos internos de la red según el gradiente del error obtenido. Por ejemplo, en el caso de una capa convolucional, se modificarían los pesos del filtro aplicado. El ajuste de los pesos se realiza propagando un gradiente desde la salida de la red hasta la capa de entrada. Sin embargo, cuando se emplean funciones de activación tradicionales como Sigmoido o Tangente hiperbólica (Figura 13), el gradiente se desvanece para redes profundas (con muchas capas), haciéndose más pequeño tras cada capa [59]. Esto es debido a los valores límites de ambas funciones, entre 0 y 1 y -1 y 1, respectivamente. Como consecuencia, las primeras capas de la red precisan más iteraciones para entrenarse, ya que el gradiente del error recibido, con el que actualizar los pesos, es demasiado pequeño.

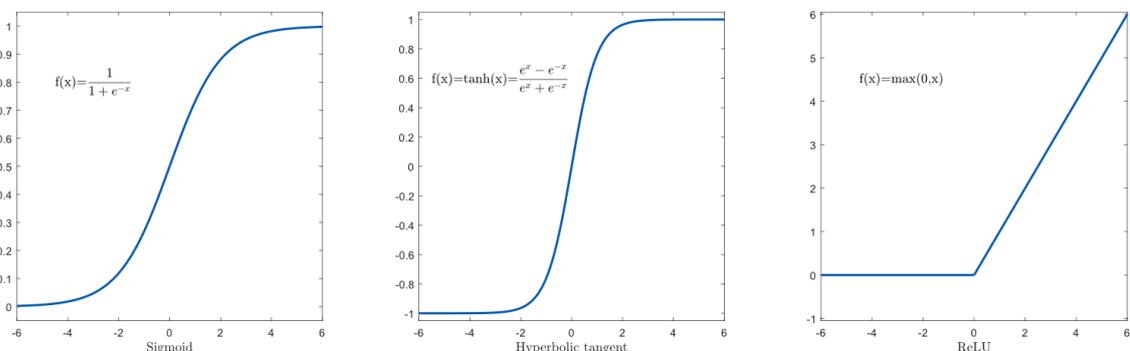


Figura 13. Comparativa entre distintas funciones de activación.

Con el objetivo de conservar el valor del gradiente del error para todas las capas de la red, surge Rectified linear activation unit (ReLU), el tercer ejemplo en la Figura 13. Esta función anula el

gradiente para entradas negativas y transfiere su valor original para entradas positivas [60]. La conservación del valor del gradiente, junto con su velocidad de cómputo, debido a la sencillez de su expresión, reduce notablemente la velocidad de la fase de entrenamiento de la red.

5.6.3 Capa MaxPooling

Para reducir el tamaño de la señal de salida de la capa convolucional, se aplica una capa MaxPooling, la cual permite conservar la información más importante. Para ello, se define un filtro de N muestras y se aplica a toda la señal de entrada. Para cada posición, se almacena únicamente el dato de mayor valor de las N muestras. Este proceso permite reducir el coste computacional de entrenamiento de la red, al reducir por un factor de N el número de muestras de la señal de entrada a la capa MaxPooling.

5.6.4 Capa recurrente

Existen otro tipo de capas denominadas recurrentes (RNN). Su cualidad principal consiste en su capacidad para persistir la información, de forma que puede realizar predicciones en base no solo de los datos actuales sino también sobre aquellos pasados. Este proceso se realiza tomando como parte de la entrada a cada nodo, la salida del anterior.

En concreto, la capa Long Short-Term Memory (LSTM) [61] se perfila como una de las mejores opciones dentro de las capas recurrentes, puesto que resuelve el problema del desvanecimiento del gradiente del error durante la fase de entrenamiento, presente en capas recurrentes de muchos nodos. En dicha capa, se define un tamaño de máscara, el cual abarca N muestras de la señal de entrada (Figura 14). Entonces, se calcula la señal de salida, en función de los valores pasados de las anteriores $N-1$ muestras. Para cada etapa de entrenamiento, se ajusta la información retenida por la capa LSTM.

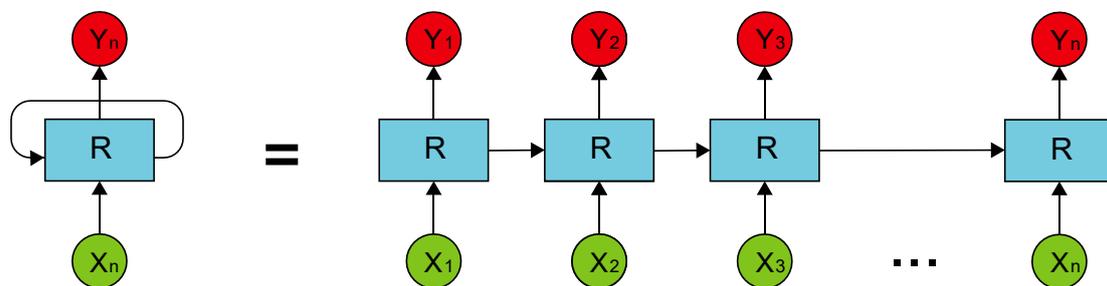


Figura 14. Capa LSTM descompuesta en N unidades fundamentales

Existe otra capa recurrente denominada Gated Recurrent Layer (GRU), introducida por *Cho et al.* [62], basada en LSTM, pero con un diseño más sencillo. Permite entrenamientos de menor duración en comparación a la LSTM, ofreciendo resultados satisfactorios [63].

5.6.5 Capa Densa

Para terminar, se emplean capas neuronales, también denominadas capas densas. El objetivo de estas últimas capas es partir de las características aprendidas bien por una Convolutional Neural Network (CNN) o por una RNN y clasificar la señal, en nuestro caso el ritmo. Cada capa densa se compone de un número concreto de neuronas, que actúan como nodos. Estos nodos se encuentran conectados a todos los nodos de las capas anterior y posterior (Figura 15). La influencia que tiene cada entrada en el valor devuelto por cada neurona (valores altos indican una mayor activación de la neurona), se ajusta durante el entrenamiento mediante pesos individuales para cada nexo. La última capa de la red de clasificación dispone de tantas neuronas

como ritmos clasificables, de forma que se toma como predicción la clase asociada a la neurona de mayor activación.

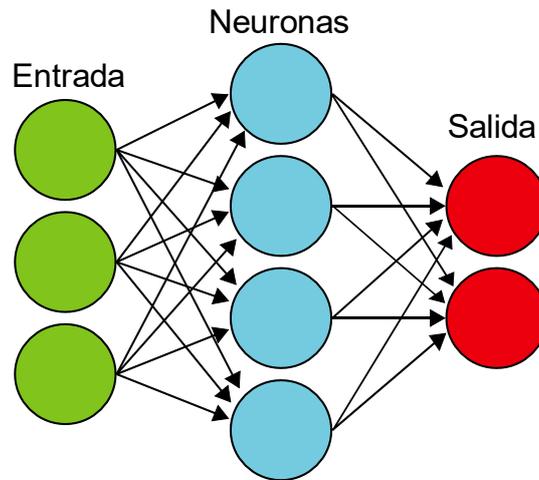


Figura 15. Ejemplo capa densa de 4 neuronas

En los últimos años, se han realizado numerosas investigaciones empleando sistemas DL en el ámbito de la detección de arritmias cardiacas, con resultados satisfactorios [64]–[66]. Todos estos estudios hacen uso de CNNs, basadas en capas convolucionales unidimensionales.

Resulta interesante el trabajo de *Picón et al.* [67], en el cual se presenta una arquitectura DL basada en CNN, como solución a un problema de clasificación binaria entre ritmos ventriculares letales y aquellos no letales. A su vez, añade una capa recurrente LSTM a la salida de los bloques convolucionales (formados por capas Conv-1D, ReLU y Maxpooling-1D), antes de la capa densa. Los resultados obtenidos mejoraron el estado del arte previo en más de dos puntos, para datos de tipo PCREH [10], [68], por lo que puede resultar un buen punto de partida para las versiones iniciales de la red neuronal propuesta en este trabajo.

Otra propuesta a mencionar es la descrita por *Elola et al.* [69], en la que se analizan el desempeño de dos redes neuronales: una basada en capas Conv-1D y otra mediante capas GRU. En este caso tratan de resolver otro problema binario: la clasificación de ritmos RP y AESP durante episodios PCREH. Además, *Elola et al.* introduce el uso de optimización Bayesiana [70] a la hora de escoger los hiperparámetros de la red neuronal, con el objetivo de diseñar la mejor arquitectura posible.

6. Análisis de alternativas

En este apartado se presentan estudios sobre las alternativas consideradas en el desarrollo del proyecto, en particular el lenguaje de programación, los algoritmos de clasificación y la librería empleada para implementar la solución mediante Deep Learning.

6.1 Lenguaje de programación

Originalmente, se proponen 4 lenguajes de programación: Matlab, Octave, Python y R. En los siguientes apartados se analizan las características fundamentales de los mismos, y el beneficio que conlleva su uso.

6.1.1 Matlab

MATLAB o MATrix LABoratory es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) propio, basado en un lenguaje de programación también propio, el lenguaje M. MATLAB es una herramienta enfocada a la manipulación de matrices, representación de datos y funciones, creación de interfaces gráficas de usuario (GUI) e implementación de algoritmos.

Dispone de un sistema de *toolboxes*, con el que ampliar sus capacidades. Estos paquetes de código convierten a MATLAB en una herramienta modular con opciones específicas para la computación en paralelo, procesamiento de señal, procesamiento de imagen y visión artificial, ML y DL, conectividad con bases de datos, herramientas de verificación, validación y test, etc.

MATLAB como IDE, está programado en C (entre otros lenguajes) y por tanto dispone de interfaz con este lenguaje. Por eso, es posible llamar a funciones y subrutinas escritas en C, haciendo uso de compiladores internos de la herramienta.

MATLAB es un software propietario de MathWorks y por lo tanto su uso se halla ligado a la compra de su correspondiente licencia. Sin embargo, existen versiones gratuitas para estudiantes, las cuales disponen de los *toolbox* básicos, suficientes como para resolver los problemas a surgir en este proyecto.

6.1.2 Octave

Octave es una herramienta de cálculo numérico que surge como alternativa *open source* a MATLAB. Por lo tanto, ofrece también una IDE y lenguaje de programación propio, muy similar a MATLAB, pero con ciertas modificaciones, de acuerdo a las peticiones más populares de la comunidad *open source*.

Entre sus mayores inconvenientes, destacar que su desarrollo se limita a seguir las actualizaciones y nuevas funciones de MATLAB. Por ello, en el mejor de los casos ofrece unas características similares a MATLAB, pero el soporte al usuario es deficiente en comparación. Además, los bugs tardan más tiempo en ser resueltos con nuevos parches de actualización.

Su interfaz gráfica resulta menos intuitiva y manejable, puesto que su diseño se enfoca a la funcionalidad. En el ámbito de este proyecto, en el que es preciso generar GUIs para examinar distintos segmentos de señal ECG, es una cualidad importante.

6.1.3 Python

Python es un lenguaje de programación multiparadigma, de forma que no se diseñó para una tarea en concreto, como MATLAB u Octave, sino que permite escoger las técnicas y estilos de programación que mejor se adaptan al problema a resolver.

La sintaxis de Python está diseñada para facilitar un rápido aprendizaje. Uno de sus puntos fuertes es su legibilidad, ya que al prescindir de llaves, punto y coma y lenguaje tipado, entre otros puntos, propensa una lectura de código amigable.

Debido a su elevada popularidad, Python dispone de una inmensa cantidad de librerías. En concreto, para problemas de ML y DL existen multitud de librerías ampliamente utilizadas y mantenidas por la comunidad. Este soporte facilita el desarrollo de soluciones mediante Python.

Posee licencia de código abierto, de forma que su uso no se encuentra ligado a pago alguno.

6.1.4 R

R es un lenguaje de programación multiparadigma, al igual que Python, pero en este caso se halla más enfocado a los ámbitos de investigación, matemáticas y manejo de datos.

Se posiciona como el lenguaje más complicado de aprender y utilizar, debido a excentricidades en su sintaxis, que no es intuitiva. Por ejemplo, existen distintas formas de realizar las mismas tareas, como analizar, nombrar o seleccionar variables. Sin embargo, esta sintaxis convierte este lenguaje en una herramienta poderosa y versátil.

Aun definiéndose como multiparadigma, R es empleado principalmente para análisis estadístico, por lo que sus librerías más importantes pertenecen a este ámbito, útil para el procesado de los datos, pero más pobre para el desarrollo de algoritmos.

6.2 Criterios de selección: lenguaje de programación

En este apartado se enumeran y explican los criterios seguidos durante el proceso de selección del lenguaje de programación. Estos son: sencillez de uso, capacidad de procesado y gestión de datos, capacidad de desarrollo de los distintos algoritmos existentes, eficiencia del sistema y precio de la licencia de uso.

Con el objetivo de procurar una comparación objetiva, se asigna un sistema de pesos a cada una de las características analizadas.

6.2.1 Sencillez de uso

Las plataformas con mayor cantidad de opciones y recursos a menudo conllevan una curva de aprendizaje para el usuario más complicada que aquellas herramientas focalizadas en tareas más concretas. El dominio del software genera soluciones más elaboradas, ya que facilita materializar ideas en forma de código. A su vez, el desarrollo del proyecto se produce con mayor rapidez.

Importancia: 2/10

6.2.2 Procesado y gestión de datos

Debido al carácter del proyecto, es preciso gestionar los datos médicos. Se deben filtrar los casos con ruido externo o transitorios, clasificar los ritmos ORG como RP o AESP y etiquetar cada segmento de grabación con el identificador de paciente correspondiente, entre otras tareas. Después, de cara a la extracción de características necesarias para proceder a la fase de

entrenamiento, es necesario procesar los datos. Es por tanto la cualidad de mayor importancia a la hora de escoger el lenguaje de programación.

Importancia: 3/10

6.2.3 Desarrollo de algoritmos

Las distintas opciones disponibles ofrecen en su mayoría los mismos algoritmos de clasificación. Sin embargo, lo que si difiere es la cantidad de parámetros disponibles para modificar. Existen opciones simples que facilitan el desarrollo de soluciones simples, pero un mayor abanico de opciones permite una solución más ajustada al problema planteado.

Importancia: 2,5/10

6.2.4 Eficiencia

La eficiencia en el tiempo de ejecución de las distintas soluciones diseñadas durante el proyecto influye directamente en el tiempo necesario para entrenar los modelos de clasificación.

Importancia: 2/10

6.2.5 Precio licencia

El precio de la licencia influye directamente en el coste final del proyecto. Sin embargo, en relación al coste de las horas de recursos humanos, representa una fracción pequeña, por lo que se le aplica el menor peso de todas las características.

Importancia: 0,5/10

6.3 Selección de alternativas: lenguaje de programación

Habiendo definido las características de evaluación de cada alternativa, se procura la Tabla 2 como resumen de la distribución de los pesos.

Tabla 2. Comparativa lenguajes de programación

<i>Criterio</i>	<i>Peso</i>	<i>MATLAB</i>	<i>Octave</i>	<i>Python</i>	<i>R</i>
<i>Sencillez de uso</i>	2/10	2	1	1	0,5
<i>Procesado y gestión de datos</i>	3/10	3	2	2	2,5
<i>Desarrollo de algoritmos</i>	2,5/10	2	1,5	2,5	2
<i>Eficiencia</i>	2/10	1,5	1,5	2	2
<i>Precio licencia</i>	0,5/10	0	0,5	0,5	0,5
<i>Total</i>	10/10	8,5/10	7/10	8/10	7,5/10

Tras realizar el análisis establecido, se ha escogido MATLAB como lenguaje de programación para el grueso del proyecto. Dispone de sintaxis enfocada al procesado de señal, gracias al diseño orientado a la representación de datos y manejo de matrices, con múltiples funcionalidades. Estas características facilitan la primera fase del proyecto, en la que se precisa de un análisis de los datos médicos. Mediante las múltiples herramientas de representación de datos, resulta relativamente sencillo elaborar una GUI que permita analizar segmentos de señal ECG. Por ejemplo, para clasificar ritmos ORG como RP o AESP.

Algunas de las características clásicas de la señal ECG (sobre todo aquellas enfocadas a la detección de FV) fueron diseñadas hace varias décadas [41], de forma que el código disponible para su cálculo está redactado en C. Aunque es posible traducir las operaciones necesarias a

otro lenguaje, la capacidad de poder ejecutar código escrito en C dentro de un script de MATLAB facilita la integración de las múltiples características empleadas en este proyecto.

Por último, también se ha tenido en cuenta la experiencia del alumno con el propio lenguaje de programación, siendo MATLAB el único del que se disponía de conocimientos previos.

Sin embargo, para la fase de diseño de soluciones basadas en DL, se ha optado por emplear Python. Este lenguaje dispone de una amplia comunidad de desarrolladores e investigadores que fomentan el uso de librerías especializadas como TensorFlow y Keras, las cuales están provistas de fuerte soporte en forma de foros y ejemplos prácticos sencillos en numerosas páginas web. Dado que existen varias alternativas para la implementación de métodos DL en Python, en la sección 6.7 Librerías modelos Deep Learning se realiza el análisis de alternativas para escoger la librería mejor adaptada a las características del proyecto.

Además, resulta una buena oportunidad para el alumno de aprender un nuevo lenguaje, siendo este más sencillo de aprender que su alternativa para el desarrollo de modelos DL, el lenguaje R.

6.4 Algoritmos de clasificación

Se proponen únicamente 3 algoritmos, aquellos más usados en la literatura referida en el apartado 5. Análisis del estado del arte. Los algoritmos analizados son KNN, Random Forest y SVM.

6.4.1 KNN

KNN (K-Nearest Neighbors) [71] es uno de los algoritmos de clasificación más simples empleados en ML. KNN agrupa las distintas clases de acuerdo a sus características. Durante el entrenamiento, recuerda asociaciones entre las distintas características de cada clase. En la Figura 16 se observa un ejemplo sencillo. Si se trata de clasificar perros y gatos (clasificación binaria) de acuerdo a peso y altura (características), durante el entrenamiento el algoritmo aprende a asociar peso y estatura bajos, con gatos; viceversa para perros. Cuando trate de clasificar un nuevo dato de entrada como perro o gato, observa sus características. Seguido, busca un número K de puntos aprendidos (vecinos o neighbors), lo más cercanos posibles al dato de entrada. Ahora, clasifica el nuevo dato, como la clase con mayor número de vecinos cercanos.

Los parámetros ajustables se reducen al número de vecinos K, el peso de cada clase y el algoritmo empleado para el cálculo de 'cercanía' de los vecinos. Es por tanto un algoritmo, muy interpretable, rápido y sencillo, que permite realizar análisis preliminares en poco tiempo. Sin embargo, su rendimiento no es el más alto.

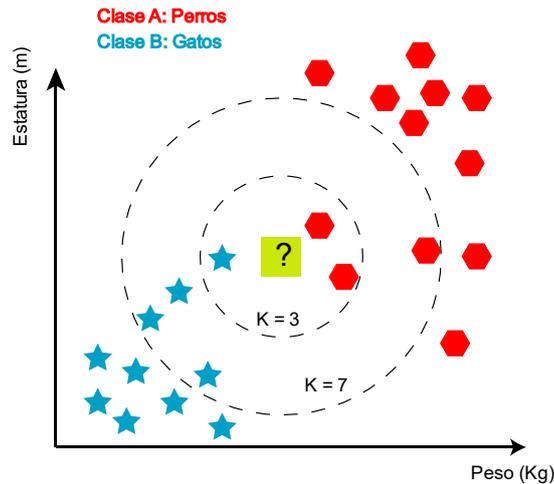


Figura 16. Ejemplo clasificador perros/gatos basado en KNN. Obsérvese la importancia del parámetro K en la decisión tomada por el algoritmo.

6.4.2 SVM

SVM (Support Vector Machine) [72] es un algoritmo muy potente, que ofrece buenos resultados una vez se han ajustado correctamente sus parámetros. Su funcionamiento se basa en el aprendizaje de hiperplanos que permitan separar las distintas clases disponibles en regiones. A la hora de clasificar un dato nuevo, simplemente se ubica en el hiperespacio (de acuerdo a sus características) y se observa a que región pertenece.

En la Figura 17 se observa un ejemplo, con los mismos datos de la Figura 16. Primero, se ubican en el hiperespacio los datos disponibles para entrenar el algoritmo, de acuerdo a sus características. Luego, se proponen hiperplanos aleatoriamente, y se selecciona el mejor, que permita separar las distintas clases (de nuevo, perros y gatos). Después, se ajusta dicho hiperplano para maximizar la distancia del mismo, respecto de unos vectores de soporte (calculados mediante puntos en el hiperespacio cercanos al hiperplano). De esta forma se procura no favorecer a ninguna de las clases, manteniendo un margen equidistante.

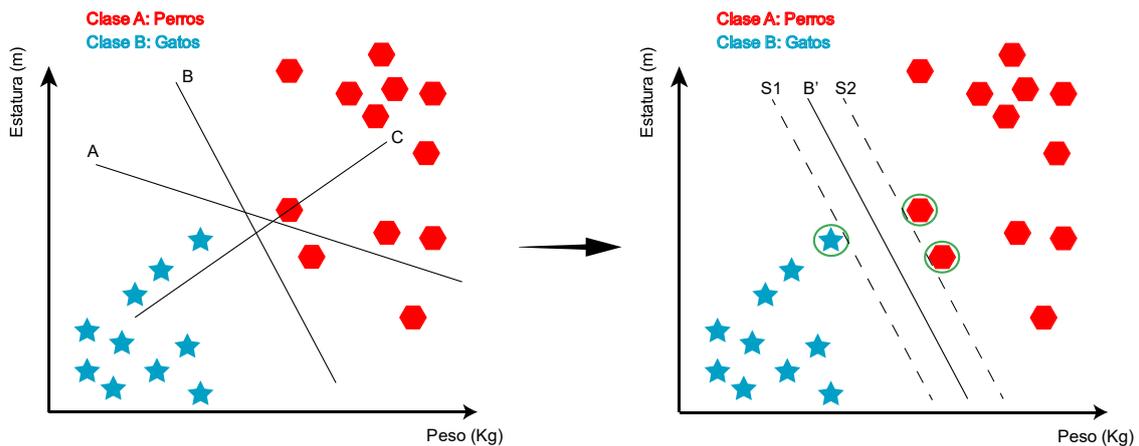


Figura 17. Selección de hiperplano óptimo por parte del algoritmo SVM. Vectores de soporte (S1 y S2) permiten calcular el margen óptimo del hiperplano B'.

Sin embargo, en la mayoría de los casos, no es posible realizar una separación lineal, como es el caso de la Figura 18. En este ejemplo se puede observar como un círculo separaría perfectamente ambas clases. Para resolver este problema, SVM emplea el método 'kernel trick', que permite transformar los datos, realizando operaciones entre ellos mismos. Esta herramienta

hace posible distinguir mediante un plano lineal puntos en el espacio que, con una dimensión menos, resultaría imposible. Existen multitud de kernels disponibles, en función de las operaciones realizadas sobre los datos de entrada.

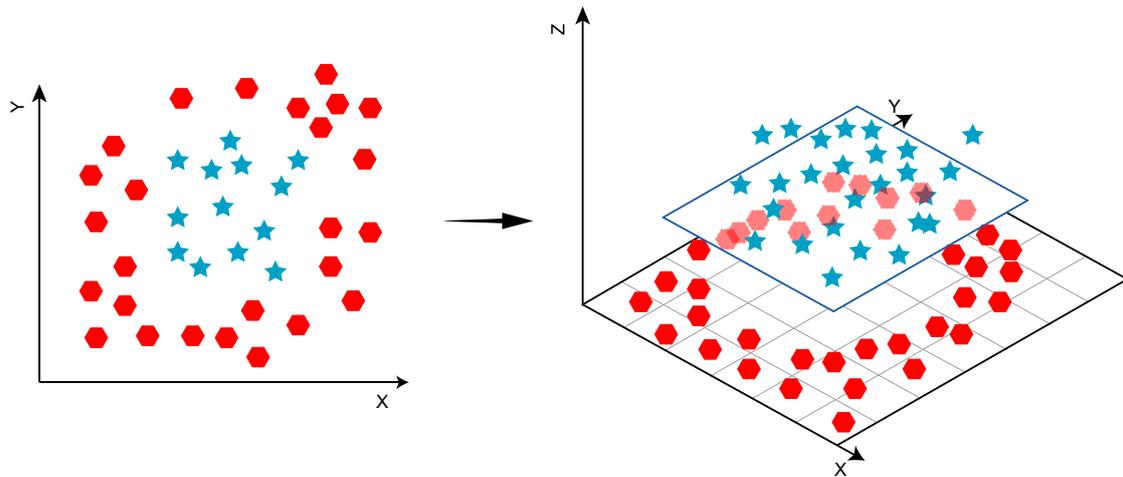


Figura 18. Ejemplo de 'kernel trick' aplicado sobre un espacio bidimensional (dos características X e Y), para convertirlo en tridimensional. La tercera dimensión se obtiene como $Z = X^2 + Y^2$. De esta forma es posible separar ambas clases mediante un plano lineal.

Gracias a esta técnica, SVM es un método capaz de encontrar hiperplanos que distingan datos de distintas clases de forma bastante efectiva. No obstante, debido a su funcionamiento, SVM es un algoritmo que solo consigue un buen rendimiento en caso de estar bien ajustado a los datos, ya que, de lo contrario, las regiones designadas por los hiperplanos establecidos durante el entrenamiento no separarían bien las distintas clases. Entre sus parámetros más importantes, se encuentran:

- **C:** Penalización por cada error cometido.
- **Kernel:** Tipo de kernel empleado, pudiendo ser lineal, polinomial, gaussiano o sigmoideal.
- **Degree, gamma y coef0:** Permiten ajustar los distintos tipos de kernel.

6.4.3 Random Forest

En el apartado 5.5 Algoritmo de clasificación, se adelanta la selección de Random Forest como el algoritmo de clasificación empleado en el proyecto. En ese mismo apartado se ofrece una explicación exhaustiva de su funcionamiento.

6.5 Criterios de selección: lenguaje de programación

Al igual que en 6.2 Criterios de selección: lenguaje de programación, se describen las propiedades a tener en cuenta por parte de los algoritmos, con el fin de escoger el que mejor se ajuste a las necesidades del proyecto. Se tiene en cuenta: velocidad de aprendizaje del algoritmo, interpretabilidad, flexibilidad del modelo y precisión del sistema.

Por último, se asigna un peso a cada característica para simplificar el análisis realizado.

6.5.1 Velocidad de aprendizaje

Debido al diseño del algoritmo, se producen divergencias en la velocidad de aprendizaje. En ocasiones, será necesario aumentar el número de etapas de entrenamiento necesarias para que el algoritmo converja en una solución estable. Por lo tanto, influye directamente en el número de ensayos realizables en un periodo de tiempo finito.

Importancia: 2/10

6.5.2 Interpretabilidad

Los algoritmos de clasificación emplean los datos de entrenamiento para aprender estructuras que identifiquen las distintas clases posibles. Sin embargo, cada algoritmo emplea distintos métodos de aprendizaje. Una vez finalizada la fase de entrenamiento, resulta útil poder interpretar las reglas y condiciones seguidas por el algoritmo durante el proceso de clasificación. De esta forma es posible mejorar el diseño de las características extraídas del ECG.

Importancia: 2/10

6.5.3 Flexibilidad del modelo

Flexibilidad del modelo, entendida como la cantidad de parámetros disponibles para ajustar el comportamiento del algoritmo. Habitualmente, aquellas alternativas con más opciones de configuración permiten desarrollar soluciones con mejores resultados.

Importancia: 3/10

6.5.4 Precisión

Capacidad del algoritmo para ofrecer una clasificación con el menor número de fallos. Ligado al apartado anterior, algoritmos simples tienden a ofrecer soluciones rápidas, pero con peores resultados que aquellos más complejos. Estos últimos disponen de mayor cantidad de parámetros, resultando en un modelo más ajustado a la problemática planteada.

Importancia: 3/10

6.6 Selección de alternativas: algoritmo de clasificación

A continuación, se ofrece la Tabla 3, la cual indica las ponderaciones asignadas a cada algoritmo.

Tabla 3. Comparativa algoritmos de clasificación

Criterio	Peso	KNN	SVM	Random Forest
Velocidad de aprendizaje	2/10	2	1	1,5
Interpretabilidad	2/10	1,5	1	2
Flexibilidad del modelo	3/10	1,5	3	2,5
Precisión	3/10	1	3	3
Total	10/10	6/10	8/10	9/10

Tal y como se ha introducido en 5.5 Algoritmo de clasificación, el algoritmo escogido es Random Forest. En la literatura se emplean habitualmente SVM y RF, obteniendo rendimientos similares. No obstante, debido al ámbito médico de los datos, siempre es conveniente poder explicar el funcionamiento del clasificador. Las decisiones del algoritmo influyen directamente en la calidad del tratamiento ofrecido al paciente, de forma que los profesionales demandan aclaraciones respecto a los métodos seguidos. En este ámbito, RF resulta más sencillo de entender, ya que es posible mostrar qué criterios, sobre qué características y la resolución tomada por cada árbol de decisión. SVM es sencillo de interpretar para pocas dimensiones (cada característica añade una nueva dimensión al hiperespacio), pero como se ha visto en la revisión del estado del arte, en este ámbito se emplean decenas de características. Es por ello que la interpretabilidad de SVM es reducida, al menos en el caso que ocupa.

SVM es el algoritmo más flexible de todos, puesto que contempla multitud de métodos para dividir el espacio de muestras con distintos hiperplanos. Es por ello que, si los datos disponen de baja varianza, se pueden dar resultados inmejorables. Sin embargo, es un algoritmo que tiende a sobreajustarse (overfitting) a los datos (Figura 19) si se realiza un entrenamiento demasiado agresivo. Por el contrario, RF resulta bastante robusto frente al overfitting. En el caso de que algunos árboles de decisión tomaran decisiones muy ajustadas a los datos de entrenamiento, puesto que la decisión final del algoritmo se toma como ponderación de los votos de todos los árboles, el resultado seguiría siendo satisfactorio. Cuanto mayor sea el número de árboles empleados, mayor es la resistencia al sobreajuste por parte de RF.

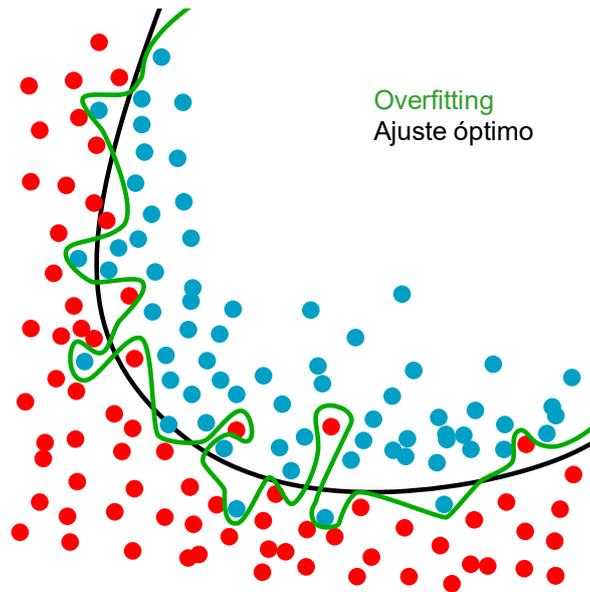


Figura 19. Ejemplo sobreajuste del algoritmo a los datos de entrenamiento (verde), frente al ajuste óptimo deseado (negro).

Para terminar, RF ofrece tiempos de entrenamientos menores respecto a SVM. Este último precisa de numerosas iteraciones para encontrar el hiperplano indicado, mientras que el tiempo de entrenamiento de RF es directamente proporcional al número de árboles empleado.

6.7 Librerías modelos Deep Learning

Python es uno de los lenguajes de programación más empleados actualmente. Gracias a ello, dispone de una amplia comunidad de investigadores, programadores y adeptos en general que fomentan actualizaciones y el mantenimiento de multitud de librerías. En concreto, para el desarrollo de soluciones de aprendizaje máquina, como DL existen librerías muy potentes, creadas por universidades o empresas privadas. Es el caso de TensorFlow, Theano, Keras y Pytorch.

6.7.1 TensorFlow

TensorFlow (TF) [73] es una librería de código abierto para aprendizaje automático, desarrollada por Google en 2015. Permite diseñar y entrenar redes neuronales para detectar patrones en datos y poder ofrecer un razonamiento por parte de la máquina, análogo a la de un ser humano. Actualmente se posiciona como la librería más popular, en gran medida debido al excelente soporte ofrecido por Google, en forma de actualizaciones periódicas y solución a errores.

Durante el desarrollo de soluciones DL, resulta interesante poder visualizar de forma gráfica la estructura de la red en forma de distintos filtros convolucionales, capas neuronales, etc. Es en

este ámbito donde TF ofrece los esquemas más intuitivos y esclarecedores respecto del resto de soluciones existentes, en particular mediante la herramienta TensorBoard.

6.7.2 Theano

Theano [74] es una librería de código abierto desarrollada por la universidad de Montreal en 2007, convirtiéndola en una de las librerías más maduras. Con el paso de los años, y en concreto debido a la aparición de nuevas librerías como TF, ha perdido popularidad, de forma que actualmente no se inician proyectos en base a esta tecnología. A día de hoy, la mayor parte de ecosistemas Theano activos son mantenidos ante la imposibilidad de migrar a otras soluciones.

Una de las causas de su declive se explica por lo reducido del grupo de personas dedicadas a su mantenimiento y desarrollo, formado tan solo por 50 personas. Deep Learning es un campo en auge, de forma que los avances se producen rápidamente, favoreciendo aquellas tecnologías que reciben soporte de grandes empresas. De lo contrario las actualizaciones se retrasan, ofreciendo peor servicio a los usuarios.

6.7.3 Keras

Keras [75] es una librería de código abierto, creada en 2015 con el objetivo de ofrecer una interfaz capaz de ejecutar distintas librerías (denominadas backend) como TF, Theano o PlaidML. Keras permite elaborar implementaciones rápidas, probando distintas soluciones, acelerando así el proceso de investigación. De esta forma, se posiciona como la solución ideal ante nuevos estudios en los que no se dispone de un punto de partida inicial.

Su diseño está enfocado a ofrecer un código legible y sencillo, que facilite la introducción de neófitos en la construcción de redes neuronales. Su capa de abstracción está, por ende, más limitada respecto a las librerías que emplea por debajo, de forma que una práctica habitual consiste en realizar una implementación inicial en Keras, y en fases más maduras del proyecto migrar a la librería concreta empleada. Al desprenderse de la capa de abstracción se consigue una solución más adaptada al problema, mejorando tiempos de ejecución, calidad de los modelos predictivos, etc.

6.7.4 PyTorch

PyTorch [76] es una librería de código abierto desarrollada por Facebook en 2018. Surge en base a la librería Torch, la cual emplea lenguaje Lua. Debido al escaso número de usuarios de dicho lenguaje, se decidió realizar una migración a lenguaje Python. Teniendo en cuenta lo joven que es esta tecnología, PyTorch dispone de un elevado número de usuarios, el cual no para de crecer, pero sigue siendo inferior a TF, el cual dispone de mayor cantidad de recursos, foros y tutoriales.

Su código resulta fácil de aprender, debido a que su sintaxis es semejante a la de Python, muy intuitiva. Sin embargo, no dispone de una herramienta nativa de visualización de estructuras como TensorBoard.

6.8 Criterios de selección: librerías modelos Deep Learning

Las características a tener en cuenta de las distintas librerías analizadas son: capacidad para visualizar la estructura de la red neuronal diseñada, soporte ofrecido por los desarrolladores, así como por parte de la comunidad de usuarios y la interpretabilidad del código.

6.8.1 Visualización de la estructura de red

Durante el diseño de la solución (la red neuronal), resulta de gran ayuda disponer de una herramienta de visualización que identifique el número de nodos, relaciones entre distintas capas, parámetros computados por la red... así como otras propiedades. De lo contrario es

preciso disponer de elevada experiencia en el diseño de redes neuronales, para poder reconocer los puntos de fallo de la arquitectura.

Importancia: 2,5/10

6.8.2 Soporte

El punto de mayor importancia. Puesto que cada solución es única, es probable que surjan errores. Cuanto mayor sea la comunidad de usuarios que emplean una tecnología, existirán mayor cantidad de foros y páginas web dedicadas a la resolución y compartición de conocimiento, facilitando el proceso de programación. A su vez, es indispensable que los desarrolladores de la librería ofrezcan de forma periódica actualizaciones que propongan nuevas aptitudes y funciones. También es importante que resuelvan los errores que sean detectados tras cada actualización.

Importancia: 4/10

6.8.3 Interpretabilidad del código

Debido a que el alumno no dispone de experiencia previa sobre ninguna de las librerías analizadas, un código legible resulta de ayuda. La curva de aprendizaje será menos acusada para aquellas tecnologías más sencillas.

Importancia: 3,5/10

6.9 Selección de alternativas: librerías modelos Deep Learning

La Tabla 4 muestra las puntuaciones asignadas a las distintas librerías de DL analizadas.

Tabla 4. Comparativa librerías Deep Learning

<i>Criterio</i>	<i>Peso</i>	<i>TF</i>	<i>Theano</i>	<i>Keras</i>	<i>PyTorch</i>
<i>Visualización de la estructura de red</i>	2,5/10	2,5	1	2	1,5
<i>Soporte</i>	4/10	4	2,5	3,5	3
<i>Interpretabilidad del código</i>	3,5/10	2	2,5	3,5	3
<i>Total</i>	10/10	8,5/10	6/10	9/10	7,5/10

La librería escogida es Keras, empleada como capa de abstracción, habiendo escogido TF como backend. Las funciones de Keras se pueden configurar para llamar a métodos de TF y, por tanto, emplear sus herramientas de visualización, como TensorBoard. Sin embargo, Keras dispone de una sintaxis sencilla que facilita la introducción del alumno en el campo de las soluciones basadas en Deep Learning.

El soporte es también bastante notable, respecto de TF, puesto que Keras ha sido diseñada en parte también por Google, y ofrece actualizaciones con la misma atención de TF. La comunidad de usuarios es también bastante grande, aunque la página oficial de Keras dispone de todo el conocimiento necesario para implementar redes sencillas.

7. Análisis de riesgos

El análisis de riesgos de un proyecto trata de aumentar la probabilidad e impacto de los eventos positivos, y disminuir la probabilidad e impacto de los negativos. El éxito del proyecto depende directamente de los eventos surgidos durante la ejecución del mismo. Una pronta gestión de riesgos permite analizar e identificar estos últimos, procurando ofrecer el mejor desempeño posible, mediante un enfoque consciente y proactivo.

El cometido de esta sección es, por tanto, la identificación, predicción, análisis, gestión y control de los posibles riesgos que pueden surgir durante el desarrollo de este proyecto. Es decir, se procura evitar las amenazas negativas y amortiguar los daños ocasionados, así como propiciar los sucesos positivos que tengan el potencial de mejorar el trabajo.

La propia definición de riesgo conlleva analizar dos variables fundamentales: por un lado, la probabilidad de los riesgos, y, por otro lado, el impacto que este riesgo puede tener en nuestro proyecto, entendido como las pérdidas ocasionadas. Para llevar a cabo este estudio, primero se realiza un análisis cualitativo, seguido de un plan de respuesta de cara al suceso, también denominado, plan de contingencia.

Seguidamente, se muestran Tabla 5 y Tabla 6, las cuales resumen los criterios a tener en cuenta para medir las probabilidades e impactos:

Tabla 5. Descripción de criterios de medición de probabilidad de riesgos.

<i>Probabilidad</i>	<i>Peso</i>	<i>Descripción</i>
<i>Baja</i>	0,2	Riesgo a tener en cuenta, pero su frecuencia de aparición es baja.
<i>Media</i>	0,5	Riesgos producidos eventualmente. Es necesario disponer de las medidas oportunas para paliar sus efectos perjudiciales.
<i>Alta</i>	0,8	Probabilidad alta de que se produzca el suceso que ocasiona el riesgo.

Tabla 6. Descripción de criterios de medición del impacto ocasionado por los riesgos.

<i>Impacto</i>	<i>Peso</i>	<i>Descripción</i>
<i>Bajo</i>	0,1	Aunque se produzca el riesgo, no es necesario tomar medidas al respecto, puesto que el daño producido en el proyecto es mínimo.
<i>Medio</i>	0,5	Los efectos producidos por el riesgo podrían llegar a afectar ligeramente a los plazos y calidad del proyecto.
<i>Alto</i>	0,9	Si se produce el riesgo, el proyecto se vería gravemente dañado. Si no son evitables se debe procurar una solución efectiva.

A continuación, se presentan los riesgos del proyecto y los eventos que pueden verse afectados, incluida la probabilidad de ocurrencia:

A. Riesgos de gastos

Entre los diferentes riesgos que pueden surgir dentro del ámbito de riesgos de gastos, los costes de desarrollo pueden ser imprevisibles y aumentar durante el ejercicio del proyecto, de forma que el coste planteado durante la fase inicial no fuese apropiado. Es el caso de subidas de precios en equipamiento, adjudicación de actividades adicionales debido a modificaciones de última hora en el plan de tareas, cambio de los materiales empleados debido a fallos en el equipamiento original, etc.

Con el fin de evitar estos incidentes y lograr una estimación razonable de los costes del proyecto, se evalúa exhaustivamente cada una de las tareas realizadas. Además, se dispone de un 5% del presupuesto original reservado para posibles imprevistos, dispuesto a paliar ligeras variaciones en los costes establecidos. Sin embargo, mencionar que el desarrollo del proyecto depende de herramientas software mayoritariamente, de forma que las posibles variaciones de precio son menos susceptibles de ocurrir frente a un proyecto con elevado coste material. En conclusión, el caso más probable de aumento del coste del proyecto vendría dado por un aumento en las horas contratadas de los recursos humanos, pero como se ha descrito, se dispone de un fondo para imprevistos pensado para estos casos.

Probabilidad: Baja	Impacto: Bajo
---------------------------	----------------------

B. Riesgos de horario

Durante el proyecto, las fechas especificadas en el plan de trabajo pueden verse afectadas por distintos riesgos como problemas técnicos durante el desarrollo y resolución de ciertas tareas, asignación de recursos no apropiados o falta de asignación de recursos al no haber sido considerados. Si bien el plan de trabajo se realiza de antemano, los problemas externos y los problemas técnicos pueden ocasionar demoras inevitables.

La probabilidad de que sucedan estos eventos es muy alta, y además es muy difícil lograr evitarlos si se producen de forma súbita. Por tanto, el mejor plan de contingencia se basa en reducir el impacto y su probabilidad, ajustando correctamente la duración de cada asignación.

Por todo ello, se definen paquetes de trabajo para facilitar la organización del proyecto. En este caso, dado el reducido número de integrantes del equipo del proyecto, los retrasos en una tarea comprometen directamente la ejecución del proyecto en plazo. Al no disponer de una plantilla extensa, no es posible recuperar los tiempos perdidos con recursos sin tareas asignadas. El riesgo de no cumplir las fechas establecidas se posiciona de esta manera como uno de los más importantes a tener en cuenta.

Probabilidad: Alta	Impacto: Medio
---------------------------	-----------------------

C. Riesgos técnicos

Los riesgos técnicos están conformados por aquellos que surgen como fruto de la posible incapacidad del alumno para resolver los problemas planteados. Aunque se identifique el problema y se plantee una solución, la implementación de esta última puede suponer un reto mayor al esperado. Puede ser el caso del procesado de los datos médicos, la extracción de las características de la señal ECG, etc.

En resumen, todos los problemas técnicos que pueden surgir durante el desarrollo de la solución planteada, retrasan directamente el proyecto si no se logran solventar en plazo. Debido al carácter del trabajo propuesto, son pocas las tareas que se pueden realizar paralelamente, y no se puede iniciar una fase sin resolver aquella previa.

En este apartado, los tutores a cargo del alumno realizan un papel fundamental, ya que su experiencia facilita en caso de necesidad, la resolución de los problemas técnicos por parte del alumno, asegurando de esta forma la consecución del proyecto. Sin embargo, debido a su carácter didáctico, es una solución de último recurso, priorizando el desarrollo de las capacidades del docente durante la realización del trabajo.

Probabilidad: Media Impacto: Alto

D. Riesgos tecnológicos

El desconocimiento en el uso de tecnologías necesarias para el desarrollo del proyecto o las dificultades surgidas para comprender la complejidad de las mismas, el uso de herramientas que no son adecuadas para las funcionalidades requeridas... en resumen, una incorrecta elección de alternativas, hace peligrar la integridad del proyecto.

A su vez, es probable que surjan problemas software como incompatibilidad entre versiones de librerías, paquetes o sistemas operativos. Son problemas indirectos que, de ser descuidados, tienen el potencial de retrasar el trabajo semanas, ya que los programas informáticos son las herramientas empleadas para implementar la solución.

Por último, las simulaciones y fases de entrenamiento de los algoritmos, se realizan en máquinas compartidas con otros usuarios (de otros proyectos ajenos). Los recursos de estos servidores no estarán, por tanto, siempre disponibles, ocasionando retrasos variables e incontrolables. Se debe tener en cuenta a su vez, el periodo de configuración de estas máquinas de acuerdo a las especificaciones de la investigación.

Probabilidad: Media Impacto: Medio

E. Riesgos de operación

Las siguientes situaciones conforman los posibles riesgos de operación: la falta de liderazgo en el equipo, la falta de comunicación, la motivación que debe tener el equipo, los problemas de monitorización para comprobar el correcto avance del proyecto y problemas durante la implementación y desarrollo.

Los problemas de liderazgo quedan prácticamente descartados, puesto que ambos tutores poseen un amplio historial de trabajos de fin de grado y master completados de forma satisfactoria. En este sentido, disponen de la destreza necesaria para organizar y asignar las tareas requeridas al alumno. La comunicación y motivación de los integrantes se asegura por lo reducido del tamaño del grupo de trabajo, que permite un trato más cercano.

Centrándose en este proyecto, en el caso de eventos que traigan problemas en su desarrollo, está claro que el impacto en el proyecto sería alto. Sin embargo, es posible manejar estos problemas de una manera más cómoda y tranquila, especialmente porque el equipo de trabajo es muy pequeño. Es por ello que la probabilidad es baja.

Probabilidad: Baja Impacto: Alto

F. Riesgos de mercado y competencia

El objetivo de este proyecto consiste en desarrollar un anotador automático de ritmos cardiacos de 5 ritmos. En el análisis del estado del arte no se han encontrado apenas trabajos con este propósito, a excepción de la línea de investigación de *Rad et al.* [56], cuyo departamento ha colaborado con el Grupo de investigación BioRes de la UPV/EHU. Este hecho, sumado a las innovaciones ofrecidas en este trabajo procuran una probabilidad baja de que otro grupo de

investigación publique resultados similares. Además, aunque se produjese esta situación, los avances realizados sirven de base para futuras investigaciones por parte de BioRes, por lo que el impacto de este riesgo es moderado.

Probabilidad: Baja **Impacto:** Medio

G. Riesgos de cambio de leyes

Se debe de tener en cuenta posibles cambios en las leyes, normativas, regulaciones y estándares a cumplir, que puedan afectar a este proyecto. No obstante, las regulaciones en torno al campo de la medicina son estables en el tiempo y solo cambian en base a evidencias clínicas muy concluyente. Por ello la probabilidad de cambio a corto plazo es muy baja. De producirse, sin embargo, conformaría un impacto importante, ya que podría forzar a modificar todo el trabajo, de cara a cumplir la nueva normativa vigente.

Probabilidad: Baja **Impacto:** Alto

A continuación, se muestra en la Figura 20 una matriz que presenta la relación probabilidad-impacto de los riesgos anteriormente descritos.

		Impacto		
		Bajo 0,1	Medio 0,5	Alto 0,9
Probabilidad	Baja 0,2	A Bajo (0,02)	F Bajo (0,1)	E,G Medio (0,18)
	Media 0,5	Bajo (0,05)	D Medio (0,25)	C Alto (0,45)
	Alta 0,8	Bajo (0,08)	B Alto (0,4)	Extremo (0,72)

Figura 20. Matriz probabilidad/impacto de los riesgos presentes en el proyecto.

La matriz permite analizar a visualmente la probabilidad y riesgo combinado de cada riesgo, computando el valor final como el producto de los pesos de ambas características.

Los riesgos más importantes son los de horario y los técnicos. Al final, la mayoría de proyectos que dispongan de plazo van a tener en cuenta los horarios como factor de riesgo. Cuanto mayor sea la duración del proyecto, mayor es la probabilidad de que se produzcan alteraciones en las jornadas estipuladas. Cuando el plazo de entrega del ejercicio del proyecto es inamovible, el impacto se agrava, como es el caso de un Trabajo de Fin de Master, de forma que se consolida como una de las principales amenazas. Por otra parte, al ser un proyecto de investigación, se van a producir una serie de retos técnicos de notable dificultad, en las que se pondrá a prueba

las capacidades del alumno para resolverlos, dentro de los periodos de tiempo establecidos inicialmente. Su probabilidad es menor respecto de los riesgos asociados al cumplimiento de las fechas, puesto que los tutores facilitan ayuda al alumno en caso de necesidad. Sin embargo, una solución pobre puede desmerecer la calidad final del proyecto, de forma que el impacto del riesgo es más elevado.

En definitiva, el control de riesgos fomenta el éxito del proyecto al identificar los posibles retrasos y fallos a ocurrir durante el trabajo propuesto. Durante la realización de las tareas, se deberá tener presente los riesgos de mayor importancia para tratar de evitarlos, y de suceder actuar en consecuencia.

8. Descripción de la solución propuesta. Diseño (de alto nivel)

En este apartado se describen los pasos realizados durante el desarrollo del proyecto. Con ánimo de facilitar la comprensión del trabajo, se ha dividido este apartado en cuatro fases, de acuerdo a los puntos indicados en la sección 3. Objetivos y alcance del proyecto. Las distintas fases consisten en preparar una base de datos auditada y anotada por expertos médicos, diseñar un clasificador de 5 ritmos cardiacos empleando únicamente la señal ECG, mejorar el rendimiento del anterior clasificador partiendo de un enfoque multimodal en el que se añade la señal de impedancia transtorácica del paciente y por último desarrollar clasificadores a partir de técnicas de Deep Learning, con el objetivo de superar el rendimiento de las soluciones diseñadas mediante métodos clásicos.

8.1 Base de datos auditada y anotada

El proyecto parte de una colección de 855 archivos electrónicos de pacientes PCREH recolectados mediante DEAs, los cuales fueron transformados a formato MATLAB, con el fin de realizar una anotación entre distintos ritmos cardiacos. La colección de archivos generó 4409 intervalos ECG de duración variable, los cuales fueron etiquetados bajo cuatro categorías: AS, ORG, TV y FV

8.1.1 Análisis de los datos

Primero, se analizan los campos disponibles en los intervalos ECG, con el objetivo de que el alumno se familiarice con su estructura. En la Tabla 7 se describen las características de los datos.

Tabla 7. Descripción de los campos contenidos en la base de datos original

<i>Campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
reg_name	string	Identificativo del paciente al que pertenece el registro
cut_name	string	Identificativo del segmento analizado para un registro concreto
reg_date	string	Fecha en la que se grabó el registro
analysis	struct	Conjunto de datos empleados por los expertos clínicos durante la anotación de los segmentos de los registros
r_type	int	Tipo de arritmia presente en el segmento
s_ecg	array<double>	Señal ECG digitalizada
s_imp	array<double>	Señal de impedancia transtorácica digitalizada

A su vez, el campo 'analysis', contiene los datos empleados por los expertos clínicos a la hora de clasificar los intervalos ECG (Tabla 8). Resulta una información esencial a la hora de resolver ambigüedades tales como ritmos transitorios o tiempos de inicio o final de segmento mal grabados (por ejemplo, aquellos casos en los que se interrumpe el complejo QRS).

Tabla 8. Descripción de los campos contenidos en 'analysis'.

<i>Campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
number	int	Número de segmento dentro de un mismo registro
int_aed	array<double>	Tiempo de inicio y fin del intervalo analizado por los expertos clínicos
int_cut	array<double>	Tiempo de inicio y fin del segmento extraído del registro
clin_diag	array<int>	Diagnóstico dado por cada experto clínico

Esta es la información original disponible de los 4409 intervalos ECG recolectados por los DEAs.

8.1.2 Anotación de intervalos RP y AESP

Inicialmente la base de datos dispone únicamente de cuatro clases. Los ritmos cardiacos AESP y RP se encuentran anotados como ORG, pero no se distingue la presencia de actividad mecánica de miocardio. Con el objetivo de poder discriminar entre ambas clases, se hace uso de una interfaz gráfica diseñada para la visualización de los segmentos de análisis [18], introducida en la Figura 5.

Durante el análisis, la característica principal empleada para discriminar entre AESP y RP es la impedancia transtorácica. Los DEAs miden la impedancia transtorácica con el objetivo de determinar si los parches se encuentran bien conectados al paciente o si se encuentra en circuito abierto y bloquear por tanto la descarga. El oxígeno presente en la sangre actúa como dieléctrico, de forma que durante las palpitations se intercalan momentos de sangre oxigenada y poco oxigenada. Estas diferencias producen variaciones a lo largo del tiempo en la señal de impedancia transtorácica, coincidiendo las fluctuaciones de impedancia con los complejos QRS del corazón, instante en el que se produce la contracción ventricular.

Por tanto, se clasifican como RP aquellos intervalos con fluctuaciones de ritmo semejante a la frecuencia cardiaca del paciente, con valores mínimos de pico de 50 mΩ [10]. En caso contrario la señal muy probablemente se identifique como AESP, especialmente si proviene o desemboca en AS, la cual se corresponde como un ECG plano. Adicionalmente se pone especial atención a la morfología de los complejos QRS, los cuales suelen presentar mayor duración en el tiempo y forma aberrante para intervalos AESP, respecto de complejos estrechos y mejor definidos en casos RP. Por último, aquellos casos de duda, ocasionados por morfologías extrañas o transiciones entre estados, son catalogados como indefinidos y por tanto excluidos de la base de datos al aportar más ruido que información relevante de cara a entrenar el sistema.

A continuación se muestra en Figura 21 y Figura 22 distintos ejemplos de intervalos ORG, con el objetivo de esclarecer las pautas seguidas durante el proceso de anotación.

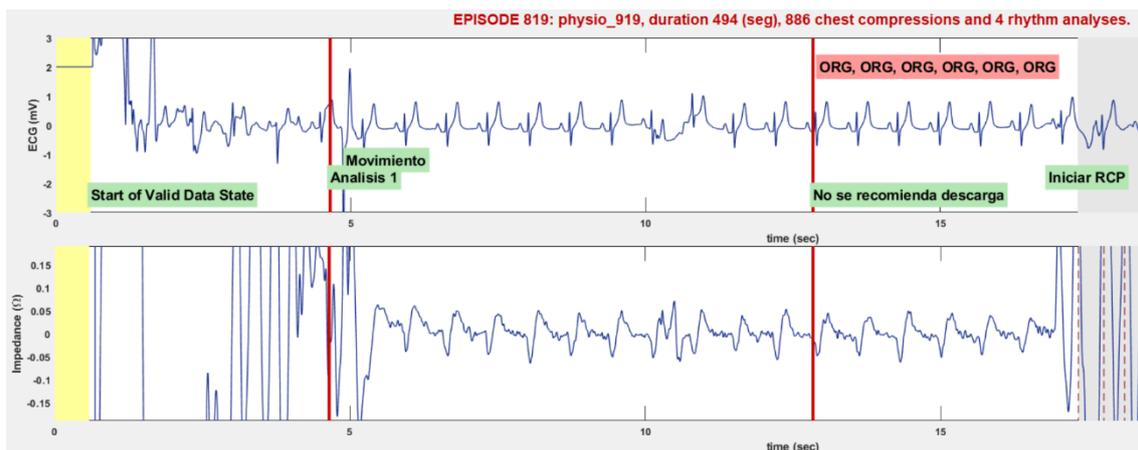


Figura 21. Ejemplo caso RP. Variaciones de impedancia en fase con complejos QRS.

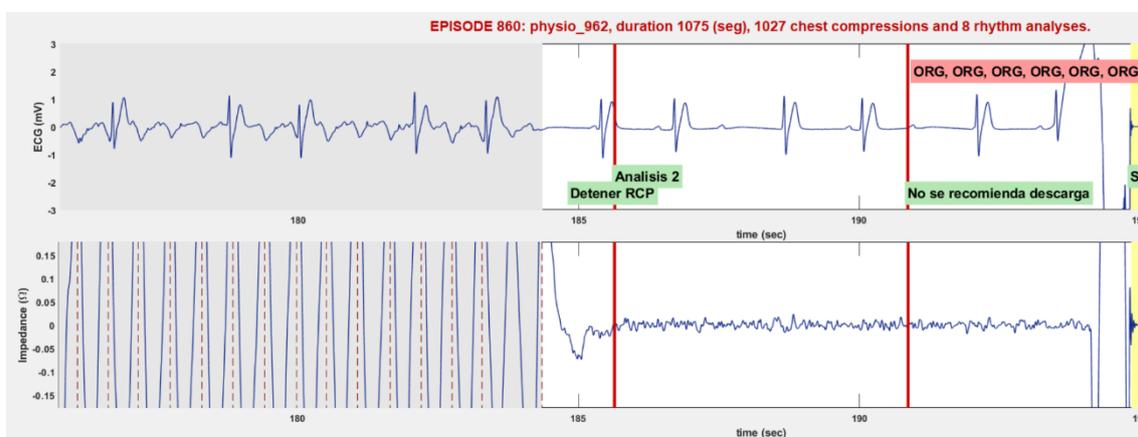


Figura 22. Ejemplo caso AESP. Sin variaciones de impedancia.

En total se analizan 1305 intervalos ECG, de los cuales se catalogan 1004 como AESP, 292 como RP y 9 como indefinidos.

Adicionalmente, se analizaron el resto de intervalos de otras clases, con el fin de excluir aquellas grabaciones que capturen pulsos transitorios intermedios o artefactados (durante la RCP la grabación no refleja el pulso del paciente), resultando en 4214 intervalos ECG disponibles para este estudio.

8.1.3 Preprocesado de los datos

Una vez se ha definido a que ritmo cardiaco pertenece cada intervalo ECG, se procede a realizar un preprocesado básico de las señales.

Primero, se genera un nuevo campo '*patID*' que identifique de forma unívoca, mediante una nomenclatura homogénea, el paciente al que pertenece cada intervalo ECG. Para ello, se asigna el mismo número identificativo a todos los intervalos con mismo valor en el campo '*reg_name*'.

Después, se aplica un filtrado frecuencial. Aunque es costumbre que los DEA analicen la señal ECG del paciente en el intervalo de 0,5 a 30 Hz, por haber empleado distintos equipos durante la recolección de los datos, es conveniente volver a filtrar la señal.

Puesto que se dispone de la señal ECG completa discretizada (muestras finitas), es posible aplicar un filtro de respuesta infinita (IIR) pero sin distorsión de fase, empleando un filtrado bidireccional. Los filtros IIR son computacionalmente más eficientes, y ofrecen mejor respuesta

en frecuencia para la amplitud. Por ello, se emplean dos filtros Chebyshev tipo 1 [77], de cuarto orden (cuatro coeficientes), con un rizado en la banda de paso de 1 dB. El primero es un filtro paso alto de frecuencia de corte 0,5 Hz mientras que el segundo es un filtro paso bajo de frecuencia de corte 30 Hz, la respuesta frecuencial de amplitud del conjunto se muestra en la Figura 23.

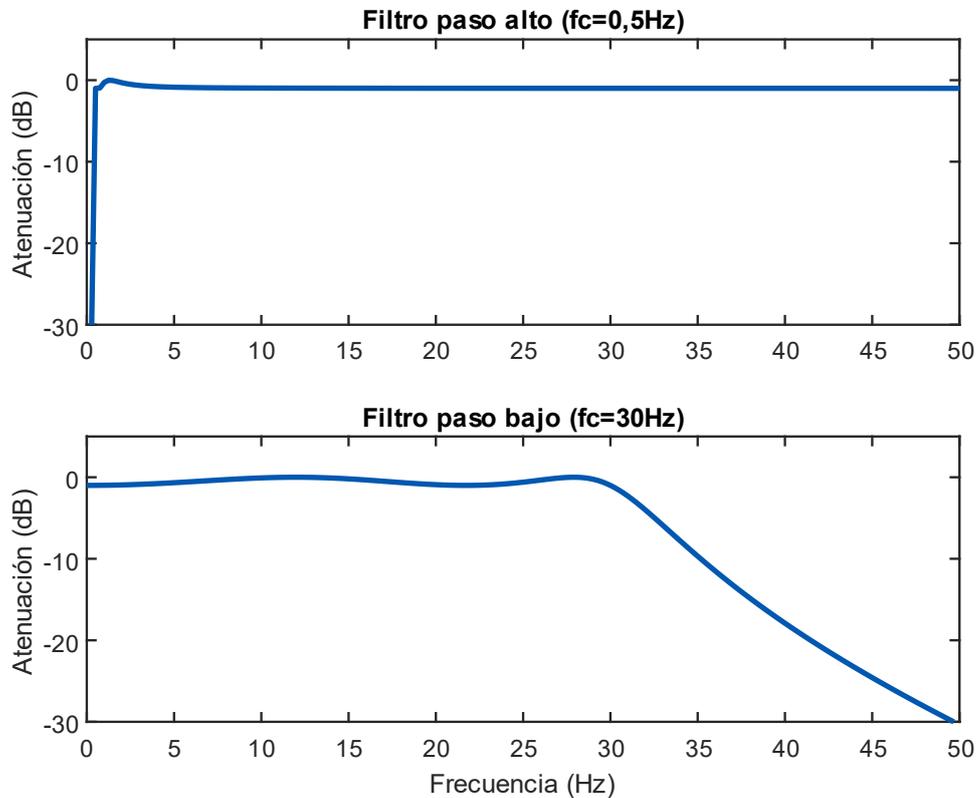


Figura 23. Respuesta frecuencial filtros paso alto y paso bajo.

Una vez completado el preprocesado de la señal ECG, el paso restante para disponer de un formato homogéneo de los datos, consiste en igualar la duración de los intervalos.

8.1.4 Segmentación de los datos

Los intervalos ECG ofrecen una duración heterogénea, de forma que, con el fin de lograr igualarlos, para cada registro se toman dos segmentos de 4 segundos. De esta forma se obtiene a su vez una mayor cantidad de datos con los que entrenar los algoritmos de clasificación. Como criterio de selección, se parte del punto central indicado en el campo *'int_aed'*. Entonces se toman los segmentos de 4 segundos anterior y posterior.

En función de los tiempos indicados en cada intervalo ECG, existen cuatro casuísticas posibles, ejemplificadas en la Figura 24:

- Existen muestras de tiempo suficientes como para tomar los segmentos propuestos.
- El punto central indicado en *'int_aed'* se encuentra a menos de cuatro segundos del inicio del intervalo. Entonces, se reubica el punto central a una distancia de cuatro segundos respecto del inicio del intervalo.
- El punto central de *'int_aed'* se encuentra a menos de cuatro segundos del final del segmento. Entonces, se reubica el punto central a una distancia de cuatro segundos respecto del final del intervalo.

- El intervalo contiene menos de 8 segundos de grabación, de forma que solo se obtiene un único segmento. Dicho segmento será el del instante anterior o posterior dependiendo del caso.

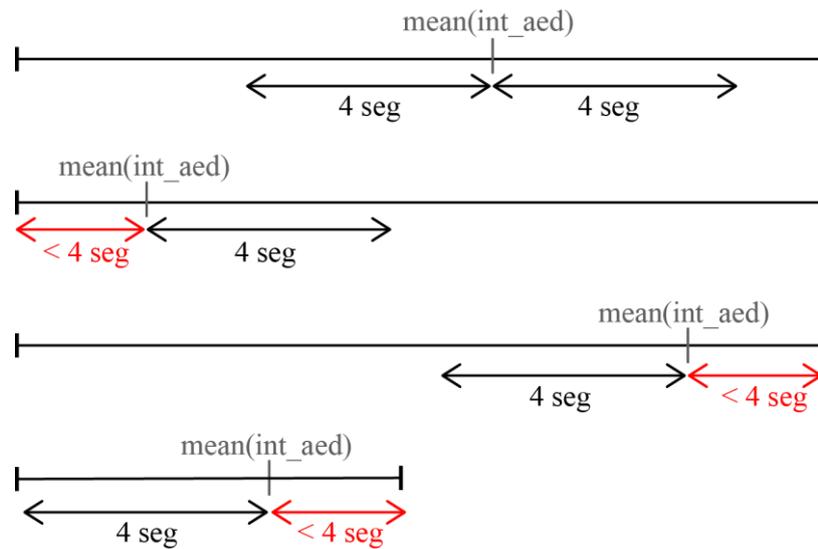


Figura 24. Casos posibles durante la extracción de tramos de señal.

8.1.5 Reserva de datos para test

Con el objetivo de disponer de datos con los que realizar un test de rendimiento del desempeño ofrecido por el clasificador de ritmos cardiacos, se reserva un 30% de los segmentos obtenidos mediante el proceso definido en el apartado anterior.

Debido a la escasa disponibilidad de intervalos ECG etiquetados como TV (22 intervalos ECG), todos ellos se reservan para test, dejando el conjunto de datos de entrenamiento sin representación de esta arritmia cardiaca. Puesto que el algoritmo necesita de datos para aprender sus características y con ello poder identificarlos, se añaden 195 intervalos TV, grabados sobre 21 pacientes distintos. Los datos se obtienen de un repositorio público denominado Physionet [78].

Por último, se extraen segmentos de 4 segundos de dichos intervalos TV. De esta forma, se disponen de 5694 segmentos de señal ECG con los que entrenar el algoritmo de clasificación, siendo 390 etiquetados como TV.

8.1.6 Particiones de datos

Siguiendo los principios explicados en 5.2 Validación cruzada, para realizar un entrenamiento eficiente del algoritmo, es necesario generar particiones de datos, de forma que se disponga de un conjunto de segmentos de señal ECG para validar cada etapa de entrenamiento. Se escoge reservar un 10% de los datos de entrenamiento para validación, tratando de mantener un compromiso entre aumentar, en la medida de lo posible, los datos disponibles para entrenar el algoritmo, sin descuidar la cantidad necesaria de datos para realizar una validación que refleje el comportamiento real del clasificador. Por tanto, se emplea una validación cruzada de $K = 10$ particiones (Figura 7), de forma que se realiza el entrenamiento con 9 particiones y se valida con la restante. El proceso se repite 10 veces, una por cada combinación posible.

Durante el proceso de creación de las particiones de datos, dos criterios fueron seguidos: los datos de un mismo paciente solo pueden hallarse en una única partición y cada partición debe representar una proporción mínima de cada tipo de ritmo cardiaco (cuasi-estratificado).

Los datos de un mismo paciente solo pueden hallarse en una misma partición, puesto que los ritmos almacenados de dicho paciente guardan una alta correlación entre ellos. Por tanto, para validar con datos independientes respecto de los datos empleados para entrenar, se debe mantener esta característica. Se impone como condición, que aquellos segmentos con mismo valor en el campo 'patID' solo puedan encontrarse en una misma partición. De esta forma, durante el proceso de validación, se comprueba el rendimiento del clasificador con datos de pacientes distintos a los empleados para entrenar.

Por otra parte, aunque una práctica común consiste en mantener al menos un 75% de representación de cada clase para entrenamiento, tras reservar una partición para validación, no siempre es posible mantener dicha condición para todas las clases. En este estudio, los datos se encuentran fuertemente desbalanceados, como se indica en la Tabla 9.

Tabla 9. Representación datos entrenamiento.

<i>Clase</i>	<i>Segmentos</i>	<i>Proporción (%)</i>
AS	3037	53,3
RP	388	6,8
AESP	1295	22,7
FV	584	10,3
TV	390	6,9

Se observa que los segmentos de tipo RP y TV representan una proporción muy baja, respecto del resto de tipos de ritmos cardiacos. Sumado a la dificultad añadida que supone respetar los pacientes bajo la misma partición, se rebaja la proporción mínima impuesta para las clases 2 y 5 a 60% y 40% respectivamente. El resto de clases mantienen una representación mínima del 75%.

Manteniendo estas condiciones, se realizan 100 conjuntos de validación cruzada distintos e incorrelados, de forma que se puedan promediar las métricas obtenidas para cada uno de ellos. Esto permite disminuir la varianza de los resultados obtenidos durante entrenamiento.

La búsqueda de particiones de datos se realiza en distintas fases. Primero se agrupan todos los segmentos de un mismo paciente, obteniendo un grupo por cada paciente distinto. Luego, se combinan dichos grupos aleatoriamente y se comprueba que las 10 particiones logren la representación mínima asignada. Si cumplen todas las condiciones, se almacena una lista en la que se identifica para cada segmento de señal ECG, a qué partición pertenece. El proceso se repite, cambiando la semilla de aleatoriedad, hasta obtener todas las particiones requeridas.

8.2 Clasificador 5 clases mediante señal ECG

En este apartado se diseña el clasificador de 5 clases, empleando únicamente la señal ECG. Para ello, primero se extraen características de la señal ECG, las cuales serán empleadas por el algoritmo para identificar las distintas clases. Luego, se realizan las primeras pruebas en las que se busca la mejor configuración de hiperparámetros para RF. Una vez escogida la combinación óptima, se entrena el algoritmo con todos los datos disponibles y se comprueba su desempeño contra los datos reservados para test. Después, se ordenan las características empleadas, de acuerdo a su capacidad para identificar las distintas clases. De esta forma se puede estudiar el comportamiento del algoritmo el emplear únicamente un conjunto finito de las mejores características. Por último, se trata de optimizar los valores escogidos en las frecuencias de corte de los filtros aplicados en el preprocesado de la señal ECG. También se busca la función *mother wavelet* que ofrezca el mejor desempeño posible.

8.2.1 Extracción de características de la señal ECG

El primer paso en el diseño del clasificador de 5 clases, consiste en extraer características de los segmentos de 4 segundos de señal ECG. En Machine Learning, se obtienen mejores resultados al entrenar los algoritmos en base a propiedades de los datos a clasificar, respecto de entrenar directamente con los datos. Por ello, se realiza un estudio sobre el estado del arte, con el objetivo de escoger características de la señal ECG, las cuales permitan elaborar un modelo que produzca el mejor clasificador posible.

Sobre los datos disponibles, previamente a la extracción de características, se aplica una descomposición sub-banda a la señal ECG, mediante el Stationary Wavelet Transform (SWT) [79]. Esta técnica proyecta la señal de interés $s(n)$, sobre una colección de funciones, generadas a partir de una función básica, denominada *mother wavelet*. En esencia, para cada nivel de descomposición SWT se filtra la señal pasabajo/pasoalto mediante un par de filtros en cuadratura $g_i(n)/h_i(n)$ que dividen el espectro en dos bandas simétricas (sub-bandas). Se obtienen así los coeficientes de aproximación $a_i(n)$ (pasabajo) y detalle $d_i(n)$ (pasoalto). Repitiendo el proceso para varios niveles se obtiene la descomposición en sub-bandas de la señal. Los filtros para cada nivel de descomposición se generan a partir de la *mother wavelet*.

En una aplicación típica al procesamiento de ECG en PCREH, *Rad et al.* [56] e *Iraia et al.* [79], proponen una descomposición sub-banda que permita el análisis del ECG en la banda típica en desfibriladores, de 0,5-30Hz. Para una frecuencia de muestreo de $f_s = 250\text{Hz}$, esto se corresponde con 8 niveles de descomposición, de forma que el análisis se realiza con los coeficientes de detalle $d_3 - d_8$, es decir igualando d_1 , d_2 y a_0 . En la Figura 25 se detalla el proceso de extracción de las distintas sub-bandas [79].

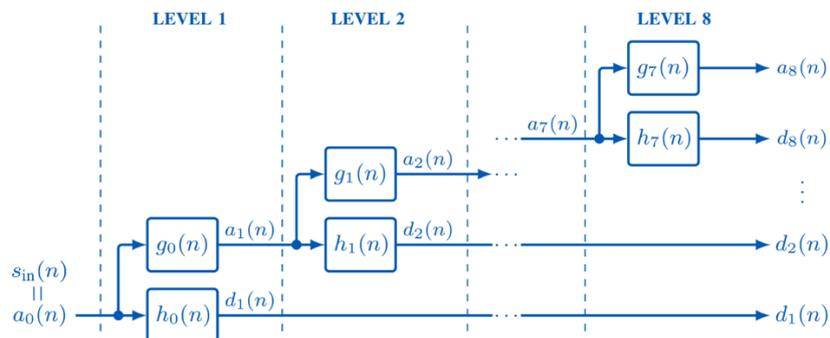


Figura 25. Descomposición SWT de ocho niveles (adaptada de [79]). En el primer nivel se obtiene el primer detalle d_1 y se propaga el coeficiente de aproximación a_1 , el cual se corresponde con la señal de entrada del segundo nivel. Progresivamente se obtienen los distintos detalles $d_1 - d_8$ y un último coeficiente de aproximación a_8 .

Las primeras 38 características están basadas en el trabajo de *Figuera et al.* [33] e *Iraia et al.* [79]. Estas características se calculan a partir de los coeficientes $d_3 - d_8$, junto con características típicas empleadas en la detección de FV, extraídas de a_0 [33]:

- **Dominio del tiempo.** describen características como amplitud, forma de la señal o ritmo cardíaco: **TCSC** (threshold crossing sample count) [36], **Expmod** (modified exponential) [37], **bCP** y **bWT** [4], **X1** y **X2** [40], **count1**, **count2** y **count3** [39], **MAV** (mean absolute value) [38]
- **Dominio frecuencial.** Describen las bandas frecuenciales presentes, la separación entre dichas bandas o su potencia espectral: **VFleak** (leakage of VF filter) [41], **Spec_SWT** (spectral algorithm) [37],

- **Complejidad de señal.** describen lo predecible que es una señal o si esta repite ciertos patrones en intervalos: **SampEn** (sample entropy) [33], **CM** (complexity measure) [45], **Cvbin** (binary covariance) [46], **Abin** (binary area) [46], **Freqbin** (binary frequency) [46], **Kurt** (kurtosis) [46], **HILB** (Hilbert transform) [44].

Adicionalmente, se calculan 11 características a partir de $s(n)$, su primera y segunda derivada. Primero se modela $s(n)$ como un proceso autoregresivo de orden 4, AR(4), tomando como características sus coeficientes y la varianza del error. Después, se calculan los momentos centrales de ambas derivadas de $s(n)$.

Por último, en base al trabajo de *Elola et al.* [48], se calculan 5 características especializadas en la detección de pulso (típicas en problemas de clasificación RP/AESP) a partir de la señal $s(n)$. Estas son, la media del valor absoluto de la pendiente, el valor medio de los intervalos entre latidos efectivos, dos medidas espectrales y la entropía difusa [80].

En total, se emplean 54 características extraídas a partir de la señal ECG.

8.2.2 Diseño clasificador 5 clases

En el apartado 5.5.2 Random Forest, se explica el funcionamiento del algoritmo, así como el significado de los parámetros de optimización del mismo, también denominados hiperparámetros. Dichos hiperparámetros regulan el rendimiento del algoritmo, de forma que, con el objetivo de procurar el mayor rendimiento posible, se realiza una búsqueda de la mejor configuración.

Primero se analiza el efecto de variar el número de árboles (N_t) que componen RF. Para ello, se entrena el algoritmo con cada uno de los 100 conjuntos de validación, dejando el resto de hiperparámetros por defecto. Se entrena con 9 particiones y se realiza el test de rendimiento sobre la partición de validación restante. En este test, el algoritmo trata de clasificar los datos de validación, anotando cada segmento ECG como una de las 5 clases posibles. Se repite el proceso, cambiando la partición de validación, hasta lograr una predicción de cada uno de los datos de entrenamiento. Por consiguiente, se obtiene para distintos valores de N_t , las predicciones realizadas por el algoritmo. Ahora con dichas predicciones, se calculan las métricas UMS y F1-score, definidas en 5.3 Evaluación del algoritmo. Al emplear 100 conjuntos de validación para cada valor de N_t , es posible dibujar para cada configuración, su valor medio y varianza, resultando en la gráfica de la Figura 26.

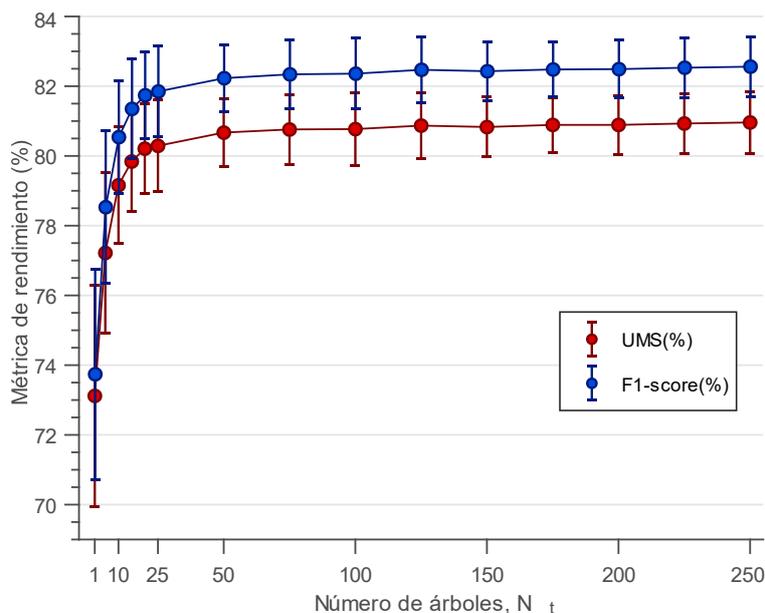


Figura 26. Influencia del número de árboles en el rendimiento del algoritmo, expresado en UMS y F1-score.

Una vez obtenida la gráfica, se observa un comportamiento lógico por parte del algoritmo. A medida que aumenta el número de árboles, se reduce el error cometido, ya que la representación de aquellos árboles que producen predicciones incorrectas se reduce. A modo de ejemplo, si se dispone de 10 árboles de decisión, RF clasifica un nuevo dato como aquella clase con más votos, de forma que puede suceder que la mitad de los árboles se equivoquen y se produzca, por tanto, un error de clasificación. Por el contrario, en el caso de emplear 100 árboles de decisión, la mitad de los votos se corresponde con 50 árboles. Es más improbable que se produzca un error en 50 árboles respecto de solo 5. Es por ello que se observa un aumento notable del rendimiento del clasificador (expresado como el valor medio de cada métrica) al variar N_t entre 1 y 50, así como una disminución en la varianza de los resultados. Sin embargo, a partir de $N_t = 100$ los resultados no varían. Aunque se pueden mejorar ligeramente los resultados mediante la optimización de otros hiperparámetros, los errores de clasificación restantes son debidos fundamentalmente a una falta de datos durante la fase de entrenamiento. Cada paciente solo se encuentra en una misma partición de datos, de forma que, si un paciente muestra una señal ECG anómala, puesto que el algoritmo no ha sido entrenado con datos semejantes, será más propenso a realizar una mala predicción de la clase.

Una vez escogido $N_t = 150$ como valor fijo (debido a que ofrece resultados ligeramente superiores a $N_t = 100$), se repite el proceso anterior variando únicamente el número de predictores empleados por cada árbol de decisión (N_p). Puesto que existen 54 características disponibles, el valor por defecto de N_p es 8, de forma que se emplearía esa cantidad de características, escogidas de forma aleatoria, para elaborar cada árbol de decisión. Tras realizar el estudio, se obtiene la Figura 27, de forma análoga al anterior análisis.

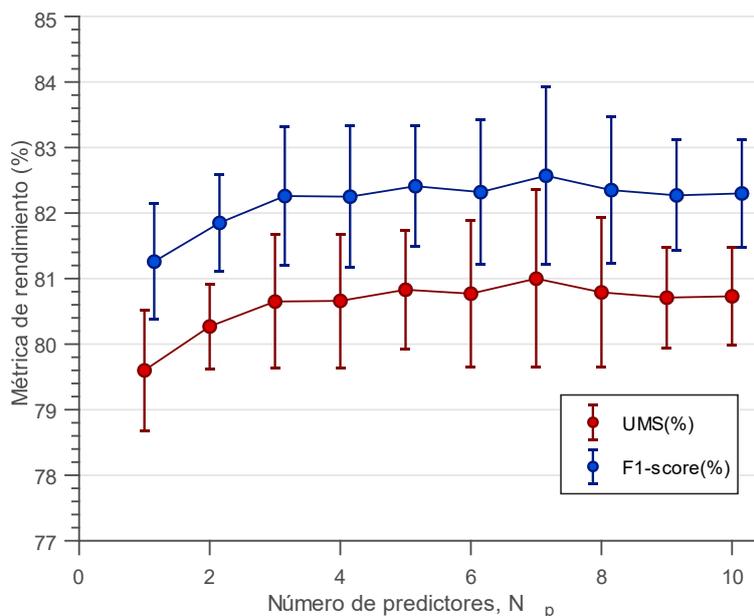


Figura 27. Influencia del número de predictores en el rendimiento del algoritmo, expresado en UMS y F1-score.

Si se emplean entre 1 y 3 predictores, es factible que un árbol de decisión no emplee las mejores características para clasificar. Por otra parte, un número demasiado elevado puede introducir ruido al tomar características semejantes, sobreajustando el modelo. Finalmente, puesto que los resultados muestran poca variación, se escoge tomar el valor por defecto, $N_p = 8$.

Por último, comentar otro hiperparámetro disponible, Prior, el cual indica las probabilidades de presenciar cada clase. Existen dos valores disponibles:

- **Uniform.** Establece la misma prevalencia para todas las clases.
- **Empirical.** Calcula la prevalencia de acuerdo a la cantidad de datos empleados para entrenamiento de cada clase.

De acuerdo a la Tabla 9, la presencia de un segmento TV frente a aquella con mayor representación, AS, se encuentra fuertemente desbalanceada. Por ello, el valor 'Uniform' no resulta interesante, puesto que asignaría una prevalencia del 20% a cada clase. De ser así, el algoritmo mejoraría su rendimiento a la hora de clasificar las clases con mayor representación, a costa de empeorar aquellas con menor cantidad de datos disponibles. Debido al carácter médico del problema, el valor 'Empirical' mejora el comportamiento del algoritmo, al focalizar el entrenamiento en las clases con menos representación, puesto que su prevalencia es menor. Este proceso se consigue al penalizar severamente los errores cometidos a la hora de clasificar clases poco comunes (RP, VF) durante la fase de entrenamiento. Para aquellas clases con alta representación en la colección de datos de entrenamiento, se les asigna una penalización más reducida, ya que su alta prevalencia resulta suficiente para obtener un buen rendimiento durante el test de clasificación.

8.2.3 Resultados obtenidos para 10 ejecuciones del algoritmo

Una vez escogida la mejor configuración de Random Forest para el problema propuesto ($N_t = 150, N_p = 8, Prior = 'Empirical'$), se somete a prueba el algoritmo. Los modelos RF obtenidos en la fase de entrenamiento son distintos para cada iteración. Se debe a que el algoritmo emplea una fracción de los datos reservados para entrenar cada árbol de decisión. A su vez, las características escogidas en cada nodo de cada árbol también varían, de forma que el

comportamiento del clasificador difiere ligeramente entre distintas ejecuciones. Es por ello que, para obtener unos resultados fiables, se entrenan diez modelos RF, con todos los datos disponibles para entrenamiento, y se obtienen los errores cometidos por todos ellos, en forma de matriz de confusión, mostrada en la Figura 28.

		Ground truth (expertos clínicos)						
		AS	RP	AESP	FV	TV		
Predicción del algoritmo	AS	12592 49.2%	0 0.0%	547 2.1%	18 0.1%	0 0.0%	95.7% 4.3%	
	RP	10 0.0%	822 3.2%	389 1.5%	0 0.0%	29 0.1%	65.8% 34.2%	
	AESP	490 1.9%	1038 4.1%	5649 22.1%	52 0.2%	44 0.2%	77.7% 22.3%	
	FV	48 0.2%	0 0.0%	15 0.1%	3306 12.9%	5 0.0%	98.0% 2.0%	
	TV	0 0.0%	40 0.2%	20 0.1%	134 0.5%	352 1.4%	64.5% 35.5%	
		95.8% 4.2%	43.3% 56.7%	85.3% 14.7%	94.2% 5.8%	81.9% 18.1%	88.8% 11.2%	

Figura 28. Matriz de confusión para 10 ejecuciones del clasificador RF ($N_t = 150, N_p = 8, \text{Prior} = \text{'Empirical'}$) sobre los datos de test, empleando 54 características de la señal ECG.

Cada casilla indica el número de segmentos de una clase concreta, clasificados por el algoritmo, respecto de las anotaciones realizadas por los expertos clínicos. Los aciertos se muestran en verde, en la diagonal, puesto que se corresponden con los segmentos clasificados con el mismo criterio que el personal SVA. En los extremos de la matriz, con fondo blanco, se muestra para cada ritmo, el porcentaje de aciertos en verde y los fallos restantes en rojo.

El valor medio (desviación estándar) de UMS y F1-score obtenido en la fase de test es de 80,15% (0,64) y 79,57% (0,60), respectivamente. Conforma una mejora considerable respecto de resultados anteriores [56], [81], con valores UMS de 78%. Recalcar que en la **¡Error! No se encuentra el origen de la referencia.** se puede observar que todas las clases son correctamente clasificadas, a excepción de RP, el cual presenta un valor UMS y F1-score del 45% y 66%, respectivamente. Esto conlleva que el rendimiento global del clasificador sea notablemente inferior respecto de la eficiencia mostrada para cada clase individual. La mayoría de fallos obtenidos al clasificar RP, es causado al anotarlo como AESP, debido a la falta de información distintiva en la señal ECG.

Con el fin de ilustrar la causa mayoritaria de los fallos cometidos por el clasificador, la Figura 29 recopila algunos casos de segmentos de señal ECG, de acuerdo a la decisión tomada por el clasificador, respecto a la anotación de los expertos clínicos.

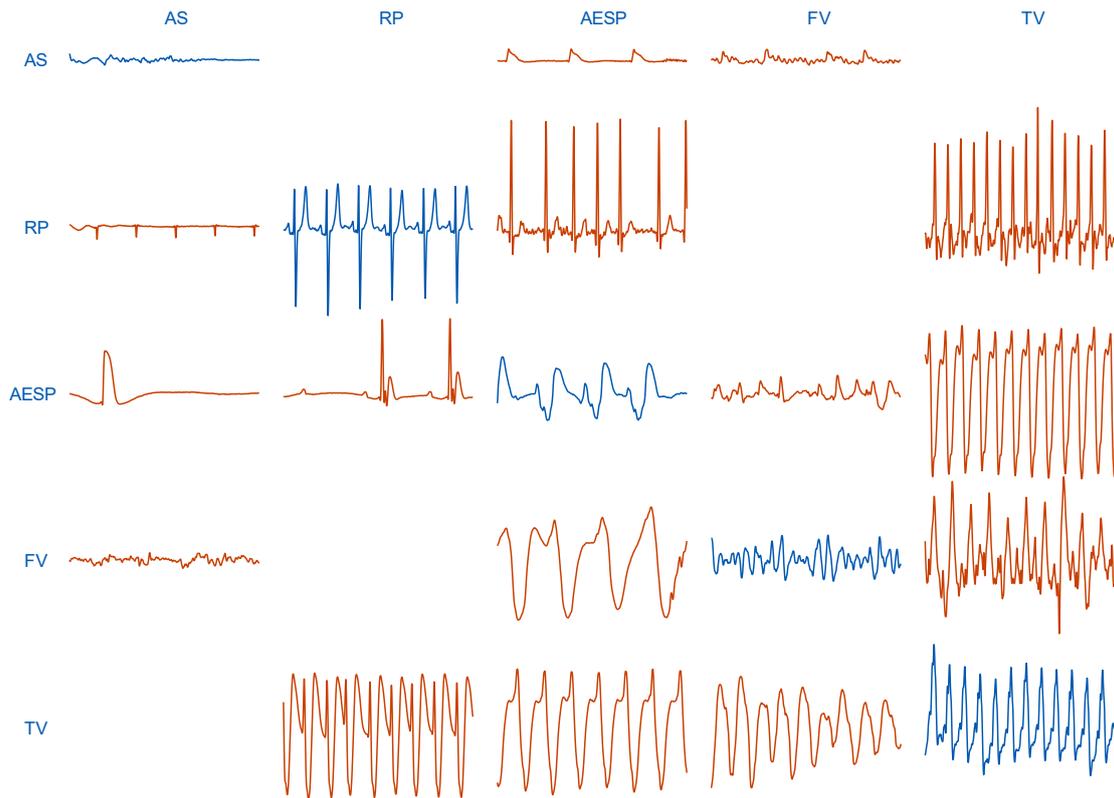


Figura 29. Ejemplo de segmentos ECG de la colección de datos reservada para test. Arriba se muestra el ritmo clasificado por los expertos clínicos. A la izquierda las predicciones del algoritmo. Los casos acertados se muestran en azul, los errores en rojo.

La mayor parte de los errores son producidos al clasificar RP como AESP y viceversa, AS como AESP y TV como FV. Debido al carácter fisiológico de los datos, existen numerosos casos frontera en los que un ritmo AESP de baja amplitud se clasifique como AS o una AS presente un complejo QRS esporádico que confunda al algoritmo, identificándolo como AESP. Los casos de TV irregulares se confunden con FV y periodos regulares de FV como TV. Por tanto, se reconocen las limitaciones que presenta la clasificación de ritmos cardiacos empleando únicamente la información presente en la señal ECG.

8.2.4 Ranking de características señal ECG

A continuación, se realiza un estudio con el objetivo de reducir el número de características empleadas en el modelo, manteniendo un rendimiento similar al obtenido durante las pruebas de test. Con este fin, el algoritmo RF dispone de una herramienta que indica la importancia de cada característica, entendida como la capacidad de esa característica para identificar las distintas clases, por si sola. Tras obtener la importancia de cada característica (expresada en tanto por uno), se ordenan menor a mayor, permitiendo escoger las que ofrecen un mejor rendimiento.

La Figura 30 muestra las distintas características (numeradas de acuerdo a nomenclatura propia), ordenadas según la importancia obtenida. Para ello, se entrena el algoritmo haciendo uso de diez particiones de datos distintas. Se promedian los valores obtenidos, con el objetivo de reducir posibles varianzas en los datos.

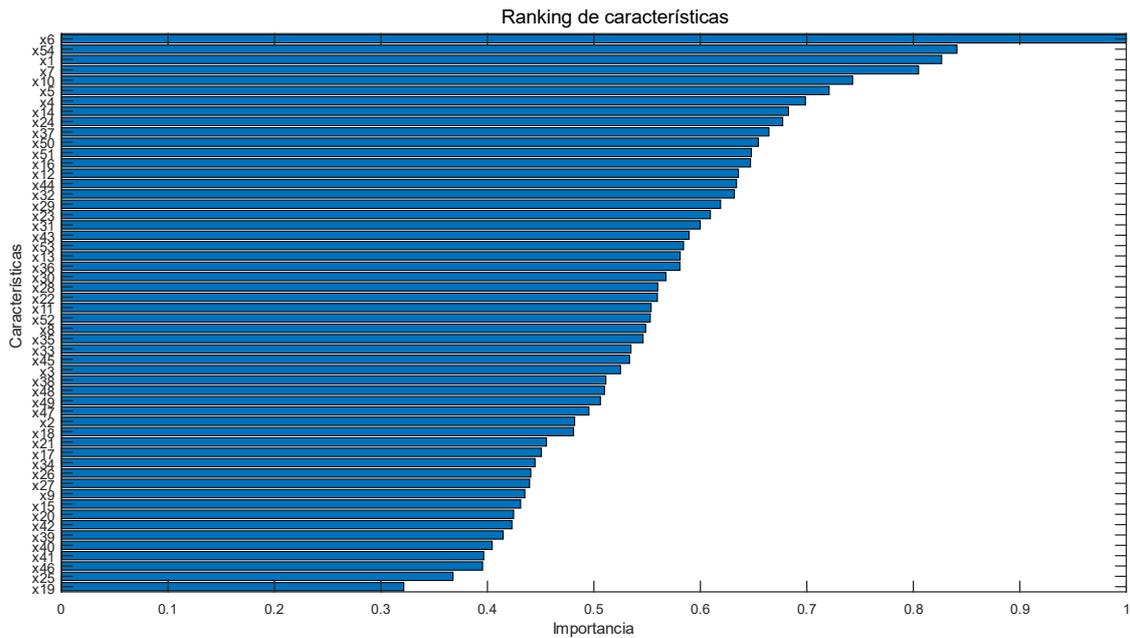


Figura 30. Se muestran 54 características de la señal ECG, ordenadas según su importancia a la hora de clasificar 5 ritmos cardiacos distintos.

Luego, se procede a entrenar el algoritmo, empleando únicamente las K mejores características, según su importancia. De nuevo, se entrena el algoritmo empleando cien particiones de datos distintas. Con los resultados obtenidos, se dibuja la Figura 31, que ilustra el valor medio y la varianza del rendimiento del algoritmo.

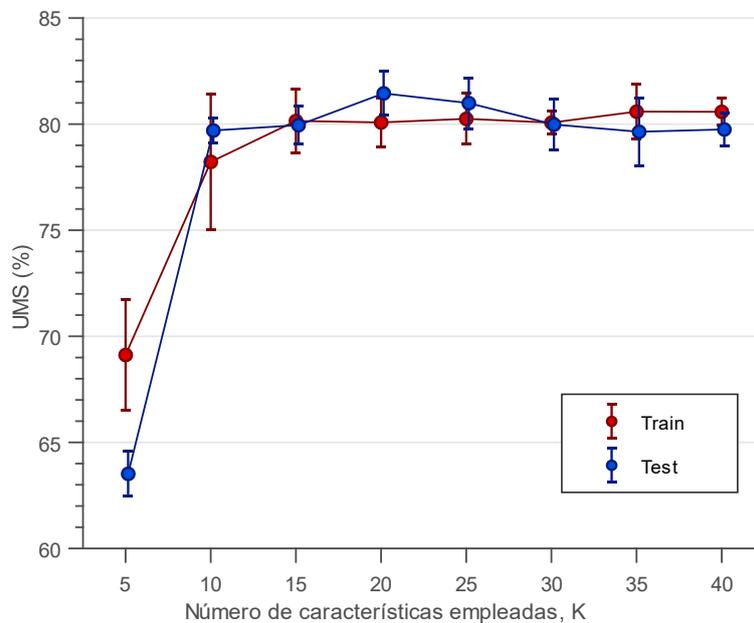


Figura 31. Rendimiento del algoritmo expresado en UMS, empleando las K mejores características de la señal ECG, para entrenamiento y test.

El valor medio del UMS mejora ligeramente para 20 características, respecto de emplear las 54 características originales. Esto es debido a que, al eliminar las características de menor

importancia, las probabilidades de que las mejores características sean escogidas en los árboles de decisión aumentan, procurando una clasificación más precisa.

A su vez, al emplear 15 características, se obtiene un valor medio UMS del 80% en entrenamiento y test. Una ventaja de usar una menor cantidad de características es que la fase de extracción de las mismas también se reduce, interesante de cara a su aplicación directa en un tratamiento o para reducir el tiempo requerido al clasificar grandes bases de datos PCREH.

8.2.5 Optimización del preprocesado de la señal ECG

Para terminar, se prueban distintas configuraciones del preprocesado de la señal ECG, con el objetivo de tratar de mejorar los resultados obtenidos.

Primero se estudia el efecto producido por los filtros Chebyshev. Algunas de las características extraídas varían fuertemente en función de las frecuencias de paso del filtro, de forma que resulta un parámetro ajustable de cara a potenciar el rendimiento del sistema. Por ello, se varía la frecuencia de corte de los filtros paso alto y paso bajo, entre 0,5-1 Hz y 25-40 Hz respectivamente. En la Tabla 10 se muestran las configuraciones posibles, empleando las frecuencias de corte establecidas, junto con el rendimiento del algoritmo obtenido, expresado en UMS.

Tabla 10. Valor UMS para distintas configuraciones del filtro de señal.

f_L (Hz)	f_H (Hz)	UMS	f_L (Hz)	f_H (Hz)	UMS
0,5	25	80,08 (0,69)	0,8	25	79,96 (0,66)
0,5	30	80,13 (0,69)	0,8	30	80,59 (0,63)
0,5	35	79,87 (0,50)	0,8	35	80,30 (0,62)
0,5	40	80,01 (0,59)	0,8	40	80,02 (0,56)
0,65	25	79,95 (0,53)	1	25	80,52 (0,53)
0,65	30	79,95 (0,46)	1	30	80,79 (0,76)
0,65	35	80,19 (0,76)	1	35	80,99 (0,65)
0,65	40	80,02 (0,65)	1	40	80,32 (0,69)

Las variaciones en el rendimiento del clasificador son de un punto, en el mejor de los casos, de forma que se mantiene la configuración original en la que se emplea la banda de 0,5 a 30 Hz. El mejor de los casos, emplea una frecuencia de corte del filtro paso alto de 1 Hz, demasiado alta para una señal ECG, puesto que aquellos ritmos cardiacos con bajas pulsaciones por minuto poseen mucha información en la banda de 0,5 a 1 Hz. Puede que, para los datos de test escogidos, el rendimiento sea ligeramente superior a la configuración por defecto, pero su rendimiento en un escenario real sería inferior, al prescindir de tanta información para ciertos casos.

Luego, se analiza el efecto producido sobre el rendimiento del algoritmo al variar la función mother wavelet [79] empleada durante la descomposición sub-banda de la señal ECG. La Tabla 11 recoge los valores obtenidos, expresados en UMS.

Tabla 11. Valor UMS para distintas configuraciones de mother wavelet.

<i>Mother wavelet</i>	<i>UMS</i>	<i>Mother wavelet</i>	<i>UMS</i>
<i>haar</i>	80,22 (0,51)	<i>sym8</i>	80,50 (0,56)
<i>db2</i>	80,76 (0,71)	<i>coif1</i>	80,78 (0,44)
<i>db3</i>	80,08 (0,47)	<i>coif2</i>	80,44 (0,68)
<i>db4</i>	80,69 (0,49)	<i>coif3</i>	80,59 (0,68)
<i>db5</i>	80,24 (0,56)	<i>coif4</i>	80,54 (0,51)
<i>db6</i>	80,33 (0,51)	<i>coif5</i>	80,62 (0,70)
<i>db7</i>	80,42 (0,40)	<i>fk4</i>	80,55 (0,41)
<i>db8</i>	80,48 (0,46)	<i>fk6</i>	80,18 (0,33)
<i>db9</i>	80,24 (0,55)	<i>fk8</i>	80,17 (0,56)
<i>sym4</i>	80,84 (0,64)	<i>fk14</i>	80,70 (0,47)
<i>sym5</i>	80,61 (0,58)	<i>fk18</i>	80,33 (0,55)
<i>sym6</i>	80,43 (0,53)	<i>fk22</i>	79,95 (0,37)
<i>sym7</i>	80,28 (0,43)	<i>dmey</i>	80,10 (0,50)

Al igual que en el análisis anterior, no se observa ninguna mejora importante, de forma que se mantiene la función Daubechies-2 (db2) como mother wavelet empleada durante la descomposición frecuencial por SWT.

8.3 Clasificador 5 clases multimodal: señal ECG y señal impedancia

Tras obtener los resultados del clasificador de 5 clases, se propone diseñar un segundo clasificador, el cual combine la información presente en la señal ECG con la ofrecida por la señal de impedancia transtorácica. Para ello, primero se diseña un clasificador de 4 clases, agrupando RP y AESP como una misma clase ORG, empleando únicamente la señal ECG. Después, se plantea un clasificador binario, especializado en la detección de RP y AESP, mediante características extraídas de las señales ECG e impedancia transtorácica. Por último, se integran ambos clasificadores, para formar el clasificador de 5 clases multimodal, de forma que todo ritmo clasificado como ORG en primera instancia, es catalogado posteriormente como RP o AESP por el clasificador binario.

8.3.1 Diseño clasificador 4 clases

El primer paso consiste en agrupar todos los segmentos RP y AESP bajo una misma clase, denominada ORG. Para ello, se modifica la etiqueta asociada a cada segmento de señal ECG. Después, es necesario volver a calcular cien particiones de datos, puesto que se dispone de una clase menos. De acuerdo a la Tabla 9, la nueva clase ORG conforma el 29,5% de la base de datos (suma de la representación de las clases RP y AESP). Por ello, a la clase ORG se le impone la misma restricción de representación mínima del 75%, aplicada a AS y FV. Finalmente, se mantiene como representación mínima para TV un 40%. Con todo ello, se dispone de las particiones de datos necesarias para validar el entrenamiento del clasificador de 4 clases.

Luego, se procede a buscar la configuración óptima del algoritmo Random Forest, puesto que, al reducir el número de clases, aunque los datos sean idénticos y se empleen las mismas características, puede que la configuración anterior no siga siendo la mejor. Al igual que en el diseño del clasificador de 5 clases, se optimiza el número de árboles y el número de predictores, empleando cien particiones de datos.

Una vez escogida la mejor configuración de Random Forest ($N_t = 150, N_p = 8, Prior = 'Empirical'$), se somete a prueba el algoritmo, obteniendo el UMS de cada clase, mostrado en la Tabla 12.

Tabla 12. Sensibilidad de cada clase obtenida para el clasificador de 4 clases.

Clase	Sin pesos (Se)
AS	95,59
ORG	92,49
FV	94,3
TV	86,05
Total	92,11

8.3.2 Ranking características clasificador 4 clases

De nuevo, se calcula la importancia de cada característica y se ordenan de mayor a menor, representado en la Figura 32.

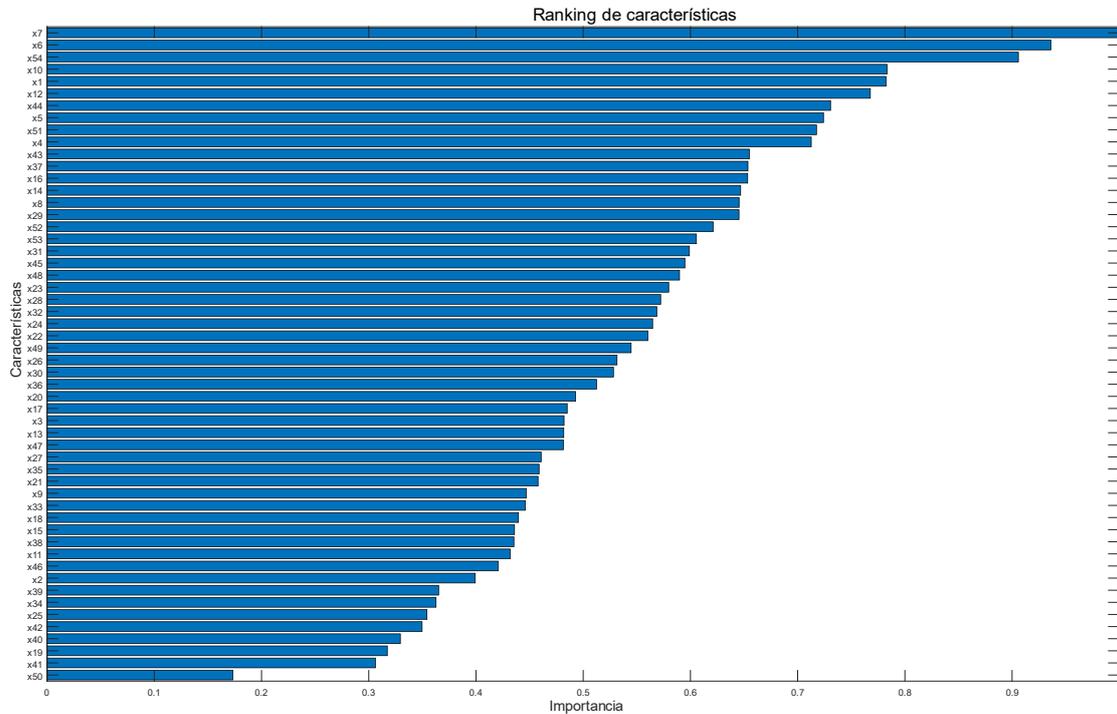


Figura 32. Características empleadas en el clasificador de 4 clases, ordenadas por importancia.

Como es de esperar, las características con importancias más altas coinciden con las de la Figura 30, puesto que los ritmos a clasificar son los mismos, a excepción de ORG. Sin embargo, al agrupar esta etiqueta RP y AESP, las características que definen estos últimos, también contienen información que describe ritmos ORG.

Para terminar, se entrena el algoritmo con un número limitado de las mejores características. De esta forma se consigue un rendimiento similar al emplear las 10 mejores características, respecto de usar las 54 propuestas, posibilitando el uso de clasificadores más sencillos al requerir menos potencia de cálculo. Sin embargo, en esta investigación se hace uso de todas las características disponibles, con ánimo de maximizar los resultados.

8.3.3 Filtrado casos ORG fallidos

El rendimiento global del clasificador multimodal, depende fuertemente del UMS obtenido al clasificar ritmos ORG. Puesto que los ritmos clasificados como tal, son discriminados posteriormente entre RP y AESP, los fallos producidos en el clasificador de 4 clases generan errores directos en el clasificador binario.

De acuerdo a este planteamiento, resulta interesante descartar los segmentos ORG erróneamente clasificados por el clasificador de 4-clases. Para ello, se repiten las simulaciones del anterior apartado, empleando únicamente diez particiones de datos y se anota para cada partición aquellos segmentos ORG mal clasificados.

Como norma, aquellos casos con menos de 3 fallos sobre las 10 particiones analizadas son igualmente añadidos a la colección de datos de entrenamiento. Para aquellos casos con 4, 5 o 6 particiones mal clasificadas se visualiza su forma de onda. De esta forma se decide si su inclusión al conjunto de datos de entrenamiento conforma un ejemplo de interés o aporta ruido (por ejemplo, puede ser una transición entre ritmos). Los segmentos con más de 7 fallos son descartados directamente. Destacar que la mayor parte de fallos se producen al clasificar AESP como AS, que tal y como se comprobó en el clasificador de 5 clases es una de las principales fuentes de errores de clasificación.

Al finalizar el proceso, se disponen de 384 segmentos RP y 1230 AESP para entrenar el algoritmo de clasificación binario. La Tabla 13 recoge el proceso de selección, mostrando la cantidad de ritmos fallados y los finalmente descartados

Tabla 13. Proceso de selección ritmos RP/AESP.

<i>Clase</i>	<i>Originales</i>	<i>Fallidos</i>	<i>Descartados</i>
<i>RP</i>	388	5	4
<i>AESP</i>	1295	130	65
Total	1683	135	69

8.3.4 Cálculo componente de circulación

Una vez filtrados los casos ORG, es necesario calcular una nueva señal a partir de la señal ECG y la impedancia transtorácica. Dicha señal, introducida en 5.4.2 Detección de circulación sanguínea, se denomina CCI y asocia los latidos del corazón con las variaciones de impedancia del paciente.

Primero, es necesario obtener las posiciones de los complejos QRS de cada segmento ECG, puesto que denotan la contracción ventricular. Para ello, se procesa la señal mediante un detector de pendiente, el cual devuelve los instantes en los que comienza cada latido.

Después, se aplica una versión modificada del algoritmo empleado en *Alonso et al.* [15] para la extracción de la señal CCI. La Figura 33 ilustra el proceso de extracción de la señal CCI a partir de la señal de IT, $s[n]$.

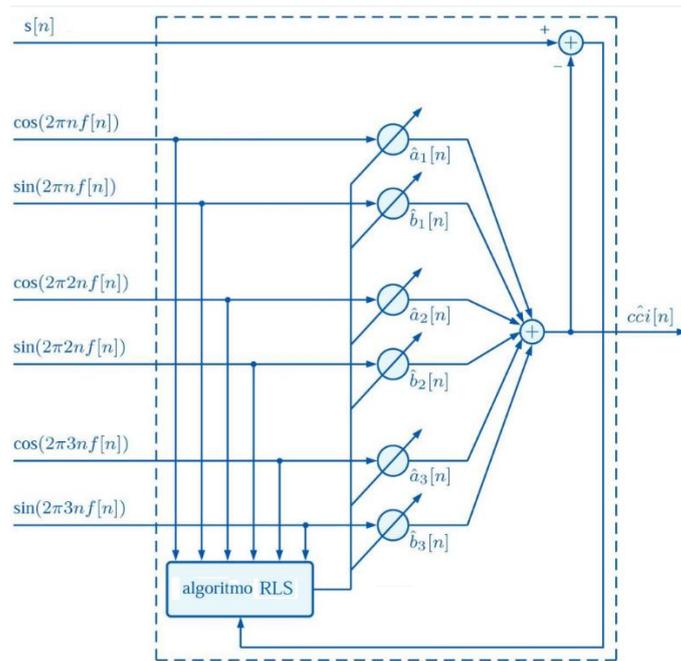


Figura 33. Esquema extractor señal CCI (adaptada de [15]), siendo $s[n]$ la señal de impedancia transtorácica.

Mediante las posiciones de los complejos QRS se obtienen los intervalos de interés de $s[n]$. Después, se modela la señal CCI mediante tres armónicos y su diferencia con $s[n]$.

De esta forma se dispone de las 3 señales sobre las que realizar la extracción de características especializadas.

8.3.5 Extracción características especializadas

En esta sección se describen las características empleadas por el clasificador binario. Debido a la similitud de los ritmos RP y AESP, es necesario disponer de características especializadas que diferencien entre las propiedades más distintas de ambos tipos.

Características basadas en señal impedancia transtorácica

Basadas en el trabajo de *Cromie et al.* [82], se extraen dos características a partir de la señal de impedancia transtorácica:

- **X1.** Valor medio de la primera derivada de la impedancia. Valores altos indican variaciones en la señal de impedancia, asociados a la circulación de sangre durante ritmos RP.
- **X2.** Potencia espectral máxima de la impedancia. Valores altos asociados a ritmos RP.

Ambas características disponen de parámetros ajustables, de forma que se realiza un estudio con el fin de escoger la mejor configuración.

Para X1, se modifican las frecuencias de filtrado de señal mínima y máxima entre 0,5-3 Hz y 3,5-8 Hz, respectivamente, en saltos de 0,5 Hz. El mejor rendimiento se obtiene para un filtrado entre 1 y 5 Hz.

Por otra parte, X2 dispone de dos parámetros ajustables: el ancho de banda sobre el que se calcula la potencia espectral máxima y el rango frecuencial sobre el que se realiza dicha

búsqueda. De nuevo, se analizan como límites frecuenciales mínimo y máximo 0,5-3 Hz y 3,5-8 Hz, respectivamente. El ancho de banda se limita a 0,1, 0,25 y 0,4 Hz. El mejor resultado se obtiene para un ancho de banda de 0,25 Hz entre 1 y 4,5 Hz.

Después, a partir del trabajo de Ruiz *et al.* [83], se calculan dos nuevas características, de nuevo, mediante la señal de impedancia transtorácica:

- **X3.** Potencia media de la impedancia. Valores elevados asociados a ritmos RP.
- **X4.** Potencia media cruzada entre señal ECG e impedancia. Valores elevados asociados a ritmos RP.

Características basadas en señal componente de circulación

En base al trabajo de Alonso *et al.* [15], se obtienen tres características a partir de la señal CCI:

- **X5.** Área media de la primera derivada de la señal CCI. Valores elevados asociados a ritmos RP.
- **X6.** Desviación estándar de la amplitud de la señal CCI. Valores elevados asociados a ritmos RP.
- **X7.** Área media de la señal CCI. Valores elevados asociados a ritmos RP.

Características basadas en señal ECG

A partir de la señal ECG se extraen nueve características (**X8 – X16**), basadas en el trabajo de Elola *et al.* [48]. Se pretende describir el ritmo cardiaco, la amplitud y pendiente, su energía espectral y la entropía de la señal.

Adicionalmente, se diseñan dos características originales:

- **X17.** Coherencia espectral entre señal ECG y señal impedancia. Valores elevados asociados a ritmos RP.
- **X18.** Potencia cruzada entre señal ECG y señal impedancia. Valores elevados asociados a ritmos RP.

Por último, se analiza el rendimiento de las características empleadas en el clasificador de 4 clases y se añaden aquellas que ofrecen un AUC mayor a 0,75. Así, se computan dieciséis características tradicionales.

Finalmente, se disponen de 34 características con las que diseñar un clasificador RP/AESP.

8.3.6 Diseño clasificador RP/AESP

Primero, se computan 100 particiones de datos, con el fin de realizar una validación cruzada durante la fase de entrenamiento. En este caso, se disponen únicamente de 2 clases, de forma que la representación mínima se fija a un valor alto, de 80% para ambas clases.

Seguidamente, se busca la configuración óptima del algoritmo RF ($N_t = 100$, $N_p = default$, $Prior = 'Empirical'$), empleando las 34 características definidas en el apartado anterior.

Puesto que la cantidad de datos disponible para cada clase difiere por mucho (384 casos RP y 1230 casos AESP), se aplican pesos distintos para cada clase. Durante el aprendizaje del algoritmo, cada caso fallado genera un *penalty* o castigo que reconduce la conducta del mismo. Al aplicar pesos, calculados como la cantidad de datos de una clase concreta respecto del total de los datos, se procura un *penalty* más severo al fallar clases con poca representación, respecto de otras con mayor cantidad de datos. De esta forma, el algoritmo se focaliza en aprender a identificar los casos RP. Aunque se reduce la sensibilidad de los casos AESP, la mejora producida

al clasificar casos RP es mucho mayor, mejorando por tanto el rendimiento global del clasificador, como se observa en la Tabla 14.

Tabla 14. Comparativa sensibilidad sin aplicar pesos y aplicando pesos.

Clase	Sin pesos (Se)	Con pesos (Se)
RP	58,44	80,60
AESP	95,15	82,86
Total	76,79	81,73

8.3.7 Ranking características clasificador RP/AESP

Por último, se calcula la importancia de las características. Para ello, no se aplican pesos a las clases, puesto que se trata de obtener la capacidad que posee cada característica, por si sola, para discriminar ambas clases. De esta forma, se obtiene la Figura 34.

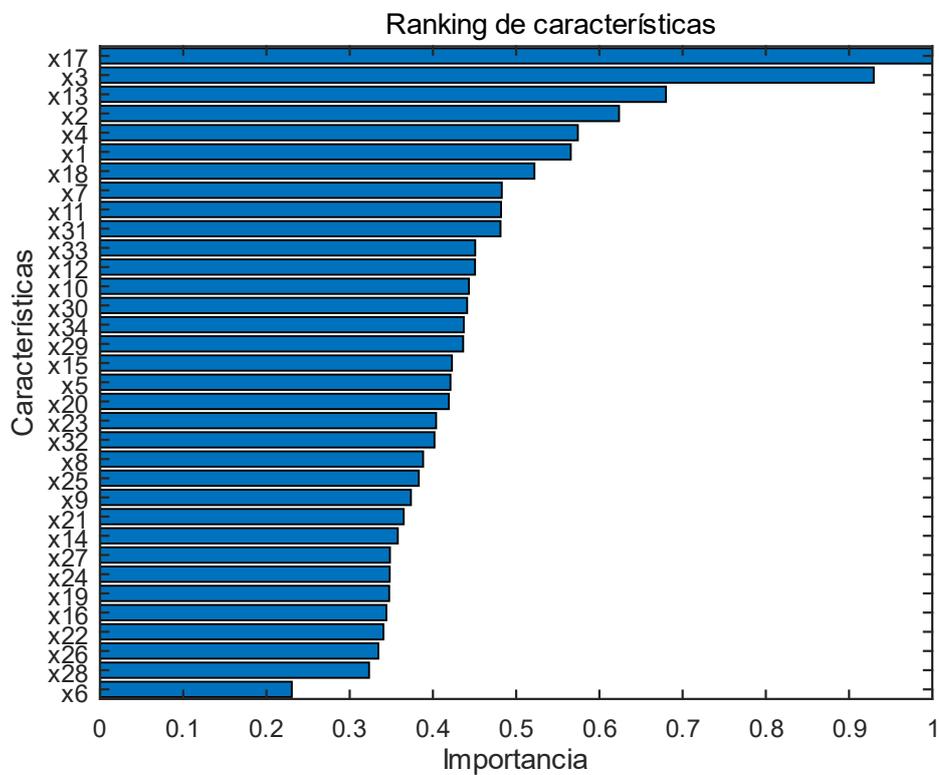


Figura 34. Características empleadas en el clasificador RP/AESP, ordenadas por importancia.

En este caso, se observa que únicamente la importancia de las nueve mejores características, supera el 50%. Esto significa que una vez empleadas las nueve mejores, el resto de características no aportan información útil.

Con esta idea en mente, se entrena el algoritmo empleando únicamente las nueve mejores características. A su vez, se aplican los pesos de cada clase, mejorando los resultados anteriores (Tabla 15).

Tabla 15. Comparativa sensibilidad al emplear todas las características y empleando las nueve mejores.

Clase	34 características (Se)	9 mejores características (Se)
RP	80,60	84,17
AESP	82,86	83,93
Total	81,73	84,05

8.3.7 Integración clasificador 5 clases

Para terminar, se colocan en cascada los clasificadores de 4 clases y RP/AESP (Figura 35). De esta forma, el primer clasificador discrimina los ritmos AS, FV y TV. En caso de identificar el ritmo como ORG, se procesa la señal en el segundo clasificador, obteniendo las dos clases restantes RP y AESP.

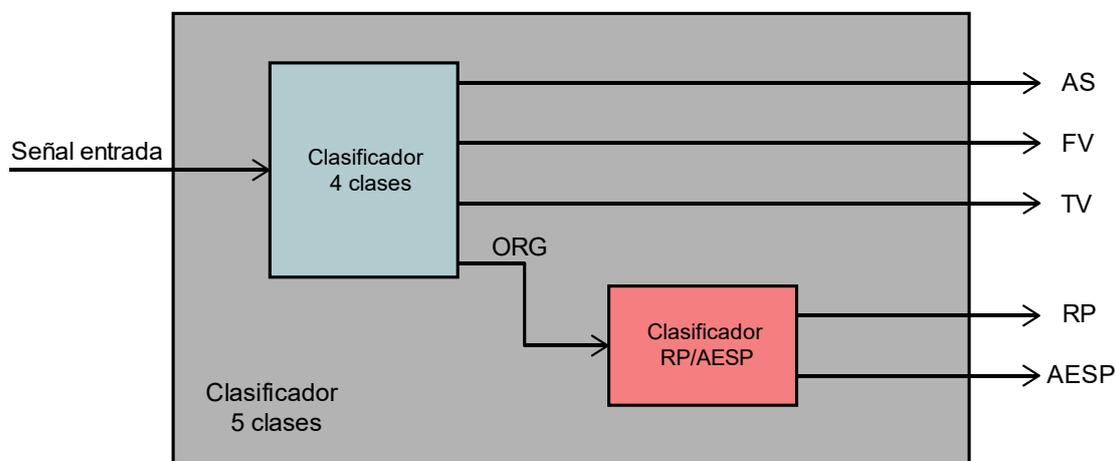


Figura 35. Diseño clasificador 5 clases multimodal.

Se entrenan los clasificadores con todos los datos disponibles para entrenamiento, y se clasifican los datos reservados para test. La Tabla 16 compara los resultados obtenidos en el clasificador multimodal respecto del clasificador basado en la señal ECG.

Tabla 16. Comparativa sensibilidad entre clasificador mediante señal ECG y clasificador multimodal.

Clase	Clasificador ECG (Se)	Clasificador Multimodal (Se)	Mejora (Se)
AS	95,8	95,6	-0,2
RP	43,3	76,3	33
AESP	85,3	74,5	-10,8
FV	94,2	94,3	0,1
TV	81,9	86,1	4,2
Total	80,1	85,4	5,26

Al añadir información presente en la señal de impedancia transtorácica se mejora la sensibilidad de las clases RP y AESP. Sin embargo, debido a la prevalencia de casos AESP en los datos, se aplican pesos para favorecer el aprendizaje del algoritmo sobre casos RP, reduciendo por tanto la sensibilidad de casos AESP. A su vez, la clasificación de ritmos TV mejora ligeramente, puesto que el clasificador identifica cuatro clases en vez de cinco, facilitando el aprendizaje.

Con todo ello, se ofrece un clasificador de cinco clases que mejora cinco puntos de UMS respecto del clasificador anterior, basado únicamente en la señal ECG. Resulta una mejora considerable e interesante, para aquellos casos en los que se disponga de la señal de impedancia.

8.4 Clasificador 5 clases mediante técnicas de Deep Learning

En este apartado, se trata de mejorar los resultados obtenidos en 8.2 Clasificador 5 clases mediante señal ECG, mediante técnicas Deep Learning. Previamente, se ha obtenido dicho clasificador mediante técnicas de Machine Learning, en las que el algoritmo RF aprende a clasificar datos a partir de una serie de características. Por ello, el rendimiento del clasificador depende directamente del diseño de dichas características, de acuerdo a su capacidad para describir los datos a clasificar. Deep Learning, sin embargo, extrae automáticamente las características que permiten distinguir entre los distintos tipos de datos de entrada, por lo que se omite todo el trabajo de investigación necesario para diseñar las características empleadas.

Por todo ello, en este apartado se trata de obtener un clasificador de 5 clases, tomando como entrada directamente la señal ECG, sin hacer uso de características. Si se obtiene un rendimiento igual o superior puede resultar una línea de investigación interesante al prescindir de la fase de ingeniería de características.

8.4.1 Diseño red convolucional preliminar

El primer paso consiste en diseñar una CNN que ofrezca resultados similares al clasificador basado en RF. Para ello, se adapta la red definida en *Picón et al.* [67] de acuerdo a la estructura de los datos disponibles en este trabajo.

La Figura 36 muestra un esquema de la arquitectura de red propuesta. Primero, se aplican dos bloques convolucionales, formados por una capa convolucional para extraer las características de la señal ECG y una capa MaxPooling para reducir la dimensionalidad al conservar únicamente los elementos más relevantes. A continuación, la capa recurrente LSTM obtiene nuevas características a partir de las relaciones temporales entre muestras consecutivas de la señal ECG. Por último, dos capas densas o neuronales permiten seleccionar las características más importantes, al restringir la transferencia de datos irrelevantes. Finalmente, la capa Softmax, formada por una neurona para cada tipo de arritmia posible, clasifica el ritmo como la clase más probable.

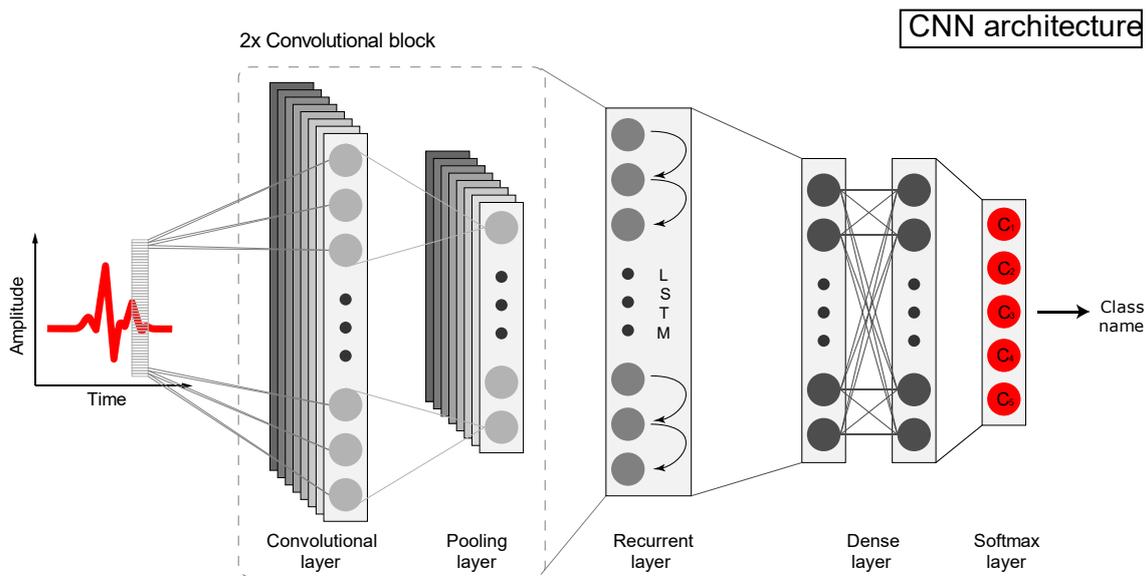


Figura 36. Arquitectura CNN: capa convolucional de 24 filtros y 16 muestras, con MaxPooling cada 3 muestras. Capa recurrente formada por 20 células LSTM, seguida por dos capas densas de 8 neuronas. Finalmente, capa Softmax de 5 neuronas. En todas las capas se emplea Relu como función de activación.

Al realizar las simulaciones, la CNN ofrece un UMS de 76, cuatro puntos por debajo de los resultados obtenidos en el clasificador de 5 clases mediante el algoritmo RF.

8.4.2 Búsqueda de mejor configuración mediante optimización Bayesiana

Debido a los resultados insuficientes, se aplica optimización Bayesiana [70] a la CNN. Este algoritmo, permite probar distintas combinaciones de hiperparámetros de la red (tales como tamaño del filtro convolucional o número de neuronas), de una forma más eficiente respecto de una búsqueda aleatoria.

Primero, se define un rango de búsqueda para los distintos hiperparámetros a optimizar. Por ejemplo, para las capas convolucionales se limitan el número de filtros entre 16 y 24 y el tamaño de los mismos entre 12 y 20 muestras. Después, se define un número de iteraciones del algoritmo de optimización, así como una función objetivo: en este caso aumentar el UMS del clasificador.

Ahora, para cada iteración del algoritmo, se emplea una combinación de hiperparámetros, conservando la mejor configuración de acuerdo a la función objetivo. La selección de hiperparámetros comienza de forma aleatoria, pero de acuerdo a los resultados obtenidos, el algoritmo converge en aquellas configuraciones que ofrecen mejores resultados.

La Figura 37 ilustra el funcionamiento del optimizador. La línea roja indica el mejor resultado obtenido tras cada iteración del algoritmo, mientras que los rombos muestran el valor de cada

evaluación.

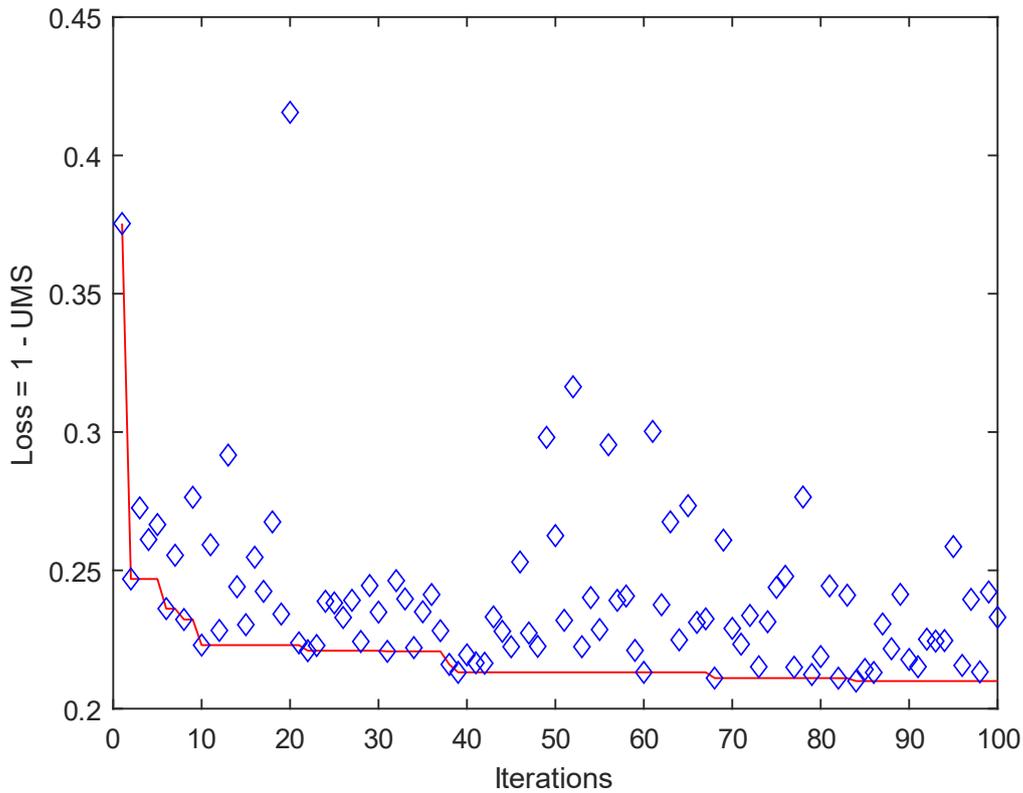


Figura 37. Resultados función objetivo para 100 iteraciones del algoritmo de optimización Bayesiana.

De esta forma, se consigue mejorar tres puntos el desempeño de la CNN, logrando un UMS de 79. Sin embargo, se mantiene un punto por debajo del rendimiento ofrecido por el clasificador de 5 clases mediante la señal ECG.

8.5 Resumen de resultados

Para concluir esta sección, se ofrece un compendio de los resultados obtenidos. De esta forma, se trata de ofrecer una vista general de la evolución del trabajo realizado.

En primer lugar, se diseña un clasificador de cinco clases, haciendo uso del algoritmo Random Forest. Este algoritmo genera sus decisiones en base a características extraídas de la señal ECG, expresadas mediante valores numéricos. Dichas características, agrupan características clásicas disponibles en el estado del arte y características destinadas a la detección de pulso [48]. Por último, se diseñan nuevas características, a partir de una descomposición sub-banda SWT.

El siguiente paso, consiste en aprovechar la información presente en la componente de impedancia transtorácica del paciente, recogida por los parches del desfibrilador durante un episodio PCREH. Puesto que facilita distinguir entre ritmos RP y AESP, permite abordar la principal fuente de errores del problema planteado. En este caso, se diseñan dos clasificadores: uno de cuatro ritmos, en el que se identifican ritmos RP y AESP como ritmos ORG, y un segundo clasificador especializado en distinguir ritmos ORG como RP y AESP. Este último emplea características extraídas de la señal ECG y de la señal de impedancia. Finalmente, se conectan en serie ambos clasificadores, obteniendo un clasificador de cinco clases.

El tercer clasificador emplea un enfoque distinto e innovador, basado en técnicas Deep Learning. Esta tecnología permite extraer por si misma las características de la señal ECG, de forma que se omite el trabajo destinado al diseño de características. De esta forma, se diseña una red neuronal adaptada en base al trabajo de *Picón et al.* [67], empleando únicamente la información presente en la señal ECG.

Por último, se aplica optimización Bayesiana en la red neuronal anterior, con el ánimo de maximizar el potencial de la arquitectura.

Con todo ello, los resultados de cada clasificador, expresados como puntos UMS, se ofrecen en la Tabla 17:

Tabla 17. Comparativa valores UMS entre distintos clasificadores.

<i>Clasificador</i>	<i>Original (ECG)</i>	<i>Multimodal (ECG + Imp)</i>	<i>Deep Learning (ECG)</i>	<i>Deep Learning optimizado (ECG)</i>
UMS	80,1	85,4	76	79

El mejor resultado disponible es ofrecido por el clasificador multimodal, puesto que dispone de más información que el resto de modelos planteados. Para aquellos casos en los que solo se disponga de la señal ECG, el clasificador original ofrece una mejora de dos puntos respecto al mejor modelo disponible en el estado del arte hasta la fecha. Por último, los modelos basados en técnicas Deep Learning, exploran un nuevo campo de investigación, sin llegar a superar a los clasificadores basados en Machine Learning. Sin embargo, en futuros trabajos sería interesante ampliar la colección de datos disponible para entrenamiento, puesto que influye directamente en la capacidad de aprendizaje del algoritmo basado en Deep Learning. Por último, queda pendiente probar técnicas Deep Learning empleando a su vez la información presente en la señal de impedancia transtorácica.

9. Metodología

Previamente a la realización del proyecto, es necesario detallar las tareas a realizar, así como designar los responsables de las mismas, dentro de los miembros del proyecto. A su vez, es preciso definir los límites y duraciones de los paquetes de trabajo de cada participante, con el fin de poder supervisar y controlar el avance del proyecto de una forma eficiente y ordenada. Con todo ello, esta sección describe el equipo de trabajo y los paquetes de trabajo realizados por cada miembro del equipo. Adicionalmente, se identifican y explican las distintas metas y entregables del proyecto.

9.1 Descripción de los recursos materiales

La Tabla 18 describe los componentes hardware y software empleados durante el proyecto. Dichos recursos son propiedad de Tecnalía y son identificados como amortizables posteriormente en el documento.

Tabla 18. Recursos hardware y software utilizados.

<i>ID</i>	<i>Material</i>
MA1	Ordenador portátil
MA2	Sistema Operativo Windows 10
MA3	Tarjeta Gráfica
MA4	Software MATLAB
MA5	Software Python
MA6	Microsoft Office 2013

9.2 Descripción de los paquetes de trabajo

Esta sección describe el grupo de trabajo, sus tareas y funciones a llevar a cabo. Además, se presentan los paquetes de trabajo definidos en el proyecto. En cada paquete de trabajo se define la tarea indicada en dicha fase, especificando su duración, fecha de inicio y de finalización.

9.2.1 Grupo de trabajo

El grupo de trabajo responsable de la consecución y finalización del proyecto es definido en este apartado, indicando las tareas y responsabilidades de cada persona. La Tabla 19 muestra dicho contenido:

Tabla 19. Grupo de trabajo del proyecto.

<i>ID</i>	<i>Material</i>	<i>Nombre</i>	<i>Función</i>
G1	Director de proyecto (UPV/EHU)	Unai Irusta Zarandona	Propuesta de proyecto, diseño general y corrección del documento
G2	Director de proyecto (Tecnalía JRL)	Javier Del Ser Lorente	Supervisión general del proyecto y experto en técnicas de Machine Learning
G3	Ingeniero Junior	Eric López Manibardo	Responsable del desarrollo del proyecto y su documentación. Autor de la memoria del proyecto

9.2.2 Paquetes de trabajo

Las siguientes tablas describen los paquetes de trabajo, junto a una descripción de las fases y el trabajo a realizar entre el inicio y fin de la fase.

Fase 1 del proyecto:

Tabla 20. Primer paquete de trabajo.

<i>WP1</i>	<i>Fecha de inicio</i>	<i>Fecha de finalización</i>	<i>Duración</i>
Gestión del proyecto: Monitorización y administración necesaria para asegurar un desarrollo satisfactorio del proyecto.	2018-09-17	2019-09-16	365
WP1.1: Gestión, monitorización y supervisión del trabajo: Coordinación, supervisión y monitorización del proyecto, desde el inicio hasta la finalización del mismo.	2018-09-17	2019-09-16	365

Fase 2 del proyecto:

Tabla 21. Segundo paquete de trabajo.

<i>WP2</i>	<i>Fecha de inicio</i>	<i>Fecha de finalización</i>	<i>Duración</i>
Preparación del proyecto: Adquisición del conocimiento necesario previo al desarrollo de la solución planteada.	2018-09-15	2018-10-31	47
WP2.1: Conocimientos previos: Estudio básico preliminar sobre técnicas de Machine Learning, Deep Learning y procesado de señal.	2018-09-15	2018-09-29	15
WP2.2: Definición del proyecto: Descripción del ámbito del proyecto y el flujo de trabajo a seguir.	2018-09-30	2018-10-01	2
WP2.3: Estado del arte: Búsqueda de información en forma de estudios, proyectos y publicaciones relacionados con el proyecto.	2018-10-02	2018-10-31	30

Fase 3 del proyecto:

Tabla 22. Tercer paquete de trabajo.

<i>WP3</i>	<i>Fecha de inicio</i>	<i>Fecha de finalización</i>	<i>Duración</i>
Desarrollo del proyecto: Agrupa las distintas etapas realizadas para ejecutar la solución del problema planteado.	2018-11-01	2019-06-06	218
WP3.1: Preparación de los datos:	2018-11-01	2018-12-10	40
3.1.1 Análisis de los datos: estudio preliminar sobre la estructura de los datos disponibles.	2018-11-01	2018-11-05	5
3.1.2 Anotación de intervalos RP y AESP: se clasifican mediante inspección visual los segmentos tipo ORG como RP o AESP.	2018-11-06	2018-11-20	15
3.1.3 Preprocesado y segmentación de los datos: filtrado y segmentación uniforme de los intervalos de señal ECG.	2018-11-21	2018-11-30	10
3.1.4 Particiones de datos: se generan particiones de datos para realizar validación cruzada durante el entrenamiento de los algoritmos.	2018-12-01	2018-12-10	10
WP3.2: Clasificador 5 clases mediante señal ECG:	2018-12-11	2019-02-17	69
3.2.1 Extracción de características ECG: se calculan las distintas características de la señal ECG, destinadas a diferenciar los 5 ritmos cardiacos.	2018-12-11	2019-01-09	30
3.2.2 Diseño clasificador 5 clases: se ajustan los hiperparámetros del algoritmo RF para obtener el mejor desempeño posible.	2019-01-10	2019-02-08	30
3.2.3 Optimización del preprocesado de señal ECG: se prueba el efecto producido en el rendimiento del algoritmo al variar el preprocesado de la señal ECG.	2019-02-09	2019-02-17	9
WP3.3 Clasificador 5 clases multimodal:	2019-02-18	2019-04-08	50

3.3.1 Diseño clasificador 4 clases: se ajustan los hiperparámetros del algoritmo RF para obtener el mejor desempeño posible.	2019-02-18	2019-02-22	5
3.3.2 Filtrado de casos ORG fallidos: se descartan los segmentos ORG mal identificados por el clasificador de 4 clases.	2019-02-23	2019-03-04	10
3.3.3 Extracción de características especializadas: se calculan características dedicadas a distinguir ritmos RP y AESP, mediante las señales ECG, impedancia y CCI.	2019-03-05	2019-03-25	21
3.3.4 Diseño clasificador RP/AESP: se ajustan los hiperparámetros del algoritmo RF para obtener el mejor rendimiento posible.	2019-03-26	2019-04-04	10
3.3.5 Integración clasificador 5 clases multimodal: se conectan en cascada el clasificador de 4 clases y el clasificador RP/AESP.	2019-04-05	2019-04-08	4
WP3.4 Clasificador 5 clases mediante técnicas Deep Learning:	2019-04-09	2019-06-06	59
3.4.1 Diseño red convolucional preliminar: se genera una arquitectura de red que ofrezca un rendimiento cercano al clasificador de 5 clases mediante señal ECG.	2019-04-09	2019-05-03	25
3.4.2 Optimización Bayesiana: se mejora el rendimiento de la CNN al optimizar los hiperparámetros de la red mediante el algoritmo Bayesiano.	2019-05-04	2019-06-06	34

Fase 4 del proyecto:

Tabla 23. Cuarto paquete de trabajo.

<i>WP4</i>	<i>Fecha de inicio</i>	<i>Fecha de finalización</i>	<i>Duración</i>
Documentación y presentación del proyecto:			
Escritura de la memoria del proyecto y presentación oral.	2019-06-07	2019-09-14	100
WP4.1: Documentación del proyecto:	2019-06-07	2019-08-15	70

Elaboración del documento que define el contexto del proyecto, objetivos, beneficios, descripción de la solución y conclusiones.			
WP4.2: Presentación oral del proyecto:			
Elaboración y ejecución de la presentación del proyecto frente al tribunal.	2019-08-16	2019-09-14	30

La descripción de la organización de los tiempos de trabajo se muestra en la Tabla 24:

Tabla 24. Organización de tiempos de trabajo.

<i>Unidad</i>	<i>Descripción</i>
Proyecto	52 semanas
Semana	7 días
Día	4 horas

9.2.3 Hitos y entregables

En esta sección se definen los hitos y entregables a completar durante la realización del proyecto. Cada uno de ellos viene definido en Tabla 25 y Tabla 26:

Tabla 25. Hitos del proyecto.

<i>ID</i>	<i>Hito</i>	<i>Fecha</i>
H1	Comienzo del proyecto	2018-09-15
H2	Definición del proyecto	2018-10-02
H3	Estado del arte concluido	2018-10-31
H4	Base de datos preparada	2018-12-10
H5	Desarrollo de algoritmos basados en RF	2019-04-08
H6	Arquitectura DL definida	2019-06-07
H7	Documentación terminada	2019-08-16
H8	Presentación terminada	2019-09-15
H9	Proyecto terminado	2019-09-15

Tabla 26. Entregables del proyecto.

<i>ID</i>	<i>Entregable</i>	<i>Fecha</i>
D1	Algoritmos de aprendizaje automático para la anotación de ritmos en parada cardiorrespiratoria	2019-06-07
D2	Memoria del proyecto	2019-08-16
D3	Presentación del proyecto	2019-09-15

9.3 Diagrama de Gantt

En la Figura 38 se muestran las tareas e hitos del proyecto, así como su duración y las fechas de inicio y fin.

WP1	↵ Gestión del proyecto	365 días	sáb 15/09/18	sáb 14/09/19
WP1.1	Gestión, monitorización y supervisión del trabajo	365 días	sáb 15/09/18	sáb 14/09/19
H1	Comienzo del proyecto	0 días	sáb 15/09/18	sáb 15/09/18
WP2	↵ Preparación del proyecto	47 días	sáb 15/09/18	mié 31/10/18
WP2.1	Conocimientos previos	15 días	sáb 15/09/18	sáb 29/09/18
WP2.2	Definición del proyecto	2 días	dom 30/09/18	lun 01/10/18
H2	Definición del proyecto concluida	0 días	lun 01/10/18	lun 01/10/18
WP2.3	Estado del arte	30 días	mar 02/10/18	mié 31/10/18
H3	Estado del arte concluido	0 días	mié 31/10/18	mié 31/10/18
WP3	↵ Desarrollo del proyecto	218 días	jue 01/11/18	jue 06/06/19
WP3.1	↵ Preparación de los datos	40 días	jue 01/11/18	lun 10/12/18
3.1.1	Análisis de los datos	5 días	jue 01/11/18	lun 05/11/18
3.1.2	Anotación de intervalos RP y AESP	15 días	mar 06/11/18	mar 20/11/18
3.1.3	Preprocesado y segmentación de los datos	10 días	mié 21/11/18	vie 30/11/18
3.1.4	Particiones de datos	10 días	sáb 01/12/18	lun 10/12/18
H4	Base de datos preparada	0 días	lun 10/12/18	lun 10/12/18
WP3.2	↵ Clasificador 5 clases mediante señal ECG	69 días	mar 11/12/18	dom 17/02/19
3.2.1	Extracción de características ECG	30 días	mar 11/12/18	mié 09/01/19
3.2.2	Diseño clasificador 5 clases	30 días	jue 10/01/19	vie 08/02/19
3.2.3	Optimización del preprocesado de señal ECG	9 días	sáb 09/02/19	dom 17/02/19
WP3.3	↵ Clasificador 5 clases multimodal	50 días	lun 18/02/19	lun 08/04/19
3.3.1	Diseño clasificador 4 clases	5 días	lun 18/02/19	vie 22/02/19
3.3.2	Filtrado de casos ORG fallidos	10 días	sáb 23/02/19	lun 04/03/19
3.3.3	Extracción de características especializadas	21 días	mar 05/03/19	lun 25/03/19
3.3.4	Diseño clasificador RP/AESP	10 días	mar 26/03/19	jue 04/04/19
3.3.5	Integración clasificador 5 clases multimodal	4 días	vie 05/04/19	lun 08/04/19
H5	Desarrollo de algoritmos basados en RF	0 días	lun 08/04/19	lun 08/04/19
WP3.4	↵ Clasificador 5 clases mediante técnicas Deep Learning	59 días	mar 09/04/19	jue 06/06/19
3.4.1	Diseño red convolucional preliminar	25 días	mar 09/04/19	vie 03/05/19
3.4.2	Optimización Bayesiana	34 días	sáb 04/05/19	jue 06/06/19
H6	Arquitectura DL definida	0 días	jue 06/06/19	jue 06/06/19
WP4	↵ Documentación y presentación del proyecto	100 días	jue 06/06/19	sáb 14/09/19
WP4.1	Documentación del proyecto	70 días	vie 07/06/19	jue 15/08/19
H7	Documentación terminada	0 días	jue 06/06/19	jue 06/06/19
WP4.2	Presentación oral del proyecto	30 días	vie 16/08/19	sáb 14/09/19
H8	Presentación terminada	0 días	sáb 14/09/19	sáb 14/09/19
H9	Proyecto terminado	0 días	sáb 14/09/19	sáb 14/09/19

Figura 38. Tareas e hitos del proyecto.

La Figura 39 muestra el diagrama de Gantt del proyecto:

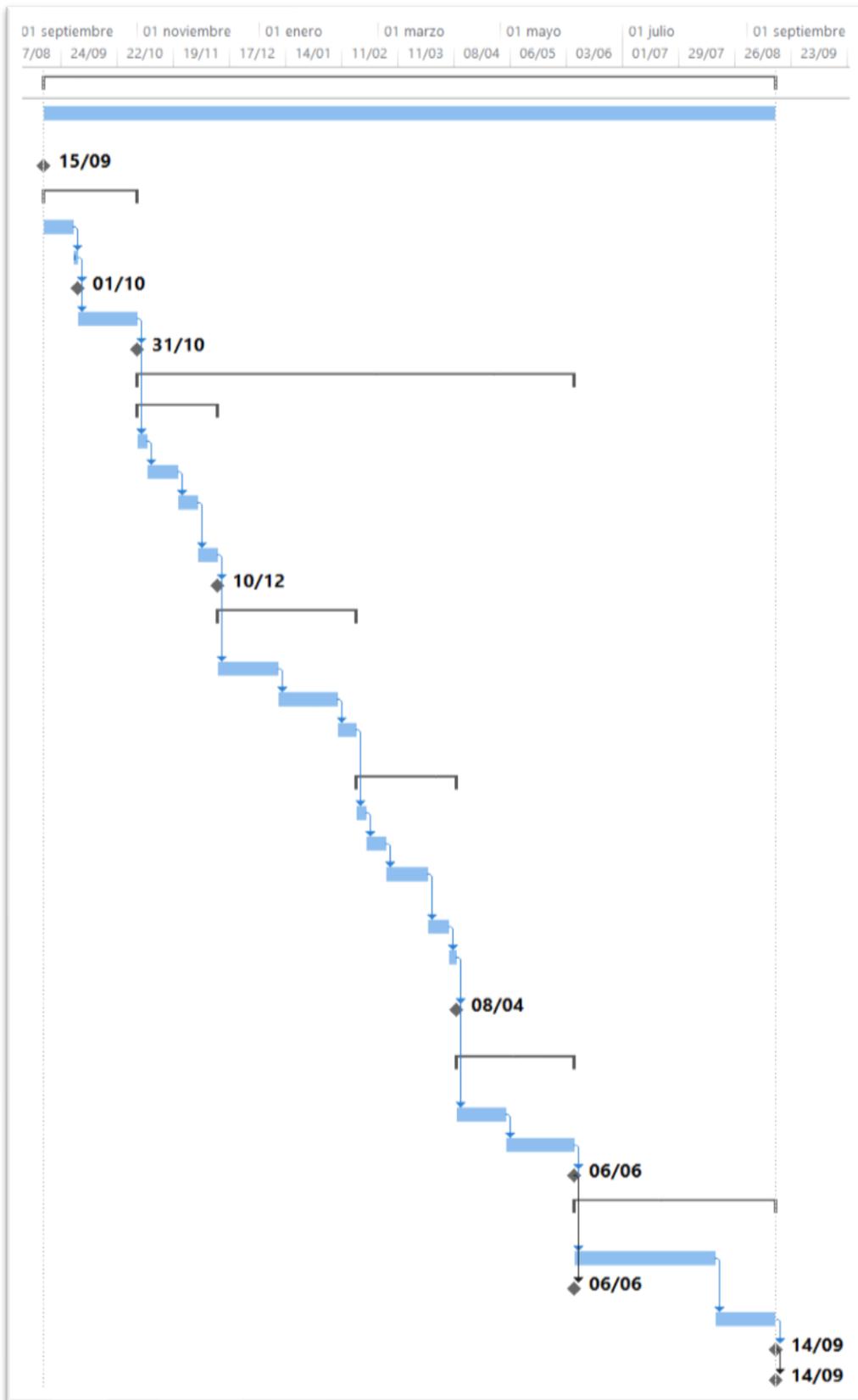


Figura 39. Diagrama de Gantt del proyecto.

10. Presupuesto del Proyecto

Esta sección presenta los recursos y presupuesto empleados durante el proyecto. Primero se calcula el coste de los recursos humanos. Seguidamente se obtiene el coste de los recursos materiales, considerando amortizaciones y consumibles.

10.1 Recursos humanos

La Tabla 27 indica el coste relacionado a cada miembro del grupo de trabajo, expresado como coste por hora

Tabla 27. Coste unitario del grupo de trabajo.

ID	Puesto	Coste unitario (€/h)
G1	Director de proyecto (UPV/EHU)	70
G2	Director de proyecto (Tecnalia JRL)	70
G3	Ingeniero Junior	30

La Tabla 28 muestra el coste total de los recursos humanos. Siguiendo la duración de los paquetes de trabajo, mostrados en el diagrama de Gantt de la Figura 39, se calculan las horas correspondientes a cada grupo. Para calcular el coste total, se computan las horas ejecutadas por el coste unitario definido en Tabla 27.

Tabla 28. Coste total de los recursos humanos.

Descripción	G1	G2	G3
Horas	150	120	1460
Coste unitario (€/h)	70	70	30
Coste total (€)	10.500	8.400	43.800
		TOTAL	62.700 €

10.2 Recursos materiales

Las tablas siguientes indican el coste material de los recursos empleados durante el desarrollo del proyecto, organizados como amortizables y fungibles.

10.2.1 Materiales amortizables

La Tabla 29 muestra el coste de los materiales amortizables, teniendo en cuenta su vida útil, tiempo de uso y precio de compra.

Tabla 29. Coste total de los materiales amortizables.

ID	Cantidad	Precio de compra (€)	Vida útil (meses)	Tiempo de uso (meses)	Coste total (€)
MA1	1	900	36	12	300
MA2	1	150	36	12	50
MA3	1	800	36	12	266,67
MA4	1	800	12	12	800
MA5	1	0	-	10	0
MA6	1	0	-	3	0
Subtotal					1.416,67 €

10.2.2 Materiales fungibles

La Tabla 30 muestra el coste asociado a los materiales fungibles.

Tabla 30. Coste total de los materiales fungibles.

ID	Material	Coste unitario (€)
CM1	Material de oficina	70
CM2	Electricidad	60
Subtotal		130 €

10.3 Coste total del Proyecto

La Tabla 31 resume el coste total del proyecto, incluyendo los costes asociados a los recursos humanos y los costes materiales, distinguidos entre amortizables y fungibles.

Tabla 31. Resumen del coste total del proyecto.

Concepto	Coste (€)
Recursos humanos	62.700
Materiales amortizables	1.416,67
Materiales fungibles	130
TOTAL	64.246,67

Teniendo en cuenta todas las partes del presupuesto, los costes totales para el desarrollo del proyecto ascienden a un total de **sesenta y dos mil doscientos cuarenta y seis euros con sesenta y siete céntimos**.

11. Conclusiones y trabajo futuro

Como se indica a lo largo del documento, el objetivo principal de este proyecto es desarrollar un algoritmo de clasificación de arritmias cardiacas mediante técnicas de aprendizaje automático, que permita anotar grandes colecciones de datos.

Durante el desarrollo del proyecto, se han completado de forma satisfactoria todos los objetivos establecidos. Primero se realizó un estudio del estado del arte. De esta forma, unido a una preparación previa por parte del estudiante, se conformó una base sólida con la que comenzar el desarrollo del proyecto. Después, se procesaron los datos médicos de los pacientes, otorgándoles unas características homogéneas. Por último, se diseñaron tres clasificadores de ritmos cardiacos: el primero emplea únicamente la señal ECG para distinguir distintos ritmos; el segundo añade como señal adicional la impedancia transtorácica; el tercero investiga el potencial de las técnicas basadas en Deep Learning, y el último trata de optimizar la red neuronal desarrollada.

El primer clasificador mejora 2 puntos de sensibilidad, respecto del mejor clasificador de cinco ritmos del estado del arte actual [56], ofreciendo una anotación más precisa. Debido a la elevada cantidad de clases presentes, resulta complicado mejorar el desempeño global del anotador automático. Por tanto, se realizó una publicación científica [84] en la que se detallan las novedades introducidas en el estudio.

El segundo clasificador mejora 5 puntos de sensibilidad respecto del primer clasificador, puesto que la señal de impedancia contiene información útil para distinguir las clases RP y AESP. Dichas clases muestran morfologías parecidas, posicionándose como la mayor fuente de errores del clasificador. Sin embargo, la señal de impedancia no se halla disponible en todos los desfibriladores, reduciendo su capacidad de implementación.

El tercer clasificador emplea únicamente la señal ECG, pero no mejora el desempeño del primer clasificador. Sin embargo, se trata de una nueva línea de investigación, en la cual el objetivo es comprobar la eficacia de nuevas técnicas, en este caso Deep Learning, puesto que elimina la carga de trabajo necesaria para diseñar nuevas características, siendo el propio algoritmo el responsable de calcularlas.

Con todo ello, se ofrecen tres soluciones distintas al problema. Las dos primeras pueden implementarse en sistemas reales y de esta forma agilizar el anotado de las colecciones de datos recogidas durante eventos PCREH, impulsando el desarrollo de nuevos sistemas de prevención y tratamiento de paros cardiacos. Por otro lado, los algoritmos de Deep Learning precisan de grandes cantidades de datos, de forma que en futuras investigaciones se podrá mejorar el desempeño de estos algoritmos si se consigue ampliar las bases de datos anotadas con las que entrenar la red neuronal y mejorar su rendimiento. A su vez, resta realizar investigaciones mediante técnicas Deep Learning que aprovechen la información presente en la señal de impedancia transtorácica.

12. Bibliografía

- [1] F. Rosell-Ortiz *et al.*, “Out-of-hospital cardiac arrest (OHCA) attended by mobile emergency teams with a physician on board. Results of the Spanish OHCA Registry (OSHCAR),” *Resuscitation*, 2017.
- [2] K. Iburguren *et al.*, “Difference in survival from pre-hospital cardiac arrest between cities and villages in the Basque Autonomous Community,” *Resuscitation*, 2015.
- [3] E. Skogvoll *et al.*, “Dynamics and state transitions during resuscitation in out-of-hospital cardiac arrest,” *Resuscitation*, 2008.
- [4] U. Irusta, J. Ruiz, E. Aramendi, S. Ruiz de Gauna, U. Ayala, and E. Alonso, “A high-temporal resolution algorithm to discriminate shockable from nonshockable rhythms in adults and children,” *Resuscitation*, 2012.
- [5] J. P. Didon, V. Krasteva, S. Ménétré, T. Stoyanov, and I. Jekova, “Shock advisory system with minimal delay triggering after end of chest compressions: Accuracy and gained hands-off time,” *Resuscitation*, 2011.
- [6] F. Alonso-Atienza, E. Morgado, L. Fernandez-Martinez, A. Garcia-Alberola, and J. L. Rojo-Alvarez, “Detection of life-threatening arrhythmias using feature selection and support vector machines,” *IEEE Trans. Biomed. Eng.*, 2014.
- [7] Q. Li, C. Rajagopalan, and G. D. Clifford, “Ventricular fibrillation and tachycardia classification using a machine learning approach,” *IEEE Trans. Biomed. Eng.*, 2014.
- [8] Y. Li, J. Bisera, M. H. Weil, and W. Tang, “An algorithm used for ventricular fibrillation detection without interrupting chest compression,” *IEEE Trans. Biomed. Eng.*, 2012.
- [9] T. Eftestøl, K. A. H. Thorsen, E. Tøssebro, C. Rong, and P. A. Steen, “Representing resuscitation data-Considerations on efficient analysis of quality of cardiopulmonary resuscitation,” *Resuscitation*, 2009.
- [10] L. Wik *et al.*, “Quality of cardiopulmonary resuscitation during out-of-hospital cardiac arrest,” *J. Am. Med. Assoc.*, 2005.
- [11] J. T. Kvaløy *et al.*, “Which factors influence spontaneous state transitions during resuscitation?,” *Resuscitation*, 2009.
- [12] T. Nordseth *et al.*, “Clinical state transitions during advanced life support (ALS) in in-hospital cardiac arrest,” *Resuscitation*, 2013.
- [13] J. Risdal, Martin, Aase, Sven Ole, Kramer-Johansen and T. Eftestøl, “Automatic identification of return of spontaneous circulation during cardiopulmonary resuscitation,” *IEEE Trans. Biomed. Eng.*, 2008.
- [14] J. Ruiz *et al.*, “Reliable extraction of the circulation component in the thoracic impedance measured by defibrillation pads,” *Resuscitation*, 2013.
- [15] E. Alonso *et al.*, “Circulation detection using the electrocardiogram and the thoracic impedance acquired by defibrillation pads,” *Resuscitation*, 2016.
- [16] C. Corcuera *et al.*, “Assessment of the diagnoses of automated external defibrillators

- operated by basic life support personnel," *Resuscitation*, 2017.
- [17] G. Galdos *et al.*, "Challenges for clinicians in ECG based retrospective resuscitation rhythm annotation," *Resuscitation*, 2017.
- [18] M. Pitarque Ordóñez, "Berpizteko bihotz erritmoak sailkatzeko ikasketa automatikoan oinarritutako algoritmo baten garapena," 2018. [Online]. Available: <http://hdl.handle.net/10810/29791>.
- [19] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Dev.*, 2000.
- [20] R. Kohavi and F. Provost, "Glossary of Terms - Journal of Machine Learning," *Mach. Learn.*, 1998.
- [21] P. Aghion, B. Jones, and C. Jones, "Artificial Intelligence and Economic Growth," 2017.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [23] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search.," *Nature*, 2016.
- [24] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," *Artif. Intell. Med.*, 2001.
- [25] M. Luu, W. G. Stevenson, L. W. Stevenson, K. Baron, and J. Walden, "Diverse mechanisms of unexpected cardiac arrest in advanced heart failure," *Circulation*, 1989.
- [26] J. Berdowski, M. T. Blom, A. Bardai, H. L. Tan, J. G. P. Tijssen, and R. W. Koster, "Impact of onsite or dispatched automated external defibrillator use on survival after out-of-hospital cardiac arrest," *Circulation*, 2011.
- [27] R. E. Kerber *et al.*, "Automatic external defibrillators for public access defibrillation: Recommendations for specifying and reporting arrhythmia analysis algorithm performance, incorporating new waveforms, and enhancing safety," *Circulation*, 1997.
- [28] Y. Xiong *et al.*, "Out-of-hospital cardiac arrest without return of spontaneous circulation in the field: Who are the survivors?," *Resuscitation*, vol. 112, pp. 28–33, Mar. 2017.
- [29] J. Soar *et al.*, "Adult advanced life support: Section 3 of the European Resuscitation Council Guidelines for Resuscitation 2015," *Notfall und Rettungsmedizin*, 2015.
- [30] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*. 2001.
- [31] S. D. Greenwald, "The development and analysis of a ventricular fibrillation detector," *Massachusetts Inst. Technol.*, 1986.
- [32] F. Nolle, F. Badura, J. Catlett, R. Bowser, and C.-G. Sketch MH., "a new concept in computerized arrhythmia monitoring systems," *Comput. Cardiol.*, 1986.
- [33] C. Figuera *et al.*, "Machine learning techniques for the detection of shockable rhythms in automated external defibrillators," *PLoS One*, 2016.
- [34] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," *Stat. Comput.*, 2011.

- [35] N. V. Thakor, Y. S. Zhu, and K. Y. Pan, "Ventricular Tachycardia and Fibrillation Detection by a Sequential Hypothesis Testing Algorithm," *IEEE Trans. Biomed. Eng.*, 1990.
- [36] M. A. Arafat, A. W. Chowdhury, and M. K. Hasan, "A simple time domain algorithm for the detection of ventricular fibrillation in electrocardiogram," *Signal, Image Video Process.*, 2011.
- [37] A. Amann, R. Tratnig, and K. Unterkofler, "Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators," *Biomed. Eng. Online*, 2005.
- [38] E. M. A. Anas, S. Y. Lee, and M. K. Hasan, "Sequential algorithm for life threatening cardiac pathologies detection based on mean signal strength and EMD functions," *Biomed. Eng. Online*, 2010.
- [39] I. Jekova and V. Krasteva, "Real time detection of ventricular fibrillation and tachycardia," *Physiol. Meas.*, 2004.
- [40] U. Ayala *et al.*, "A reliable method for rhythm analysis during cardiopulmonary resuscitation," *Biomed Res. Int.*, 2014.
- [41] S. Kuo and R. Dillman, *Computer Detection of Ventricular Fibrillation*. 1978.
- [42] S. Barro, R. Ruiz, D. Cabello, and J. Mira, "Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: a diagnostic system," *J. Biomed. Eng.*, 1989.
- [43] D. E. Lake, J. S. Richman, M. P. Griffin, and J. R. Moorman, "Sample entropy analysis of neonatal heart rate variability.," *Am. J. Physiol. Regul. Integr. Comp. Physiol.*, 2002.
- [44] A. Amann, R. Tratnig, and K. Unterkofler, "Detecting ventricular fibrillation by time-delay methods," *IEEE Trans. Biomed. Eng.*, 2007.
- [45] X. S. Zhang, Y. S. Zhu, N. V. Thakor, and Z. Z. Wang, "Detecting ventricular tachycardia and fibrillation by complexity measure," *IEEE Trans. Biomed. Eng.*, 1999.
- [46] I. Jekova, "Shock advisory tool: Detection of life-threatening cardiac arrhythmias and shock success prediction by means of a common parameter set," *Biomed. Signal Process. Control*, 2007.
- [47] H. Losert *et al.*, "Thoracic-impedance changes measured via defibrillator pads can monitor signs of circulation," *Resuscitation*, 2007.
- [48] A. Elola, E. Aramendi, U. Irusta, J. Del Ser, E. Alonso, and M. Daya, "ECG-based pulse detection during cardiac arrest using random forest classifier," *Med. Biol. Eng. Comput.*, 2019.
- [49] P. S. Hamilton and W. J. Tompkins, "Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database," *IEEE Trans. Biomed. Eng.*, 2007.
- [50] N. A. Paradis, H. R. Halperin, K. B. Kern, V. Wenzel, and D. A. Chamberlain, *Cardiac arrest: The science and practice of resuscitation medicine*. 2007.
- [51] G. Ristagno, Y. Li, F. Fumagalli, A. Finzi, and W. Quan, "Amplitude spectrum area to guide resuscitation-A retrospective analysis during out-of-hospital cardiopulmonary resuscitation in 609 patients with ventricular fibrillation cardiac arrest," *Resuscitation*, 2013.
- [52] B. Chicote *et al.*, "Application of entropy-based features to predict defibrillation

- outcome in cardiac arrest," *Entropy*, 2016.
- [53] A. B. Rad *et al.*, "Automatic cardiac rhythm interpretation during resuscitation," *Resuscitation*, 2016.
- [54] A. B. Rad *et al.*, "Probabilistic Classification Approaches for Cardiac Arrest Rhythm Interpretation during Resuscitation," in *2013 Computing in Cardiology Conference*, 2013.
- [55] A. B. Rad, T. Eftestol, J. T. Kvaloy, U. Ayala, J. Kramer-Johansen, and K. Engan, "Nearest-manifold classification approach for cardiac arrest rhythm interpretation during resuscitation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2014.
- [56] A. B. Rad *et al.*, "ECG-Based classification of resuscitation cardiac rhythms for retrospective data analysis," *IEEE Trans. Biomed. Eng.*, 2017.
- [57] S. G. Mallat, "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$," *Trans. Am. Math. Soc.*, 2006.
- [58] L. Breiman, "Random Forests," *Mach. Learn.*, 2001.
- [59] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, 2003.
- [60] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," *JMLR W&CP*, 2011.
- [61] S. Hochreiter and J. Schmidhuber, *Long Short-term Memory*, vol. 9. 1997.
- [62] Y. Cho, Kyunghyun and Van Merriënboer, Bart and Gulcehre, Caglar and Bahdanau, Dzmitry and Bougares, Fethi and Schwenk, Holger and Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *J. Biol. Chem.*, 2006.
- [63] B. He, Y. Guan, and R. Dai, "Convolutional Gated Recurrent Units for Medical Relation Classification," in *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018*, 2019.
- [64] U. R. Acharya *et al.*, "Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network," *Futur. Gener. Comput. Syst.*, 2018.
- [65] M. M. Al Rahhal, Y. Bazi, M. Al Zuair, E. Othman, and B. BenJdira, "Convolutional Neural Networks for Electrocardiogram Classification," *J. Med. Biol. Eng.*, 2018.
- [66] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," *IEEE Trans. Biomed. Eng.*, 2016.
- [67] A. Picon *et al.*, "Mixed convolutional and long short-term memory network for the detection of lethal ventricular arrhythmia," *PLoS One*, vol. 14, no. 5, p. e0216756, May 2019.
- [68] J. Kramer-Johansen *et al.*, "Quality of out-of-hospital cardiopulmonary resuscitation with real time automated feedback: A prospective interventional study," *Resuscitation*, 2006.
- [69] A. Elola *et al.*, "Deep Neural Networks for ECG-Based Pulse Detection during Out-of-Hospital Cardiac Arrest," *Entropy*, 2019.

- [70] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian Optimization with Robust Bayesian Neural Networks," in *Advances in Neural Information Processing Systems 29*, 2016.
- [71] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," 2010.
- [72] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, 2002.
- [73] F. Nelli and F. Nelli, "Deep Learning with TensorFlow," in *Python Data Analytics*, 2018.
- [74] D. Lyons, "Theano," in *The Homer Encyclopedia*, 2012.
- [75] Keras, "Keras Documentation," *Loss functions*, 2017. .
- [76] N. Ketkar and N. Ketkar, "Introduction to PyTorch," in *Deep Learning with Python*, 2017.
- [77] S. W. Smith, "Chebyshev Filters," in *Digital Signal Processing*, 2013.
- [78] G. B. Moody, "PhysioNet," in *Encyclopedia of Computational Neuroscience*, 2014.
- [79] I. Isasi *et al.*, "A Machine Learning Shock Decision Algorithm for Use During Piston-Driven Chest Compressions," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 6, pp. 1752–1760, 2019.
- [80] B. Chicote *et al.*, "Fuzzy and sample entropies as predictors of patient survival using short ventricular fibrillation recordings during out of hospital cardiac arrest," *Entropy*, 2018.
- [81] A. B. Rad *et al.*, "An automatic system for the comprehensive retrospective analysis of cardiac rhythms in resuscitation episodes," *Resuscitation*, 2018.
- [82] N. A. Cromie, J. D. Allen, C. Navarro, C. Turner, J. M. C. Anderson, and A. A. J. Adgey, "Assessment of the impedance cardiogram recorded by an automated external defibrillator during clinical cardiac arrest," *Crit. Care Med.*, 2010.
- [83] J. M. Ruiz *et al.*, "Circulation assessment by automated external defibrillators during cardiopulmonary resuscitation," *Resuscitation*, 2018.
- [84] E. Manibardo, U. Irusta, J. Del Ser, E. Aramendi, I. Isasi, and M. Olabarria, "ECG-based Random Forest Classifier for Cardiac Arrest Rhythms," 2019.